
Morpheus Documentation

Morpheus

Apr 19, 2024

MORPHEUS UI

- 1 v7.0.0 LTS Release Notes** **1**
- 2 New Features** **3**
- 3 Fixes** **5**
- 4 Appliance Updates** **9**
- 5 Worker** **11**
 - 5.1 Getting Started 11
 - 5.2 Operations 114
 - 5.3 Provisioning 169
 - 5.4 Library 211
 - 5.5 Infrastructure 342
 - 5.6 Backups 483
 - 5.7 Monitoring 488
 - 5.8 Tools 497
 - 5.9 Administration 515
 - 5.10 Personas 629
 - 5.11 Diagrams 638
 - 5.12 Troubleshooting 646
 - 5.13 Guides 684
 - 5.14 v7.0.0 LTS Release Notes 1185
 - 5.15 v7.0.0 Compatibility & Breaking Changes 1189
 - 5.16 Morpheus v7 Highlights 1196

V7.0.0 LTS RELEASE NOTES

- Compatible Plugin API version: 1.1.0
- Compatible Morpheus Worker version: 5.4.8+
- Minimum upgrade version: v6.0.0

Note: Items appended with ^{6.x.x} are also included in that version

Release Dates

- v7.0.0 Mar 20 2024

NEW FEATURES

API & CLI

- Added API support for optionally specifying a stack name when provisioning from CloudFormation templates ^{6.2.8}
- Added API support for specifying an S3 bucket to read CloudFormation templates from during provisioning. This is necessary when provisioning from CF templates greater than 50 KB ^{6.2.8}

CloudFormation

- Provisioning from CloudFormation templates now includes a “STACK NAME” configuration. By default, this will be the same as the Instance or App name but can be overridden ^{6.2.8}
- When provisioning from CloudFormation Spec Templates, added a configuration to specify an S3 bucket to read the Spec Template from. This is required for CF templates greater than 50 KB ^{6.2.8}

Dashboard

- Added support for Spanish-language localizations for Morpheus Dashboard ^{6.2.8}

Identity Sources

- “Post RelayState” field added for For SAML SSO Identity Sources using “Post Binding Mode” for defining RelayState post parameter. ^{6.2.8}

Import/Export

- Forms can now be used with the import/export feature. Export Forms as code into an integrated Git repository and import them back into any other appliance with the same repository integrated

Kubernetes

- System Kubernetes 1.29 Layouts added ^{6.2.8}

Policies

- The Roles and Policies list pages have been updated to give the user control over visible output columns and page size

Roles

- Added a Cluster Types tab to the Role detail page to control the Cluster types available to the Role

Security

- Upgraded spring-web to version 5.3.32 to mitigate CVE-2024-22243 ^{6.2.8}

Terraform

- For licensing reasons, automated Terraform installs handled by Morpheus are now capped at version 1.5.5. Other versions may be utilized in Morpheus through manual installation ^{6.2.8}

VMware

- When Snapshot names are changed in VMware, the name change is now reflected in Morpheus following the next Cloud sync ^{6.2.8}

API & CLI

- Adding or updating Identity Sources via Morpheus API, will no longer fail when adding a mapping for a built-in role (such as System Admin) ^{6.2.8}
- After updating the CODE value on a Cloud (via UI or API), GET calls to `/api/billing/zones` no longer return the old `zoneCode` value ^{6.2.8}
- Fixed situations where a 200 status and a `success=false` message could be received in the same response when creating virtual servers for load balancers via Morpheus API ^{6.2.8}
- Service Plans created via API without a `config.ranges` value in the payload can now be updated normally via API ^{6.2.8}
- Updating Catalog Item icon logos via Morpheus API now works correctly ^{6.2.8}

Agent

- Fixed an issue with the Windows Agent that could cause CPU usage to be reported incorrectly for VMs within the UI ^{6.2.8}
- Linux Agent upgrades now update the appliance URL (if applicable) to account for situations in which this has changed since the last Agent upgrade ^{6.2.8}

Amazon

- Fixed an issue causing actions such as adding nodes or reconfiguring at the Instance or VM level from working correctly for discovered nodes which were converted to managed ^{6.2.8}
- Fixed an issue that prevented AWS Clouds scoped to all regions from reading from region-scoped S3 buckets ^{6.2.8}
- Fixed an issue with snapshot sync when more than 1000 snapshots exist in an aws account. ^{6.2.8-2}
- When creating RDS Instances on AWS Clouds, the subnet groups now populate correctly for Clouds scoped to all regions ^{6.2.8}
- When scoping AWS Clouds to all VPCs, DB subnets available on the AWS Cloud are now discovered properly and are available for selection during RDS provisioning. Previously, this only worked correctly when scoping to a specific VDC ^{6.2.8}

Apps

- When adding an Instance to an existing App, the display name of the Instance is now shown rather than the Instance name ^{6.2.8}

Catalog

- The default FORM TYPE value for catalog items is now “Select” (that is, unselected) rather than “Form” ^{6.2.8}

Cluster Layouts

- Hidden-type Inputs on Cluster Layouts now show the `inputName` value along with “(hidden)” when the Cluster Layout is later edited ^{6.2.8}
- On custom Cluster Layouts, Inputs are no longer shown in an order when many Inputs are used (approximately five or more) ^{6.2.8}

Clusters

- When additional nodes are added to Clusters, they now have the custom options values (Input values) that nodes created at initial cluster provision had ^{6.2.8}

Costing

- Fixed costs shown on Instance detail pages within Subtenants showing costs in USD rather than the configured Tenant currency ^{6.2.9}
- Fixed the “Cost this Month” figure displayed for a Cloud integrated in the Master Tenant and shared down with a Subtenant ^{6.2.8}

Cypher

- Removed the “sys” mountpoint tool tip from the “ADD KEY” modal that appears when adding a new entry to Cypher ^{6.2.8}
- When File Templates containing calls to Cypher are set on a Node Type created in the Master Tenant, the Cypher values are now properly decrypted when provisioning involving the Node Type takes place in a subtenant ^{6.2.8}

Hosts

- Improved the backend logic for handling server records moved from one Cloud type to another (Change Cloud functionality) ^{6.2.8}

Import/Export

- Fixed a situation where “Export All” could cause the UI to become unresponsive and require restart ^{6.2.8}

Inputs

- Fixed Input dependency and visibility not working as configured when the Inputs were created by different methods (API vs UI) ^{6.2.8}

Instances

- Fixed Layout and Version fields not appearing on “Convert to Instance” modal (for converting servers to managed Instances) ^{6.2.9}
- Fixed reconfigures to add disks dropping the name(s) of the new disk(s) under specific conditions ^{6.2.8}
- Improved how CPU representation graphs are displayed to prevent misinterpretation of total CPU capacity being used ^{6.2.8}

Kubernetes

- Fixed 500 errors being thrown when Kubernetes Apps were deleted ^{6.2.8}

NSX-T

- Fixed network resources being visible in other Subtenants when NSX-T integrations created in one Subtenant were scoped to a public Cloud integrated from the Master Tenant ^{6.2.9}

- Load balancer virtual server protocol configuration can now be successfully updated via Morpheus API and CLI ^{6.2.8}

Network

- A friendly UI error message is now given when attempting to save a network display name larger than 255 characters ^{6.2.8}

Nutanix Prism Central

- Discovered hypervisor hosts now correctly display the OS rather than defaulting statically to display “ESXi” ^{6.2.8}
- Fixed Instances not deleting when they were provisioned to Nutanix networks utilizing Morpheus IP Pools ^{6.2.8}
- Network interfaces on Prism Central VMs no longer show a null MAC address until after the first Cloud sync following provisioning ^{6.2.8}

Option Lists

- Setting a proxy that requires authentication no longer causes 407 errors and fetch failures for REST-populated Option Lists ^{6.2.8}

Plugins

- Updated Ansible Tower and Bluecat plugins to honor global proxy settings ^{6.2.8}

Proxies

- The Morpheus Windows Agent now uses a proxy if one is configured ^{6.2.8}

SAML

- There are no longer two POST BINDING MODE fields on Add/Edit modals for SAML SSO Identity Sources. Additionally, the INCLUDES SAML REQUEST PARAMETER field no longer toggles back to “Yes” on edit ^{6.2.8}

SCVMM

- Fixed SCVMM Clouds scoped to specific clusters discovering workloads from outside that scoped cluster ^{6.2.8}
- Fixed an issue that could cause cloned Veeam backups for SCVMM Instances to not be created properly ^{6.2.8}

Security

- Fixed cache files exposing cloud credentials in plaintext under certain conditions when local credentials (not stored credentials) were used to authenticate the cloud integration ^{6.2.8}

VMware

- Any changes to minimum memory values for an image in vCenter are now properly synced over to Morpheus on the next Cloud sync ^{6.2.8}
- Fixed tag values being updated via Morpheus API being wiped out on the next Cloud sync ^{6.2.8}

APPLIANCE UPDATES

Agents

- Linux Agent: Morpheus linux agent updated to v2.5.4
- Windows Agent: Morpheus Windows Agent updated to v2.5.0.0

Appliance

- mysql: Updated default jdbc url used for db cluster connections to include connectTimeout, maxReconnects, queriesBeforeRetrySource and secondsBeforeRetrySource. Default values can be updated in morpheus.rb. ^{6.2.8}
- nginx: Logging requests to the /ping endpoint of Morpheus app nodes are now disabled by default. Logging can be reenabled by adding nginx['ssl_access_ping_log'] = true &

nginx['access_ping_log'] = true to morpheus.rb ^{6.2.8}

- Package repo cleanup: The appliance installer has been updated to clean /var/opt/morpheus/package-repos/ directory after package install when it is larger than 5GB. The current Morpheus package repo files will be added during reconfigure or supplemental package install ^{6.2.8}
- SLES 15 FIPS: - Added a FIPS-compliant Morpheus installer for SLES 15 ^{6.2.8}

Embedded Plugins

- Bluecat: bluecat-plugin updated to v1.2.1
- DigitalOcean: digital-ocean-plugin updated to v1.2.3
- Infoblox: infoblox-plugin updated to v1.3.5
- Solarwinds: solarwinds-plugin updated to v1.1.2
- phpIPAM: phpipam-plugin updated to v1.2.3
- Efficient IP: efficient-ip-plugin updated to v1.2.1

Node Package

- Morpheus Node & VM Node Packages updated to 3.2.22

WORKER

Distributed Worker

- Morpheus Worker v7.0.0 now available
- Distributed worker v7.0.0 can now be deployed in a highly-available configuration. See distributed worker documentation for additional details

5.1 Getting Started

5.1.1 Requirements

Morpheus is a software-based appliance installation capable of orchestrating many clouds and hypervisors. Before an installation is started it is important to understand some of the base requirements.

In the simplest configuration Morpheus needs one Appliance Server. The Appliance Server, by default, contains all the components necessary to orchestrate both VMs and containers. To get started some base requirements are recommended:

Base Requirements

Table 1: Supported Appliance Operating Systems

OS	Version(s)	Notes
Amazon Linux	2	
CentOS	7.x, 8.x (stream) 9.x (stream)	
Debian	10, 11	
RHEL	7.x, 8.x, 9.x	
Oracle Enterprise Linux (OEL)	7.x, 8.x	
SUSE SLES	12, 15	
Ubuntu	18.04, 20.04, 22.04	14.04 is no longer supported for Appliance OS. Note: 14.04 is still supported by the Morpheus Agent.

- **Memory:** 16 GB recommended for default installations. 8 GB minimum required with 4 GB+ available storage swap space
- **Storage:** 200 GB storage minimum (see Storage Considerations below)
- **CPU:** 4-core, 1.4 GHz (or better), 64-bit CPU recommended for all-in-one systems. For a distributed-tier installation, it's recommended each tier have 2-core, 1.4 GHz (or better), 64-bit CPU

- Network connectivity from your users to the appliance over TCP 443 (HTTPS)
- Superuser privileges via the `sudo` command for the user installing the Morpheus appliance package
- Morpheus service nodes must be configured to use accurate NTP servers. A service node may be an app node, database node, RabbitMQ, or Elasticsearch node (see Morpheus system architecture details further on in the installation section for more details)
- **Required repository access:**
 - Prior to installing the Morpheus Appliance you will need to ensure that the target server or virtual machine has access to the base YUM/DNF or APT repositories
 - A RHEL 8 server requires the `codeready` (`codeready-builder-for-rhel-8-x86_64-rpms`) repository be enabled and accessible
 - A RHEL 7 server requires access to Optional RPMs repo. The repository need to be enabled and accessible
- An appliance license is required for any operations involving provisioning
- Current major web browsers supporting modern standards, such as Google Chrome, Mozilla Firefox, Apple Safari, and Microsoft Edge are supported
- **Internet Connectivity (optional)**
 - Access to `https://downloads.morpheusdata.com`, `https://share.morpheusdata.com`, and `https://d2u3hdjdx56gx.cloudfront.net` (the `share.morpheusdata.com` cloudfront domain that share package requests will redirect to) over port 443 required on app nodes when reconfiguring to download embedded packages and plugins.
 - Access to `https://morpheus-images.morpheusdata.com`, `https://registry.morpheusdata.com`, and `https://playbooks.morpheusdata.com` To download Morpheus' system images and playbooks.
 - Offline installation requires installing the supplemental package in addition to the regular installation package. Local yum/apt repo access still required for offline installations.

Note: Access to yum and apt repos is still required for offline installations.

- **VM and Host Agent Install (optional)**
 - Inbound connectivity access from provisioned VMs and container hosts on port 443 (Agent install and communication). Port 80 may be required for older apt distros.
 - An Appliance URL that is accessible/resolvable to all managed hosts. It is necessary for all hosts that are managed by Morpheus to be able to communicate with the appliance server ip on port 443. This URL is configured under Administration > Settings.

Storage Considerations

Upon initial installation Morpheus takes up less than 10 GB of space, however Morpheus Services, Virtual Images, Backups, Logs, stats, and user-uploaded and imported data require adequate space on the Morpheus Appliance(s) based on appliance configuration and activity. Morpheus recommends at least 200 GB as a general figure to start from but storage needs will vary dramatically based on each specific use case. In some cases, significantly more space will be needed.

Important: It is the customer's responsibility to ensure adequate storage space per configuration and use case. The appliance should be properly monitored to ensure it does not run low on disk space.

Default Paths

/opt/morpheus Morpheus Application and Services Files

/var/opt/morpheus User, Application and Services Data, including default config Elasticsearch, RabbitMQ and Database data, default Virtual Image path, and working directory for Backups

/var/opt/morpheus/morpheus-ui (HA Installations Only) In an HA Installation, a NFS share is required and mounted to this location. Virtual Images, Plugins, and other shared files are stored at this path to be accessible for all nodes

/home Morpheus service account home data, such as configuration files

/var/log Morpheus Service logs

/tmp Storage for any temp files created by Morpheus or the operating system

Storage Calculator

Images

Virtual Images can be uploaded to Morpheus Storage Providers for use across Clouds. By default when no Storage Provider has been added, images will write to `/var/opt/morpheus/morpheus-ui/vms`. Please ensure adequate space when uploading Images using local file paths.

Backups

Morpheus can offload snapshots when performing backups to local or other Storage Providers. By default when no Storage Provider has been added, backups will write to `/var/opt/morpheus/bitcan/backups/`. When using none NFS Storage providers, the backup file(s) must be written to `/var/opt/morpheus/bitcan/working/` before they can be zipped, sent to the destination Storage provider such as S3, and removed from `/var/opt/morpheus/bitcan/working/`. Please ensure adequate space in `/var/opt/morpheus/bitcan/` when offloading Backups.

Note: The backup `/working` and `/backups` paths are configurable in `morpheus.rb` with `bitcan['working_directory'] = '$path'` and `bitcan['backup_directory'] = '/tmp'`

VM Logs and Stats

When using Morpheus with a locally-installed Elasticsearch configuration, VM, Container, Host and Appliance logs and stats are stored in Elasticsearch. Please ensure adequate space in `/var`, specifically `/var/opt/morpheus/elasticsearch` in relation to the number of Instances reporting logs, log frequency, and log retention count. With partition space at 85% filled or higher (by default), Elasticsearch will enter an unhealthy state and the Morpheus appliance will not function properly.

Morpheus Services Logs

Logs for services local to the Morpheus Appliance, such as the Morpheus UI, elasticsearch, rabbitmq, mysql, nginx and guacd are written to `/var/log/morpheus/`. Current logs are rotated nightly, zipped, and files older than 30 days are automatically removed. Misconfigured services, ports and permissions can cause excessive log file sizes.

Network Connectivity

Morpheus primarily operates via communication with its Agent that is installed on all managed VMs or docker hosts. This is a lightweight agent responsible for aggregating logs and stats and sending them back to the client with minimal network traffic overhead. It also is capable of processing instructions related to provisioning and deployments instigated by the appliance server.

The Morpheus Agent exists for both Linux and Windows-based platforms and opens NO ports on the guest operating system. Instead it makes an outbound SSL (HTTPS/WSS) connection to the appliance server. This is what is known as the `appliance url` during configuration (in Administration > Settings). When the Agent is started it automatically makes this connection and securely authenticates. Therefore, it is necessary for all VMs and docker based hosts that are managed by Morpheus to be able to reach the appliance server IP on port 443.

Morpheus has numerous methods to execute Agent installation, including zero open port methods.

Components

The Appliance Server automatically installs several components for the operation of Morpheus. This includes:

- RabbitMQ (Messaging)
- MySQL (Logistical Data store)
- Elasticsearch (Logs / Metrics store)
- Tomcat (Morpheus Application)
- Nginx (Web frontend)
- Guacamole (Remote console service for clientless remote console)
- Check Server (Monitoring Agent for custom checks added via UI)

All of these are installed in an isolated way using chef zero to `/opt/morpheus`. It is also important to note these services can be offloaded to separate servers or clusters as desired. For details check the installation section and high availability.

Common Ports & Requirements

The following chart is useful for troubleshooting Agent install, Static IP assignment, Remote Console connectivity, and Image transfers.

Table 2: Common Ports & Requirements

Feature	Method	OS	Source	Destination	Port	Requirement
Agent Communication	All	All	Node	Appliance	443	DNS Resolution from node to appliance url
Agent Install	All	Linux	Node	Appliance	80	Used for appliance yum and apt repos
	SSH	Linux	Appliance	Node	22	DNS Resolution from node to appliance url Virtual Images configured SSH Enabled on Virtual Image
	WinRM	Windows	Appliance	Node	5985	Not required for agent installation in VMware vCenter and vCloud Director type clouds. Otherwise, access from Morpheus App Nodes to Instance Node on 5985 Virtual Images configured WinRM Enabled on Virtual Image(<i>winrm quickconfig</i>)
	Cloud-init	Linux				Cloud-init installed on template/image Cloud-init settings populated in User Settings or in <i>Administration > Settings > Provisioning</i> Agent install mode set to Cloud-Init in Cloud Settings
	Cloudbase-init	Windows				Cloudbase-init installed on template/image Cloud-init settings populated in User Settings or in <i>Administration > Settings > Provisioning</i> Agent install mode set to Cloud-Init in Cloud Settings
	VMtools	All				VMtools installed on template Cloud-init settings populated in Morpheus user settings or in <i>Administration > Settings > Provisioning</i> when using Static IP's Existing User credentials entered on Virtual Image when using DHCP RPC mode set to VMtools in VMware cloud settings.
Static IP Assignment & IP Pools	Cloud-Init	All				Network configured in Morpheus (Gateway, Primary and Secondary DNS, CIDR populated, DHCP disabled) Cloud-init/Cloudbase-init installed on

Communication Data

The following table contains communication information, including frequency and configurability between Morpheus and its supported technology integrations.

Source	Push/Pull	Destination
Cloud Foundry App Check	Server Pull	Cloud Foundry Applications that exist within Morpheus
Docker Container Check	Server Pull	Docker containers that exist within Morpheus
Elastic Search Check	Server Pull	Elastic Search application
Microsoft SQL Server Check	Server Pull	Microsoft SQL application
Mongo Check	Server Pull	Mongo DB application
MySQL Check	Server Pull	MySQL application
Postgres Check	Server Pull	Postgres application
Push API Check	Client Push	Morpheus API
Rabbit MQ Check	Server Pull	Rabbit MQ application
Redis Check	Server Pull	Redis application
Riak Check	Server Pull	Riak application
SNMP Check	Server Pull	SNMP
Socket Check	Server Pull	Web Socket
Virtual Machine Check	Server Pull	Virtual Machine that exists within Morpheus
Web Check	Server Pull (GET) or Server Push (POST)	Web application
Public Cloud Integration	Server Pull	Alibaba Cloud
Public Cloud Integration	Server Pull	Amazon AWS
Public Cloud Integration	Server Pull	Amazon AWS GovCloud
Public Cloud Integration	Server Pull	DigitalOcean
Public Cloud Integration	Server Pull	Google Cloud Platform
Public Cloud Integration	Server Pull	Huawei Cloud
Public Cloud Integration	Server Pull	IBM Cloud
Public Cloud Integration	Server Pull	Microsoft Azure
Public Cloud Integration	Server Pull	Open Telekom Cloud
Public Cloud Integration	Server Pull	Oracle Public Cloud
Public Cloud Integration	Server Pull	UpCloud
Public Cloud Integration	Server Pull	VMware on AWS
Private Cloud Integration	Server Pull	Cisco UCS Manager
Private Cloud Integration	Server Pull	Dell EMC
Private Cloud Integration	Server Pull	HPE
Private Cloud Integration	Server Pull	HPE OneView
Private Cloud Integration	Server Pull	KVM
Private Cloud Integration	Server Pull	MacStadium
Private Cloud Integration	Server Pull	Microsoft Azure Stack
Private Cloud Integration	Server Pull	Microsoft Hyper-V
Private Cloud Integration	Server Pull	Microsoft SCVMM
Private Cloud Integration	Server Pull	Nutanix Acropolis
Private Cloud Integration	Server Pull	Openstack
Private Cloud Integration	Server Pull	Oracle VM
Private Cloud Integration	Server Pull	Pivotal Cloud Foundry
Private Cloud Integration	Server Pull	Supermicro
Private Cloud Integration	Server Pull	Vmware vCloud Director
Private Cloud Integration	Server Pull	Vmware ESXi
Private Cloud Integration	Server Pull	VMware Fusion

Source	Push/Pull	Destination
Private Cloud Integration	Server Pull	VMware vCenter
Private Cloud Integration	Server Pull	Xen Server
Automation Integration		Ansible
Automation Integration	Server Pull	Ansible Tower
Automation Integration	Server Pull	Chef
Automation Integration	Server Pull	Puppet
Automation Integration	Server Pull	Salt
Automation Integration		Terraform
Automation Integration	Server Pull	vRealize Orchestrator
Backup Integration	Server Pull	Commvault
Backup Integration	Server Pull	Veeam
Backup Integration	Server Pull	Rubrik
Backup Integration	Server Pull	Zerto
Backup Integration	Server Pull	Avamar
Build Integration	Server Pull	Jenkins
Container Integration	Server Pull	Docker
Container Integration		Docker Registry
Container Integration	Server Pull	Kubernetes
Deployment Integration	Server Pull	Git/Github
DNS Integration	Server Pull	AWS Route53
DNS Integration	Server Pull	Microsoft DNS
DNS Integration	Server Pull	PowerDNS
Identity Management Integration	Server Pull	Microsoft AD
Identity Management Integration	Server Pull	OneLogin
Identity Management Integration	Server Pull	Okta
Identity Management Integration	Server Pull	Jump Cloud
Identity Management Integration	Server Pull	LDAP
Identity Management Integration	Server Pull	SAML
IPAM Integration	Server Pull	Infoblox
IPAM Integration	Server Pull	phpIPAM
IPAM Integration	Server Pull	Bluecat
IPAM Integration	Server Pull	SolarWinds
ITSM Integration	Server Pull	ServiceNow
ITSM Integration	Server Pull	Cherwell
ITSM Integration	Server Pull	Remedy
Key & Certificate Integration	Server Pull	Venafi
Load Balancer Integration	Server Pull	AzureLB
Load Balancer Integration	Server Pull	F5 BigIP
Load Balancer Integration	Server Pull	Citrix NetScaler
Logging Integration		Syslog
Monitoring Integration	Server Pull	ServiceNow
Monitoring Integration		AppDynamics
Monitoring Integration		NewRelic
Network Integration	Server Pull	NSX-T
Network Integration	Server Pull	Cisco ACI
Network Integration	Server Pull	Unisys Stealth
Storage Integration	Server Pull	3Par
Storage Integration	Server Pull	Azure Storage
Storage Integration	Server Pull	Dell ECS

Source	Push/Pull	Destination
Storage Integration	Server Pull	Isilon
Morpheus Agent	Agent Pull	Application Tier

SELinux

If not required by organizational policy, we recommend setting SELinux to “Permissive” or “Disabled” modes to prevent any unnecessary security-related issues. Morpheus versions 3.6.0 and higher do support “Enforcing” mode if it is required by your organization due to IT policies. Set the mode appropriately prior to running the Morpheus installer and it will make the required changes based on your chosen SELinux context.

Important: Setting SELinux to “Enforcing” mode requires policies to be configured correctly in order for the Morpheus appliance to function correctly.

Supported Locales

Morpheus supports a number of different UI locales, including:

- English
- French
- German
- Spanish
- Chinese (Simplified)
- Portuguese (Brazil)
- Korean
- Italian

The full list of supported locales can be viewed within the Morpheus UI via the “Default Appliance Locale” select list within Administration Settings.

Switching Locales

The language of text in the Morpheus UI can be changed by telling the Morpheus server which locale to use. A Locale is made up of a language code and a country code. For example “en_US” is the code for US English, whilst “en_GB” is the code for British English. When using the Morpheus UI software through a client, the Morpheus UI software server will send the UI text back to the client using the appropriate static Language Pack. All translations returned via Morpheus Internationalization have been made and approved prior to the Morpheus UI version release.

Via Locale Settings in the Morpheus UI: An application wide default locale can be configured within the Master Tenant under Administration -> Settings -> Default Appliance Locale. An individual Morpheus user can configure a preferred locale within their user settings. Using a locale setting is the preferred method for switching the Morpheus UI language.

Via the Accept-Language HTTP header: The Accept-Language request HTTP header advertises which languages the client is able to understand, and which locale variant is preferred. Most browsers will by default send a “Accept-Language” HTTP header when visiting websites. The value of Accept-Language will match the configured preference of the browser’s user language/locale settings. The Morpheus UI will automatically detect the Accept-Language header

and return the UI text from the corresponding Language Pack where possible. If there is no matching Language Pack the English US language pack is displayed in the UI. If an application locale or user locale setting has been configured in the Morpheus UI, then this will override the user's browser preferences.

Via the “lang” query parameter: Morpheus has the capability to switch locales by passing a parameter called “lang” to the Morpheus UI request. For example, “operations/dashboard?lang=de” will switch the locale preference to German. Morpheus will automatically switch the user's locale and store it in a cookie so subsequent requests will have the new header. If a user is unauthenticated then the user's locale will be reset. To switch a locale back to English pass “?lang=en” as a query parameter in the Morpheus UI request.

Note: Many of Morpheus' language packs are generated by our clients. For that reason, we cannot guarantee accuracy and completeness of the translation. As new UI elements are added, existing language sets may not have immediate updates to keep pace with application changes. If you would like to contribute to a new or existing language pack, contact your account team or Morpheus support. Contributed content will be included within the next application version.

5.1.2 Capacity and Planning

There are many different architectures Morpheus can be deployed as. However, the most common architectures are an All-In-One (AIO) single node and a 3-Node Highly Available (HA) cluster. These architectures can have many variables to determine the capacity of each node, as each users' environment will be different. Example factors that determine the capacity are:

- Number of virtual machines (VMs)/systems/instances that Morpheus will manage, also know as Workload Elements (WLEs)
- If the WLEs will have agents installed
- The number of clouds added to Morpheus
- The technologies used, such as: Kubernetes, Terraform, ARM, etc.
- The number of concurrent workflows
- Number of active users

Although there are many factors that can contribute to the capacity planning, even outside the list above, below are recommended initial specifications.

Table 4: **Recommendations**

Architecture	# of CPUs per node	Memory (GB) per node	Local Storage (GB) per node	Shared Storage (NFS)	Supported WLEs
AIO	4	16	200	N/A	5,000
AIO	4	32	400	N/A	10,000
3-Node HA	4	16	400	50	10,000
3-Node HA	8	32	400	50	20,000

In the above recommendations, an AIO can support ~5,000 WLEs with agents installed at the base requirements. However, the AIO architecture cannot tolerate failure and will be unavailable during upgrades, unlike the 3-Node HA, which is the likely choice for ensuring zero down time upgrades. A 3-Node HA architecture *could* support ~15,000 WLEs with agents installed (3 x 5,000) at the base requirements but it is best to consider the possible loss of a node, which would be an effective support of ~10,000 WLEs with agents installed (2 x 5,000).

In the case of both the AIO and 3-Node HA architectures, Morpheus nodes can be **scaled up** to a maximum of 32GB of memory, which can support more WLEs per node. Adding more CPU or memory is possible, without adding additional nodes. Also, in the case of the 3-Node HA architecture, Morpheus can be **scaled out**, which can support additional WLEs per node added and provides additional redundancy, whereas the AIO configuration cannot add additional nodes. The 3-Node HA architecture should always have an odd number of nodes for quorum.

Important: Customer architectures and requirements will vary. Please contact your account manager if you wish to deploy or transition to a HA environment, which can help right-size the environment

Additional information around the various architectures can be found here:

[Installation Overview](#)

[HA Installation Overview](#)

5.1.3 Installation

Installation Overview

Important: Morpheus v4.2.0 enhanced security configuration restricts incoming appliance connections to TLS v1.2, potentially impacting front-end load balancer monitoring/health checks that support only TLS v1.1 or lower, as well as Morpheus Agent installations for Windows nodes using .net versions that do not support TLS v1.2. Refer to TLS

Morpheus comes packaged as a debian or yum based package. The default configuration installs all required services on a single vm or bare metal Host. Morpheus can be configured in a distributed architecture to use one or multiple external services, and multiple application Hosts can be configured for High Availability configurations.

All components required for Morpheus are installed and configured by default during the Morpheus **reconfigure** command. The Morpheus config file, `morpheus.rb`, can optionally be configured to point the Morpheus App to external services (distributed configuration).

Morpheus can optionally be configured to use external Database, Messaging, and/or Search Tiers. This means instead of installing, for example, MySQL on the same host as the Morpheus App, the Morpheus configuration file (`morpheus.rb`) is setup to point to an external MySQL host, cluster or service, and MySQL will not be installed or configured on the Appliance Host.

Install Packages

Morpheus Release Package urls can be obtained from <https://morpheushub.com>

Configuration Options

- **Single Host (All-In-One/default)** All tiers running on a single host. The reconfigure process installs all required services. This is the default configuration.
- **Single Hosts with Distributed Service(s)** Transactional Database, Non-Transactional Database, and/or Message tiers are externalized, with the remaining services on a single host. The reconfigure process installs all services not set to false in `/etc/morpheus/morpheus.rb`
- **Clustered Hosts with Distributed Transactional Database (3-Node HA)** Application, Message and Non-Transactional tiers are installed and clustered on three or more hosts, with all three hosts pointing to externalized database tier. The reconfigure process installs all services except mySQL.

- **App Host(s) with Distributed Services (Full HA)** Application tier is installed on one or more hosts. All UI hosts point to externalized Transactional Database, Non-Transactional Database, and Message Tiers. The reconfigure process installs only Application services.

Distributed Configurations

Morpheus provides a wide array of options when it comes to deployment architectures. It can start as a simple one machine instance where all services run on the same machine, or it can be split off into individual services per machine and configured in a high availability configuration, either in the same region or cross-region. Naturally, high availability can grow more complicated, depending on the configuration you want to do and this article will cover the basic concepts of the Morpheus HA architecture that can be used in a wide array of configurations.

There are four primary tiers of services represented within the Morpheus appliance. They are the Application Tier, Transactional Database Tier, Non-Transactional Database Tier, and Message Tier. Each of these tiers have their own recommendations for High availability deployments that we need to cover.

Application Tier

The application tier is easily installed with the same Debian or yum repository package that Morpheus is normally distributed with. Advanced configuration allows for the additional tiers to be skipped and leave only the “stateless” services that need run. These stateless services include Nginx and Tomcat. These machines should also have at least 8gb of Memory. They can be configured across all regions and placed behind a central load-balancer or Geo based load-balancer. They typically connect to all other tiers as none of the other tiers talk to each other besides through the central application tier. One final piece when it comes to setting up the Application tier is a shared storage means is necessary when it comes to maintaining things like deployment archives, virtual image catalogs, backups, etc. These can be externalized to an object storage service such as Amazon S3 or Openstack Swiftstack as well. If not using those options a simple NFS cluster can also be used to handle the shared storage structure.

Transactional Database Tier

The Transactional database tier usually consists of a MySQL compatible database. It is recommended that a lockable clustered configuration be used (Currently Percona XtraDB Cluster is the most recommended in Permissive Mode). There are several documents online related to configuring and setting up an XtraDB Cluster but it most simply can be laid out in a many master configuration. There can be some nodes setup with replication delay as well as some with no replication delay. It is common practice to have no replication delay within the same region and allow some replication delay cross region. This does increase the risk of job run overlap between the 2 regions however, the concurrent operations typically self-correct and this is a non-issue.

Non-Transactional Database Tier

The Non-Transactional tier consists of an ElasticSearch (version 7.6.0) cluster. Elastic Search is used for log aggregation data and temporal aggregation data (essentially stats, metrics, and logs). This enables for a high write throughput at scale. ElasticSearch is a Clustered database meaning all nodes no matter the region need to be connected to each other over what they call a “Transport” protocol. It is fairly simple to get setup as all nodes are identical. It is also a Java-based system and does require a sizable chunk of memory for larger data sets. 8 GB is recommended and more nodes can be added to scale either horizontally or vertically.

Messaging Tier

The Messaging tier is an AMQP based tier along with STOMP Protocol (used for agent communication). The primary model recommended is to use RabbitMQ for queue services. RabbitMQ is also a cluster-based queuing system and needs at least 3 instances for HA configurations. This is due to elections in the failover scenarios RabbitMQ can manage. If doing a cross-region HA RabbitMQ cluster, it is recommended to have at least 3 Rabbit queue clusters per region. Typically to handle HA a RabbitMQ cluster should be placed between a load balancer and the front-end application server to handle cross host connections. The ports necessary to forward in a Rabbit MQ cluster are (5672, and 61613). A RabbitMQ cluster can run on smaller memory machines depending on how frequent large requests bursts occur. 4 - 8 GB of Memory is recommended to start.

Pros/Cons

Single Host

Advantages

- Simple Installation - Morpheus Installs all required services
- Simple Configuration - Morpheus configures all required services
- Simple Maintenance - All service connections and credentials are local - All logs are local - All Data is local (by default)
- Not dependent on network connections for vital services - Facilitates speed and reliability

Disadvantages

- Single point of failure
- Individual services cannot be scaled
- Upgrades require (minimal) downtime
- Single region

Single Hosts with Distributed Service(s)

Advantages

- Individual services can be scaled
- Managed Services such as RDS can be utilized

Disadvantages

- Single region
- External services require additional configuration and maintenance
- Morpheus is subject to network performance, configuration and availability
- Increased Installation time possible

Clustered Hosts with Distributed Transactional Database

Advantages

- Database can be scaled vertically and/or horizontally
- Managed Services such as RDS can be utilized
- Zero down time upgrades
- No single point of failure
- RabbitMQ and Elasticsearch Clusters

Disadvantages

- External Database services requires additional configuration and maintenance
- App Host Clustering requires additional configuration and maintenance
- Extended Installation time
- Increased Infrastructure requirements
- Load Balancer required to front App Hosts
- Shared Storage configuration required

App Host(s) with Distributed Services

Advantages

- Individual services can be scaled vertically and/or horizontally
- Managed Services such as RDS can be utilized
- Zero down time upgrades
- No single point of failure
- Multi region support

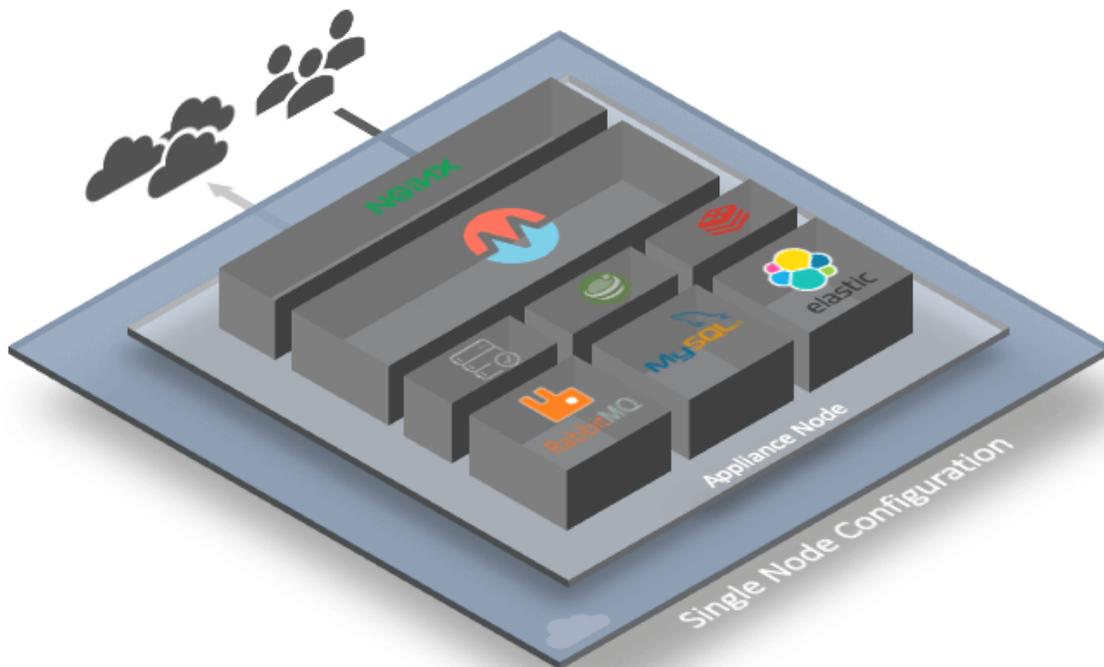
Disadvantages

- External services require additional configuration and maintenance
- Extended Installation time
- Increased Infrastructure Requirements
- Increased Networking requirements
- Load Balancer required to front App Hosts
- Shared Storage configuration required
- Rabbit Load balancer required

Single Node Installation

Single Node Install Overview

In a Single Host/All-in-one configuration, all components required for Morpheus are automatically installed and configured during the Morpheus `reconfigure` command.



Appliance Host

- **Application**
 - Morpheus App
- **Web Server/Proxy**
 - NGINX
- **Database**
 - MySQL
- **Messaging**
 - RabbitMQ
- **Search**
 - Elasticsearch
- **Console**
 - Guacamole

- **Monitoring**
 - Check Server

Default Paths

Morpheus follows several install location conventions. Below is a list of the system paths.

Important: Altering the default system paths is not supported and may break functionality.

- Installation Location: `/opt/morpheus`
- Log Location: `/var/log/morpheus`
 - Morpheus-UI: `/var/log/morpheus/morpheus-ui`
 - MySQL: `/var/log/morpheus/mysql`
 - NginX: `/var/log/morpheus/nginx`
 - Check Server: `/var/log/morpheus/check-server`
 - Elastic Search: `/var/log/morpheus/elasticsearch`
 - RabbitMQ: `/var/log/morpheus/rabbitmq`
- User-defined install/config: `/etc/morpheus/morpheus.rb`

Single Node Install on CentOS

Note: Appliance Package links are available at <https://morpheushub.com> in the downloads section.

Quick Install

1. Install the Appliance package and run `sudo morpheus-ctl reconfigure`.

That is it. After the reconfigure completes, Morpheus will start and be available at `https://your_machine_name` in a minute or few.

Step-by-step Install Instructions

1. Ensure the Morpheus Appliance host meets the minimum *Requirements*
2. Download the target version `.rpm` package for installation in a directory of your choosing. The package can be removed after successful installation.

```
wget https://downloads.morpheusdata.com/path/to/morpheus-appliance-$version.rpm
```

3. Validate the package checksum matches source checksums. For example:

```
sha256sum morpheus-appliance-$version.rpm
```

4. Next install the rpm package

```
sudo rpm -ihv morpheus-appliance-x.x.x-1.x86_64.rpm
```

5. By default the `appliance_url` uses the machines hostname, ie `https://your_machine_name`. The default url can be changed by editing `/etc/morpheus/morpheus.rb` and changing the value of `appliance_url`. Additional Appliance configuration options are available below.

Appliance Configuration Options [Click to Expand/Hide](#)

Morpheus allows for additional advanced customizations for system managed services within the `morpheus.rb` file located in `/etc/morpheus/morpheus.rb`. Below is a list of the supported items available in the `morpheus.rb` file.

Note: Service configuration settings are not applicable for externalized services such as external mysql/percona, elasticsearch or rabbitmq clusters. Only connection settings are applicable for external services. Additionally, to configure Morpheus to utilize alternate ports for SSL, you may have to take additional configuration steps. If simply appending a port to your `appliance_url` value doesn't work, consult the related article in our [KnowledgeBase](#).

```
app['encryption_key_suffix'] = '$suffix'
  # Replace $suffix with the suffix string of your choice. See `https://
  ↪ docs.morpheusdata.com/en/latest/getting_started/additional/encryption.
  ↪ html` for important details and warnings.
appliance_url 'https://morpheus.appliance-url.com'
  # Appending alternate port to appliance_url is supported. ie 'https://
  ↪ morpheus.appliance-url.com:8443'
  # The appliance_url cannot exceed 64 characters
  # The appliance_url must not contain a trailing `\/`.

bitcan['backup_directory'] = '/var/opt/morpheus/bitcan/backups'
bitcan['working_directory'] = '/var/opt/morpheus/bitcan/working'

elasticsearch['auth_password'] = 'xxxxxxxxxxxxxxxxxxxx'
elasticsearch['auth_user'] = 'morpheus-es-user'
elasticsearch['enable'] = true
elasticsearch['es_hosts'] = {'127.0.0.1' => 9200}
elasticsearch['host'] = "127.0.0.1"
elasticsearch['port'] = "9200"
elasticsearch['tmp_dir'] = '/var/tmp/elasticsearch'
elasticsearch['use_tls'] = false
↓ The following elasticsearch settings are only valid for Internal/Embedded
  ↪ elasticsearch services
elasticsearch['log_dir'] = '/var/log/morpheus/elasticsearch'
elasticsearch['memory_alloc_arena_max'] = 2
elasticsearch['memory_map_max'] = 65536
elasticsearch['memory_map_threshold'] = 131072
elasticsearch['memory_top_pad'] = 131072
elasticsearch['memory_trim_threshold'] = 131072
elasticsearch['open_files'] = 204800
elasticsearch['secure_mode'] = false
elasticsearch['xms'] = 0.25 #### Configurable for customers running into
  ↪ high memory issues, values are the percentage of total RAM. Both ui and
  ↪ es xms/xmx config only apply if Elasticsearch is enabled.
```

(continues on next page)

(continued from previous page)

```

elasticsearch['xmx'] = 0.25

guacd['guacamole_enabled'] = false
guacd['host'] = localhost #### The host guacd is listening on, if not
↳ configured 'localhost' is the default value

logging['svlogd_num'] = 30 #### keep 30 rotated log files
logging['svlogd_size'] = 209715200 #### 200 MB in bytes
logging['svlogd_timeout'] = 86400 #### rotate after 24 hours in seconds

mysql['enable'] = true
mysql['host'] = {'127.0.0.1' => 3306}
mysql['use_tls'] = false
mysql['morpheus_db_user'] = 'morpheus-db-user'
mysql['morpheus_password'] = 'morpheus-db-password'
mysql['morpheus_db'] = 'xxxxxxxxxxxxxxxxxxx'
mysql['mysql_url_override'] = 'jdbc:mysql://10.30.20.10:3306,10.30.20.
↳ 11:3306,10.30.20.12:3306/morpheusdb?autoReconnect=true&useUnicode=true&
↳ characterEncoding=utf8&failOverReadOnly=false&useSSL=false'
↓ The following mysql settings are only valid for Internal/Embedded mysql
↳ services
mysql['tmp_dir'] = '/tmp/mysql'
mysql['log_dir'] = '/var/log/morpheus/mysql'
mysql['max_active'] = 150 # The combined value off all app node max_active
↳ values must be lower than max_connections setting in mysql
mysql['max_connections'] = 151
mysql['max_allowed_packet'] = 67108864
mysql['mysql_connect_timeout'] = 60000
mysql['mysql_max_reconnects'] = 2
mysql['mysql_queries_before_retry_source'] = 0
mysql['mysql_seconds_before_retry_source'] = 300

nginx['cache_max_size'] = '5000m'
nginx['enable'] = true
nginx['loading_pages']['failure_page_h1'] = 'Morpheus Server Error'
nginx['loading_pages']['failure_page_h2'] = 'Please contact your system
↳ administrator for assistance.'
nginx['loading_pages']['failure_page_title'] = 'Morpheus Server Error'
nginx['loading_pages']['iteration_time'] = 10000 # milliseconds
nginx['loading_pages']['loading_page_h1'] = 'Morpheus is Loading...'
nginx['loading_pages']['loading_page_h2'] = 'please wait'
nginx['loading_pages']['loading_page_title'] = 'Morpheus Loading'
nginx['loading_pages']['max_loops'] = 60 # seconds
nginx['loading_pages']['timeout_page'] = '/timeout.html'
nginx['loading_pages']['timeout_page_h1'] = 'Timeout waiting for Morpheus to
↳ load, click below to try again.'
nginx['loading_pages']['timeout_page_title'] = 'Morpheus timeout, please try
↳ again...'
nginx['log_format_name'] = 'custom'
nginx['log_format'] = '\$remote_addr - \$remote_user [\$time_local] \"$request
↳ \" \" \"\$status \$body_bytes_sent \"\$http_referer\" \" \"\$http_user_agent\" \"
↳ \"\$http_x_forwarded_for\" \" \"\$request_time uct=\"\$upstream_connect_time\"
↳ uht=\"\$upstream_header_time\" urt=\"\$upstream_response_time\"';'

```

(continued from previous page)

```

nginx['ssl_ciphers'] = "ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-
↳SHA256:DHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-
↳AES256-SHA384:ECDHE-RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA:ECDHE-RSA-
↳AES128-SHA:DHE-RSA-AES256-SHA256:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-
↳SHA:DHE-RSA-AES128-SHA:ECDHE-RSA-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:AES256-
↳GCM-SHA384:AES128-GCM-SHA256:AES256-SHA256:AES128-SHA256:AES256-
↳SHA:AES128-SHA:DES-CBC3-SHA:HIGH:!aNULL:!eNULL:!EXPORT:!DES:!MD5:!PSK:!RC4
↳"
nginx['ssl_company_name'] = "Morpheus, LLC"
nginx['ssl_country_name'] = "US"
nginx['ssl_email_address'] = "personal@email.com"
nginx['ssl_locality_name'] = "San Mateo"
nginx['ssl_organizational_unit_name'] = "DevOps"
nginx['ssl_protocols'] = "TLSv1 TLSv1.1 TLSv1.2"
nginx['ssl_session_cache'] = "builtin:1000 shared:SSL:10m"
nginx['ssl_session_timeout'] = "5m"
nginx['ssl_state_name'] = "CA"
nginx['worker_connections'] = 10240
nginx['workers'] = integer calculated from number of cpus
nginx['ssl_access_ping_log'] = false #### false by default, when true GET
↳requests to the ``/ping`` endpoint are logged in the ``/var/log/morpheus/
↳nginx/morpheus-ssl-access.log`` file on the appliance
nginx['access_ping_log'] = false #### false by default, when true GET
↳requests to the ``/ping`` endpoint are logged in the ``/var/log/morpheus/
↳nginx/morpheus-ssl-access.log`` file on the appliance

rabbitmq['enable'] = true
rabbitmq['host'] = '127.0.0.1'
rabbitmq['port'] = '5672'
rabbitmq['queue_user_password'] = 'xxxxxxxxxxxxxxxxxxx'
rabbitmq['queue_user'] = 'morpheus-rmq-user'
rabbitmq['vhost'] = 'morpheus'
↓ The following rabbitmq settings are only valid for Internal/Embedded
↳rabbitmq services
rabbitmq['heartbeat'] = nil
rabbitmq['log_dir'] = '/var/log/morpheus/rabbitmq'
rabbitmq['nodename'] = 'rabbit@localhost'
rabbitmq['port'] = '5672'
rabbitmq['use_tls'] = false

repo['repo_host_url'] = 'https://downloads.morpheusdata.com'

ui['http_client_connect_timeout'] = 10000 #### milliseconds
ui['jobs_enabled'] = true #### This option disables the appliance jobs
↳service on the appliance node when set to false. This should be disabled
↳only when configuring jobs to run on specific app nodes in HA
↳environments.
ui['kerberos_config'] = nil
ui['kerberos_login_config'] = nil
ui['log_dir'] = '/var/log/morpheus/morpheus-ui'
ui['max_memory_mb'] = nil
ui['memory_alloc_arena_max'] = 2

```

(continues on next page)

(continued from previous page)

```

ui['memory_map_max'] = 65536
ui['memory_map_threshold'] = 131072
ui['memory_top_pad'] = 131072
ui['memory_trim_threshold'] = 131072
ui['pxe_boot_enabled'] = false ##### This option disables the PXE service.
↳ within the app
ui['vm_images_cdn_url'] = 'https://morpheus-images.morpheusdata.com'
ui['xms'] = 0.25 ##### Configurable for customers running into high memory.
↳ issues, values are the percentage of total RAM. Both ui and es xms/xmx.
↳ config only apply if Elasticsearch is enabled.
ui['xmx'] = 0.25

```

6. After all configuration options have been set, run:

```
sudo morpheus-ctl reconfigure
```

Note: Configuration options can be updated after the initial reconfigure by editing `/etc/morpheus/morpheus.rb` and running `sudo morpheus-ctl reconfigure` again. Appliance and other services may need to be restarted depending on configuration changes.

7. Once the installation is complete the `morpheus-ui` service will automatically start up and be available shortly. To monitor the UI startup process, run `morpheus-ctl tail morpheus-ui` and look for the ascii logo accompanied by the install version and start time:

```

timestamp:  __  __
timestamp:  /  | /  /__  _____  / /  __  __  _____
timestamp:  /  /|_ /  /  \  __/  _ \  _ \  _ \  -) //  (  -<
timestamp:  /_ /  /_/\___/_ / /  .__/_//_/\___/\_,_/_/___/
timestamp:  *****
timestamp:  Version: |morphver|
timestamp:  Start Time: xxx xxx xxx 00:00:00 UTC 2021
timestamp:  *****

```

There are additional install settings that can be viewed in the [Additional Configuration Options](#) section.

Once the browser is pointed to the appliance a first time setup wizard will be presented. Please follow the on screen instructions by creating the master account. From there you will be presented with the license settings page where a license can be applied for use (if a license is required you may request one or purchase one by contacting your sales representative).

More details on setting up infrastructure can be found throughout this guide.

Tip: If any issues occur it may be prudent to check the morpheus log for details at `/var/log/morpheus/morpheus-ui/current`.

Single Node Install on Debian/Ubuntu

To get started installing Morpheus on Ubuntu or Debian a few preparatory items should be addressed first.

1. First make sure the apt repository is up to date by running `sudo apt-get update`. It is advisable to verify the assigned hostname of the machine is self-resolvable.

Important: If the machine is unable to resolve its own hostname `nslookup hostname` some installation commands will be unable to verify service health during installation and fail.

1. Next simply download the relevant `.deb` package for installation. This package can be acquired from <https://morpheushub.com> downloads section.

Tip: Use the `wget` command to directly download the package to your appliance server. i.e. `wget https://downloads.morpheusdata.com/path/to/package/morpheus-appliance_x.x.x-1_amd64.deb`

1. Next we must install the package onto the machine and configure the morpheus services:

```
sudo dpkg -i morpheus-appliance_x.x.x-1_amd64.deb
sudo morpheus-ctl reconfigure
```

2. Once the installation is complete the web interface will automatically start up. By default it will be resolvable at `https://your_machine_name` and in many cases this may not be resolvable from your browser. The url can be changed by editing `/etc/morpheus/morpheus.rb` and changing the value of `appliance_url`. After this has been changed simply run:

```
sudo morpheus-ctl reconfigure
sudo morpheus-ctl stop morpheus-ui
sudo morpheus-ctl start morpheus-ui
```

Note: The `morpheus-ui` can take 2-3 minutes to startup before it becomes available.

There are additional install settings that can be viewed in the [Additional Configuration Options](#) section.

Once the browser is pointed to the appliance a first time setup wizard will be presented. Please follow the on screen instructions by creating the master account. From there you will be presented with the license settings page where a license can be applied for use (if a license is required you may request one or purchase one by contacting your sales representative).

More details on setting up infrastructure can be found throughout this guide.

Tip: If any issues occur it may be prudent to check the morpheus log for details at `/var/log/morpheus/morpheus-ui/current`.

Single Node Install on RHEL

To get started installing Morpheus on RHEL/RedHat a few prerequisite items are required.

1. Configure firewalld to allow access from users on 443 (Or remove firewall if not required).
2. Make sure the machine is self resolvable to its own hostname.
3. For RHEL 7.x, the Optional RPMs repo needs to be added for Reconfigure to succeed. It does not need to be added For RHEL 8.x, as the Optional RPMs repo is now part of the appstream repo that is enabled by default in RHEL 8.x.
 - **RHEL 7.x Amazon:** `yum-config-manager --enable rhel-7-server-rhui-optional-rpms`
 - **RHEL 7.x:** `yum-config-manager --enable rhel-7-server-optional-rpms`

Note: For Amazon users a Redhat subscription is not required if the appropriate yum REGION repository is added instead as demonstrated above.

The RedHat Enterprise Linux server needs to be registered and activated with Redhat subscription. The server optional rpms repo needs to be enabled as well.

To check if the server has been activated please run the subscription-manager version. Subscription manager will return the version plus the python dependency version.

If the server has not been registered and activated then the subscription manager version will return the below message.

```
sudo subscription-manager version
server type: This system is currently not registered
subscription management server: 0.9.51.24-1
subscription-manager: 1.10.14-7.e17 python-rhsm: 1.10.12-2.e17
```

When a server has been registered and activated with Redhat the subscription manager will return the below message.

```
sudo subscription-manager version
server type: Red Hat Subscription Management
subscription management server: 0.9.51.24-1
subscription-manager: 1.10.14-7.e17 python-rhsm: 1.10.12-2.e17
```

If the subscription manager re-returns the message `This system is currently not registered` please follow the below steps to register the server.

Tip: To register the server you will need to have sudo permissions [Member of the Wheel group] or root access to the server. You will also need your Redhat registered email address and password.

subscription-manager register

```
sudo subscription-manager register
Username: redhat@example.com
Password: . subscription-manager auto --attach
```

Note: This can take a minute to complete

```
sudo subscription-manager attach --auto
```

```
Installed Product Current Status: Product Name: Red Hat Enterprise Linux  
Server Status: Subscribed
```

To check to see if the RHEL server has the Red Hat Enterprise Linux 7 Server - Optional (RPMs) repo enabled please run the following command to return the repo status.

Tip: To check the server repos you will need to have sudo permissions [Member of the Wheel group] or root access to the server.

```
sudo yum repolist all | grep "rhel-7-server-optional-rpms" rhel-7-server-optional-rpms/  
↪7Server/x86_64 disabled
```

If the repo status was returned as disabled then you will need to enable the repo using the subscription manager like below.

```
sudo subscription-manager repos --enable rhel-7-server-optional-rpms  
Repository 'rhel-7-server-optional-rpms' is enabled for this system.
```

The message Repo 'rhel-7-server-optional-rpms' is enabled for this system. will appear after enabling the repo. This will confirm that the repo has been enabled.

Next simply download the relevant .rpm package for installation. This package can be acquired from morpheushub.com.

Tip: Use the wget command to directly download the package to your appliance server. i.e. `wget https://downloads.morpheusdata.com/path/to/package.rpm`

Next we must install the package onto the machine and configure the morpheus services:

```
sudo rpm -i morpheus-appliance_x.x.x-1_amd64.rpm  
sudo morpheus-ctl reconfigure
```

Once the installation is complete the web interface will automatically start up. By default it will be resolvable at `https://your_machine_name` and in many cases this may not be resolvable from your browser. The url can be changed by editing `/etc/morpheus/morpheus.rb` and changing the value of `appliance_url`. After this has been changed simply run:

```
sudo morpheus-ctl reconfigure  
sudo morpheus-ctl stop morpheus-ui  
sudo morpheus-ctl start morpheus-ui
```

Note: The `morpheus-ui` can take 2-3 minutes to startup before it becomes available.

There are additional install settings that can be viewed in the *Additional Configuration Options* section.

Once the browser is pointed to the appliance a first time setup wizard will be presented. Please follow the on screen instructions by creating the master account. From there you will be presented with the license settings page where a license can be applied for use (if a license is required you may request one or purchase one by contacting your sales representative).

More details on setting up infrastructure can be found throughout this guide.

Tip: If any issues occur it may be prudent to check the morpheus log for details at `/var/log/morpheus/morpheus-ui/current`.

HA Installation Overview

Morpheus provides a wide array of options when it comes to deployment architectures. It can start as a simple one machine instance where all services run on the same machine, or it can be split off into individual services per machine and configured in a high availability (HA) configuration, either in the same region or cross-region. Naturally, high availability can grow more complicated, depending on the configuration you want to do and this article will cover the basic concepts of the Morpheus HA architecture that can be used in a wide array of configurations.

There are four primary tiers of services represented within the Morpheus appliance. They are the Application Tier, Transactional Database Tier, Non-Transactional Database Tier, and Messaging Tier. Each of these tiers have their own recommendations for High availability deployments.

Important: This is a sample configuration only. Customer configurations and requirements will vary. Please contact your account manager if you wish to deploy or transition to a HA environment.

Application Tier

The application tier is easily installed with the same apt or yum repository package that Morpheus is normally distributed with. Advanced configuration allows for the additional tiers to be skipped and leave only the “stateless” services that need run. These stateless services include Nginx and Tomcat. They can be configured across all regions and placed behind a central load-balancer or geo-based load balancer. They typically connect to all other tiers as none of the other tiers talk to each other besides through the central application tier. One final piece when it comes to setting up the Application tier is shared storage, which is necessary when it comes to maintaining deployment archives, virtual image catalogs, backups, etc. These can be externalized to an object storage service such as Amazon S3 or Openstack Swiftstack as well. Alternatively, a simple NFS cluster can also be used to handle the shared storage structure. Alternatively, supported (Platform as a Service) PaaS offerings can be used that provide a NFS compatible shared storage, eliminating the need to manage the underlying infrastructure.

Transactional Database Tier

The Transactional Database tier consists of a MySQL compatible database. It is recommended that a lockable clustered configuration be used, such as a Galera cluster, which can also provide high availability. Alternatively, supported PaaS offerings can be used that provide a mySQL compatible database, eliminating the need to manage the underlying infrastructure.

Non-Transactional Database Tier

The Non-Transactional tier consists of an Elasticsearch cluster. Elastic Search is used for log aggregation data and temporal aggregation data (essentially stats, metrics, and logs). This enables for a high write throughput at scale. ElasticSearch is a Clustered database meaning all nodes no matter the region need to be connected to each other over what they call a “Transport” protocol. It is fairly simple to get setup as all nodes are identical. It is also a java based system and does require a sizable chunk of memory for larger data sets. Alternatively, supported PaaS offerings can be used that provide an Elasticsearch compatible deployment, eliminating the need to manage the underlying infrastructure.

Messaging Tier

The Messaging tier is an AMQP based tier along with STOMP Protocol (used for agent communication). The primary model recommended is to use RabbitMQ for queue services. RabbitMQ is also a cluster-based queuing system and needs at least 3 instances for HA configurations. This is due to elections in the failover scenarios RabbitMQ can manage. If doing a cross-region HA RabbitMQ cluster it is recommended to have at least 3 Rabbit queue clusters per region. Typically to handle HA a RabbitMQ cluster should be placed between a load balancer and the front-end application server to handle cross host connections. Alternatively, supported PaaS offerings can be used that provide a RabbitMQ compatible deployment, eliminating the need to manage the underlying infrastructure.

HA Installation Architectures

3-Node HA (Recommended) In this architecture, all tiers are deployed on three machines by Morpheus during the installation, with the exception of the Transactional Database Tier. This provides HA not just for the Morpheus Application Tier but all underlying tiers that support Morpheus. The Transactional Database Tier will remain external, either as a separate cluster or PaaS, following the supported services. An external Percona cluster must still be set up outside of the Morpheus app nodes.

Distributed HA In this architecture, the tiers do not need to reside on the same machines, each can be hosted by a supported cluster or PaaS offering. This provides flexibility and reuse of already existing technologies such as RabbitMQ or Elasticsearch. Each tier should be architected to provide HA following the vendor’s documentation, to ensure no downtime for the Morpheus Application Tier.

Supported PaaS Offerings

The Morpheus Application Tier can be deployed to any public or private cloud. Morpheus has underlying technologies that support the Application Tier, which can be automatically installed and embedded on the applications nodes. Alternatively, the services can be provided externally and not be embedded. Many cloud providers have Platform as a Service (PaaS) offerings of these underlying technologies but some are not native or don’t provide the performance required. Below is a list of PaaS offerings that are supported to be used for Morpheus:

Table 5: PaaS Offering Support

Cloud	Messaging (RabbitMQ)	Log (Elasticsearch)	Database (mySQL)	Shared Storage (NFS)
AWS	Amazon MQ	Amazon OpenSearch	Amazon Aurora	Elastic File System
GCP	N/A	N/A	mySQL Instance	Cloud Filestore
Azure	N/A	N/A	N/A	Storage Account
OCI	N/A	N/A	N/A	N/A
Alibaba	N/A	N/A	N/A	N/A

5.1.4 Upgrades & Maintenance

Upgrading

Warning: Morpheus v7.0.0 contains new node and VM node packages that require 3.5GB of storage. It is safe to run `sudo rm -Rf /var/opt/morpheus/package-repos/*` after v7.0.0 package installation and before reconfigure to clean old node and vm node packages from the package-repo when room is needed.

Upgrading Overview

Important: Known issue with embedded Elasticsearch upgrade: When upgrading to v5.4.8, v5.4.9 or v5.5.1, there is a potential issue with embedded Elasticsearch clustering on rolling upgrades and existing data migration for all embedded Elasticsearch architectures. Refer to the [v7.0.0 LTS Release Notes](#) for additional information.

Morpheus Packages

Morpheus Release Package urls can be obtained from <https://morpheushub.com>

Upgrade Requirements

Warning: Morpheus v7.0.0 contains new node and VM node packages that require 3.5GB of storage. It is safe to run `sudo rm -Rf /var/opt/morpheus/package-repos/*` after v7.0.0 package installation and before reconfigure to clean old node and VM node packages from the package-repo when appliance free space is needed.

Important: BACKUP YOUR DATABASE before the upgrade! You can use the [appliance backup job in Morpheus](#), then [rollback and restore your appliance](#) if needed. Make sure you download the backup before you do the upgrade!

- For firewall/proxy/acl considerations, the domain for Appliance, Supplemental and Agent packages was changed recently to <https://downloads.morpheusdata.com> from <https://downloads.gomorpheus.com>. Please update ACL's to allow access to <https://downloads.morpheusdata.com> when necessary.

v7.0.0 Compatibility & Breaking Changes

When installing and upgrading to Morpheus v7.0.0, refer to the following to ensure compatibility.

Breaking Changes

- 6.3.0: Version 6.3.0 is the first version to require Plugin API 1.0.0+. Small changes will need to be made in order to make plugins created for prior versions of Morpheus compatible with 6.3.0+. See the [related article in our KnowledgeBase](#) on the small changes that will need to be made to ensure plugin compatibility
- 6.2.2, 6.0.7 v7.0.0 contains embedded MySQL v8 upgrade when upgrading from v6.0.0 - v6.0.6 or 6.1.0 - 6.2.1. BACKUP YOUR DATABASE PRIOR TO UPGRADE when using embedded MySQL (all-in-one appliances)
- 6.2.2, 6.0.7 Minimum v6.x required to upgrade to v7.0.0 for environments using embedded RabbitMQ. Environments running 5.5.x or earlier using embedded RabbitMQ must upgrade to v6.0.0 - v6.0.6, or 6.1.0 - 6.2.1 prior to upgrading to v7.0.0
- 6.2.2, 6.0.7 Rolling upgrades for HA environments using embedded RabbitMQ and/or embedded Elasticsearch services are not supported when upgrading from v6.0.0 - v6.0.6 or 6.1.0 - 6.2.1
- 6.1.1 contains database datatype modifications on account_invoice and account_invoice_item that may cause long initial ui start up times while the modifications are ran in MySQL for environments with over 100k invoice records when upgrading from any version other than 6.0.3
- 6.1.1 relocates the embedded plugin folder and remove the previous folder. For HA environments using shared storage, rolling upgrades from any version other than 6.0.3 are not advised as the embedded plugins will not be found on non-upgraded nodes after one node is upgraded.
- 6.1.1: NSX-V networking integration support is removed and no longer supported as of Morpheus 6.1.1
- 6.0.3 contains database datatype modifications on account_invoice and account_invoice_item that may cause long initial ui start up times while the modifications are ran in MySQL for environments with over 100k invoice records.
- 6.0.3 relocates the embedded plugin folder and remove the previous folder. For HA environments using shared storage, rolling upgrades are not advised as the embedded plugins will not be found on non-upgraded nodes after one node is upgraded.
- 6.0.0: NSX-V support is deprecated though still supported as of Morpheus 6.0.0. It will be removed and unsupported in 6.1.1 and higher.
- 6.0.0+: In Morpheus 6.0.0+, many third party integrations have been moved out of the core installer package and converted to Morpheus plugins. As a result, during the upgrade process your appliance will need to be able to access share.morpheusdata.com, the online repository for all Morpheus plugins. Where this is not possible, users may instead apply the supplemental installer package which is also available at Morpheus Hub alongside the main installer package.
- 6.0.0+: In Morpheus 6.0.0+, older service specific system provided Instance Types and Layouts were deprecated and disabled. Updating to 6.0.0 will not affect existing Instances that are associated with the disabled types, however existing catalog item configurations, blueprints and api requests that use disabled Instance Types and layouts will need to be updated.
- 5.5.2: VM Node Packages: Due to build java version requirements, the i386.deb and i386.rpm (32-bit) VM Node Packages can no longer be updated, and remain on v3.2.9.
- 5.4.12: Guacd: Guacd is now compiled with libssh2-1.10.0 on all platforms. Appliances on SLES15 may need openssl-devel manually installed for guacd to successfully compile.
- 5.4.12: Session Manager: Morpheus features a new session manager that was necessary in order to resolve expiring connections from the agents due to a Spring framework update. This new session manager no longer requires Sticky Sessions and they can now be turned off at the load balancer if so desired. However, keeping them on is totally reasonable as well as it reduces overall system load. Rolling restarts no longer kick you out of your session if sticky sessions are off as it distributes your session data across the morpheus nodes in an HA environment. Additionally, overall system load is reduced as a result of the new session manager.

- 5.4.9: Morpheus 5.4.9 adds the “Provisioning: State” Role permission. This permission determines access to the State tab for Terraform-backed Instances and is set to “None” by default. On upgrade, only System Admin users will be able to see the State tab for these Instances. For other users who should have this access, edit their Roles to include “Provisioning: State” permissions.
- 5.4.5: Warning: Database indexes added for account_usage and metadata_tag tables. Customers with very large account_usage and/or metadata_tag tables (10 million+) may experience slower initial morpheus-ui loading time after upgrading to 5.4.5, as well as additional database load.
- 5.4.5: ‘AVI Load Balancer’ renamed to ‘NSX Advanced Load Balancer’
- 5.4.5: Cloud Types disabled by default: Dell, HPE (NOT HPE Oneview), Supermicro and Cloud Foundry. Users would still be able to re-enable these clouds in the appliance settings. Does not affect existing Clouds.
- 5.4.5: A10 Load Balancer type has been disabled, and will no longer be an option when adding new Load Balancers. Contact Morpheus if you need to re-enable A10 Load Balancer option. This does not affect existing Load Balancers.
- 5.4.5: Morpheus Cluster type “Combo Cluster” renamed to “KVM/Docker Cluster”
- 5.4.5: Greenfield managed vm’s (provisioned with Morpheus) can no longer be deleted in Morpheus without removing the actual vm/infrastructure. Restriction does not apply to brownfield vm’s that have been converted to managed.
- 5.4.4: The Venafi and AppDynamics integrations are deprecated in v5.4.4 and will be removed in v5.4.5. App-Dynamic will return as a plugin at a later date.
- 5.4.4: The morpheus-ui logging configuration file has changed from logback.groovy to logback.xml in v5.4.4 (/opt/morpheus/conf/logback.xml). The logback.groovy file from previous versions can be removed, and any updates to logback.groovy will not result in any logging configuration changes.
- 5.4.3: vCloud Director: Support for integrations with vCD 9 ended
- 5.4.3: Morpheus Worker/Gateway v5.4.3 packages are now available. Existing Worker & Gateway nodes must be upgraded to v5.4.3 for compatibility with Morpheus v5.4.3 Appliances.
- 5.4.2: vCloud Director: vCD 9.x will no longer be supported by Morpheus
- 5.4.2: ServiceNow: Instance and Blueprint specific exposures will be removed from ServiceNow plugin support. More advanced configurations of Instances and Blueprints, in addition to Workflows, can be exposed utilizing Catalog Items
- 5.4.2: After upgrading, it is recommended that you manually perform one “Daily” refresh Amazon Clouds to ensure availability of Amazon Service Plans for each region. To manually refresh a Cloud, navigate to Infrastructure > Clouds > (Selected Amazon Cloud) and select “Daily” from the REFRESH dropdown menu. If this is not done, Morpheus may not show Amazon Service Plans in the provisioning wizard until after Midnight UTC following the upgrade when the next automatic Daily sync would run.
- 5.3.4: Major UI navigation structure changes. Refer to the Navigation Updates reference table
- 5.3.3: Support for OpenStack v2 Identity API is removed
- 5.3.2+: The local code repository path moved from /var/opt/morpheus/morpheus-ui/repo to /var/opt/morpheus/morpheus-local/repo to reduce potential shared storage issues and performance restrictions. The reconfigure process creates the folders and sets the paths in application.yml, no manual intervention is needed unless symlinks exist on /var/opt/morpheus/morpheus-ui/repo/git which will need to be removed prior to reconfiguring - 5.3.2+ The deprecated /var/opt/morpheus/morpheus-ui/repo path will be automatically deleted in a future release but can be manually recursively deleted at any time for storage reclamation.
- 5.3.2+: *Provisioning* → *Deployments* has been moved to *Provisioning* → *Code* → *Deployments*
- 5.2.9: OpenStack v2 Identity API is deprecated as of v5.2.9 (and is removed as of v5.3.3)

- 5.2.6, 5.3.1: Appliance & Agent java version updated to 8u292-b10. jdk8u292 disables TLS 1.0 and 1.1 by default
- 5.2.3+: codeready (codeready-builder-for-rhel-8-x86_64-rpms) repo access required for RHEL 8+ Appliances, replacing the previous PowerTools/powertools requirement
- 5.2.1 & 4.2.5: API: Metadata: Metadata tags now referred to as `tags` and labels now referred to as `labels`. Previously metadata tags were referred to as `metadata` and labels were referred to as `tags`
- 5.0.0+: When upgrading to 5.0.0+ from 4.x.x, any bearer tokens that have been generated are deleted which requires users to request new bearer tokens
- 4.2.4: For appliances with externalized MySQL databases, due to MySQL deprecation of the “EDT” timezone you may need to update your database timezone to UTC or another compatible value. If this is not done, you will receive errors referencing timezone and Morpheus will not start. Morpheus should handle this change automatically for all-in-one appliances.
- 4.2.1+: Tasks: Python: Virtual environment are now used for Python Tasks. **Note:** `virtualenv` is required on all Appliance App nodes
- 4.2.1+: Puppet: Morpheus integration now supports version 6+. Puppet versions prior to 6 are no longer supported
- 4.2.1+: Clouds: VirtualBox, VirtuSteam, and MetaCloud Cloud Types are no longer supported or available
- 4.2.1+: Appliance: OS: Ubuntu 14.04 has reached its end of life (EOL) and is no longer supported as a Morpheus Appliance Host Operating System. Any Morpheus Appliance running on 14.04 must be upgraded to 16.04, 18.04, 20.04 or 22.04 BEFORE upgrading to 4.2.1+. Upgrades on 14.04 will not succeed

Morpheus Application OS

Morpheus can be installed on the following platforms. Please note the table below is for Morpheus Application OS support, not Morpheus Agent OS Support.

Note: If CentOS 8.2 is pinned to 8.2.2004 vault, the PowerTools repository will need to be pinned to 8.2.2004 to access freerdp-libs 2.0.0

Table 6: Supported Appliance Operating Systems

OS	Version(s)	Notes
Amazon Linux	2	
CentOS	7.x, 8.x (stream) 9.x (stream)	
Debian	10, 11	
RHEL	7.x, 8.x, 9.x	
Oracle Enterprise Linux (OEL)	7.x, 8.x	
SUSE SLES	12, 15	
Ubuntu	18.04, 20.04, 22.04	14.04 is no longer supported for Appliance OS. Note: 14.04 is still supported by the Morpheus Agent.

Services

v7.0.0 Service Versions & Compatibility

Table 7: v7.0.0 Service Versions & Compatibility

Service	Compatible Branch	Morpheus Installer Version	Updated in v7.0.0
Plugin API	1.1.0	1.1.0	
Morpheus Worker	5.4.8+		
MySQL	v5.7, v8.0	v8.0.35	
MySQL (FIPS)	v5.7, v8.0	v8.0.34	
Elasticsearch	v8.11+	v8.11.2	
RabbitMQ	v3.5-3.12	v3.12.9	
Tomcat		v9.0.83	
Nginx		v1.25.1	
OpenSSL		1.1.1t, 1.0.2u (FIPS)	
Java		11.0.20+8	
Java (macOS agent)		11.0.14+9	

Important: Elasticsearch 8.10.1 was released within days of Morpheus v7.0.0. It's expected to be compatible but has not been tested. Customers managing their own Elasticsearch clusters should be aware that internal testing of the Morpheus v7.0.0 was conducted on Elasticsearch 8.9.2 and we cannot recommend deploying a higher version at this time.

Morpheus Agent & Node Package Versions

Table 8: v7.0.0 Agent & Node Package Versions

Package	Version	v7.0.0 changes from v6.3.4
Morpheus Node and VM Node Packages	3.2.22	Node and VM Node Packages update to 3.2.22 with v2.5.4 Linux Agent
Morpheus Linux Agent	v2.5.4	Morpheus Linux Agent updated to v2.5.4
Morpheus Windows Agent	v2.5.0.0	Morpheus Windows Agent updated to v2.5.0.0
Morpheus macOS Agent	v2.4.0	No changes

Security

Security Advisories

Advisory ID	Severity	Description	Updated On
MOR20220721-01	Critical	Morpheus through 5.4.3 (which run Java 8) are confirmed to be impacted, Morpheus through 5.5.1-1 (for customers on 5.5.x Standard installations) and 5.4.8-2 (for customers on 5.4.x LTS installations) are potentially impacted if the vulnerability is found on Java 11.	07-21-2022
MOR20220524-01	High	An XXE issue was discovered in Morpheus through 5.2.16 and 5.4.x through 5.4.4. A successful attack requires a SAML identity provider to be configured.	06-08-2022

Upgrade Paths & Methods

The following table shows supported version upgrade paths and methods.

* Some Features and Fixes in the From version may not be included in the To version due to From version being released after the To version.

Integrations

Note: Current iterations of Amazon AWS, Microsoft Azure, Google Cloud Platform, Digital Ocean, OpenTelekom Cloud, IBM Bluemix, Softlayer and UpCloud are all supported.

Table 9: Integration Compatibility

Integration	Supported Version(s)	Notes
Ansible	2.7.x or higher	
Ansible Tower	3.8.x	
App Dynamics	4.5.x	
Ava-mar	18	
Azure Stack	2002 back to 1908	2019-03-01-hybrid api-profile version used which is supported in 1908 and later Azure Stack versions
Cisco ACI	3.x,4.x,5.x	
Citrix Netscaler	v12.1	
Comm-vault	v11 sp 19	
F5 Big-IP	11.4+	
In-foblox	Latest Versions Supported	
Kubernetes	1.21+	
Microsoft Hyper-V	2012R2, 2016, 2019	
Nutanix AHV	5.0+	In 5.5 - 5.7 if Prism Central is managing Prism Element, image creation will not function due to PC Image Management.
Openstack	Latest Versions Supported	When creating an OpenStack integration, select the latest available from the OS Version dropdown menu when running a later version
Puppet	6+	Puppet Agent version will be latest 6 version from yum.puppetlabs.com or apt.puppetlabs.com
Rubrik	Up to 7.x	
SCVMM	2016, 2019	
ServiceNow	Per servicenow plugin version	Refer to the servicenow store for Morpheus Catalog compatibility
Terraform	Latest	When Morpheus is handling Terraform installation, the version used will be 1.5.5 for licensing reasons. If a later version is required, you will need to manually handle Terraform installation. See the “Terraform Installation” section of Morpheus docs for more details.
vCloud Director	10.0, 10.2, 10.3, 10.4	When upgrading a vCD environment, you should update the API Version setting on the Morpheus Cloud configuration first

Note: Non-listed versions may be compatible but are not verified.

Single Node/AIO Appliance Upgrades

Important: Only appliances running Morpheus v6.0.0 or higher can upgrade to v7.0.0. Always backup your database before running any upgrade.

The following covers upgrading Single Node or AIO (All-In-One) Appliance configurations.

Debian / Ubuntu

Warning: Morpheus v7.0.0 contains new node and VM node packages that require 3.5GB of storage. It is safe to run `sudo rm -rf /var/opt/morpheus/package-repos/*` after v7.0.0 package installation and before reconfigure to clean old node and vm node packages from the package-repo when room is needed.

To upgrade Morpheus running on Ubuntu/Debian, download new deb package, stop the morpheus-ui, install the new deb package, then reconfigure.

```
sudo wget https://packageUrl.morpheus-appliance_x.x.x-x_amd64.deb
sudo morpheus-ctl stop morpheus-ui
sudo dpkg -i morpheus-appliance_x.x.x-1_amd64.deb
sudo morpheus-ctl reconfigure
```

All services will automatically start during the reconfigure process. After the reconfigure has succeeded, tail the ui service to watch ui startup logs with `morpheus-ctl tail morpheus-ui`.

Note: Services will be stopped during package installation and started during the reconfigure process, including the morpheus-ui service. If the reconfigure process is interrupted or fails, the morpheus-ui service may need to be manually started or restarted. In certain situations if another service hangs on starting during reconfigure, run `systemctl restart morpheus-runsrvdir` then reconfigure and restart morpheus-ui if successful.

After the morpheus-ui service finishes loading, the upgrade is complete.

CentOS / RHEL / Amazon / SLES

Warning: Morpheus v7.0.0 contains new node and VM node packages that require 3.5GB of storage. It is safe to run `sudo rm -rf /var/opt/morpheus/package-repos/*` after v7.0.0 package installation and before reconfigure to clean old node and vm node packages from the package-repo when room is needed.

To upgrade Morpheus running on CentOS, RHEL, Amazon or SLES, download and install the new rpm package, stop the morpheus-ui, reconfigure and then start the morpheus-ui:

```
sudo wget https://packageUrl.morpheus-appliance-x.x.x-x.x86_64.rpm
sudo morpheus-ctl stop morpheus-ui
sudo rpm -Uhv morpheus-appliance-x.x.x-x.x86_64.rpm
sudo morpheus-ctl reconfigure
```

All services will automatically start during the reconfigure process. After the reconfigure has succeeded, tail the ui service to watch ui startup logs with `morpheus-ctl tail morpheus-ui`.

Note: Services will be stopped during package installation and started during the reconfigure process, including the morpheus-ui service. If the reconfigure process is interrupted or fails, the morpheus-ui service may need to be manually started or restarted. In certain situations if another service hangs on starting during reconfigure, run `systemctl restart morpheus-runsvdir` then reconfigure and restart morpheus-ui if successful.

After the morpheus-ui service finishes loading, the upgrade is complete.

3-Node HA Upgrade

3-Node HA Appliances represent 3 App nodes with local RabbitMQ and Elasticsearch services clustered across the app nodes, and an external Galera/Percona MySQL cluster.

Morpheus Packages

Morpheus Release Package urls can be obtained from <https://morpheushub.com>

Refer to *v7.0.0 Compatibility & Breaking Changes* for any 3-node variations using externalized MySQL, Elasticsearch and/or RabbitMQ version requirements.

Upgrade Instructions

3-Node HA Debian / Ubuntu Upgrade

The following covers upgrading the Morpheus App nodes in 3 Node HA configurations to v7.0.0.

Warning: As a best practice, always backup your database prior to any upgrade.

Important: The following is only for “3 Node HA” Architecture configurations.

Morpheus Packages

Morpheus Release Package urls can be obtained from <https://morpheushub.com>

v6.0.6 or lower -> v7.0.0 Upgrade

Warning: Rolling upgrades from v6.0.6 or lower to v7.0.0 are not supported

Important: It is important to stop the morpheus-ui service on all app nodes prior to upgrade. Failure to do so will result in a flood of log errors due to previous message serializaiton conflict. The messages will eventually expire and the logs will clear.

Warning: Morpheus v7.0.0 contains new node and VM node packages that require 3.5GB of storage. It is safe to run `sudo rm -Rf /var/opt/morpheus/package-repos/*` after v7.0.0 package installation and before reconfigure to clean old node and vm node packages from the package-repo when room is needed.

1. Starting with Node 3, on **All App Nodes**, stop the morpheus-ui services via `morpheus-ctl stop morpheus-ui`. If you receive a timeout, run `morpheus-ctl graceful-kill morpheus-ui`.

```
[root@app-server-3 ~]# morpheus-ctl stop morpheus-ui
```

```
[root@app-server-2 ~]# morpheus-ctl stop morpheus-ui
```

```
[root@app-server-1 ~]# morpheus-ctl stop morpheus-ui
```

2. Upgrade the deb package on Node 1, then run a Reconfigure on Node 1

```
[root@app-server-1 ~]# sudo wget https://packageUrl.morpheus-appliance_x.x.x-x_
↪amd64.deb
[root@app-server-1 ~]# sudo dpkg -i morpheus-appliance_x.x.x-1_amd64.deb
[root@app-server-1 ~]# sudo morpheus-ctl reconfigure
```

Note: All services will automatically be stopped and started during the reconfigure process. After the reconfigure has succeeded, tail the ui service to watch ui startup logs with `morpheus-ctl tail morpheus-ui`.

3. Once Node 1 upgrade has completed and the ui is available, upgrade the deb package on Node 2, then run a Reconfigure on Node 2.

```
[root@app-server-2 ~]# sudo wget https://packageUrl.morpheus-appliance_x.x.x-x_
↪amd64.deb
[root@app-server-2 ~]# sudo dpkg -i morpheus-appliance_x.x.x-1_amd64.deb
[root@app-server-2 ~]# sudo morpheus-ctl reconfigure
```

4. Then upgrade the deb package on Node 3, and run a Reconfigure on Node 3

```
[root@app-server-3 ~]# sudo wget https://packageUrl.morpheus-appliance_x.x.x-x_
↪amd64.deb
[root@app-server-3 ~]# sudo dpkg -i morpheus-appliance_x.x.x-1_amd64.deb
[root@app-server-3 ~]# sudo morpheus-ctl reconfigure
```

5. The upgrade is complete and the Morpheus-ui services should be running with clustered Elasticsearch and RabbitMQ services across the 3 nodes.

Important: If reconfigure after a rpm package upgrade stalls or hangs on starting a service (mysql, rabbitmq, elasticsearch ...) it is possible the `morpheus-runsvdir` service did not start or a process it was managing was manually shutdown or killed. To resolve, run `systemctl stop morpheus-runsvdir` then `systemctl start morpheus-runsvdir`, then run reconfigure again, `morpheus-ctl reconfigure`.

v6.0.7 -> v7.0.0 Upgrade

Note: Rolling upgrades are supported for v6.0.7 -> v7.0.0 only.

Warning: Morpheus v7.0.0 contains new node and VM node packages that require 3.5GB of storage. It is safe to run `sudo rm -Rf /var/opt/morpheus/package-repos/*` after v7.0.0 package installation and before reconfigure to clean old node and vm node packages from the package-repo when room is needed.

1. Upgrade the deb package on Node 1, then run a Reconfigure on Node 1

```
[root@app-server-1 ~]# sudo wget https://packageUrl.morpheus-appliance_x.x.x-x_
↪amd64.deb
[root@app-server-1 ~]# sudo dpkg -i morpheus-appliance_x.x.x-1_amd64.deb
[root@app-server-1 ~]# sudo morpheus-ctl stop morpheus-ui
[root@app-server-1 ~]# sudo morpheus-ctl reconfigure
[root@app-server-1 ~]# sudo morpheus-ctl start morpheus-ui
```

After the reconfigure has succeeded, tail the ui service to watch ui startup logs with `morpheus-ctl tail morpheus-ui`. Once morpheus-ui is started, proceed to the next node.

2. Once Node 1 upgrade has completed and the ui is available, upgrade the deb package on Node 2, then run a Reconfigure on Node 2.

```
[root@app-server-2 ~]# sudo wget https://packageUrl.morpheus-appliance_x.x.x-x_
↪amd64.deb
[root@app-server-2 ~]# sudo dpkg -i morpheus-appliance_x.x.x-1_amd64.deb
[root@app-server-2 ~]# sudo morpheus-ctl stop morpheus-ui
[root@app-server-2 ~]# sudo morpheus-ctl reconfigure
[root@app-server-2 ~]# sudo morpheus-ctl start morpheus-ui
```

After the reconfigure has succeeded, tail the ui service to watch ui startup logs with `morpheus-ctl tail morpheus-ui`. Once morpheus-ui is started, proceed to the next node.

3. Once Node 2 upgrade has completed and the ui is available, upgrade the deb package on Node 3, and run a Reconfigure on Node 3

```
[root@app-server-3 ~]# sudo wget https://packageUrl.morpheus-appliance_x.x.x-x_
↪amd64.deb
[root@app-server-3 ~]# sudo dpkg -i morpheus-appliance_x.x.x-1_amd64.deb
[root@app-server-3 ~]# sudo morpheus-ctl stop morpheus-ui
[root@app-server-3 ~]# sudo morpheus-ctl reconfigure
[root@app-server-3 ~]# sudo morpheus-ctl start morpheus-ui
```

After the reconfigure has succeeded, tail the ui service to watch ui startup logs with `morpheus-ctl tail morpheus-ui`.

4. The upgrade is complete and the Morpheus-ui services should be running with clustered Elasticsearch and RabbitMQ services across the 3 nodes.

Important: If reconfigure after a rpm package upgrade stalls or hangs on starting a service (mysql, rabbitmq, elasticsearch ...) it is possible the `morpheus-runsvdir` service did not start or a process it was managing was manually shutdown or killed. To resolve, run `systemctl stop morpheus-runsvdir` then `systemctl start morpheus-runsvdir`, then run reconfigure again, `morpheus-ctl reconfigure`.

3-Node HA CentOS / RHEL Upgrade

The following covers upgrading the Morpheus App nodes in 3 Node HA configurations to v7.0.0.

Warning: As a best practice, always backup your database prior to any upgrade.

Important: The following is only for “3 Node HA” Architecture configurations.

Morpheus Packages

Morpheus Release Package urls can be obtained from <https://morpheushub.com>

v6.0.6 or lower -> v7.0.0 Upgrade

Warning: Rolling upgrades from v6.0.6 or lower to v7.0.0 are not supported

Important: It is important to stop the morpheus-ui service on all app nodes prior to upgrade. Failure to do so will result in a flood of log errors due to previous message serializaiton conflict. The messages will eventually expire and the logs will clear.

Warning: Morpheus v7.0.0 contains new node and VM node packages that require 3.5GB of storage. It is safe to run `sudo rm -Rf /var/opt/morpheus/package-repos/*` after v7.0.0 package installation and before reconfigure to clean old node and vm node packages from the package-repo when room is needed.

1. Starting with Node 3, on **All** App Nodes, stop the morpheus-ui services via `morpheus-ctl stop morpheus-ui`. If you receive a timeout, run `morpheus-ctl graceful-kill morpheus-ui`.

```
[root@app-server-3 ~]# morpheus-ctl stop morpheus-ui
```

```
[root@app-server-2 ~]# morpheus-ctl stop morpheus-ui
```

```
[root@app-server-1 ~]# morpheus-ctl stop morpheus-ui
```

2. Upgrade the RPM package on Node 1, then run a Reconfigure on Node 1

```
[root@app-server-1 ~]# sudo wget https://packageUrl.morpheus-appliance-x.x.x-x.x86_
↪64.rpm
[root@app-server-1 ~]# sudo rpm -Uhv morpheus-appliance-x.x.x-x.x86_64.rpm
[root@app-server-1 ~]# sudo morpheus-ctl reconfigure
```

Note: All services will automatically be stopped and started during the reconfigure process. After the reconfigure has succeeded, tail the ui service to watch ui startup logs with `morpheus-ctl tail morpheus-ui`.

3. Once Node 1 upgrade has completed and the ui is available, upgrade the RPM package on Node 2, then run a Reconfigure on Node 2.

```
[root@app-server-2 ~]# sudo wget https://packageUrl.morpheus-appliance-x.x.x-x.x86_
↪64.rpm
[root@app-server-2 ~]# sudo rpm -Uhv morpheus-appliance-x.x.x-x.x86_64.rpm
[root@app-server-2 ~]# sudo morpheus-ctl reconfigure
```

4. Then upgrade the RPM package on Node 3, then run a Reconfigure on Node 3

```
[root@app-server-3 ~]# sudo wget https://packageUrl.morpheus-appliance-x.x.x-x.x86_
↪64.rpm
[root@app-server-3 ~]# sudo rpm -Uhv morpheus-appliance-x.x.x-x.x86_64.rpm
[root@app-server-3 ~]# sudo morpheus-ctl reconfigure
```

- The upgrade is complete and the Morpheus-ui services should be running with clustered Elasticsearch and RabbitMQ services across the 3 nodes.

Important: If reconfigure after a rpm package upgrade stalls or hangs on starting a service (mysql, rabbitmq, elasticsearch ...) it is possible the `morpheus-runsvdir` service did not start or a process it was managing was manually shutdown or killed. To resolve, run `systemctl stop morpheus-runsvdir` then `systemctl start morpheus-runsvdir`, then run reconfigure again, `morpheus-ctl reconfigure`.

v6.0.7 -> v7.0.0 Upgrade

Note: Rolling upgrades are supported for v6.0.7 -> v7.0.0 only.

Warning: Morpheus v7.0.0 contains new node and VM node packages that require 3.5GB of storage. It is safe to run `sudo rm -Rf /var/opt/morpheus/package-repos/*` after v7.0.0 package installation and before reconfigure to clean old node and vm node packages from the package-repo when room is needed.

- Upgrade the RPM package on Node 1, then run a Reconfigure on Node 1

```
[root@app-server-1 ~]# sudo wget https://packageUrl.morpheus-appliance-x.x.x-x.x86_
↪64.rpm
[root@app-server-1 ~]# sudo rpm -Uhv morpheus-appliance-x.x.x-x.x86_64.rpm
[root@app-server-1 ~]# sudo morpheus-ctl stop morpheus-ui
[root@app-server-1 ~]# sudo morpheus-ctl reconfigure
[root@app-server-1 ~]# sudo morpheus-ctl start morpheus-ui
```

After the reconfigure has succeeded, tail the ui service to watch ui startup logs with `morpheus-ctl tail morpheus-ui`. Once morpheus-ui is started, proceed to the next node.

- Once Node 1 upgrade has completed and the u is available, upgrade the RPM package on Node 2, then run a Reconfigure on Node 2.

```
[root@app-server-2 ~]# sudo wget https://packageUrl.morpheus-appliance-x.x.x-x.x86_
↪64.rpm
[root@app-server-2 ~]# sudo rpm -Uhv morpheus-appliance-x.x.x-x.x86_64.rpm
[root@app-server-2 ~]# sudo morpheus-ctl stop morpheus-ui
[root@app-server-2 ~]# sudo morpheus-ctl reconfigure
[root@app-server-2 ~]# sudo morpheus-ctl start morpheus-ui
```

After the reconfigure has succeeded, tail the ui service to watch ui startup logs with `morpheus-ctl tail morpheus-ui`. Once morpheus-ui is started, proceed to the next node.

3. Then upgrade the RPM package on Node 3, then run a Reconfigure on Node 3

```
[root@app-server-3 ~]# sudo wget https://packageUrl.morpheus-appliance-x.x.x-x.x86_
↪64.rpm
[root@app-server-3 ~]# sudo rpm -Uhv morpheus-appliance-x.x.x-x.x86_64.rpm
[root@app-server-3 ~]# sudo morpheus-ctl stop morpheus-ui
[root@app-server-3 ~]# sudo morpheus-ctl reconfigure
[root@app-server-3 ~]# sudo morpheus-ctl start morpheus-ui
```

After the reconfigure has succeeded, tail the ui service to watch ui startup logs with `morpheus-ctl tail morpheus-ui`. Once morpheus-ui is started, proceed to the next node.

4. The upgrade is complete and the Morpheus-ui services should be running with clustered Elasticsearch and RabbitMQ services across the 3 nodes.

Important: If reconfigure after a rpm package upgrade stalls or hangs on starting a service (mysql, rabbitmq, elasticsearch ...) it is possible the `morpheus-runsvdir` service did not start or a process it was managing was manually shutdown or killed. To resolve, run `systemctl stop morpheus-runsvdir` then `systemctl start morpheus-runsvdir`, then run reconfigure again, `morpheus-ctl reconfigure`.

Warning: Upgrades can add additional storage load on database nodes. Please refer to database storage requirements within `3nodeinstall` before upgrading.

Full HA Upgrade

Full HA configurations represent multiple app nodes with external (non-system) MySQL, RabbitMQ and Elasticsearch Clusters or Services.

Morpheus Packages

Morpheus Release Package urls can be obtained from <https://morpheushub.com>

Overview

When upgrading to v7.0.0 only services local to the Morpheus App node(s) will be upgraded. For fully distributed configurations (Full HA) where MySQL, RabbitMQ and Elasticsearch are external, the upgrade process will not upgrade these services.

Refer to *v7.0.0 Compatibility & Breaking Changes* for externalized MySQL, Elasticsearch and/or RabbitMQ version requirements.

Upgrade Instructions

Full HA Debian / Ubuntu Upgrade

The following covers upgrading the Morpheus App nodes in Full HA Architecture configurations to v7.0.0.

Important: The following is only for Full HA Architecture configurations, where MySQL, Elasticsearch and RabbitMQ services are external to the App nodes. The following steps assume 3 app nodes, adjust accordingly.

Morpheus Packages

Morpheus Release Package urls can be obtained from <https://morpheushub.com>

v6.0.6 or lower -> v7.0.0 Upgrade

Warning: Rolling upgrades from v6.0.6 or lower to v7.0.0 are not supported

Important: It is important to stop the morpheus-ui service on all app nodes prior to upgrade. Failure to do so will result in a flood of log errors due to previous message serializaiton conflict. The messages will eventually expire and the logs will clear.

Warning: Morpheus v7.0.0 contains new node and VM node packages that require 3.5GB of storage. It is safe to run `sudo rm -Rf /var/opt/morpheus/package-repos/*` after v7.0.0 package installation and before reconfigure to clean old node and vm node packages from the package-repo when room is needed.

1. Starting with App Node 3, on **All** App Nodes, stop the morpheus-ui services via `morpheus-ctl stop morpheus-ui`. If you receive a timeout, run `morpheus-ctl graceful-kill morpheus-ui`.

```
[root@app-server-3 ~]# morpheus-ctl stop morpheus-ui
```

```
[root@app-server-2 ~]# morpheus-ctl stop morpheus-ui
```

```
[root@app-server-1 ~]# morpheus-ctl stop morpheus-ui
```

2. Upgrade the deb package on App Node 1, then run a Reconfigure on App Node 1

```
[root@app-server-1 ~]# sudo wget https://packageUrl.morpheus-appliance_x.x.x-x_
↔amd64.deb
[root@app-server-1 ~]# sudo dpkg -i morpheus-appliance_x.x.x-1_amd64.deb
[root@app-server-1 ~]# sudo morpheus-ctl stop morpheus-ui
[root@app-server-1 ~]# sudo morpheus-ctl reconfigure
[root@app-server-1 ~]# sudo morpheus-ctl start morpheus-ui
```

After the reconfigure has succeeded, tail the ui service to watch ui startup logs with `morpheus-ctl tail morpheus-ui`.

3. Once App Node 1 upgrade has completed and the ui is available, upgrade the deb package on App Node 2, then run a Reconfigure on App Node 2.

```
[root@app-server-2 ~]# sudo wget https://packageUrl.morpheus-appliance_x.x.x-x_
↳amd64.deb
[root@app-server-2 ~]# sudo dpkg -i morpheus-appliance_x.x.x-1_amd64.deb
[root@app-server-1 ~]# sudo morpheus-ctl stop morpheus-ui
[root@app-server-1 ~]# sudo morpheus-ctl reconfigure
[root@app-server-1 ~]# sudo morpheus-ctl start morpheus-ui
```

After the reconfigure has succeeded, tail the ui service to watch ui startup logs.
↳with ``morpheus-ctl tail morpheus-ui``.

4. Once App Node 2 upgrade has completed and the ui is available, upgrade the deb package on App Node 3, and run a Reconfigure on App Node 3

```
[root@app-server-3 ~]# sudo wget https://packageUrl.morpheus-appliance_x.x.x-x_
↳amd64.deb
[root@app-server-3 ~]# sudo dpkg -i morpheus-appliance_x.x.x-1_amd64.deb
[root@app-server-3 ~]# sudo morpheus-ctl reconfigure
```

5. The upgrade is complete and the Morpheus-ui services should be running on the three allocated nodes.

Important: If reconfigure after a rpm package upgrade stalls or hangs on starting a local service it is possible the morpheus-runsvdir service did not start or a process it was managing was manually shutdown or killed. To resolve, run `systemctl stop morpheus-runsvdir` then `systemctl start morpheus-runsvdir`, then run `reconfigure` again, `morpheus-ctl reconfigure`.

v6.0.7 -> v7.0.0 Upgrade

Note: Rolling upgrades are supported for v6.0.7 -> v7.0.0 only.

Warning: Morpheus v7.0.0 contains new node and VM node packages that require 3.5GB of storage. It is safe to run `sudo rm -Rf /var/opt/morpheus/package-repos/*` after v7.0.0 package installation and before reconfigure to clean old node and vm node packages from the package-repo when room is needed.

1. Starting with App Node 3, on **All** App Nodes, stop the morpheus-ui services via `morpheus-ctl stop morpheus-ui`. If you receive a timeout, run `morpheus-ctl graceful-kill morpheus-ui`.
2. Upgrade the deb package on App Node 1, then run a Reconfigure on App Node 1

```
[root@app-server-1 ~]# sudo wget https://packageUrl.morpheus-appliance_x.x.x-x_
↳amd64.deb
[root@app-server-1 ~]# sudo dpkg -i morpheus-appliance_x.x.x-1_amd64.deb
[root@app-server-1 ~]# sudo morpheus-ctl stop morpheus-ui
```

(continues on next page)

(continued from previous page)

```
[root@app-server-1 ~]# sudo morpheus-ctl reconfigure
[root@app-server-1 ~]# sudo morpheus-ctl start morpheus-ui
```

After the reconfigure has succeeded, tail the ui service to watch ui startup logs with `morpheus-ctl tail morpheus-ui`. Once morpheus-ui is started, proceed to the next node.

3. Once App Node 1 upgrade has completed and the ui is available, upgrade the deb package on App Node 2, then run a Reconfigure on App Node 2.

```
[root@app-server-2 ~]# sudo wget https://packageUrl.morpheus-appliance_x.x.x-x_
↳amd64.deb
[root@app-server-2 ~]# sudo dpkg -i morpheus-appliance_x.x.x-1_amd64.deb
[root@app-server-2 ~]# sudo morpheus-ctl stop morpheus-ui
[root@app-server-2 ~]# sudo morpheus-ctl reconfigure
[root@app-server-2 ~]# sudo morpheus-ctl start morpheus-ui
```

After the reconfigure has succeeded, tail the ui service to watch ui startup logs with `morpheus-ctl tail morpheus-ui`. Once morpheus-ui is started, proceed to the next node.

4. Once App Node 2 upgrade has completed and the ui is available, upgrade the deb package on App Node 3, and run a Reconfigure on App Node 3

```
[root@app-server-3 ~]# sudo wget https://packageUrl.morpheus-appliance_x.x.x-x_
↳amd64.deb
[root@app-server-3 ~]# sudo dpkg -i morpheus-appliance_x.x.x-1_amd64.deb
[root@app-server-3 ~]# sudo morpheus-ctl stop morpheus-ui
[root@app-server-3 ~]# sudo morpheus-ctl reconfigure
[root@app-server-3 ~]# sudo morpheus-ctl start morpheus-ui
```

After the reconfigure has succeeded, tail the ui service to watch ui startup logs with `morpheus-ctl tail morpheus-ui`. Once morpheus-ui is started, proceed to the next node.

5. The upgrade is complete and the Morpheus-ui services should be running on the three allocated nodes.

Important: If reconfigure after a rpm package upgrade stalls or hangs on starting a local service it is possible the `morpheus-runsvdir` service did not start or a process it was managing was manually shutdown or killed. To resolve, run `systemctl stop morpheus-runsvdir` then `systemctl start morpheus-runsvdir`, then run `reconfigure` again, `morpheus-ctl reconfigure`.

Full HA RPM Upgrade

The following covers upgrading the Morpheus App nodes in Full HA Architecture configurations to v7.0.0.

Important: The following is only for Full HA Architecture configurations, where MySQL, Elasticsearch and RabbitMQ services are external to the App nodes.

v6.0.6 or lower -> v7.0.0 Upgrade

Warning: Rolling upgrades from v6.0.6 or lower to v7.0.0 are not supported

Important: It is important to stop the morpheus-ui service on all app nodes prior to upgrade. Failure to do so will result in a flood of log errors due to previous message serializaiton conflict. The messages will eventually expire and the logs will clear.

Warning: Morpheus v7.0.0 contains new node and VM node packages that require 3.5GB of storage. It is safe to run `sudo rm -Rf /var/opt/morpheus/package-repos/*` after v7.0.0 package installation and before reconfigure to clean old node and vm node packages from the package-repo when room is needed.

1. Starting with App Node 3, on **All** App Nodes, stop the morpheus-ui services via `morpheus-ctl stop morpheus-ui`. If you receive a timeout, run `morpheus-ctl graceful-kill morpheus-ui`.

```
[root@app-server-3 ~]# morpheus-ctl stop morpheus-ui
```

```
[root@app-server-2 ~]# morpheus-ctl stop morpheus-ui
```

```
[root@app-server-1 ~]# morpheus-ctl stop morpheus-ui
```

2. Upgrade the RPM package on App Node 1, then run a Reconfigure on App Node 1

```
[root@app-server-1 ~]# sudo wget https://packageUrl.morpheus-appliance-x.x.x-x.x86_
↪64.rpm
[root@app-server-1 ~]# sudo rpm -Uhv morpheus-appliance-x.x.x-x.x86_64.rpm
[root@app-server-1 ~]# sudo morpheus-ctl reconfigure
```

Note: All services will automatically be stopped and started during the reconfigure process. After the reconfigure has succeeded, tail the ui service to watch ui startup logs with `morpheus-ctl tail morpheus-ui`.

3. Once App Node 1 upgrade has completed and the ui is available, upgrade the RPM package on App Node 2, then run a Reconfigure on App Node 2.

```
[root@app-server-2 ~]# sudo wget https://packageUrl.morpheus-appliance-x.x.x-x.x86_
↪64.rpm
[root@app-server-2 ~]# sudo rpm -Uhv morpheus-appliance-x.x.x-x.x86_64.rpm
[root@app-server-2 ~]# sudo morpheus-ctl reconfigure
```

4. Then upgrade the RPM package on App Node 3, then run a Reconfigure on App Node 3

```
[root@app-server-3 ~]# sudo wget https://packageUrl.morpheus-appliance-x.x.x-x.x86_
↪64.rpm
[root@app-server-3 ~]# sudo rpm -Uhv morpheus-appliance-x.x.x-x.x86_64.rpm
[root@app-server-3 ~]# sudo morpheus-ctl reconfigure
```

5. The upgrade is complete and the Morpheus-ui services should be running with clustered Elasticsearch and RabbitMQ services across the 3 nodes.

Important: If reconfigure after a rpm package upgrade stalls or hangs on starting a local service it is possible the morpheus-runsvdir service did not start or a process it was managing was manually shutdown or killed. To resolve, run `systemctl stop morpheus-runsvdir` then `systemctl start morpheus-runsvdir`, then run reconfigure again, `morpheus-ctl reconfigure`.

v6.0.7 -> v7.0.0 Upgrade

Note: Rolling upgrades are supported for v6.0.7 -> v7.0.0 only.

Warning: Morpheus v7.0.0 contains new node and VM node packages that require 3.5GB of storage. It is safe to run `sudo rm -Rf /var/opt/morpheus/package-repos/*` after v7.0.0 package installation and before reconfigure to clean old node and vm node packages from the package-repo when room is needed.

1. Upgrade the RPM package on App Node 1, then run a Reconfigure on App Node 1

```
[root@app-server-1 ~]# sudo wget https://packageUrl.morpheus-appliance-x.x.x-x.x86_
↪64.rpm
[root@app-server-1 ~]# sudo rpm -Uhv morpheus-appliance-x.x.x-x.x86_64.rpm
[root@app-server-1 ~]# sudo morpheus-ctl stop morpheus-ui
[root@app-server-1 ~]# sudo morpheus-ctl reconfigure
[root@app-server-1 ~]# sudo morpheus-ctl start morpheus-ui
```

After the reconfigure has succeeded, tail the ui service to watch ui startup logs with `morpheus-ctl tail morpheus-ui`.

2. Once App Node 1 upgrade has completed and the ui is available, upgrade the RPM package on App Node 2, then run a Reconfigure on App Node 2.

```
[root@app-server-2 ~]# sudo wget https://packageUrl.morpheus-appliance-x.x.x-x.x86_
↪64.rpm
[root@app-server-2 ~]# sudo rpm -Uhv morpheus-appliance-x.x.x-x.x86_64.rpm
[root@app-server-2 ~]# sudo morpheus-ctl stop morpheus-ui
[root@app-server-2 ~]# sudo morpheus-ctl reconfigure
[root@app-server-2 ~]# sudo morpheus-ctl start morpheus-ui
```

After the reconfigure has succeeded, tail the ui service to watch ui startup logs with `morpheus-ctl tail morpheus-ui`.

3. Once App Node 2 upgrade has completed and the u is available, upgrade the RPM package on App Node 3, then run a Reconfigure on App Node 3

```
[root@app-server-3 ~]# sudo wget https://packageUrl.morpheus-appliance-x.x.x-x.x86_
↪64.rpm
[root@app-server-3 ~]# sudo rpm -Uhv morpheus-appliance-x.x.x-x.x86_64.rpm
[root@app-server-3 ~]# sudo morpheus-ctl stop morpheus-ui
```

(continues on next page)

(continued from previous page)

```
[root@app-server-3 ~]# sudo morpheus-ctl reconfigure
[root@app-server-3 ~]# sudo morpheus-ctl start morpheus-ui
```

After the reconfigure has succeeded, tail the ui service to watch ui startup logs with `morpheus-ctl tail morpheus-ui`.

4. The upgrade is complete and the Morpheus-ui services should be running with clustered Elasticsearch and RabbitMQ services across the 3 nodes.

Important: If reconfigure after a rpm package upgrade stalls or hangs on starting a local service it is possible the `morpheus-runsrvdir` service did not start or a process it was managing was manually shutdown or killed. To resolve, run `systemctl stop morpheus-runsrvdir` then `systemctl start morpheus-runsrvdir`, then run reconfigure again, `morpheus-ctl reconfigure`.

Warning: Upgrades can add additional storage load on database nodes. Please refer to database storage requirements within `perconainstall` before upgrading.

morpheus-ctl tips

`morpheus-ctl` is useful beyond reconfigures and starting the ui, and many commands can be run across all services, or scoped to a single service.

Some common commands include:

morpheus-ctl status Lists all the installed services and their current status

morpheus-ctl start (service) This starts all services if no service is specified, or starts the specified service. For example,

- `morpheus-ctl start/stop/restart/kill` on an all-in-one appliance will start, stop, restart or kill `mysql`, `elasticsearch`, `rabbitmq`, `check-server`, `guacd` and the `morpheus-ui`, one by one.
- `morpheus-ctl start/stop/restart/kill morpheus-ui` will only start, stop, restart or kill the `morpheus-ui` service, leaving the other service in their current state. Same goes for `morpheus-ctl start/stop/restart/kill mysql`, `morpheus-ctl start/stop/restart/kill elasticsearch` etc.

`morpheus-ctl` commands:

General Commands:

```
cleanse
  Delete *all* morpheus data, and start from scratch.
help
  Print this help message.
reconfigure
  Reconfigure the application.
show-config
  Show the configuration that would be generated by reconfigure.
uninstall
  Kill all processes and uninstall the process supervisor (data will be preserved).
```

Service Management Commands:

(continues on next page)

(continued from previous page)

```
graceful-kill
  Attempt a graceful stop, then SIGKILL the entire process group.
hup
  Send the services a HUP.
int
  Send the services an INT.
kill
  Send the services a KILL.
once
  Start the services if they are down. Do not restart them if they stop.
restart
  Stop the services if they are running, then start them again.
service-list
  List all the services (enabled services appear with a *.)
start
  Start services if they are down, and restart them if they stop.
status
  Show the status of all the services.
stop
  Stop the services, and do not restart them.
tail
  Watch the service logs of all enabled services.
term
  Send the services a TERM.
```

Elasticsearch Commands:

```
elastic-util
  Backup/Restore ElasticSearch data
```

Firewall Commands:

```
firewall-enable-blocking
  Enables firewall blocking mode.
```

Morpheus DB Migration

If your new installation is part of a migration or you need to move the data from your original Morpheus database, this is easily accomplished by using a stateful dump.

To begin this, stop the Morpheus UI on your original Morpheus server:

```
[root@app-server-old ~] morpheus-ctl stop morpheus-ui
```

Once this is done you can safely export. To access the MySQL shell we will need the password for the Morpheus DB user. We can find this in the morpheus-secrets file:

```
[root@app-server-old ~] cat /etc/morpheus/morpheus-secrets.json | grep morpheus_password
"morpheus_password": "451e122cr5d122asw3de5e1b", <-----this one
"morpheus_password": "9b5vdj4de5awf87d",
```

Take note of the first `morpheus_password` as it will be used to invoke a dump. Morpheus provides embedded binaries for this task. Invoke it via the embedded path and specify the host. In this example we are using the morpheus database on the MySQL listening on localhost. Enter the password copied from the previous step when prompted:

```
[root@app-server-old ~] /opt/morpheus/embedded/mysql/bin/mysqldump -u morpheus -h 127.0.0.1 morpheus -p > /tmp/morpheus_backup.sql
Enter password:
```

This file needs to be pushed to the new Morpheus Installation's backend. Depending on the GRANTS in the new MySQL backend, this will likely require moving this file to one of the new Morpheus frontend servers.

Once the file is in place it can be imported into the backend. Begin by ensuring the Morpheus UI service is stopped on all of the application servers:

```
[root@app-server-new ~] morpheus-ctl stop morpheus-ui
```

Then you can import the MySQL dump into the target database using the embedded MySQL binaries, specifying the database host, and entering the password for the morpheus user when prompted:

```
[root@app-server-new ~] /opt/morpheus/embedded/mysql/bin/mysql -u morpheus -h 127.0.0.1 morpheus -p < /tmp/morpheus_backup.sql
Enter password:
```

The data from the old appliance is now replicated on the new appliance. Simply start the UI to complete the process:

```
[root@app-server-new ~] morpheus-ctl start morpheus-ui
```

With the migration complete, you will also need to update the stored password for the appliance backup job as the destination appliance will have a different dynamically-generated MySQL password. We can update that password value by altering the backup directly in the Morpheus database.

```
select * from backup where `name` = 'Morpheus Appliance';
UPDATE `morpheus`.`backup` SET `ssh_host` = '127.0.0.1', `target_password` = 'its-a-secret' WHERE `id` = '1';
```

Important: After the migration it is important to reset the unique ID of the Morpheus Appliance. This will ensure your new installation will communicate correctly with the Morpheus Hub.

The final step is to generate a new unique ID for the Morpheus Appliance. Firstly run the following SQL command on the database for the new installation:

```
UPDATE appliance_instance SET hub_unique_id = NULL;
```

Secondly, re-apply your Morpheus license key within the Morpheus UI via the “Upgrade A License” action within Administration -> Settings -> License

Scaling Morpheus Nodes

Morpheus App nodes can be scaled to accommodate additional load. Appliance nodes can be scaled vertically in centralized architectures, and both vertically and horizontally in distributed architectures.

Vertical Scaling

In all Appliance Architectures, Application nodes can be vertically scaled at any time, however a reconfigure must be performed for additional resources to be utilized by Morpheus on a node, which will result in the `morpheus-ui` restarting on the reconfiguring node.

Morpheus configures memory/ram utilization for services during the `reconfigure` process. If additional memory/ram is added to a Host or VM running the Morpheus App, the additional memory/ram will not be utilized by the Morpheus Application until a `morpheus-ctl reconfigure` command is ran and the additional memory/ram is recognized.

Important: When the `morpheus-ctl reconfigure` command detects changes on available memory/ram, it will restart the `morpheus-ui` service.

The impact on Availability depends on the Morpheus Appliance Architecture.

- **Centralized Appliances** Morpheus will be unavailable while the `morpheus-ui` restarts.
- **Distributed Appliances** Zero-down time can be achieved by Reconfiguring one App Node at a time, with proper Front-End Load Balancer configuration.

Horizontal Scaling

Additional Morpheus App Nodes can be added at any time to Fully Distributed Architectures.

- Configure Shared Storage paths for the new App Node(s)
- Install, but do not run the `morpheus-ctl reconfigure` command on the new App Node(s), using the same Morpheus version as the existing Appliance nodes.
- Copy the `morpheus.rb` from an existing App Node to the new App Node(s)
- Ensure permissions and network configuration for the new App Node(s) to access all MySQL and Elasticsearch nodes, and the RabbitMQ VIP.
- Ensure permissions and network configuration for all required UI services and Integrations, such as network access to ESXi hosts over 443 for Hypervisor console and/or image transfers.
- Add associated SSL files and configuration which is not on shared storage.
- Reconfigure the new App Node(s) via `morpheus-ctl reconfigure`
- Verify UI startup succeeded
- Add New App Node(s) to Front End Morpheus UI Load Balancer pool.

During `morpheus-ctl reconfigure`, the new App Node(s) will validate and be configured to use the existing tiers for the UI service. Upon successful reconfigure, the Morpheus service will be available on the App Node(s) with consistent data and capabilities.

Note: No services, including `morpheus-ui`, are required to be shut down on existing nodes when adding new App Nodes

Morpheus UI war files

Pre-release or patched versions of the Morpheus UI are sometimes provided. To deploy the ui war on a Morpheus Appliance:

1. Download the war file to the target appliance

```
wget https://url/war_file
```

Note: If the war file is provided via a droplr link, ensure a + is added to end of droplr url or the file will not download

2. Backup current war file

```
sudo mv /opt/morpheus/lib/morpheus/morpheus-ui.war /opt/morpheus/lib/morpheus/  
↪morpheus-ui.bak.`date +%m-%d-%Y`
```

3. Move and rename new war file

```
sudo mv <file> /opt/morpheus/lib/morpheus/morpheus-ui.war
```

4. Ensure war is owned by morpheus-app

```
sudo chown morpheus-app:morpheus-app /opt/morpheus/lib/morpheus/morpheus-ui.war
```

5. Restart the Morpheus UI

```
sudo morpheus-ctl restart morpheus-ui
```

The new ui war will load on startup!

5.1.5 Additional Configuration Options

Advanced morpheus.rb Settings

Morpheus allows for additional advanced customizations for system managed services within the morpheus.rb file located in /etc/morpheus/morpheus.rb. Below is a list of the supported items available in the morpheus.rb file.

Note: Service configuration settings are not applicable for externalized services such as external mysql/percona, elasticsearch or rabbitmq clusters. Only connection settings are applicable for external services. Additionally, to configure Morpheus to utilize alternate ports for SSL, you may have to take additional configuration steps. If simply appending a port to your appliance_url value doesn't work, consult the related article in our [KnowledgeBase](#).

```
app['encryption_key_suffix'] = '$suffix'  
  # Replace $suffix with the suffix string of your choice. See `https://docs.  
↪morpheusdata.com/en/latest/getting_started/additional/encryption.html` for important  
↪details and warnings.  
appliance_url 'https://morpheus.appliance-url.com'  
  # Appending alternate port to appliance_url is supported. ie 'https://morpheus.  
↪appliance-url.com:8443'  
  # The appliance_url cannot exceed 64 characters
```

(continues on next page)

(continued from previous page)

```

# The appliance_url must not contain a trailing `\/`.

bitcan['backup_directory'] = '/var/opt/morpheus/bitcan/backups'
bitcan['working_directory'] = '/var/opt/morpheus/bitcan/working'

elasticsearch['auth_password'] = 'xxxxxxxxxxxxxxxxxxx'
elasticsearch['auth_user'] = 'morpheus-es-user'
elasticsearch['enable'] = true
elasticsearch['es_hosts'] = {'127.0.0.1' => 9200}
elasticsearch['host'] = "127.0.0.1"
elasticsearch['port'] = "9200"
elasticsearch['tmp_dir'] = '/var/tmp/elasticsearch'
elasticsearch['use_tls'] = false
↓ The following elasticsearch settings are only valid for Internal/Embedded
↳ elasticsearch services
elasticsearch['log_dir'] = '/var/log/morpheus/elasticsearch'
elasticsearch['memory_alloc_arena_max'] = 2
elasticsearch['memory_map_max'] = 65536
elasticsearch['memory_map_threshold'] = 131072
elasticsearch['memory_top_pad'] = 131072
elasticsearch['memory_trim_threshold'] = 131072
elasticsearch['open_files'] = 204800
elasticsearch['secure_mode'] = false
elasticsearch['xms'] = 0.25 ##### Configurable for customers running into high memory
↳ issues, values are the percentage of total RAM. Both ui and es xms/xmx config only
↳ apply if Elasticsearch is enabled.
elasticsearch['xmx'] = 0.25

guacd['guacamole_enabled'] = false
guacd['host'] = localhost ##### The host guacd is listening on, if not configured
↳ 'localhost' is the default value

logging['svlogd_num'] = 30 ##### keep 30 rotated log files
logging['svlogd_size'] = 209715200 ##### 200 MB in bytes
logging['svlogd_timeout'] = 86400 ##### rotate after 24 hours in seconds

mysql['enable'] = true
mysql['host'] = {'127.0.0.1' => 3306}
mysql['use_tls'] = false
mysql['morpheus_db_user'] = 'morpheus-db-user'
mysql['morpheus_password'] = 'morpheus-db-password'
mysql['morpheus_db'] = 'xxxxxxxxxxxxxxxxxxx'
mysql['mysql_url_override'] = 'jdbc:mysql://10.30.20.10:3306,10.30.20.11:3306,10.30.20.
↳ 12:3306/morpheusdb?autoReconnect=true&useUnicode=true&characterEncoding=utf8&
↳ failOverReadOnly=false&useSSL=false'
↓ The following mysql settings are only valid for Internal/Embedded mysql services
mysql['tmp_dir'] = '/tmp/mysql'
mysql['log_dir'] = '/var/log/morpheus/mysql'
mysql['max_active'] = 150 # The combined value off all app node max_active values must
↳ be lower than max_connections setting in mysql
mysql['max_connections'] = 151
mysql['max_allowed_packet'] = 67108864

```

(continues on next page)

(continued from previous page)

```

mysql['mysql_connect_timeout'] = 60000
mysql['mysql_max_reconnects'] = 2
mysql['mysql_queries_before_retry_source'] = 0
mysql['mysql_seconds_before_retry_source'] = 300

nginx['cache_max_size'] = '5000m'
nginx['enable'] = true
nginx['loading_pages']['failure_page_h1'] = 'Morpheus Server Error'
nginx['loading_pages']['failure_page_h2'] = 'Please contact your system administrator.
↳for assistance.'
nginx['loading_pages']['failure_page_title'] = 'Morpheus Server Error'
nginx['loading_pages']['iteration_time'] = 10000 # milliseconds
nginx['loading_pages']['loading_page_h1'] = 'Morpheus is Loading...'
nginx['loading_pages']['loading_page_h2'] = 'please wait'
nginx['loading_pages']['loading_page_title'] = 'Morpheus Loading'
nginx['loading_pages']['max_loops'] = 60 # seconds
nginx['loading_pages']['timeout_page'] = '/timeout.html'
nginx['loading_pages']['timeout_page_h1'] = 'Timeout waiting for Morpheus to load, click.
↳below to try again.'
nginx['loading_pages']['timeout_page_title'] = 'Morpheus timeout, please try again...'
nginx['log_format_name'] = 'custom'
nginx['log_format'] = '\'$remote_addr - $remote_user [$time_local] "$request" \' \'
↳$status $body_bytes_sent "$http_referer" \' \' "$http_user_agent" "$http_x_forwarded_for
↳" \' \' \rt=$request_time uct="$upstream_connect_time" uht="$upstream_header_time" urt="
↳$upstream_response_time"\'';'
nginx['ssl_ciphers'] = "ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:DHE-RSA-
↳AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-AES128-
↳SHA256:ECDHE-RSA-AES256-SHA:ECDHE-RSA-AES128-SHA:DHE-RSA-AES256-SHA256:DHE-RSA-AES128-
↳SHA256:DHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA:ECDHE-RSA-DES-CBC3-SHA:EDH-RSA-DES-CBC3-
↳SHA:AES256-GCM-SHA384:AES128-GCM-SHA256:AES256-SHA256:AES128-SHA256:AES256-SHA:AES128-
↳SHA:DES-CBC3-SHA:HIGH:!aNULL:!eNULL:!EXPORT:!DES:!MD5:!PSK:!RC4"
nginx['ssl_company_name'] = "Morpheus, LLC"
nginx['ssl_country_name'] = "US"
nginx['ssl_email_address'] = "personal@email.com"
nginx['ssl_locality_name'] = "San Mateo"
nginx['ssl_organizational_unit_name'] = "DevOps"
nginx['ssl_protocols'] = "TLSv1 TLSv1.1 TLSv1.2"
nginx['ssl_session_cache'] = "builtin:1000 shared:SSL:10m"
nginx['ssl_session_timeout'] = "5m"
nginx['ssl_state_name'] = "CA"
nginx['worker_connections'] = 10240
nginx['workers'] = integer calculated from number of cpus
nginx['ssl_access_ping_log'] = false #### false by default, when true GET requests to.
↳the ``/ping`` endpoint are logged in the ``/var/log/morpheus/nginx/morpheus-ssl-access.
↳log`` file on the appliance
nginx['access_ping_log'] = false #### false by default, when true GET requests to the ``/
↳ping`` endpoint are logged in the ``/var/log/morpheus/nginx/morpheus-ssl-access.log``.
↳file on the appliance

rabbitmq['enable'] = true
rabbitmq['host'] = '127.0.0.1'

```

(continues on next page)

(continued from previous page)

```

rabbitmq['port'] = '5672'
rabbitmq['queue_user_password'] = 'xxxxxxxxxxxxxxxxxxxx'
rabbitmq['queue_user'] = 'morpheus-rmq-user'
rabbitmq['vhost'] = 'morpheus'
↓ The following rabbitmq settings are only valid for Internal/Embedded rabbitmq services
rabbitmq['heartbeat'] = nil
rabbitmq['log_dir'] = '/var/log/morpheus/rabbitmq'
rabbitmq['nodename'] = 'rabbit@localhost'
rabbitmq['port'] = '5672'
rabbitmq['use_tls'] = false

repo['repo_host_url'] = 'https://downloads.morpheusdata.com'

ui['http_client_connect_timeout'] = 10000 #### milliseconds
ui['jobs_enabled'] = true #### This option disables the appliance jobs service on the
↳ appliance node when set to false. This should be disabled only when configuring jobs
↳ to run on specific app nodes in HA environments.
ui['kerberos_config'] = nil
ui['kerberos_login_config'] = nil
ui['log_dir'] = '/var/log/morpheus/morpheus-ui'
ui['max_memory_mb'] = nil
ui['memory_alloc_arena_max'] = 2
ui['memory_map_max'] = 65536
ui['memory_map_threshold'] = 131072
ui['memory_top_pad'] = 131072
ui['memory_trim_threshold'] = 131072
ui['pxe_boot_enabled'] = false #### This option disables the PXE service within the app
ui['vm_images_cdn_url'] = 'https://morpheus-images.morpheusdata.com'
ui['xms'] = 0.25 #### Configurable for customers running into high memory issues, values
↳ are the percentage of total RAM. Both ui and es xms/xmx config only apply if
↳ Elasticsearch is enabled.
ui['xmx'] = 0.25

```

Load Balancer Configuration

For configurations with 2 or more Applications Nodes, a load balancer is recommended to ensure high availability (HA) from disruptions and upgrades. Below are the guidelines to configuring a load balancer for Morpheus but each configuration may differ based on the organization's requirements.

Requirements

- WebSockets enabled
- Load Balance 443 (optionally redirect 80 to 443)
 - SSL Termination (Offload), Bridging, and Passthrough are supported
- Round-Robin or least connection distribution
- HTTPS monitor `https://ip_address/ping` body for MORPHEUS PING or status of 200, for node operational health

Example configurations

Below are a few examples of configuring load balancers to meet the needs of a HA configuration. The examples assume SSL bridging will be used, which means an SSL (TLS) certificate is presented by the load balancer to clients and the load balancer will communicate with the backend nodes via a different (possibly same) certificate. This configuration is recommended because the Morpheus application nodes will create self-signed certificates and the load balancer will present a valid certificate to end users. Additionally, all communication will be encrypted. This reduces the overhead of maintaining the certificates on the Morpheus application nodes, as the load balancer can ignore invalid certs on the application nodes. However, the certificates on the Morpheus application nodes are not required to be self-signed, they can be replaced with other trusted certificates following the [SSL Certificates](#) documentation.

Tip: The list below is not meant to be a complete list for all load balancers. The provided examples are common deployments that can be used for reference. The settings mentioned in the examples list the primary settings that may need to be configured, other settings are based on the organization's needs requirements and own configuration.

F5 BIG-IP

Monitor

- **Type:** HTTPS
- **Send String:** GET /ping
- **Receive String:** MORPHEUS PING

Pool

- **Health Monitor:** Select monitor created in the **Monitor** section
- **Load Balancing Method:** Least Connections (member) is recommended (alternatively Round Robin)
- **Node Service Port:** 443/HTTPS

Virtual Server

- **Type:** Standard
- **Service Port:** 443/HTTPS
- **Protocol:** TCP
- **Protocol Profile (Client):** tcp
- **Protocol Profile (Server):** tcp
- **HTTP Profile (Client):** http
- **HTTP Profile (Server):** http
- **SSL Profile (Client):** clientssl (or preferred profile with a trusted certificate)
- **SSL Profile (Server):** serverssl
- **Source Address Translation:** Auto Map

AWS Application Load Balancer (ALB)

Target Group

- **Target Type:** Instances
- **Protocol/Port:** HTTPS/443
- **Health Check Protocol:** HTTPS

- **Health check path:** /ping
- **Load balancing algorithm:** Least Outstanding Requests is recommended (alternatively Round Robin)

Load Balancer

- **Security Group:** Allow HTTPS/443 Inbound and Outbound
- **Listener:** HTTPS:443 to the target group created in the **Target Group** section
- **Security Policy:** ELBSecurityPolicy-2016-08

HAProxy

Example /etc/haproxy/haproxy.cfg configuration file

```
#-----
# Example configuration for a possible web application.  See the
# full configuration options online.
#
# https://www.haproxy.org/download/1.8/doc/configuration.txt
#
#-----
#-----
# Global settings
#-----
global
    log          127.0.0.1:514 local2
    chroot       /var/lib/haproxy
    pidfile      /var/run/haproxy.pid
    maxconn      4000
    user         haproxy
    group        haproxy
    daemon

    # turn on stats unix socket
    stats socket /var/lib/haproxy/stats

    # utilize system-wide crypto-policies
    ssl-default-bind-ciphers PROFILE=SYSTEM
    ssl-default-server-ciphers PROFILE=SYSTEM

defaults
    mode          http
    log           global
    option        httplog
    option        dontlognull
    option http-server-close
    option forwardfor      except 127.0.0.0/8
    option          redispatch
    retries         3
    timeout http-request 10s
    timeout queue    1m
    timeout connect  10s
    timeout client   1m
    timeout server   1m
```

(continues on next page)

```
timeout http-keep-alive 10s
timeout check          10s
maxconn                3000

frontend main
  mode http
  bind *:443 ssl crt /etc/haproxy/ssl/combined.crt_key.pem
  default_backend      mybackend

backend mybackend
  mode http
  option                httpchk
  http-check            send meth GET uri /ping
  http-check            expect string MORPHEUS\ PING
  balance               leastconn
  server                app1 192.168.101.1:443 check ssl verify none
  server                app2 192.168.101.2:443 check ssl verify none
  server                app3 192.168.101.3:443 check ssl verify none
```

Azure Application Gateway

In this example, it is assumed **End-To-End TLS Encryption** is being used, which means the Application Gateway will present a certificate to the clients and the backend nodes will present the **same** certificate.

If a setting is not mentioned, it is assumed that the default can be maintained.

General Settings

- **Tier:** Standard V2
- **Capacity type:** Autoscale or Manual are both supported
- **HTTPS2:** Disabled

Frontend Configuration

- **Type:** Set **Public** if Morpheus should be accessible externally, otherwise choose **Private**
- **Public IP Address:** Associate a previously create public IP or create a new one

Listener

- **Frontend IP:** Choose the IP created from the Frontend Configuration above
- **Protocol:** HTTPS
- **Port:** 443
- **Certificate:**
 - Upload the public certificate in **PEM** format
 - This certificate should match the one presented by the backend nodes
 - The certificate should include the entire chain, including the private key
- **Listener type:** Basic
- **Error page URL:** No

Backend Settings

- **Backend protocol:** HTTPS

- **Backend port:** 443
- **Use well known CA certificate:**
 - If set to **Yes**, the certificate does not need to be uploaded in the settings. This must be a well known certificate provided by a well known certificate authority, not an internally generated certificate
 - If set to **No**, ensure the certificate that is present on the backend nodes is uploaded to the Application Gateway. Note that the certificate should include the entire chain (CA, Intermediates, Certificate)
- **Cookie-based affinity:** Disable
- **Connection draining:** Enable
- **Override with new host name:** No
- **Use Custom probe:** No (one will be created next and will be associated during that configuration)

Health Probe

- **Protocol:** HTTPS
- **Host:** Enter the host that is configured on the Morpheus application nodes. This same host that will be used on the Application Gateway Example: morpheus.mydomain.com
- **Pick host name from backend settings:** No
- **Pick port from backend settings:** Yes
- **Path:** /ping
- **Use probe matching conditions:** Yes
- **HTTP response status code match:** 200-399
- **Backend settings:** Choose the backend settings created above

Backend Pool

- The **Target Type** can either be **Virtual Machine** or **IP address or FQDN**
 - If Morpheus is hosted in Azure, **Virtual Machine** will likely be the choice. The load balancer will need to be able to communicate with the target
 - If Morpheus is hosted on-premise, or outside of Azure, the **IP address or FQDN** can be used but the load balancer will need to be able to communicate with the target

Important Items

- Ensure the backend virtual machines allow port 443 from the load balancer, otherwise a **502 error** may be seen
- If using a wildcard certificate, you **must** use a custom health probe, as mentioned above, otherwise you may see the following error message:

The Common Name (CN) of the backend server certificate does not match the host header entered in the health probe configuration (v2 SKU) or the FQDN in the backend pool (v1 SKU). Verify if the hostname matches with the CN of the backend server certificate.

More info: <https://techcommunity.microsoft.com/t5/fasttrack-for-azure/walkthrough-configuring-end-to-end-tls-with-application-gateway/ba-p/3269132>

- As mentioned above, ensure the complete chain for the certificate is presented by Morpheus, otherwise you may see the following error message:

The root certificate of the server certificate used by the backend does not match the trusted root certificate added to the application gateway. Ensure that you add the correct root certificate to whitelist the backend

More info: <https://learn.microsoft.com/en-us/answers/questions/150524/the-root-certificate-of-the-server-certificate-use>

- Configuring the certificate on the Morpheus nodes

More info: https://docs.morpheusdata.com/en/latest/getting_started/additional/morpheusSslCerts.html

- Additional reading:

<https://learn.microsoft.com/en-us/azure/application-gateway/certificates-for-backend-authentication>

<https://learn.microsoft.com/en-us/azure/application-gateway/end-to-end-ssl-portal>

<https://learn.microsoft.com/en-us/azure/application-gateway/ssl-overview>

Offline Installations and Upgrades

For customers that have an appliance behind a firewall/proxy that does not allow downloads from our Amazon download site, you can add the supplemental package to add the needed packages the standard Morpheus installer would have downloaded.

Offline Installation Requirements

- NTP should be correctly configured and the server is able to connect to the NTP server in the ntp.conf file
- The OS package repositories should be configured to use local LAN repository servers or the server should be able to receive packages from the configured repositories
- The standard Morpheus and supplemental packages must be downloaded from another system and transferred to the Morpheus Appliance server
- The supplemental package is additive, the full installer is also required

Note: The supplemental package is linked 1-to-1 to the appliance release. For example the supplemental package for 4.2.1-1 should be used with the appliance package 4.2.1-1

Offline Install

Ubuntu/Debian

1. Download both the regular Morpheus Appliance package and the Supplemental packages on to the appliance server:

```
wget http://example_url/morpheus-appliance_version_amd64.deb
wget http://example_url/morpheus-appliance-supplemental_version_all.deb
```

2. Install the both the Appliance package AND the Supplemental package.

```
sudo dpkg -i morpheus-appliance_version_amd64.deb
sudo dpkg -i morpheus-appliance-supplemental_version_all.deb
```

3. Set the Morpheus UI appliance url (if needed, hostname will be automatically set).

```
# edit appliance_url to resolvable url (if not configured correctly by default)

sudo vi /etc/morpheus/morpheus.rb
```

4. Reconfigure the appliance to install required packages

```
sudo morpheus-ctl reconfigure
```

The Chef run should complete successfully. There is a small pause when Chef runs the resource `remote_file[package_name]` action create while Chef verifies the checksum. After the reconfigure is complete, the `morpheus-ui` will start and be up in a few minutes.

Note: Tail the morpheus log file located at `/var/log/morpheus/morpheus-ui/current` with the command `morpheus-ctl tail morpheus-ui` and look for the Morpheus ascii logo to know when the `morpheus-ui` is up.

CentOS/RHEL

1. Download both the regular Morpheus Appliance package and the matching Supplemental package on to the Appliance server:

```
wget http://example_url/morpheus-appliance_package_url.noarch.rpm
wget http://example_url/morpheus-appliance_package_supplemental_url.noarch.rpm
```

2. Install the both the Appliance package AND the Supplemental package.

```
sudo rpm -i morpheus-appliance_package_url.noarch.rpm
sudo rpm -i morpheus-appliance_package_supplemental_url.noarch.rpm
```

3. Set the Morpheus UI appliance url (if needed, hostname will be automatically set).

```
#Edit appliance_url to resolvable url (if not configured correctly by default)

sudo vi /etc/morpheus/morpheus.rb
```

4. Reconfigure the appliance to install required packages

```
sudo morpheus-ctl reconfigure
```

The Chef run should complete successfully. There is a small pause when Chef runs the resource `remote_file[package_name]` action create while Chef verifies the checksum. After the reconfigure is complete, the `morpheus-ui` will start and be up in a few minutes.

Note: Tail the `morpheus-ui` log file with `morpheus-ctl tail morpheus-ui` and look for the Morpheus ascii logo to know when the `morpheus-ui` is up.

Proxies

Overview

In many situations, companies deploy virtual machines in proxy restricted environments for things such as PCI Compliance, or just general security. As a result of this Morpheus provides out of the box support for proxy connectivity. Proxy authentication support is also provided with both Basic Authentication capabilities as well as NTLM for Windows Proxy environments. Morpheus is even able to configure virtual machines it provisions to utilize these proxies by setting up the operating systems proxy settings directly (restricted to cloud-init based Linux platforms for now, but can also be done on windows based platforms in a different manner).

To get started with Proxies, it may first be important to configure the Morpheus appliance itself to have access to proxy communication for downloading service catalog images. To configure this, visit the Administration > Settings page where a section labeled “Proxy Settings” is located. Fill in the relevant connection info needed to utilize the proxy. It may also be advised to ensure that the Linux environment’s `http_proxy`, `https_proxy`, and `no_proxy` are set appropriately.

Defining Proxies

Proxies can be used in a few different contexts and optionally scoped to specific networks with which one may be provisioning into or on a cloud integration as a whole. To configure a Proxy for use by the provisioning engines within Morpheus we must go to **Infrastructure > Networks > Proxies**. Here we can create records representing connection information for various proxies. This includes the host ip address, proxy port, and any credentials (if necessary) needed to utilize the proxy. Now that these proxies are defined we can use them in various contexts.

Cloud Communication

When morpheus needs to connect to various cloud APIs to issue provisioning commands or to sync in existing environments, we need to ensure that those api endpoints are accessible by the appliance. In some cases the appliance may be behind a proxy when it comes to public cloud access like Azure and AWS. To configure the cloud integration to utilize a proxy, when adding or editing a cloud there is a setting called “API Proxy” under “Advanced Options”. This is where the proxy of choice can be selected to instruct the Provisioning engine how to communicate with the public cloud. Simply adjust this setting and the cloud should start being able to receive/issue instructions.

Provisioning with Proxies

Proxy configurations can vary from operating system to operating system and in some cases it is necessary for these to be configured in the blueprints as a prerequisite. In other cases it can also be configured automatically. Mostly with the use of cloud-init (which all of our out of the box service catalog utilizes on all clouds). When editing/creating a cloud there is a setting for “Provisioning Proxy” in “Provisioning Options”. If this proxy is set, Morpheus will automatically apply these proxy settings to the guest operating system.

Overriding proxy settings can also be done on the Network record. Networks (or subnets) can be configured in **Infrastructure > Networks** or on the Networks tab of the relevant Cloud detail page. Here, a proxy can also be assigned as well as additional options like the **No Proxy** rules for proxy exceptions.

Docker

When provisioning Docker based hosts within a Proxy environment it is up to the user to configure the docker host proxy configuration manually. There are workflows that can be configured via the Automation engine to make this automatic when creating docker based hosts. Please see documentation on Docker and proxies for specific information.

Proxy setups can vary widely from company to company, and it may be advised to contact support for help configuring morpheus to work in the proxy environment.

SSL Certificates

By default Morpheus generates a Self-Signed SSL Certificate. The Self-Signed SSL Certificate can be replaced with a Trusted CA Signed SSL Certificate.

Trusted CA Signed SSL Certificate Implementation

1. If you don't already have your certificate, run an OpenSSL command to generate an SSL certificate request (.csr) and private key (.key). If you need help formatting the command, [DigiCert provides a helpful tool](#)
2. Submit your certificate request (.csr) and await approval of the request and return of the certificate (.crt)
3. Copy the private key and certificate to /etc/morpheus/ssl/your_fqdn_name.key and /etc/morpheus/ssl/your_fqdn_name.crt respectively

- **Extracting Certificates in PFX Format**

```
# Extract the private key
openssl pkcs12 -in example.pfx -nocerts -nodes -out priv.key
# Extract the public key
openssl pkcs12 -in example.pfx -clcerts -nokeys -out pub.crt
# Extract the CA cert chain
openssl pkcs12 -in example.pfx -cacerts -nokeys -chain -out ca.crt
```

- **Extracting Certificates in PEM Format**

```
# Extract the private key
openssl x509 -outform der -in your-cert.pem -out your-cert.key
# Extract the public key
openssl x509 -outform der -in your-cert.pem -out your-cert.key
```

4. Edit the configuration file /etc/morpheus/morpheus.rb and add the following entries:

```
nginx['ssl_certificate'] = 'path to the certificate file'
nginx['ssl_server_key'] = 'path to the server key file'
```

Note: Both files should be owned by root and only readable by root, also if the server certificate is signed by an intermediate then you should include the signing chain inside the certificate file. The key file needs to be decrypted for Morpheus to install it properly.

5. Next simply reconfigure the appliance and restart nginx:

```
sudo morpheus-ctl reconfigure
sudo morpheus-ctl restart nginx
```

Self-Signed SSL Certificate Regeneration

When Morpheus is deployed it generates a 10 year Self-Signed SSL Certificate. Below details the process to regenerate the Certificate and Key files.

Regenerate both the Certificate and Key

1. Delete the certificate and key files in `/etc/morpheus/ssl/`.
2. Run `Reconfigure morpheus-ctl reconfigure`.
3. Restart NGINX `morpheus-ctl restart nginx`.

Regenerate only the Certificate

1. Delete the certificate file in `/etc/morpheus/ssl/`.
2. Run `Reconfigure morpheus-ctl reconfigure`.
3. Restart NGINX `morpheus-ctl restart nginx`.

Import Trusted Certificates

Important: The following applies to upgrades after modifying the java keystore.

Steps to import trusted certificates to Morpheus after an upgrade.

1. Obtain the full SSL certificate chain in PEM format.
2. Copy them to each appliance and place them in the `/etc/morpheus/ssl/trusted_certs` directory.
3. Run `morpheus-ctl reconfigure` on each appliance, note you don't need to stop Morpheus before you run this.
4. Run the following command as root:

```
export PATH=/opt/morpheus/sbin:/opt/morpheus/sbin:/opt/morpheus/embedded/  
↪sbin:/opt/morpheus/embedded/bin:$PATH
```

5. Run the following command for each certificate in the chain, adjusting the file and alias name as needed. Answer yes for the root certificate when asked if you want to trust it.

```
/opt/morpheus/embedded/java/jre/bin/keytool -import -keystore /opt/morpheus/  
↪embedded/java/jre/lib/security/cacerts -trustcacerts -file /etc/morpheus/  
↪ssl/trusted_certs/root_ca.pem -alias some_alias -storepass changeit
```

6. Verify by running:

```
openssl s_client -connect host:port -showcerts -tls1_2
```

7. You should get an output similar to:

```

New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES256-GCM-SHA384
Server public key is 2048 bit
Secure Renegotiation IS supported
No ALPN negotiated
SSL-Session:
Protocol : TLSv1.2
Cipher   : ECDHE-RSA-AES256-GCM-SHA384
Session-ID: 5D9E820E4FF2A73A9977BA663E6029AA5415FEE85F49D8B1E541F5997C8E1FB2
Session-ID-ctx:
Master-Key:↵
↵29EEC2E7750C659AECB9942902D9A87B824E571522812B718420FC08F8D2ACE68CB16EC812A7D90B12A86D1970FF
Key-Arg   : None
PSK identity: None
PSK identity hint: None
SRP username: None
Start Time: 1547219217
Timeout   : 7200 (sec)
Verify return code: 0 (ok) #<-----

```

8. If the certificates are installed correctly you should see `Verify return code: 0 (ok)`. If they were not installed correctly then you will see a return similar to: `Verify return code: 21 (unable to verify the first certificate)`
9. Repeat for all App Nodes

Data Encryption

By default, Morpheus encrypts sensitive data in the Database using AES encryption mode with GCM. Passwords and other strings in Morpheus Appliance configuration files such as `morpheus-secrets.json` and `morpheus.rb` are in plain text as they are only accessible by root.

Passwords and other strings in Morpheus Appliance configuration files can be set to an encrypted string using the Morpheus crypto utility to generate ENC strings and then using `ENC(string)` as the value in the configuration file.

Additionally a custom Encryption Key Suffix can be set in the `morpheus.rb` configuration file. This suffix will be combined with a system string to generate a SHA-256 hash, which is used to generate the AES encryption key.

Generate ENC Strings for morpheus-secrets.json

System generated passwords are set in `/etc/morpheus/morpheus-secrets.json`. These entries can be updated to ENC strings with the following steps:

1. On the Morpheus appliance, run `morpheus-ctl get-crypto-string migrate` which will output `ENC()` strings for the passwords in `morpheus-secrets.json`
2. Update the desired password strings in the `morpheus-secrets.json` config file with the matching `ENC()` string.
3. Save `morpheus-secrets.json`
4. Run `morpheus-ctl reconfigure`

Generate ENC Strings for custom morpheus.rb entries

ENC() strings can be generated for sensitive data set in morpheus.rb, such as the password to an external service.

To generate ENC(0) strings for morpheus.rb entries:

1. On the Morpheus appliance, run `morpheus-ctl get-crypto-string string $clear_text '$suffix'` which will output strings for the passwords in morpheus-secrets.json
 - Replace `$clear_text` with the string to be encrypted
 - If a suffix is defined in morpheus.rb (as described in the next section), replace `$suffix` with your suffix.

Note: It is advisable to disable bash history logging by running `unset HISTFILE` before running the `morpheus-ctl get-crypto-string` command and then `set HISTFILE=$HOME/.bash_history` to reenale.

2. Update the desired password strings in the `morpheus.rb` config file with the matching string output, using `ENC($output)` format
 - Example: `mysql['morpheus_password'] = 'ENC($ZI5Dna00quhxKe$kDFD+U2ZeJUuYiNC$F1+czPNyo+3lAdq7V0g`
3. Save `morpheus.rb`
4. Run `morpheus-ctl reconfigure`

Encrypted Key Suffix

A custom Encryption Key Suffix can be set in the `morpheus.rb` configuration file. This suffix will be combined with a system string to generate a SHA-256 Hash, which is used in the generation of the system AES encryption key.

Danger: Setting a custom Encryption Key Suffix affects all data encrypted by Morpheus, including database and cypher data. Encryption Key Suffix is required in the event data needs to be migrated or recovered. Once the Encryption Key Suffix is set, data cannot be recovered without it. Store any Encryption Key Suffix externally where it can be referenced for future scenarios.

Important: The Encryption Key Suffix cannot be changed or removed after being set. Changing or removing an existing Encryption Key Suffix will prevent data access. If an existing suffix is altered in the `morpheus.rb` file, it must be restore to its original value.

1. Add `app['encryption_key_suffix'] = '$suffix'` to `/etc/morpheus/morpheus.rb`, replacing `$suffix` with your suffix.

Danger: Once an Encryption Key Suffix is set and applied via `reconfigure`, it cannot be altered or removed and data cannot be migrated or recovered without it.

2. Run `morpheus-ctl reconfigure`
 - `Reconfigure` will generate a new encryption key using the suffix and set (ENC) values for the service password in `application.yml`

logback config

Note: This doc is for 5.4.4+ versions that use `logback.xml`. 5.4.3 and earlier versions use `logback.groovy` with a different syntax that is not compatible with this doc. Please refer to 5.4.3 and earlier documentation for `logback.groovy` configuration details.

The log output for the `morpheus-ui` service is configured in the `logback.xml` file. Log output levels can be updated when more or less log output is desired.

Setting Log Levels

To change a log level, edit the logback configuration file in `/opt/morpheus/conf/logback.xml` and save. The changes will be reflected within the configured `scanPeriod`, 30 seconds by default.

Levels:

- **OFF** (no log output)
- **ERROR** (includes error logs)
- **WARN** (includes warn and error logs)
- **INFO** (includes info, warn and error logs)
- **DEBUG** (includes info, warn, error and debug logs)
- **TRACE** (includes info, warn, error, debug and trace logs)

Warning: Use `DEBUG` and/or `TRACE` levels with caution. `DEBUG` & `TRACE` levels can produce many logs that can consume disk space quickly. Only use `DEBUG` and/or `TRACE` levels when needed and target them for specific services.

Example Logback Settings

Below are sample log configuration settings. This is not a complete list. Additional log names/paths can typically be determined from the standard `INFO`, `WARN` and `ERROR` logs.

ACI

```
<logger name="com.morpheus.integration.NetworkServersController" level=
↪ "DEBUG"/>
<logger name="com.morpheus.network.AciNetworkService" level="DEBUG"/>
<logger name="com.morpheus.network.AciUtility" level="DEBUG"/>
<logger name="com.morpheus.network.NetworkService " level="DEBUG"/>
```

Amazon

```
<logger name="com.morpheus.compute.amazon.AmazonComputeService" level=
↪ "DEBUG"/>
<logger name="com.morpheus.compute.AmazonComputeUtility" level="DEBUG"/>
<logger name="com.morpheus.costing.AmazonCostingService" level="DEBUG"/>
<logger name="com.morpheus.provision.AmazonProvisionService" level="DEBUG"/
↪ >
```

Azure

```
<logger name="com.morpheus.Azure.ServersController" level="DEBUG"/>
<logger name="com.morpheus.AzureSqlServerProvisionService" level="DEBUG"/>
<logger name="com.morpheus.compute.azure.AzureComputeService" level="DEBUG
↵"/>
<logger name="com.morpheus.compute.AzureComputeUtility" level="DEBUG"/>
<logger name="com.morpheus.compute.AzureCostingService" level="DEBUG"/>
```

Chef

```
<logger name="com.morpheus.automation.ChefClientService" level="DEBUG"/>
<logger name="com.morpheus.automation.ChefService" level="DEBUG"/>
<logger name="com.morpheus.automation.ChefTaskService" level="DEBUG"/>
```

DNS

```
<logger name="com.morpheus.dns.MicrosoftDnsService" level="DEBUG"/>
```

General

```
<logger name="com.morpheus.InstanceService" level="DEBUG"/>
<logger name="com.morpheus.util.ApiUtility" level="DEBUG"/>
<logger name="com.morpheus.AppService" level="DEBUG"/>
<logger name="com.morpheus.MorpheusComputeService" level="DEBUG"/>
<logger name="com.morpheus.RpcService" level="DEBUG"/>
<logger name="com.morpheus.network.NetworkService " level="DEBUG"/>
<logger name="com.morpheus.provision.AbstractProvisionService" level="DEBUG
↵"/>
<logger name="com.morpheus.provision.AbstractBoxProvisionService" level=
↵"DEBUG"/>
<logger name="com.morpheus.compute.ProgressUpdater" level="DEBUG"/>
```

Google

```
<logger name="com.morpheus.compute.google.GoogleComputeService" level=
↵"DEBUG"/>
<logger name="com.morpheus.compute.GoogleComputeUtility" level="DEBUG"/>
<logger name="com.morpheus.provision.GoogleProvisionService" level="DEBUG"/
↵>
```

Hyper-V

```
<logger name="com.morpheus.compute.hyperv.HypervComputeService" level=
↵"DEBUG" />
<logger name="com.morpheus.compute.HypervComputeUtility" level="DEBUG" />
```

IBM Cloud

```
<logger name="com.morpheus.compute.softlayer.SoftlayerComputeService" ↵
↵level="DEBUG"/>
<logger name="com.morpheus.compute.SoftlayerComputeUtility" level="DEBUG"/>
```

Kubernetes

```

<logger name="com.morpheus.app.KubernetesAppTemplateService" level="DEBUG"/
↪>
<logger name="com.morpheus.app.KubernetesResourceMappingService" level=
↪ "DEBUG"/>
<logger name="com.morpheus.compute.KubernetesComputeService" level="DEBUG"/
↪>
<logger name="com.morpheus.host.KubernetesHostService" level="DEBUG"/>
<logger name="com.morpheus.provision.KubernetesProvisionService" level=
↪ "DEBUG"/>
<logger name="com.morpheus.storage.KubernetesStorageService" level="DEBUG"/
↪>

```

Monitoring

```

<logger name="com.morpheus.monitoring.MonitorCheckService" level="DEBUG"/>

```

Network

```

<logger name="com.morpheusdata.infoblox.InfobloxProvider" level="DEBUG"/>
<logger name="com.morpheus.network.InfobloxNetworkPoolService" level="DEBUG
↪"/>
<logger name="com.morpheus.network.NetworkService " level="DEBUG"/>
<logger name="com.morpheus.network.PluginNetworkPoolService" level="DEBUG"/
↪>

```

Nutanix

```

<logger name="com.morpheus.compute.nutanix.NutanixComputeService" level=
↪ "DEBUG"/>
<logger name="com.morpheus.compute.NutanixComputeUtility" level="DEBUG"/>
<logger name="com.morpheus.provision.NutanixProvisionService" level="DEBUG
↪"/>

```

Openstack

```

<logger name="com.morpheus.compute.AbstractOpenStackComputeService" level=
↪ "DEBUG"/>
<logger name="com.morpheus.compute.AbstractOpenStackComputeUtility" level=
↪ "DEBUG"/>
<logger name="com.morpheus.provision.OpenStackProvisionService" level=
↪ "DEBUG"/>
<logger name="com.morpheus.storage.OpenStackSFSSStorageService" level="DEBUG
↪"/>

```

Option Types

```

<logger name="com.morpheus.OptionSourceService" level="DEBUG"/>
<logger name="com.morpheus.OptionTypeListService" level="DEBUG"/>
<logger name="com.morpheus.OptionTypeService" level="DEBUG"/>

```

Remote Console

```

<logger name="com.morpheus.remote.MorpheusGuacamoleWebsocketHandler" level=
↪ "DEBUG"/>

```

SCVMM

```
<logger name="com.morpheus.compute.scvmm.ScvmmComputeService" level="DEBUG"
↳"/>
<logger name="com.morpheus.compute.ScvmmComputeUtility" level="DEBUG"/>
<logger name="com.morpheus.provision.ScvmmProvisionService" level="DEBUG"/>
```

ServiceNow

```
<logger name="com.morpheus.cmdb.ServiceNowCmdbService" level="DEBUG"/>
<logger name="com.morpheus.integrations.ServiceNowUtility" level="DEBUG"/>
```

Tasks

```
<logger name="com.morpheus.task.AnsibleTowerTaskService" level="DEBUG"/>
<logger name="com.morpheus.task.TaskService" level="DEBUG"/>
<logger name="com.morpheus.task.WinrmTaskService" level="DEBUG"/>
```

Terraform

```
<logger name="com.morpheus.app.AbstractResourceMappingService" level="DEBUG"
↳"/>
<logger name="com.morpheus.app.TerraformAppTemplateService" level="DEBUG"/>
<logger name="com.morpheus.app.TerraformAwsResourceMappingService" level=
↳"DEBUG"/>
<logger name="com.morpheus.app.TerraformResourceMappingService" level=
↳"DEBUG"/>
<logger name="com.morpheus.provision.TerraformProvisionService" level=
↳"DEBUG"/>
```

Usage

```
<logger name="com.morpheus.AccountPriceService" level="DEBUG"/>
```

vCloud

```
<logger name="com.morpheus.compute.vmware.VcloudDirectorComputeService"
↳level="DEBUG"/>
<logger name="com.morpheus.provision.VcloudDirectorProvisionService" level=
↳"DEBUG"/>
<logger name="com.morpheus.compute.VcdComputeUtility" level="DEBUG"/>
```

Veeam

```
<logger name="com.morpheus.backup.VeeamBackupService" level="DEBUG"/>
```

Vmware

```
<logger name="com.morpheus.compute.VmwareComputeUtility" level="DEBUG"/>
<logger name="com.morpheus.compute.vmware.VmwareComputeService" level=
↳"DEBUG"/>
<logger name="com.morpheus.provision.VmwareProvisionService" level="DEBUG"/
↳>
```

vRO

```
<logger name="com.morpheus.automation.VroService" level="DEBUG"/>
```

All core logger paths

Expand below to see all core Morpheus logger paths set to INFO level.

All core logger paths [Click to Expand/Hide](#)

```
<logger name="com.bertramlabs.plugins.AccountsAuthService" level="INFO"/>
<logger name="com.bertramlabs.plugins.AccountsService" level="INFO"/>
<logger name="com.bertramlabs.plugins.ActiveDirectoryUserService" level="INFO"/>
↔>
<logger name="com.bertramlabs.plugins.AzureSamlUserService" level="INFO"/>
<logger name="com.bertramlabs.plugins.CustomApiUserService" level="INFO"/>
<logger name="com.bertramlabs.plugins.CustomExternalUserService" level="INFO"/>
<logger name="com.bertramlabs.plugins.DefaultUserService" level="INFO"/>
<logger name="com.bertramlabs.plugins.JumpCloudUserService" level="INFO"/>
<logger name="com.bertramlabs.plugins.LdapUserService" level="INFO"/>
<logger name="com.bertramlabs.plugins.OktaUserService" level="INFO"/>
<logger name="com.bertramlabs.plugins.OneLoginUserService" level="INFO"/>
<logger name="com.bertramlabs.plugins.PingUserService" level="INFO"/>
<logger name="com.bertramlabs.plugins.SamlUserService" level="INFO"/>
<logger name="com.bertramlabs.plugins.UserSourceAuthenticationProvider" level=
↔"INFO"/>
<logger name="com.morpheus.AbstractComputeService" level="INFO"/>
<logger name="com.morpheus.AbstractPriceManagerService" level="INFO"/>
<logger name="com.morpheus.AccountBudgetService" level="INFO"/>
<logger name="com.morpheus.AccountIntegrationObjectService" level="INFO"/>
<logger name="com.morpheus.AccountIntegrationService" level="INFO"/>
<logger name="com.morpheus.AccountInvoiceService" level="INFO"/>
<logger name="com.morpheus.AccountPriceService" level="INFO"/>
<logger name="com.morpheus.AccountResourceService" level="INFO"/>
<logger name="com.morpheus.AccountUsageService" level="INFO"/>
<logger name="com.morpheus.ActivityService" level="INFO"/>
<logger name="com.morpheus.analytics.AbstractAnalyticsService" level="INFO"/>
<logger name="com.morpheus.analytics.AmazonConvertibleRiAnalyticsService"
↔level="INFO"/>
<logger name="com.morpheus.analytics.CostAnalyticsService" level="INFO"/>
<logger name="com.morpheus.analytics.UtilizationAnalyticsService" level="INFO"/>
↔>
<logger name="com.morpheus.analytics.WorkloadAnalyticsService" level="INFO"/>
<logger name="com.morpheus.AnalyticsService" level="INFO"/>
<logger name="com.morpheus.api.AbstractApiService" level="INFO"/>
<logger name="com.morpheus.api.agent.CommandService" level="INFO"/>
<logger name="com.morpheus.api.agent.DownloadService" level="INFO"/>
<logger name="com.morpheus.api.agent.UploadService" level="INFO"/>
<logger name="com.morpheus.app.AbstractAppTemplateService" level="INFO"/>
<logger name="com.morpheus.app.AbstractResourceMappingService" level="INFO"/>
<logger name="com.morpheus.app.AppTemplateService" level="INFO"/>
<logger name="com.morpheus.app.HelmAppTemplateService" level="INFO"/>
<logger name="com.morpheus.app.KubernetesAppTemplateService" level="INFO"/>
<logger name="com.morpheus.app.KubernetesResourceMappingService" level="INFO"/>
```

(continues on next page)

(continued from previous page)

```
<logger name="com.morpheus.app.MorpheusAppTemplateService" level="INFO"/>
<logger name="com.morpheus.app.ScribeResourceMappingService" level="INFO"/>
<logger name="com.morpheus.app.TerraformAppTemplateService" level="INFO"/>
<logger name="com.morpheus.app.TerraformAwsResourceMappingService" level="INFO"
↳"/>
<logger name="com.morpheus.app.TerraformAzurermResourceMappingService" level=
↳"INFO"/>
<logger name="com.morpheus.app.TerraformGoogleResourceMappingService" level=
↳"INFO"/>
<logger name="com.morpheus.app.TerraformResourceMappingService" level="INFO"/>
<logger name="com.morpheus.app.TerraformVsphereResourceMappingService" level=
↳"INFO"/>
<logger name="com.morpheus.ApplianceClientService" level="INFO"/>
<logger name="com.morpheus.ApplianceDelayedJobService" level="INFO"/>
<logger name="com.morpheus.ApplianceHealthService" level="INFO"/>
<logger name="com.morpheus.ApplianceJobService" level="INFO"/>
<logger name="com.morpheus.ApplianceLicenseService" level="INFO"/>
<logger name="com.morpheus.ApplianceService" level="INFO"/>
<logger name="com.morpheus.ApplianceStorageService" level="INFO"/>
<logger name="com.morpheus.approval.ApprovalService" level="INFO"/>
<logger name="com.morpheus.approval.RemedyApprovalService" level="INFO"/>
<logger name="com.morpheus.approval.ServiceNowApprovalService" level="INFO"/>
<logger name="com.morpheus.AppService" level="INFO"/>
<logger name="com.morpheus.ArchiveService" level="INFO"/>
<logger name="com.morpheus.AsyncService" level="INFO"/>
<logger name="com.morpheus.AuditLogService" level="INFO"/>
<logger name="com.morpheus.automation.AbstractConfigManagementService" level=
↳"INFO"/>
<logger name="com.morpheus.automation.AnsibleService" level="INFO"/>
<logger name="com.morpheus.automation.AnsibleTowerService" level="INFO"/>
<logger name="com.morpheus.automation.ChefService" level="INFO"/>
<logger name="com.morpheus.automation.ConfigManagementService" level="INFO"/>
<logger name="com.morpheus.automation.HelmService" level="INFO"/>
<logger name="com.morpheus.automation.PuppetService" level="INFO"/>
<logger name="com.morpheus.automation.SaltStackService" level="INFO"/>
<logger name="com.morpheus.automation.VroService" level="INFO"/>
<logger name="com.morpheus.backup.AbstractBackupExecutionService" level="INFO"/
↳>
<logger name="com.morpheus.backup.AbstractBackupJobService" level="INFO"/>
<logger name="com.morpheus.backup.AbstractBackupProviderService" level="INFO"/>
<logger name="com.morpheus.backup.AbstractBackupRestoreService" level="INFO"/>
<logger name="com.morpheus.backup.AbstractBackupService" level="INFO"/>
<logger name="com.morpheus.backup.AbstractReplicationService" level="INFO"/>
<logger name="com.morpheus.backup.BackupExecutionInterface" level="INFO"/>
<logger name="com.morpheus.backup.BackupInterface" level="INFO"/>
<logger name="com.morpheus.backup.BackupJobInterface" level="INFO"/>
<logger name="com.morpheus.backup.BackupJobService" level="INFO"/>
<logger name="com.morpheus.backup.BackupProviderInterface" level="INFO"/>
<logger name="com.morpheus.backup.BackupProviderService" level="INFO"/>
<logger name="com.morpheus.backup.BackupRestoreInterface" level="INFO"/>
<logger name="com.morpheus.backup.BackupRestoreService" level="INFO"/>
<logger name="com.morpheus.backup.BackupService" level="INFO"/>
```

(continues on next page)

(continued from previous page)

```
<logger name="com.morpheus.backup.BackupStatus" level="INFO"/>
<logger name="com.morpheus.backup.BackupStorageService" level="INFO"/>
<logger name="com.morpheus.backup.DirectoryBackupService" level="INFO"/>
<logger name="com.morpheus.backup.FileBackupService" level="INFO"/>
<logger name="com.morpheus.backup.KarmanStorageProviderBackupService" level=
↪ "INFO"/>
<logger name="com.morpheus.backup.LvmImageBackupService" level="INFO"/>
<logger name="com.morpheus.backup.LvmSnapshotBackupService" level="INFO"/>
<logger name="com.morpheus.backup.MorpheusApplianceBackupService" level="INFO"/
↪ >
<logger name="com.morpheus.backup.MorpheusBackupService" level="INFO"/>
<logger name="com.morpheus.backup.MorpheusContainerBackupService" level="INFO"/
↪ >
<logger name="com.morpheus.backup.MysqlBackupService" level="INFO"/>
<logger name="com.morpheus.backup.PluginBackupExecutionService" level="INFO"/>
<logger name="com.morpheus.backup.PluginBackupJobService" level="INFO"/>
<logger name="com.morpheus.backup.PluginBackupProviderService" level="INFO"/>
<logger name="com.morpheus.backup.PluginBackupRestoreService" level="INFO"/>
<logger name="com.morpheus.backup.PluginReplicationService" level="INFO"/>
<logger name="com.morpheus.backup.ReplicationInterface" level="INFO"/>
<logger name="com.morpheus.backup.ReplicationService" level="INFO"/>
<logger name="com.morpheus.backup.SqlserverBackupService" level="INFO"/>
<logger name="com.morpheus.backup.TarDirectoryBackupService" level="INFO"/>
<logger name="com.morpheus.BootMacService" level="INFO"/>
<logger name="com.morpheus.builds.AbstractBuildsService" level="INFO"/>
<logger name="com.morpheus.builds.BuildsService" level="INFO"/>
<logger name="com.morpheus.builds.JenkinsBuildsService" level="INFO"/>
<logger name="com.morpheus.CapacityService" level="INFO"/>
<logger name="com.morpheus.CatalogCartService" level="INFO"/>
<logger name="com.morpheus.CatalogItemService" level="INFO"/>
<logger name="com.morpheus.CatalogItemTypeService" level="INFO"/>
<logger name="com.morpheus.certificate.AbstractCertificateService" level="INFO
↪ "/>
<logger name="com.morpheus.certificate.MorpheusCertificateService" level="INFO
↪ "/>
<logger name="com.morpheus.CertificateService" level="INFO"/>
<logger name="com.morpheus.ChefClientService" level="INFO"/>
<logger name="com.morpheus.cm.ChangeManagementService" level="INFO"/>
<logger name="com.morpheus.cm.CherwellCmService" level="INFO"/>
<logger name="com.morpheus.cmdb.AbstractCmdbService" level="INFO"/>
<logger name="com.morpheus.cmdb.CmdbService" level="INFO"/>
<logger name="com.morpheus.cmdb.RemedyCmdbService" level="INFO"/>
<logger name="com.morpheus.cmdb.ServiceNowCmdbService" level="INFO"/>
<logger name="com.morpheus.code.AbstractCodeService" level="INFO"/>
<logger name="com.morpheus.code.CodeService" level="INFO"/>
<logger name="com.morpheus.code.GitCodeService" level="INFO"/>
<logger name="com.morpheus.code.GithubCodeService" level="INFO"/>
<logger name="com.morpheus.compliance.NVDSyncService" level="INFO"/>
<logger name="com.morpheus.compliance.PackageManagementService" level="INFO"/>
<logger name="com.morpheus.compute.cisco.UcsComputeService" level="INFO"/>
<logger name="com.morpheus.compute.CloudPluginComputeService" level="INFO"/>
<logger name="com.morpheus.compute.ComputeApiService" level="INFO"/>
```

(continues on next page)

(continued from previous page)

```
<logger name="com.morpheus.compute.ComputeServiceInterface" level="INFO"/>
<logger name="com.morpheus.compute.IpmiService" level="INFO"/>
<logger name="com.morpheus.compute.KubernetesComputeService" level="INFO"/>
<logger name="com.morpheus.compute.MaasComputeService" level="INFO"/>
<logger name="com.morpheus.compute.ManualComputeService" level="INFO"/>
<logger name="com.morpheus.compute.OneviewComputeService" level="INFO"/>
<logger name="com.morpheus.compute.SelfManagedComputeService" level="INFO"/>
<logger name="com.morpheus.compute.standard.StandardComputeService" level="INFO
↪"/>
<logger name="com.morpheus.compute.unmanaged.UnmanagedComputeService" level=
↪"INFO"/>
<logger name="com.morpheus.compute.vmware.VcloudDirectorComputeService" level=
↪"INFO"/>
<logger name="com.morpheus.compute.vmware.VmwareComputeService" level="INFO"/>
<logger name="com.morpheus.ComputeService" level="INFO"/>
<logger name="com.morpheus.container.ActivemqContainerService" level="INFO"/>
<logger name="com.morpheus.container.DockerContainerService" level="INFO"/>
<logger name="com.morpheus.container.DockerContainerUpgradeService" level="INFO
↪"/>
<logger name="com.morpheus.container.ElasticsearchContainerService" level="INFO
↪"/>
<logger name="com.morpheus.container.JavaContainerService" level="INFO"/>
<logger name="com.morpheus.container.MysqlContainerService" level="INFO"/>
<logger name="com.morpheus.container.NodeContainerService" level="INFO"/>
<logger name="com.morpheus.container.PostgresContainerService" level="INFO"/>
<logger name="com.morpheus.container.RedisContainerService" level="INFO"/>
<logger name="com.morpheus.container.SqlserverContainerService" level="INFO"/>
<logger name="com.morpheus.ContainerScriptService" level="INFO"/>
<logger name="com.morpheus.ContainerService" level="INFO"/>
<logger name="com.morpheus.costing.AbstractCostingService" level="INFO"/>
<logger name="com.morpheus.costing.CostingInterface" level="INFO"/>
<logger name="com.morpheus.costing.CostingService" level="INFO"/>
<logger name="com.morpheus.costing.StandardCostingService" level="INFO"/>
<logger name="com.morpheus.CurrencyConversionService" level="INFO"/>
<logger name="com.morpheus.cypher.CypherGORMDatastoreService" level="INFO"/>
<logger name="com.morpheus.cypher.CypherService" level="INFO"/>
<logger name="com.morpheus.DashboardService" level="INFO"/>
<logger name="com.morpheus.DatastoreService" level="INFO"/>
<logger name="com.morpheus.DataViewService" level="INFO"/>
<logger name="com.morpheus.DbSchedulerService" level="INFO"/>
<logger name="com.morpheus.deploy.AbstractDeployService" level="INFO"/>
<logger name="com.morpheus.deploy.AbstractDeployTargetService" level="INFO"/>
<logger name="com.morpheus.deploy.CloudFoundryAppDeployService" level="INFO"/>
<logger name="com.morpheus.deploy.DefaultDeployService" level="INFO"/>
<logger name="com.morpheus.deploy.DockerDeployTargetService" level="INFO"/>
<logger name="com.morpheus.deploy.GrailsDeployService" level="INFO"/>
<logger name="com.morpheus.deploy.IisDeployService" level="INFO"/>
<logger name="com.morpheus.deploy.JbossDeployService" level="INFO"/>
<logger name="com.morpheus.deploy.KubernetesDeployTargetService" level="INFO"/>
<logger name="com.morpheus.deploy.NodeDeployService" level="INFO"/>
<logger name="com.morpheus.deploy.ServerDeployTargetService" level="INFO"/>
<logger name="com.morpheus.deploy.VmDeployTargetService" level="INFO"/>
```

(continues on next page)

(continued from previous page)

```

<logger name="com.morpheus.DeploymentService" level="INFO"/>
<logger name="com.morpheus.discovery.AbstractDiscoveryService" level="INFO"/>
<logger name="com.morpheus.discovery.DatastoreCapacityDiscoveryService" level=
↳ "INFO"/>
<logger name="com.morpheus.discovery.DiscoveryService" level="INFO"/>
<logger name="com.morpheus.discovery.HostBalancingDiscoveryService" level="INFO
↳ "/>
<logger name="com.morpheus.discovery.HostCapacityDiscoveryService" level="INFO
↳ "/>
<logger name="com.morpheus.discovery.ReservationRecommendationDiscoveryService
↳ " level="INFO"/>
<logger name="com.morpheus.discovery.ShutdownDiscoveryService" level="INFO"/>
<logger name="com.morpheus.discovery.SizeDiscoveryService" level="INFO"/>
<logger name="com.morpheus.dns.AbstractDnsService" level="INFO"/>
<logger name="com.morpheus.dns.BindDnsService" level="INFO"/>
<logger name="com.morpheus.dns.ConsulDnsService" level="INFO"/>
<logger name="com.morpheus.dns.DNSProvider" level="INFO"/>
<logger name="com.morpheus.dns.DnsService" level="INFO"/>
<logger name="com.morpheus.dns.MicrosoftDnsService" level="INFO"/>
<logger name="com.morpheus.dns.PluginDnsService" level="INFO"/>
<logger name="com.morpheus.dns.PowerDnsService" level="INFO"/>
<logger name="com.morpheus.ElasticCleanupService" level="INFO"/>
<logger name="com.morpheus.EnvironmentService" level="INFO"/>
<logger name="com.morpheus.EnvironmentVariableService" level="INFO"/>
<logger name="com.morpheus.ExecuteScheduleTypeService" level="INFO"/>
<logger name="com.morpheus.ExecutionRequestService" level="INFO"/>
<logger name="com.morpheus.export.AccountInvoiceExportService" level="INFO"/>
<logger name="com.morpheus.export.CodeRepositoryExportService" level="INFO"/>
<logger name="com.morpheus.export.DeploymentExportService" level="INFO"/>
<logger name="com.morpheus.export.ExecuteScheduleTypeExportService" level="INFO
↳ "/>
<logger name="com.morpheus.export.ExportService" level="INFO"/>
<logger name="com.morpheus.export.InstanceExportService" level="INFO"/>
<logger name="com.morpheus.export.integrations.AdminIntegrationExportService"
↳
↳ level="INFO"/>
<logger name="com.morpheus.export.integrations.
↳ AutomationIntegrationExportService" level="INFO"/>
<logger name="com.morpheus.export.integrations.BackupIntegrationExportService"
↳
↳ level="INFO"/>
<logger name="com.morpheus.export.integrations.
↳ CertificateIntegrationExportService" level="INFO"/>
<logger name="com.morpheus.export.integrations.DeployIntegrationExportService"
↳
↳ level="INFO"/>
<logger name="com.morpheus.export.integrations.NetworkIntegrationExportService
↳ " level="INFO"/>
<logger name="com.morpheus.export.LoadBalancerExpertService" level="INFO"/>
<logger name="com.morpheus.export.LoadBalancerInstancesExportService" level=
↳ "INFO"/>
<logger name="com.morpheus.export.NetworkDomainExportService" level="INFO"/>
<logger name="com.morpheus.export.NetworkExportService" level="INFO"/>
<logger name="com.morpheus.export.NetworkGroupExportService" level="INFO"/>
<logger name="com.morpheus.export.NetworkPoolExportService" level="INFO"/>

```

(continues on next page)

(continued from previous page)

```
<logger name="com.morpheus.export.NetworkRouterExportService" level="INFO"/>
<logger name="com.morpheus.export.NetworkSecurityGroupExportService" level=
↪ "INFO"/>
<logger name="com.morpheus.export.PowerScheduleTypeExportService" level="INFO"/
↪ >
<logger name="com.morpheus.export.ServerExportService" level="INFO"/>
<logger name="com.morpheus.export.ServerGroupExportService" level="INFO"/>
<logger name="com.morpheus.export.ServicePlanExportService" level="INFO"/>
<logger name="com.morpheus.export.TaskExportService" level="INFO"/>
<logger name="com.morpheus.export.ThresholdExportService" level="INFO"/>
<logger name="com.morpheus.export.UserExportService" level="INFO"/>
<logger name="com.morpheus.export.UserGroupExportService" level="INFO"/>
<logger name="com.morpheus.export.WorkflowExportService" level="INFO"/>
<logger name="com.morpheus.FileCopyRequestService" level="INFO"/>
<logger name="com.morpheus.GlobalSearchService" level="INFO"/>
<logger name="com.morpheus.host.AbstractHostService" level="INFO"/>
<logger name="com.morpheus.host.DockerHostService" level="INFO"/>
<logger name="com.morpheus.host.ExternalKubernetesHostService" level="INFO"/>
<logger name="com.morpheus.host.KubernetesHostService" level="INFO"/>
<logger name="com.morpheus.host.SwarmHostService" level="INFO"/>
<logger name="com.morpheus.HttpClientService" level="INFO"/>
<logger name="com.morpheus.hub.MorpheusHubQueueService" level="INFO"/>
<logger name="com.morpheus.hub.MorpheusHubService" level="INFO"/>
<logger name="com.morpheus.hub.MorpheusHubSyncService" level="INFO"/>
<logger name="com.morpheus.imagebuild.ImageBuildService" level="INFO"/>
<logger name="com.morpheus.ImageCacheService" level="INFO"/>
<logger name="com.morpheus.instance.InstanceUpgradeService" level="INFO"/>
<logger name="com.morpheus.InstanceService" level="INFO"/>
<logger name="com.morpheus.InstanceTypeService" level="INFO"/>
<logger name="com.morpheus.integration.AbstractIntegrationService" level="INFO
↪ "/>
<logger name="com.morpheus.integration.CherwellIntegrationService" level="INFO
↪ "/>
<logger name="com.morpheus.integration.GitRepoService" level="INFO"/>
<logger name="com.morpheus.integration.RemedyIntegrationService" level="INFO"/>
<logger name="com.morpheus.integration.RunDeckIntegrationService" level="INFO"/
↪ >
<logger name="com.morpheus.integration.SalesForceIntegrationService" level=
↪ "INFO"/>
<logger name="com.morpheus.integration.ScribeService" level="INFO"/>
<logger name="com.morpheus.integration.ServiceNowIntegrationService" level=
↪ "INFO"/>
<logger name="com.morpheus.integration.TerraformService" level="INFO"/>
<logger name="com.morpheus.jobs.AbstractJobExecutorService" level="INFO"/>
<logger name="com.morpheus.jobs.JobExecutor" level="INFO"/>
<logger name="com.morpheus.jobs.KubernetesJobExecutorService" level="INFO"/>
<logger name="com.morpheus.jobs.SecurityScanExecutorService" level="INFO"/>
<logger name="com.morpheus.jobs.TaskJobExecutorService" level="INFO"/>
<logger name="com.morpheus.jobs.WorkflowJobExecutorService" level="INFO"/>
<logger name="com.morpheus.JobService" level="INFO"/>
<logger name="com.morpheus.KeyPairService" level="INFO"/>
<logger name="com.morpheus.library.LayoutService" level="INFO"/>
```

(continues on next page)

(continued from previous page)

```

<logger name="com.morpheus.LicenseService" level="INFO"/>
<logger name="com.morpheus.LoadBalancerPriceManagerService" level="INFO"/>
<logger name="com.morpheus.LocalizationService" level="INFO"/>
<logger name="com.morpheus.LocalRepoService" level="INFO"/>
<logger name="com.morpheus.log.AbstractLogService" level="INFO"/>
<logger name="com.morpheus.log.LogRhythmLogService" level="INFO"/>
<logger name="com.morpheus.log.SplunkLogService" level="INFO"/>
<logger name="com.morpheus.log.SyslogLogService" level="INFO"/>
<logger name="com.morpheus.LogService" level="INFO"/>
<logger name="com.morpheus.maint.UpdateService" level="INFO"/>
<logger name="com.morpheus.MarketplaceClientService" level="INFO"/>
<logger name="com.morpheus.MarshallService" level="INFO"/>
<logger name="com.morpheus.MetadataTagService" level="INFO"/>
<logger name="com.morpheus.migration.AbstractMigrationService" level="INFO"/>
<logger name="com.morpheus.migration.HypervisorMigrationService" level="INFO"/>
<logger name="com.morpheus.migration.LvmMigrationService" level="INFO"/>
<logger name="com.morpheus.migration.MigrationService" level="INFO"/>
<logger name="com.morpheus.migration.WindowsMigrationService" level="INFO"/>
<logger name="com.morpheus.monitoring.AlerterService" level="INFO"/>
<logger name="com.morpheus.monitoring.AlertRuleService" level="INFO"/>
<logger name="com.morpheus.monitoring.AvailabilityService" level="INFO"/>
<logger name="com.morpheus.monitoring.CheckAgentService" level="INFO"/>
<logger name="com.morpheus.monitoring.IncidentService" level="INFO"/>
<logger name="com.morpheus.monitoring.MonitorAppService" level="INFO"/>
<logger name="com.morpheus.monitoring.MonitorChartingService" level="INFO"/>
<logger name="com.morpheus.monitoring.MonitorCheckManagementService" level=
↪ "INFO"/>
<logger name="com.morpheus.monitoring.MonitorCheckService" level="INFO"/>
<logger name="com.morpheus.monitoring.MonitoringService" level="INFO"/>
<logger name="com.morpheus.monitoring.MonitorService" level="INFO"/>
<logger name="com.morpheus.monitoring.MorpheusMonitorService" level="INFO"/>
<logger name="com.morpheus.monitoring.NewRelicService" level="INFO"/>
<logger name="com.morpheus.monitoring.ServiceNowService" level="INFO"/>
<logger name="com.morpheus.MorpheusComputeService" level="INFO"/>
<logger name="com.morpheus.MorpheusPackageService" level="INFO"/>
<logger name="com.morpheus.MorpheusSecurityService" level="INFO"/>
<logger name="com.morpheus.MotdService" level="INFO"/>
<logger name="com.morpheus.network.A10LoadBalancerService" level="INFO"/>
<logger name="com.morpheus.network.AbstractLoadBalancerService" level="INFO"/>
<logger name="com.morpheus.network.AbstractNetworkPoolService" level="INFO"/>
<logger name="com.morpheus.network.AbstractNetworkRegistryService" level="INFO
↪ "/>
<logger name="com.morpheus.network.AbstractNetworkSecurityService" level="INFO
↪ "/>
<logger name="com.morpheus.network.AbstractNetworkService" level="INFO"/>
<logger name="com.morpheus.network.AciNetworkSecurityService" level="INFO"/>
<logger name="com.morpheus.network.AciNetworkService" level="INFO"/>
<logger name="com.morpheus.network.AviLoadBalancerService" level="INFO"/>
<logger name="com.morpheus.network.BluecatNetworkPoolService" level="INFO"/>
<logger name="com.morpheus.network.BootService" level="INFO"/>
<logger name="com.morpheus.network.CitrixNetScalerLoadBalancerService" level=
↪ "INFO"/>

```

(continues on next page)

(continued from previous page)

```

<logger name="com.morpheus.network.CloudPluginNetworkService" level="INFO"/>
<logger name="com.morpheus.network.ConsulRegistryService" level="INFO"/>
<logger name="com.morpheus.network.ConsulService" level="INFO"/>
<logger name="com.morpheus.network.F5BigIpLoadBalancerService" level="INFO"/>
<logger name="com.morpheus.network.F5LineRateLoadBalancerService" level="INFO"/
↪>
<logger name="com.morpheus.network.FirewallService" level="INFO"/>
<logger name="com.morpheus.network.FortiADCLoadBalancerService" level="INFO"/>
<logger name="com.morpheus.network.HaproxyLoadBalancerService" level="INFO"/>
<logger name="com.morpheus.network.InfobloxNetworkPoolService" level="INFO"/>
<logger name="com.morpheus.network.InternalLoadBalancerService" level="INFO"/>
<logger name="com.morpheus.network.InternalNetworkSecurityService" level="INFO
↪"/>
<logger name="com.morpheus.network.InternalNetworkService" level="INFO"/>
<logger name="com.morpheus.network.IPAMProvider" level="INFO"/>
<logger name="com.morpheus.network.KubernetesRegistryService" level="INFO"/>
<logger name="com.morpheus.network.LoadBalancerService" level="INFO"/>
<logger name="com.morpheus.network.LocalFirewallService" level="INFO"/>
<logger name="com.morpheus.network.MorpheusNetworkPoolService" level="INFO"/>
<logger name="com.morpheus.network.MorpheusRegistryService" level="INFO"/>
<logger name="com.morpheus.network.NetScalerLoadBalancerService" level="INFO"/>
<logger name="com.morpheus.network.NetworkConfigService" level="INFO"/>
<logger name="com.morpheus.network.NetworkPoolService" level="INFO"/>
<logger name="com.morpheus.network.NetworkRegistryService" level="INFO"/>
<logger name="com.morpheus.network.NetworkSecurityService" level="INFO"/>
<logger name="com.morpheus.network.NetworkService" level="INFO"/>
<logger name="com.morpheus.network.NetworkServicesService" level="INFO"/>
<logger name="com.morpheus.network.NutanixNetworkPoolService" level="INFO"/>
<logger name="com.morpheus.network.PaloAltoNetworkService" level="INFO"/>
<logger name="com.morpheus.network.PhipamNetworkPoolService" level="INFO"/>
<logger name="com.morpheus.network.PluginNetworkPoolService" level="INFO"/>
<logger name="com.morpheus.network.PxeService" level="INFO"/>
<logger name="com.morpheus.network.SolarWindsNetworkPoolService" level="INFO"/>
<logger name="com.morpheus.network.StealthNetworkSecurityService" level="INFO"/
↪>
<logger name="com.morpheus.NetworkDomainService" level="INFO"/>
<logger name="com.morpheus.OauthProviderService" level="INFO"/>
<logger name="com.morpheus.OperationEventService" level="INFO"/>
<logger name="com.morpheus.OptionSourcePluginService" level="INFO"/>
<logger name="com.morpheus.OptionSourceService" level="INFO"/>
<logger name="com.morpheus.OptionTypeListService" level="INFO"/>
<logger name="com.morpheus.OptionTypeService" level="INFO"/>
<logger name="com.morpheus.os.LinuxOsService" level="INFO"/>
<logger name="com.morpheus.os.WindowsOsService" level="INFO"/>
<logger name="com.morpheus.PermissionService" level="INFO"/>
<logger name="com.morpheus.plugin.AbstractPluginProviderManagerService" level=
↪"INFO"/>
<logger name="com.morpheus.plugin.backup.BackupProviderPluginManagerService" ↪
↪level="INFO"/>
<logger name="com.morpheus.plugin.backup.MorpheusBackupImplService" level="INFO
↪"/>
<logger name="com.morpheus.plugin.backup.MorpheusBackupJobImplService" level=
↪"INFO"/>

```

(continues on next page)

(continued from previous page)

```

<logger name="com.morpheus.plugin.backup.MorpheusBackupRestoreImplService"
↳level="INFO"/>
<logger name="com.morpheus.plugin.backup.MorpheusBackupResultImplService"
↳level="INFO"/>
<logger name="com.morpheus.plugin.backup.MorpheusBackupTypeImplService" level=
↳"INFO"/>
<logger name="com.morpheus.plugin.backup.MorpheusReplicationGroupImplService"
↳level="INFO"/>
<logger name="com.morpheus.plugin.backup.MorpheusReplicationImplService" level=
↳"INFO"/>
<logger name="com.morpheus.plugin.backup.MorpheusReplicationSiteImplService"
↳level="INFO"/>
<logger name="com.morpheus.plugin.backup.MorpheusReplicationTypeImplService"
↳level="INFO"/>
<logger name="com.morpheus.plugin.compute.
↳MorpheusComputeServerInterfaceImplService" level="INFO"/>
<logger name="com.morpheus.plugin.compute.MorpheusComputeZoneFolderImplService
↳" level="INFO"/>
<logger name="com.morpheus.plugin.compute.MorpheusDatastoreImplService" level=
↳"INFO"/>
<logger name="com.morpheus.plugin.costing.MorpheusAccountInvoiceImplService"
↳level="INFO"/>
<logger name="com.morpheus.plugin.costing.MorpheusCostingImplService" level=
↳"INFO"/>
<logger name="com.morpheus.plugin.cypher.MorpheusCypherImplService" level="INFO
↳"/>
<logger name="com.morpheus.plugin.integration.
↳MorpheusAccountInventoryImplService" level="INFO"/>
<logger name="com.morpheus.plugin.integration.MorpheusIntegrationImplService"
↳level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusAccountCredentialImplService" level=
↳"INFO"/>
<logger name="com.morpheus.plugin.MorpheusAccountCredentialTypeImplService"
↳level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusCloudImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusComputePortImplService" level="INFO"/
↳>
<logger name="com.morpheus.plugin.MorpheusComputeServerImplService" level="INFO
↳"/>
<logger name="com.morpheus.plugin.MorpheusComputeTypeLayoutFactoryImplService"
↳level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusComputeTypeSetImplService" level=
↳"INFO"/>
<logger name="com.morpheus.plugin.MorpheusComputeZonePoolImplService" level=
↳"INFO"/>
<logger name="com.morpheus.plugin.MorpheusContainerTypeImplService" level="INFO
↳"/>
<logger name="com.morpheus.plugin.MorpheusContextImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusInstanceImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusMetadataTagImplService" level="INFO"/
↳>
<logger name="com.morpheus.plugin.MorpheusMetadataTagTypeImplService" level=
↳"INFO"/>

```

(continues on next page)

(continued from previous page)

```
<logger name="com.morpheus.plugin.MorpheusOperationNotificationImplService" ↵
↵ level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusOsTypeImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusPermissionImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusProcessImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusReportImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusServicePlanImplService" level="INFO"/
↵>
<logger name="com.morpheus.plugin.MorpheusSnapshotImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusStatsImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusStorageControllerImplService" level=
↵ "INFO"/>
<logger name="com.morpheus.plugin.MorpheusStorageControllerTypeImplService" ↵
↵ level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusStorageVolumeImplService" level="INFO
↵"/>
<logger name="com.morpheus.plugin.MorpheusStorageVolumeTypeImplService" level=
↵ "INFO"/>
<logger name="com.morpheus.plugin.MorpheusTaskImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusUsageImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusVirtualImageImplService" level="INFO
↵"/>
<logger name="com.morpheus.plugin.MorpheusVirtualImageLocationImplService" ↵
↵ level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusWikiPageImplService" level="INFO"/>
<logger name="com.morpheus.plugin.network.MorpheusNetworkDomainImplService" ↵
↵ level="INFO"/>
<logger name="com.morpheus.plugin.network.
↵MorpheusNetworkDomainRecordImplService" level="INFO"/>
<logger name="com.morpheus.plugin.network.MorpheusNetworkImplService" level=
↵ "INFO"/>
<logger name="com.morpheus.plugin.network.MorpheusNetworkPoolImplService" ↵
↵ level="INFO"/>
<logger name="com.morpheus.plugin.network.MorpheusNetworkPoolIpImplService" ↵
↵ level="INFO"/>
<logger name="com.morpheus.plugin.network.MorpheusNetworkPoolRangeImplService" ↵
↵ level="INFO"/>
<logger name="com.morpheus.plugin.network.MorpheusNetworkSubnetImplService" ↵
↵ level="INFO"/>
<logger name="com.morpheus.plugin.network.MorpheusNetworkTypeImplService" ↵
↵ level="INFO"/>
<logger name="com.morpheus.plugin.PluginManagerService" level="INFO"/>
<logger name="com.morpheus.plugin.PluginProviderManagerService" level="INFO"/>
<logger name="com.morpheus.plugin.policy.MorpheusPolicyImplService" level="INFO
↵"/>
<logger name="com.morpheus.plugin.policy.MorpheusPolicyTypeImplService" level=
↵ "INFO"/>
<logger name="com.morpheus.plugin.provisioning.MorpheusProvisionImplService" ↵
↵ level="INFO"/>
<logger name="com.morpheus.plugin.web.MorpheusWebRequestImplService" level=
↵ "INFO"/>
<logger name="com.morpheus.policy.AbstractPolicyService" level="INFO"/>
```

(continues on next page)

(continued from previous page)

```
<logger name="com.morpheus.policy.BackupStoragePolicyService" level="INFO"/>
<logger name="com.morpheus.policy.MotdPolicyService" level="INFO"/>
<logger name="com.morpheus.policy.NetworkPolicyService" level="INFO"/>
<logger name="com.morpheus.policy.PolicyServiceInterface" level="INFO"/>
<logger name="com.morpheus.policy.StorageBucketQuotaPolicyService" level="INFO
↪"/>
<logger name="com.morpheus.policy.StorageServerQuotaPolicyService" level="INFO
↪"/>
<logger name="com.morpheus.policy.StorageShareQuotaPolicyService" level="INFO"/
↪>
<logger name="com.morpheus.policy.TagCompliancePolicyService" level="INFO"/>
<logger name="com.morpheus.policy.WorkflowPolicyService" level="INFO"/>
<logger name="com.morpheus.PolicyService" level="INFO"/>
<logger name="com.morpheus.PowerScheduleService" level="INFO"/>
<logger name="com.morpheus.PowerScheduleTypeService" level="INFO"/>
<logger name="com.morpheus.PriceManagerService" level="INFO"/>
<logger name="com.morpheus.PricePlanService" level="INFO"/>
<logger name="com.morpheus.ProcessService" level="INFO"/>
<logger name="com.morpheus.ProfileService" level="INFO"/>
<logger name="com.morpheus.project.ProjectService" level="INFO"/>
<logger name="com.morpheus.provision.AbstractBoxProvisionService" level="INFO"/
↪>
<logger name="com.morpheus.provision.AbstractProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.CloudPluginProvisioningService" level=
↪"INFO"/>
<logger name="com.morpheus.provision.DockerEngineProvisionService" level="INFO
↪"/>
<logger name="com.morpheus.provision.DockerProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.ExternalProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.HelmProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.IProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.KubernetesProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.MaasProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.ManualProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.OneviewProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.ScribeProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.SelfManagedProvisionService" level="INFO"/
↪>
<logger name="com.morpheus.provision.StandardProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.SwarmProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.TerraformProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.UcsProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.UnmanagedProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.WorkflowProvisionService" level="INFO"/>
<logger name="com.morpheus.ProvisioningService" level="INFO"/>
<logger name="com.morpheus.ProxyService" level="INFO"/>
<logger name="com.morpheus.ReferenceService" level="INFO"/>
<logger name="com.morpheus.report.AbstractReportService" level="INFO"/>
<logger name="com.morpheus.report.AmazonCoverageReportService" level="INFO"/>
<logger name="com.morpheus.report.AmazonSavingsReportService" level="INFO"/>
<logger name="com.morpheus.report.AmazonUtilizationReportService" level="INFO"/
↪>
```

(continues on next page)

(continued from previous page)

```
<logger name="com.morpheus.report.CloudAppCapacityReportService" level="INFO"/>
<logger name="com.morpheus.report.CloudAppUsageReportService" level="INFO"/>
<logger name="com.morpheus.report.CloudCapacityReportService" level="INFO"/>
<logger name="com.morpheus.report.CloudInstanceTypeCapacityReportService"
↳level="INFO"/>
<logger name="com.morpheus.report.CloudInstanceTypeUsageReportService" level=
↳"INFO"/>
<logger name="com.morpheus.report.CloudInventoryReportService" level="INFO"/>
<logger name="com.morpheus.report.CloudUsageReportService" level="INFO"/>
<logger name="com.morpheus.report.CostReportService" level="INFO"/>
<logger name="com.morpheus.report.InventoryReportService" level="INFO"/>
<logger name="com.morpheus.report.InvoiceReportService" level="INFO"/>
<logger name="com.morpheus.report.MigrationReportService" level="INFO"/>
<logger name="com.morpheus.report.PluginReportService" level="INFO"/>
<logger name="com.morpheus.report.ReportService" level="INFO"/>
<logger name="com.morpheus.report.TenantUsageReportService" level="INFO"/>
<logger name="com.morpheus.report.TimeSeriesCostReportService" level="INFO"/>
<logger name="com.morpheus.RoleService" level="INFO"/>
<logger name="com.morpheus.RpcService" level="INFO"/>
<logger name="com.morpheus.scale.AbstractScaleService" level="INFO"/>
<logger name="com.morpheus.scale.MorpheusScaleService" level="INFO"/>
<logger name="com.morpheus.ScaleService" level="INFO"/>
<logger name="com.morpheus.scribe.ScribeLibraryService" level="INFO"/>
<logger name="com.morpheus.ScriptConfigService" level="INFO"/>
<logger name="com.morpheus.sdn.AbstractSdnService" level="INFO"/>
<logger name="com.morpheus.sdn.MorpheusSdnService" level="INFO"/>
<logger name="com.morpheus.sdn.OvsService" level="INFO"/>
<logger name="com.morpheus.sdn.VethSdnService" level="INFO"/>
<logger name="com.morpheus.security.AbstractSecurityScanService" level="INFO"/>
<logger name="com.morpheus.security.ScapScanService" level="INFO"/>
<logger name="com.morpheus.security.SecurityScanService" level="INFO"/>
<logger name="com.morpheus.SecurityGroupService" level="INFO"/>
<logger name="com.morpheus.SequenceService" level="INFO"/>
<logger name="com.morpheus.ServerScriptService" level="INFO"/>
<logger name="com.morpheus.ServerService" level="INFO"/>
<logger name="com.morpheus.ServicePlanService" level="INFO"/>
<logger name="com.morpheus.SettingsService" level="INFO"/>
<logger name="com.morpheus.SetupService" level="INFO"/>
<logger name="com.morpheus.SiteService" level="INFO"/>
<logger name="com.morpheus.SnapshotPriceManagerService" level="INFO"/>
<logger name="com.morpheus.SnapshotService" level="INFO"/>
<logger name="com.morpheus.StatsService" level="INFO"/>
<logger name="com.morpheus.StatusService" level="INFO"/>
<logger name="com.morpheus.storage.AbstractStorageServerService" level="INFO"/>
<logger name="com.morpheus.storage.AbstractStorageService" level="INFO"/>
<logger name="com.morpheus.storage.BasicStorageService" level="INFO"/>
<logger name="com.morpheus.storage.CephStorageService" level="INFO"/>
<logger name="com.morpheus.storage.EcsStorageService" level="INFO"/>
<logger name="com.morpheus.storage.IsilonStorageService" level="INFO"/>
<logger name="com.morpheus.storage.KubernetesStorageService" level="INFO"/>
<logger name="com.morpheus.storage.NfsStorageService" level="INFO"/>
<logger name="com.morpheus.storage.QnapFileStationService" level="INFO"/>
```

(continues on next page)

(continued from previous page)

```
<logger name="com.morpheus.storage.StorageServerService" level="INFO"/>
<logger name="com.morpheus.storage.StorageVolumeService" level="INFO"/>
<logger name="com.morpheus.storage.ThreeParStorageService" level="INFO"/>
<logger name="com.morpheus.StorageProviderService" level="INFO"/>
<logger name="com.morpheus.SubAccountService" level="INFO"/>
<logger name="com.morpheus.task.AbstractTaskService" level="INFO"/>
<logger name="com.morpheus.task.AnsibleTaskService" level="INFO"/>
<logger name="com.morpheus.task.AnsibleTowerTaskService" level="INFO"/>
<logger name="com.morpheus.task.ChefTaskService" level="INFO"/>
<logger name="com.morpheus.task.ContainerScriptTaskService" level="INFO"/>
<logger name="com.morpheus.task.ContainerTemplateTaskService" level="INFO"/>
<logger name="com.morpheus.task.EmailTaskService" level="INFO"/>
<logger name="com.morpheus.task.ExecutableTaskInterface" level="INFO"/>
<logger name="com.morpheus.task.GroovyTaskService" level="INFO"/>
<logger name="com.morpheus.task.HttpTaskService" level="INFO"/>
<logger name="com.morpheus.task.JavascriptTaskService" level="INFO"/>
<logger name="com.morpheus.task.JRubyTaskService" level="INFO"/>
<logger name="com.morpheus.task.LocalScriptTaskService" level="INFO"/>
<logger name="com.morpheus.task.PuppetTaskService" level="INFO"/>
<logger name="com.morpheus.task.PythonTaskService" level="INFO"/>
<logger name="com.morpheus.task.RestartTaskService" level="INFO"/>
<logger name="com.morpheus.task.ShellTaskService" level="INFO"/>
<logger name="com.morpheus.task.TaskConfigService" level="INFO"/>
<logger name="com.morpheus.task.TaskService" level="INFO"/>
<logger name="com.morpheus.task.VroTaskService" level="INFO"/>
<logger name="com.morpheus.task.WinrmTaskService" level="INFO"/>
<logger name="com.morpheus.task.WriteAttributesTaskService" level="INFO"/>
<logger name="com.morpheus.trust.AbstractCredentialService" level="INFO"/>
<logger name="com.morpheus.trust.CredentialProvider" level="INFO"/>
<logger name="com.morpheus.trust.CredentialService" level="INFO"/>
<logger name="com.morpheus.trust.CypherCredentialService" level="INFO"/>
<logger name="com.morpheus.trust.InternalCredentialService" level="INFO"/>
<logger name="com.morpheus.trust.PluginCredentialService" level="INFO"/>
<logger name="com.morpheus.UsageLimitService" level="INFO"/>
<logger name="com.morpheus.UserGroupService" level="INFO"/>
<logger name="com.morpheus.UserManagementService" level="INFO"/>
<logger name="com.morpheus.vdi.VdiAppService" level="INFO"/>
<logger name="com.morpheus.vdi.VdiGatewayService" level="INFO"/>
<logger name="com.morpheus.vdi.VdiPoolService" level="INFO"/>
<logger name="com.morpheus.VirtualImagePriceManagerService" level="INFO"/>
<logger name="com.morpheus.VirtualImageService" level="INFO"/>
<logger name="com.morpheus.WikiPageService" level="INFO"/>
<logger name="com.morpheus.worker.DistributedWorkerService" level="INFO"/>
<logger name="com.morpheus.ZoneFolderService" level="INFO"/>
<logger name="com.morpheus.ZoneMarketplaceService" level="INFO"/>
<logger name="com.morpheus.ZonePoolService" level="INFO"/>
<logger name="com.morpheus.ZoneRegionService" level="INFO"/>
<logger name="com.morpheus.ZoneService" level="INFO"/>
```

Audit logs

1. To set up CEF/SIEM auditing export, add the below appender above or below the other appenders in the logback.xml configuration file:

```
<appender name="AUDIT" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>/var/log/morpheus/morpheus-ui/audit.log</file>
  <rollingPolicy class="ch.qos.logback.core.rolling.
↪SizeAndTimeBasedRollingPolicy">
    <fileNamePattern>audit.%d{yyyy-MM-dd}.%i.log</fileNamePattern>
    <maxFileSize>50MB</maxFileSize>
    <maxHistory>30</maxHistory>
  </rollingPolicy>
  <encoder>
    <pattern>[%d] [%thread] %-5level %logger{15} - %maskedMsg %n</pattern>
  </encoder>
</appender>
```

.. note:: ``maxFileSize`` and ``maxHistory`` values can be updated as needed.

2. Add the below logger above or below the other loggers in the logback.xml configuration file (make sure it is below, not above, the appender from the previous step or an error will occur):

```
<logger name="com.morpheus.AuditLogService" level="INFO" additivity="false">
  <appender-ref ref="AUDIT" />
</logger>
```

3. Once you have done this, you need to restart the Morpheus Application server:

```
morpheus-ctl stop morpheus-ui
```

Note: Please be aware this will stop the web interface for Morpheus.

4. Once the service has stopped enter the following at the xml prompt to restart (if the service does not stop, replace stop with graceful-kill and retry)

```
morpheus-ctl start morpheus-ui
```

5. To know when the UI is up and running you can run the following command

```
morpheus-ctl tail morpheus-ui
```

Once you see the ASCII art show up you will be able to log back into the User Interface. A new audit file will have been created called audit.log and will be found in the default Morpheus log path which is /var/log/morpheus/morpheus-ui/

This is only an example and other configurations are possible, such as creating an appender definition for your SIEM audit database product.

IPv6

Overview

There may be situations where instances only have IPv6 routing available. Morpheus fully supports IPv6 for the appliance and agent comms.

To enable IPv6 listener on the Morpheus appliance, you must modify the NGINX listeners.

Configuring NGINX Listeners

1. Confirm IPv6 is enabled and IP address is applied within the Morpheus underlying OS.
2. Modify the `/opt/morpheus/embedded/nginx/conf/sites-available/morpheus.conf` file, and add the following listening under the server block at the top:

```
listen [::]:80;
```

3. Modify the `/opt/morpheus/embedded/nginx/conf/sites-available/morpheus-ssl.conf` file, and add the following listening under the server block at the top:

```
listen [::]:443 ssl;
```

4. Restart the NGINX service:

```
morpheus-ctl restart nginx  
ok: run: nginx: (pid 47868) 0s
```

5. The site should now be resolvable via IPv6. To test you should be able to do the following:

```
curl -k -6 https://[<IPv6 Address>]/ping  
MORPHEUS PING
```

5.1.6 Initial Appliance Setup

Appliance Setup

After installation, log into the appliance at the URL presented upon completion. An initial setup wizard walks through the first account and user creations.

1. Enter Master Account name
 - Typically, the Master Account name is your Company name.
2. Create Master User
 - First Name
 - Last Name
 - Username
 - Email Address

- Password * Must be at least 8 characters long and contain one each of the following: Uppercase letter, lowercase letter, Number, Special Character
3. Enter Appliance Name & Appliance URL
 - The Appliance Name is used for white labeling and as a reference for multi-appliance installations.
 - The Appliance URL is the URL all provisioned instances will report back to. Example: <https://example.morpheusdata.com>. The Appliance URL can be changed later, and also set to a different URL per cloud integration.
 4. Optionally Enable or Disable Backups, Monitoring, or Logs from this screen.

Note: You may adjust these settings from the Administration section.

Note: The Master Account name is the top-level admin account.

Note: The Master User is the system super user and will have full access privileges.

Upon completing of the initial appliance setup, you will be taken to the Administration > Settings page, where you will add your License Key.

Login Methods

Master Tenant

- Enter username (or email address) and password

Subtenant

To login, subtenants can either use the master tenant URL with `subtenant\username` formatting:

Example: I have a username `subuser` that belongs to a tenant with the subdomain `subaccount`. When logging in from the main login url, I would now need to enter in: `subaccount\subuser`

Or use the tenant specific URL which can be found and configured under Administration > Tenants > Select Tenant > Identity Sources.

Tenants > Morpheus-Tenant > Identity Sources

Morpheus-Tenant

ACCOUNT LOGIN URL

<https://sandbox.morpheusdata.com/login/account/79> EDIT

Search Q + ADD IDENTITY SOURCE

TYPE	NAME	DETAILS	ACTIVE
------	------	---------	--------

Important: In 3.4.0+ Subtenant users will no longer be able to login from the main login url without specifying their subdomain.

Configure Cloud-init Global Settings

When using cloud-init, cloudbase-init, VMware Tools customizations, or Nutanix Sysprep, Global Linux User and Windows Administrator credentials can be set using the settings in Administration > Settings > Provisioning. It is recommended to define these settings after installation unless credentials are defined per Virtual Image for Provisioning.

Add a License Key

In order to provision anything in Morpheus, a Morpheus License Key must be applied.

If you do not already have a license key, one may be requested from <https://www.morpheushub.com> or from your Morpheus representative.

In the Administration > Settings section, select the LICENSE tab, paste your License Key and click *UPDATE*

The screenshot shows the Morpheus Administration interface. At the top, there is a navigation bar with the Morpheus logo and the text "Morpheus Data". Below this is a menu with tabs for Dashboard, Provisioning, Infrastructure, Backups, Logs, Monitoring, Reports, and Admin. The Admin tab is selected, and a sub-menu is visible with icons for Accounts, Service Plans, Roles, Users, Integrations, Provisioning, Monitoring, Backups, Logs, and Settings. The main content area has two tabs: APPLIANCE and LICENSE. The LICENSE tab is active, showing a section titled "Purchase a License" with instructions to contact support for a license or free trial. Below this is a blue "UPGRADE" button. The next section is "Upgrade License Key" with instructions to enter a new license key. A large text input field is provided for this purpose, followed by a blue "UPDATE" button.

When the license is accepted, your license details will populate in the Current License section.

If you receive an error message and your license is not accepted, please check it was copied in full and then contact your Morpheus representative. You can also verify the License Key and expiration at <https://www.morpheushub.com>.

5.1.7 Core Functionality

Morpheus Discovery

Morpheus has the ability to ingest existing environments. Existing running workloads will be inventoried into Morpheus and displayed in the UI. In 5-7 days Morpheus will start making recommendations based off of usage and pricing.

Note: Workloads that are inventoried do not have to be converted to managed.

Once inventoried, Morpheus can provide valuable data for that instance:

- Morpheus will know about networks
- Start aggregating cost on public clouds
- Start tracking usage
- Some Clouds offer statistical details (Amazon / VMware)
- Power Status

Right away inventorying existing environments will provide you with immediate insight to that environment. Once an existing workload has been discovered it can be converted to managed. Once converted to managed, Morpheus can deliver more capabilities and features.

Note: Workloads do not need the agent installed to be managed

Once a workload is managed:

- Enforce expiration/shutdown policies. This helps reign in environments (sprawl) and reduce cost.
- Can tell what instance type it is
- Can install agent (agent is optional)
- Installing agent provides credentials and allows you to run workflows against it (day 2 operations)

Morpheus Agent

Overview

The Morpheus Agent is an important and powerful facet of Morpheus as an orchestration tool. Though it is not required, which is one unique capability of our platform versus some of our competitors, it is recommended for use as it brings many benefits. Not only does it provide statistics for the guest operating system and resource utilization, it also brings along with it monitoring and log aggregation capabilities. After an initial brownfield discovery, Users can decide to convert unmanaged VMs to managed.

Note: **Agent installation is not required to manage an Instance.** If you don't have the Agent installed, we make every effort to aggregate stats. These will vary based on the Cloud and can be more limited or less accurate without utilizing Morpheus Agent.

The Morpheus Agent is very lightweight and secure. It does not open any inbound network ports but rather only opens an outbound connection back to the Morpheus appliance over port 443 (HTTPS or WSS protocol). This allows for a bidirectional command bus where instructions can be sent to orchestrate a workload without needing access to things like SSH or WinRM. The tool can even be installed at provision time via technologies like Cloud-Init, such that the

Morpheus appliance itself doesn't even need direct network access to the VLAN under which the workload resides. By doing this we address many of the network security concerns that come with the use of an agent while demonstrating its security and analytics benefits. We can even use this statistical data at the guest OS level rather than the hypervisor level to provide extremely precise right-sizing recommendations.

Morpheus Agent Key Features

While optional, Morpheus Agent provides many benefits and features in areas of log aggregation, security, automation, monitoring, and more. This page contains the complete summary of its key features and benefits.

Logging

The installed Morpheus Agent captures application logs and sends them back to the Morpheus appliance. The Agent buffers and compresses logs, sending them in chunks to minimize packet transfers. If desired, users may set up forwarding to an external syslog platform though for most users Morpheus internal logging functionality is sufficient. Agent logs can be viewed in the UI at Monitoring > Logs. Filtering and search tools are available, even supporting [Lucene search query syntax](#). Logs may also be exported.

Monitoring and Guidance

Morpheus provides robust monitoring into the workloads it manages. For example, from Instance detail pages, usage metrics are tracked on the Summary and Monitoring tabs. The available metrics are significantly improved when Morpheus Agent is installed on the workload. Morpheus will make a best effort to gather this information in the absence of an installed Agent but for some Cloud types this is not possible. The table below shows the usage metrics Morpheus can gather with and without the Agent.

Category	Statistic	With Agent	Without Agent
Memory	Max Memory	Yes	Yes
Memory	Used Memory	Yes	No
Memory	Cache Memory	Yes	No
Storage	Max Storage	Yes	Yes
Storage	Used Memory	Yes	No
Processing	System CPU %	Yes	Yes
Processing	User CPU %	Yes	No
IOPS	Total IOPS	Yes	No
IOPS	IOPS Read	Yes	No
IOPS	IOPS Write	Yes	No
Networking	Net TX Rate	Yes	No
Networking	Net RX Rate	Yes	No
Other	Agent Command Bus	Yes	No
Other	Log Aggregation	Yes	No
Other	Health & Monitoring	Yes	No

In addition to usage and monitoring, Morpheus also provides a useful guidance feature (Operations > Guidance). Guidance analyzes your managed workloads and makes cost-saving or performance recommendations. The effectiveness of this feature is greatly enhanced when Morpheus Agent is installed on your workloads.

Script Execution

The Morpheus Agent initiates an outbound connection from the managed workload to the appliance over TCP port 443. This establishes a bidirectional command bus which allows Morpheus to orchestrate automation on managed machines without stored credentials. Many different [Task types](#) are supported and Tasks can be stacked into Operational or Provisioning Workflows to create logical automation routines. SSH or WinRM connectivity, as well as credentials, are required for Task execution when Morpheus Agent is not installed.

Setting File Templates

File Templates are stored templated text files, primarily config files (for example, `my.cnf` or `morpheus.rb`). Users have access to the full map of Morpheus variables for injecting values custom to the specific workload at provision time. Morpheus Agent can transfer files generated from templates to managed nodes.

Firewall Management

When “Local Firewall” is enabled for a Cloud (see the Advanced Options section on the Add/Edit Cloud modal), the Morpheus Agent can manage host or VM IP Table (firewall) rules for Linux workloads.

Summary and Additional benefits:

- Installation is optional for provisioned and managed VMs
- The Linux agent can be shrunk and should be less than 0.2% peak
- Provides a command bus such that Morpheus doesn't need credentials to access a box
- Can still manage workflows if credentials are changed
- SSH agent can be disabled and still get access to the box
- Agent can be installed over Cloud-init, Windows unattend.xml, VMware Tools, SSH, WinRM, Cloudbase-Init, or manually
- Makes a single, persistent connection over HTTPS websocket and runs as a service
- Buffers and compresses logs, then sends them in chunks to minimize packets
- Supports syslog forwarding
- Accepts commands, executes commands, writes files, and manipulates firewalls
- Significantly enhances Guidance recommendations through enhanced statistics

Note: The Morpheus Agent is required for managed Docker, Kubernetes, SCVMM, Hyper-V, KVM, and ESXi Hosts (for ESXi-only Cloud, not vCenter Clouds).

Morpheus Agent OS Support

Name	Bit Count	Category	Code	OS Family	OS Version	Platform	Vend
Almalinux Linux 8 64-bit	64	almalinux	almalinux.8.64	almalinux	8	linux	almali
Almalinux Linux 9 64-bit	64	almalinux	almalinux.9.64	almalinux	9	linux	almali
amazon linux 2 64-bit	64	amazonlinux	amazonlinux.2.64	centos	2	linux	amaz
centOS	32	centos	cent	rhel	all	linux	centos
centOS 6	32	centos	cent.6	rhel	6	linux	centos
centOS 6 64-bit	64	centos	cent.6.64	rhel	6	linux	centos
centOS 64-bit	64	centos	cent.64	rhel	all	linux	centos
centOS 7	32	centos	cent.7	rhel	7	linux	centos
centOS 7 64-bit	64	centos	cent.7.64	rhel	7	linux	centos
centOS 8	32	centos	cent.8	rhel	8	linux	centos
centOS 8 64-bit	64	centos	cent.8.64	rhel	8	linux	centos
centOS 8 Stream 64-bit	64	centos	cent.8.stream.64	rhel	8	linux	centos
centOS 9 64-bit	64	centos	cent.9.64	rhel	9	linux	centos
debian	32	debian	debian	debian	all	linux	debian
debian 10 64-bit	64	debian	debian.10.64	debian	10	linux	debian

Table 10 – continued from previous page

Name	Bit Count	Category	Code	OS Family	OS Version	Platform	Vendor
debian 10.9 64-bit	64	debian	debian.10.9.64	debian	10	linux	debian
debian 11 64-bit	64	debian	debian.11.64	debian	11	linux	debian
debian 6	32	debian	debian.6	debian	6	linux	debian
debian 6 64-bit	64	debian	debian.6.64	debian	6	linux	debian
debian 64-bit	64	debian	debian.64	debian	all	linux	debian
debian 7	32	debian	debian.7	debian	7	linux	debian
debian 7 64-bit	64	debian	debian.7.64	debian	7	linux	debian
debian 8	32	debian	debian.8	debian	8	linux	debian
debian 8 64-bit	64	debian	debian.8.64	debian	8	linux	debian
debian 9.13 64-bit	64	debian	debian.9.13.64	debian	9	linux	debian
debian 9.4 64-bit	64	debian	debian.9.4.64	debian	9	linux	debian
ESXi	64	esxi	esxi	NULL		ESXi	vmware
ESXi 4	64	esxi	esxi.4	NULL	4	ESXi	vmware
ESXi 5	64	esxi	esxi.5	NULL	5	ESXi	vmware
ESXi 6	64	esxi	esxi.6	NULL	6	ESXi	vmware
ESXi 7	64	esxi	esxi.7	NULL	7	ESXi	vmware
ESXi 8	64	esxi	esxi.8	NULL	8	ESXi	vmware
fedora	32	fedora	fedora	NULL	all	linux	fedora
fedora 64-bit	64	fedora	fedora.64	NULL	all	linux	fedora
linux	64	linux	linux	NULL	all	linux	linux
linux 32-bit	32	linux	linux.32	NULL	all	linux	linux
linux 64-bit	64	linux	linux.64	NULL	all	linux	linux
mac os	64	mac	mac	darwin	all	osx	apple
opensuse	32	opensuse	opensuse.32	suse	all	linux	opensuse
opensuse 15 64-bit	64	opensuse	suse.15.64	suse	15	linux	suse
opensuse 64-bit	64	opensuse	opensuse.64	suse	all	linux	opensuse
oracle enterprise	32	oracle	oracle.32	NULL	all	linux	oracle
oracle enterprise 64-bit	64	oracle	oracle.64	NULL	all	linux	oracle
oracle linux 64-bit	64	oracle	oracle.linux.64	NULL	all	linux	oracle
oracle vm server	64	oracle	ovs	NULL	all	linux	oracle
other 32-bit	32	other	other.32	NULL	all	other	other
other 64-bit	64	other	other.64	NULL	all	other	other
redhat	32	redhat	redhat	rhel	all	linux	redhat
redhat 6	32	redhat	redhat.6	rhel	6	linux	redhat
redhat 6 64-bit	64	redhat	redhat.6.64	rhel	6	linux	redhat
redhat 64-bit	64	redhat	redhat.64	rhel	all	linux	redhat
redhat 7	32	redhat	redhat.7	rhel	7	linux	redhat
redhat 7 64-bit	64	redhat	redhat.7.64	rhel	7	linux	redhat
redhat 8	32	redhat	redhat.8	rhel	8	linux	redhat
redhat 8 64-bit	64	redhat	redhat.8.64	rhel	8	linux	redhat
redhat 9 64-bit	64	redhat	redhat.9.64	rhel	9	linux	redhat
Rocky Linux 8	32	rocky	rocky.8	rocky	8	linux	rocky
Rocky Linux 8 64-bit	64	rocky	rocky.8.64	rocky	8	linux	rocky
Rocky Linux 9 64-bit	64	rocky	rocky.9.64	rocky	9	linux	rocky
solaris	32	solaris	solaris.32	NULL	all	solaris	solaris
solaris 64-bit	64	solaris	solaris.64	NULL	all	solaris	solaris
suse enterprise	32	suse	suse	suse	all	linux	suse
suse enterprise 11	32	suse	suse.11	suse	11	linux	suse
suse enterprise 11 64-bit	64	suse	suse.11.64	suse	11	linux	suse

COR

Table 10 – continued from previous page

Name	Bit Count	Category	Code	OS Family	OS Version	Platform	Vendor
suse enterprise 12	32	suse	suse.12	suse	12	linux	suse
suse enterprise 12 64-bit	64	suse	suse.12.64	suse	12	linux	suse
suse enterprise 64-bit	64	suse	suse.64	suse	all	linux	suse
ubuntu	32	ubuntu	ubuntu	debian	all	linux	canonical
ubuntu 12	32	ubuntu	ubuntu.12.04	debian	12.04	linux	canonical
ubuntu 12 64-bit	64	ubuntu	ubuntu.12.04.64	debian	12.04	linux	canonical
ubuntu 13	32	ubuntu	ubuntu.13.10	debian	13.10	linux	canonical
ubuntu 13 64-bit	64	ubuntu	ubuntu.13.10.64	debian	13.10	linux	canonical
ubuntu 14	32	ubuntu	ubuntu.14.04	debian	14.04	linux	canonical
ubuntu 14 64-bit	64	ubuntu	ubuntu.14.04.64	debian	14.04	linux	canonical
ubuntu 15	32	ubuntu	ubuntu.15.10	debian	15.10	linux	canonical
ubuntu 15 64-bit	64	ubuntu	ubuntu.15.10.64	debian	15.10	linux	canonical
ubuntu 16	32	ubuntu	ubuntu.16.04	debian	16.04	linux	canonical
ubuntu 16 64-bit	64	ubuntu	ubuntu.16.04.64	debian	16.04	linux	canonical
ubuntu 18.04	32	ubuntu	ubuntu.18.04	debian	18.04	linux	canonical
ubuntu 18.04 64-bit	64	ubuntu	ubuntu.18.04.64	debian	18.04	linux	canonical
ubuntu 20.04	32	ubuntu	ubuntu.20.04	debian	20.04	linux	canonical
ubuntu 20.04 64-bit	64	ubuntu	ubuntu.20.04.64	debian	20.04	linux	canonical
ubuntu 22.04 64-bit	64	ubuntu	ubuntu.22.04.64	debian	22.04	linux	canonical
ubuntu 64-bit	64	ubuntu	ubuntu.64	debian	all	linux	canonical
unknown	64	other	unknown	NULL	all	unknown	unknown
windows	64	windows	windows	windows	all	windows	microsoft
windows 10	32	windows	windows.10	windows	10	windows	microsoft
windows 10 64-bit	64	windows	windows.10.64	windows	10	windows	microsoft
windows 11	32	windows	windows.11	windows	11	windows	microsoft
windows 11 64-bit	64	windows	windows.11.64	windows	11	windows	microsoft
windows 7	32	windows	windows.7	windows	7	windows	microsoft
windows 7 64-bit	64	windows	windows.7.64	windows	7	windows	microsoft
windows 8	32	windows	windows.8	windows	8	windows	microsoft
windows 8 64-bit	64	windows	windows.8.64	windows	8	windows	microsoft
windows server 2003	64	windows	windows.server.2003	windows	2003	windows	microsoft
windows server 2008	64	windows	windows.server.2008	windows	2008	windows	microsoft
windows server 2008 R2	64	windows	windows.server.2008.r2	windows	2008	windows	microsoft
windows server 2012	64	windows	windows.server.2012	windows	2012	windows	microsoft
windows server 2016	64	windows	windows.server.2016	windows	2016	windows	microsoft
windows server 2019	64	windows	windows.server.2019	windows	2019	windows	microsoft
windows server 2022	64	windows	windows.server.2022	windows	2022	windows	microsoft
xen server 6.1	64	xen	xenserver.6.1	xen	6.1	linux	citrix
xen server 6.2	64	xen	xenserver.6.2	xen	6.2	linux	citrix
xen server 6.5	64	xen	xenserver.6.5	xen	6.5	linux	citrix
xen server 7.0	64	xen	xenserver.7.0	xen	7.0	linux	citrix

Note: Other Operating System types may be supported but are not tested.

Agent Installation

There are many methods to install the Morpheus Agent on supported targets. All Agent installation methods are executing a script on the target that calls back to the Morpheus appliance over port 443.

Important: All Agent installation methods require the Target (VM or Host) to resolve and reach the appliance URL over port 443. In addition to the main Appliance URL (in Administration > Settings), additional Appliance URLs can be set per cloud in the Advanced Options section of the Create/Edit Cloud modal. When this field is populated, it will override the main Appliance URL for anything provisioned into that Cloud.

Basic Installation Steps

1. An Agent installation method is used to get the install script onto the target VM or Host
2. The Agent installation script is executed on the target VM or Host, installing the Agent and all dependencies
3. The Agent is started and makes a websocket connection to the configured Appliance URL over port 443 using the target VM or Host API key

It is important to note these are three separate steps with varying requirements. The first requires a path to get the script on the target. The second requires successful execution of the Agent installation script. The third requires a successful websocket connection. Requirements for the successful execution of each step are covered in the sections below.

Tip: The Morpheus UI current log, located at `/var/log/morpheus/morpheus-ui/current`, is very helpful when troubleshooting Agent installations.

Agent Install Modes

Agent Installation Method is determined by:

- The AGENT INSTALL MODE setting on target Cloud: - SSH / WinRM / Guest Execution - Cloud Init / Unattend (when available)
- Platform / OS type on Virtual Image or target (VM or Host)
- Virtual Image configuration
- RPC Mode setting (VMware Clouds only)

Agent Installation Methods

The Morpheus Agent can be installed with a variety of automated methods. It is important to note these methods simply get the Agent install script to the target. Successful execution of the Agent install script is not directly related to the Agent install method.

SSH Morpheus makes an SSH connection to the VM or Host, CURLs, and executes the Agent install script:

```
curl -k -s "https://${applianceUrl}/api/server-script/agentInstall?apiKey=${apiKey}" | bash
```

WinRM Morpheus makes a WinRM connection to the VM or Host and executes the Agent install script

VMware Tools Morpheus executes `agentInstall.sh` or `agentInstall.ps1` via VMware Tools Guest Execution

Cloud-Init Morpheus executes agentInstall.sh via cloud-init runcmd

Cloudbase-Init Morpheus adds WindowsAgentCloudInitInstallScript to CloudbaseInitUserData

Windows Unattend Morpheus adds getWindowsAgentDownloadScript to unattend.xml (RunSynchronousCommand)

Manual User executes agentInstall.sh or agentInstall.ps1 manually on the VM or Host. These scripts can be obtained on the VM or Host detail page via Actions > Download Agent Script

SSH

Process

Morpheus makes an SSH connection to the VM or Host, CURLs, and executes the Agent install script:

```
curl -k -s "https://${applianceUrl}/api/server-script/agentInstall?apiKey=${apiKey}" |  
bash
```

Requirements

- Port 22 is open for Linux images, and SSH is enabled
- Credentials have been entered on the image if using a custom or synced image. Credentials can be entered on images in the Library > Virtual Images section

WinRM

Process

Morpheus makes a WinRM connection to the VM or Host and executes the Agent install script

Requirements

- Port 5985 must be open and WinRM enabled for Windows images
- Credentials have been entered on the image if using a custom or synced image. Credentials can be entered on images in the Library > Virtual Images section
- Administrator User (SID 500) is required for Windows Agent install

VMware Tools

Process

Morpheus executes agentInstall.sh or agentInstall.ps1 via VMware Tools Guest Execution

Requirements

- VMware Tools is installed on the template(s)
- Credentials have been entered on the Image if using an uploaded or synced image when Cloud-init, Guest Customizations, or Sysprep for Windows are not used. Credentials can be entered on Images in the Library > Virtual Images section
- Administrator User (SID 500) is required for Windows Agent install.

Cloud-Init

Process

Morpheus executes agentInstall.sh via Cloud-Init runcmd

Requirements

- Cloud-Init is installed on Virtual Image
- “IS CLOUD INIT ENABLED?” is checked (true) on the Morpheus Virtual Image record
- Cloud-Init User is configured in the Administration > Settings > Provisioning section

Cloudbase-init

Process

Morpheus adds WindowsAgentCloudInitInstallScript to CloudbaseInitUserData

Requirements

- Cloudbase-Init is installed on the Virtual Image
- “IS CLOUD INIT ENABLED?” is checked (true) on the Morpheus Virtual Image record
- Windows Administrator password defined in the Administration > Settings > Provisioning section

Windows Unattend

Process

Morpheus adds getWindowsAgentDownloadScript to unattend.xml (RunSynchronousCommand)

Requirements

VMware

- Windows Administrator password defined in the Administration > Settings > Provisioning section OR Administrator User (SID 500) and valid Windows password are defined on the Morpheus Virtual Image record
- “FORCE GUEST CUSTOMIZATION?” is checked (true) on the Morpheus Virtual Image record when using DHCP
- “IS CLOUD INIT ENABLED?” is unchecked (false) on the Morpheus Virtual Image record

Nutainx/SCVMM/Openstack

- Windows Administrator password defined in the Administration > Settings > Provisioning section OR Administrator User (SID 500) and valid Windows password are defined on the Morpheus Virtual Image record
- “ENABLED SYSPREP?” is checked (true) on the Morpheus Virtual Image record
- “IS CLOUD INIT ENABLED?” is unchecked (false) on the Morpheus Virtual Image record

Manual

Process

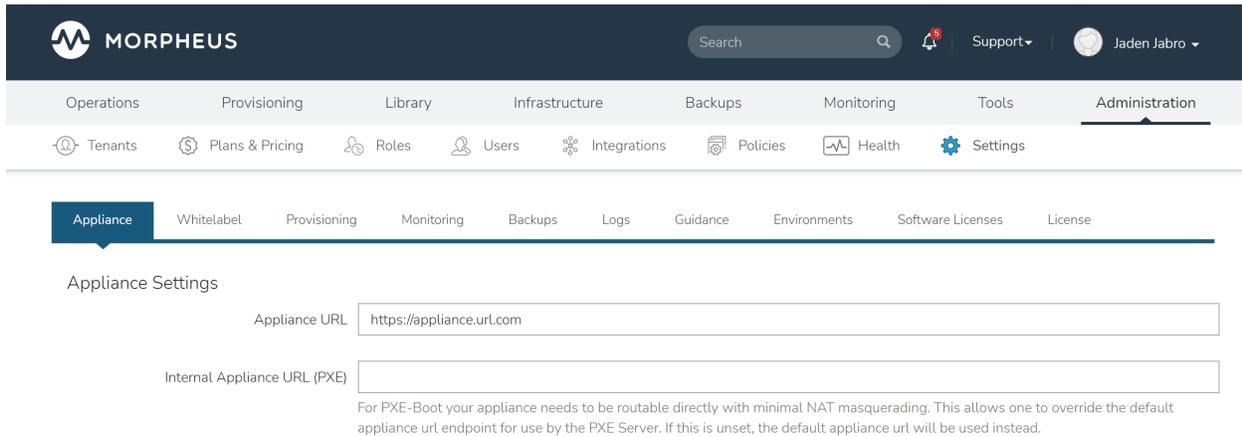
- From the VM or Host record page (`/infrastructure/servers/{id}`) run *ACTIONS* > Download Agent Script
- This will generate an Agent install script based off the target OS and platform, Appliance URL, and API key
- Manually execute the downloaded script on the Target VM or Host

Agent Install Requirements

Agent Installation Requirements							
Requirement	Agent Installation Method						
	SSH	WINRM	VMWARE TOOLS	CLOUD-INIT	CLOUD-INIT	BASE-TEND	MANUAL
Target (vm/host) to resolve and reach Appliance URL over 443	YES	YES	YES	YES	YES	YES	YES
Target (vm/host) to resolve and reach Appliance URL over 80	Ubuntu 14.04 Only		Ubuntu 14.04 Only	Ubuntu 14.04 Only			Ubuntu 14.04 Only
Websockets enabled when using load balancer	YES	YES	YES	YES	YES	YES	YES
Access to Target VM/Host on port 22	YES	NO	NO	NO	NO	NO	NO
Access to Target VM/Host on port 5985	NO	YES	NO	NO	NO	NO	NO
Vmware Tools installed and flagged on Virtual Image	NO	NO	YES	NO	NO	YES	NO
Syspreped Image and Sysprep Enabled flagged on Virtual Image (Nutanix, Openstack, SCVMM)	NO	NO	NO	NO	NO	YES	NO
Force Guest Customizations flagged on Virtual Image	NO	NO	DHCP	NO	NO	DHCP	NO
Cloud-Init installed and flagged on Virtual Image	NO	NO	NO	YES	YES	NO	NO
Global Cloud-Init user populated in Administration > Settings > Provisioning	NO	NO	NO	YES	NO	NO	NO
Windows Administrator Password populated in Administration > Settings > Provisioning	NO	NO	NO	NO	YES	YES	NO
Access to configured YUM or APT repos	NO but will cause delay in Agent Install	N/A	NO but will cause delay in Agent Install	NO but will cause delay in Agent Install	N/A	N/A	NO but will cause delay in Agent Install
.net >=4.5.2 (Windows, Morpheus >= 4.1.2)	N/A	YES	YES	N/A	YES	YES	YES
User with Sudo Access set on Virtual Image (Greenfield)	YES	N/A	YES	NO	N/A	N/A	N/A
Administrator User (SID 500) set on Virtual Image (Greenfield)	N/A	YES	YES	N/A	NO	N/A	N/A
User with Sudo Access set on VM/Host Record (Brownfield)	YES	N/A	YES	N/A	N/A	N/A	N/A
Administrator User (SID 500) set on VM/Host Record (Brownfield)	N/A	YES	YES	N/A	N/A	N/A	N/A

When provisioning an Instance, there are network and configuration requirements to consider in order to successfully install the Morpheus Agent. Typically, when a VM Instance is still in the provisioning phase long after the VM is up, the Instance is unable to reach Morpheus. Depending on the Agent install mode, it could also mean Morpheus is unable to reach the Instance.

The most common reason an Agent install fails is the provisioned Instance cannot reach the Morpheus Appliance via the Appliance URL set in Administration > Settings over port 443. When an Instance is provisioned from Morpheus, it must be able to reach the Morpheus appliance via the Appliance URL or the Agent will not be installed.



The screenshot shows the Morpheus Administration interface. The top navigation bar includes the Morpheus logo, a search bar, a notification bell with a red '5', a 'Support' dropdown, and a user profile for 'Jaden Jabro'. Below this is a main navigation menu with categories: Operations, Provisioning, Library, Infrastructure, Backups, Monitoring, Tools, and Administration (which is highlighted). Under Administration, there are sub-items: Tenants, Plans & Pricing, Roles, Users, Integrations, Policies, Health, and Settings. The 'Appliance' sub-menu is selected, showing options: Appliance, Whitelabel, Provisioning, Monitoring, Backups, Logs, Guidance, Environments, Software Licenses, and License. The 'Appliance Settings' section contains two input fields: 'Appliance URL' with the value 'https://appliance.url.com' and 'Internal Appliance URL (PXE)' which is empty. A note below the fields states: 'For PXE-Boot your appliance needs to be routable directly with minimal NAT masquerading. This allows one to override the default appliance url endpoint for use by the PXE Server. If this is unset, the default appliance url will be used instead.'

In addition to the main Appliance URL in Administration > Settings, additional Appliance URLs can be set per Cloud in the Advanced Options section of the Cloud configuration modal when creating or editing a Cloud. When this field is populated, it will override the main Appliance URL for anything provisioned into that Cloud.

Tip: The Morpheus UI current log, located at `/var/log/morpheus/morpheus-ui/current`, is very helpful when troubleshooting Agent installations.

Agent Install Methods

Morpheus Agent installation supports multiple install methods.

- SSH/WinRM
- VM Tools
- Cloud-Init & Cloudbase-Init
- Windows Unattended
- Manual

For All Agent Install Methods

When an Instance is provisioned and the Agent does not install, verify the following for any Agent install mode:

- The Morpheus Appliance URL (Administration > Settings) is both reachable and resolvable from the provisioned node
- The Appliance URL begins with <https://>, not <http://>

Note: Be sure to use <https://> even when using an IP address for the appliance.

- Inbound connectivity access to the Morpheus appliance from provisioned VMs and container hosts on port 443 (needed for Agent communication)
- Private (non-Morpheus provided) VM images and templates must have their credentials stored. These can be entered or edited in the Library > Virtual Images section by clicking the Actions dropdown on an image detail page and selecting Edit.

Note: Administrator user is required for Windows Agent install.

- The Instance does not have an IP address assigned. For scenarios without a DHCP server, static IP information must be entered by selecting the Network Type: Static in the Advanced Options section during provisioning. IP Pools can also be created in the Infrastructure > Networks > IP Pools section and added to Cloud network sections for IPAM
- DNS is not configured and the node cannot resolve the appliance. If DNS cannot be configured, the IP address of the Morpheus appliance can be used as the main or Cloud appliance

SSH

- Port 22 is open for Linux images, and SSH is enabled
- Credentials set on the image if using a custom or synced image. Credentials can be entered on images in the Library > Virtual Images section

WinRM

- Port 5985 must be open and WinRM enabled for Windows images
- Credentials have been entered on the image if using a custom or synced image. Credentials can be entered on images in the Library > Virtual Images section

Note: Administrator user is required for Windows Agent install.

VMware Tools (vmtools)

- VMware Tools is installed on the template(s)
- Credentials have been entered on the image if using custom or synced image. Credentials can be entered on images in the Library > Virtual Images section
- Sudo privileges required for Linux
- Administrator User required for Windows (SID 500)

Cloud-Init

- Cloud-Init settings configured in Administration > Settings > Provisioning section
- Cloud-Init installed on Virtual Image
- Cloud-Init enabled on Virtual Image config

Cloudbase-Init

- Windows Administrator Password defined in Administration > Settings > Provisioning section
- Cloudbase-Init installed on Virtual Image
- Cloud-Init enabled on Virtual Image config
- Cloudbase-Init is only required for OpenStack Cloud types

Note: Unattend Agent Installation and customizations are recommended over Cloudbase-Init

Windows Unattended

- Windows Administrator Password defined in Administration > Settings > Provisioning section
- VMware: Force Guest Customizations set to forced on Virtual Image config when using DHCP (Static Assignment will already force Guest Customizations)
- Nutanix & SCVMM: Virtual Image is sysprepped and shutdown, Sysprep Enabled flagged on Virtual Image config

Manual

Agent Install scripts can be downloaded from Morpheus by selecting **Actions > Download Agent Script** from an Server detail page, then run manually on the target host when required for a given managed resource. Please note the script will be unique per managed resource and should not be saved to run as needed on any arbitrary resources in the future.

When installing on Windows, continue with the steps below to complete manual installation:

- Open powershell as an administrator
- Run the `unblock-file cmdlet` against the download agent installation script:

```

Unblock-File -Path C:\Users\User01\Documents\Downloads\agentInstall.ps1

Get-ExecutionPolicy

Set-ExecutionPolicy -ExecutionPolicy Unrestricted -Scope CurrentUser

```

- After running the powershell script, ensure the script downloaded the msi and the Agent service started correctly:

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

Following setup, verify that the Agent is reporting back to the Morpheus appliance.

Restarting the Morpheus Agent

In some situations, it may necessary to restart the Morpheus Agent on the host to re-sync communication from the Agent to the Morpheus appliance.

Linux

On the target host, run `sudo morpheus-node-ctl restart morphd` and the Morpheus agent will restart. `morpheus-node-ctl status` will also show the agent status.

Windows

The Morpheus Windows Agent service can be restarted in Administrative Tools > Services.

Tip: The Morpheus Remote Console is not dependent on Agent communication and can be used to install or restart the Morpheus agent on an Instance.

Uninstall Morpheus Agent

Linux Agents

You can use the following to uninstall the linux agent (contains commands for both rpm and deb agents)

```

sudo rm /etc/apt/sources.list.d/morpheus.list \
sudo morpheus-node-ctl kill \
sudo apt-get -y purge morpheus-node \
sudo apt-get -y purge morpheus-vm-node \
sudo yum -y remove morpheus-node \
sudo yum -y remove morpheus-vm-node \
sudo yum clean all \
sudo systemctl stop morpheus-node-runsrvdir \
sudo rm -f /etc/systemd/system/morpheus-node-runsrvdir.service \
sudo systemctl daemon-reload \
sudo rm -rf /var/run/morpheus-node \
sudo rm -rf /opt/morpheus-node \

```

(continues on next page)

(continued from previous page)

```
sudo rm -rf /etc/morpheus \  
sudo rm -rf /var/log/morpheus-node \  
sudo pkill runsv \  
sudo pkill runsvdir \  
sudo pkill morphd \  
sudo usermod -l morpheus-old morpheus-node \  

```

Windows Agents

```
$app = Get-WmiObject -Class Win32_Product -Filter "Name = 'Morpheus Windows Agent'"  
$app.Uninstall()
```

CentOS/RHEL 7 Images

For custom CentOS 7 images we highly recommend setting up Cloud-Init and fixing the network device names. More information for custom CentOS images can be found in the CentOS 7 image guide.

Communication Data

The following page contains communication information between the Morpheus appliance, integrated technologies, managed machines, and services.

Communication Frequency and Configurability

The following table contains communication information, including frequency and configurability between Morpheus and its supported technology integrations.

Source	Push/Pull	Destination
Cloud Foundry App Check	Server Pull	Cloud Foundry Applications that exist within Morpheus
Docker Container Check	Server Pull	Docker containers that exist within Morpheus
Elastic Search Check	Server Pull	Elastic Search application
Microsoft SQL Server Check	Server Pull	Microsoft SQL application
Mongo Check	Server Pull	Mongo DB application
MySQL Check	Server Pull	MySQL application
Postgres Check	Server Pull	Postgres application
Push API Check	Client Push	Morpheus API
Rabbit MQ Check	Server Pull	Rabbit MQ application
Redis Check	Server Pull	Redis application
Riak Check	Server Pull	Riak application
SNMP Check	Server Pull	SNMP
Socket Check	Server Pull	Web Socket
Virtual Machine Check	Server Pull	Virtual Machine that exists within Morpheus
Web Check	Server Pull (GET) or Server Push (POST)	Web application
Public Cloud Integration	Server Pull	Alibaba Cloud
Public Cloud Integration	Server Pull	Amazon AWS

Source	Push/Pull	Destination
Public Cloud Integration	Server Pull	Amazon AWS GovCloud
Public Cloud Integration	Server Pull	DigitalOcean
Public Cloud Integration	Server Pull	Google Cloud Platform
Public Cloud Integration	Server Pull	Huawei Cloud
Public Cloud Integration	Server Pull	IBM Cloud
Public Cloud Integration	Server Pull	Microsoft Azure
Public Cloud Integration	Server Pull	Open Telekom Cloud
Public Cloud Integration	Server Pull	Oracle Public Cloud
Public Cloud Integration	Server Pull	UpCloud
Public Cloud Integration	Server Pull	VMware on AWS
Private Cloud Integration	Server Pull	Cisco UCS Manager
Private Cloud Integration	Server Pull	Dell EMC
Private Cloud Integration	Server Pull	HPE
Private Cloud Integration	Server Pull	HPE OneView
Private Cloud Integration	Server Pull	KVM
Private Cloud Integration	Server Pull	MacStadium
Private Cloud Integration	Server Pull	Microsoft Azure Stack
Private Cloud Integration	Server Pull	Microsoft Hyper-V
Private Cloud Integration	Server Pull	Microsoft SCVMM
Private Cloud Integration	Server Pull	Nutanix Acropolis
Private Cloud Integration	Server Pull	Openstack
Private Cloud Integration	Server Pull	Oracle VM
Private Cloud Integration	Server Pull	Pivotal Cloud Foundry
Private Cloud Integration	Server Pull	Supermicro
Private Cloud Integration	Server Pull	Vmware vCloud Director
Private Cloud Integration	Server Pull	Vmware ESXi
Private Cloud Integration	Server Pull	VMware Fusion
Private Cloud Integration	Server Pull	VMware vCenter
Private Cloud Integration	Server Pull	Xen Server
Automation Integration		Ansible
Automation Integration	Server Pull	Ansible Tower
Automation Integration	Server Pull	Chef
Automation Integration	Server Pull	Puppet
Automation Integration	Server Pull	Salt
Automation Integration		Terraform
Automation Integration	Server Pull	vRealize Orchestrator
Backup Integration	Server Pull	Commvault
Backup Integration	Server Pull	Veeam
Backup Integration	Server Pull	Rubrik
Backup Integration	Server Pull	Zerto
Backup Integration	Server Pull	Avamar
Build Integration	Server Pull	Jenkins
Container Integration	Server Pull	Docker
Container Integration		Docker Registry
Container Integration	Server Pull	Kubernetes
Deployment Integration	Server Pull	Git/Github
DNS Integration	Server Pull	AWS Route53
DNS Integration	Server Pull	Microsoft DNS
DNS Integration	Server Pull	PowerDNS

Source	Push/Pull	Destination
Identity Management Integration	Server Pull	Microsoft AD
Identity Management Integration	Server Pull	OneLogin
Identity Management Integration	Server Pull	Okta
Identity Management Integration	Server Pull	Jump Cloud
Identity Management Integration	Server Pull	LDAP
Identity Management Integration	Server Pull	SAML
IPAM Integration	Server Pull	Infoblox
IPAM Integration	Server Pull	phpIPAM
IPAM Integration	Server Pull	Bluecat
IPAM Integration	Server Pull	SolarWinds
ITSM Integration	Server Pull	ServiceNow
ITSM Integration	Server Push	ServiceNow
ITSM Integration	Server Pull	Cherwell
ITSM Integration	Server Pull	Remedy
Key & Certificate Integration	Server Pull	Venafi
Load Balancer Integration	Server Pull	AzureLB
Load Balancer Integration	Server Pull	F5 BigIP
Load Balancer Integration	Server Pull	Citrix NetScaler
Logging Integration		Syslog
Monitoring Integration	Server Pull	ServiceNow
Monitoring Integration		AppDynamics
Monitoring Integration		NewRelic
Network Integration	Server Pull	NSX-T
Network Integration	Server Pull	Cisco ACI
Network Integration	Server Pull	Unisys Stealth
Storage Integration	Server Pull	3Par
Storage Integration	Server Pull	Azure Storage
Storage Integration	Server Pull	Dell ECS
Storage Integration	Server Pull	Isilon
Morpheus Agent	Agent Pull	Application Tier

Ports and Protocols

The following table contains communication port and protocol data between Morpheus appliance tiers, managed machines, and services. All communication to and from Morpheus goes thru the application tier with exception of inter-cluster communications for each of the Morpheus tiers when using a distributed architecture.

Ports used to communicate with integrated technologies are those defined for the integration’s API. They are not represented in this table as many of these are configurable and may be different in each customer environment. Additionally, ports used to complete Morpheus checks are customizable and may vary for each check configured. They are also not represented in this table.

Table 12: **Ports and Protocols**

Source	Destination	Port	Protocol	Description
User	Application Tier	443	TCP	User Access
Morpheus Servers	DNS Servers	53	TCP	Domain Name Resolution
Morpheus Servers	Time Source	123	TCP	Time Resolution
Morpheus Servers	Web or Offline Installer	80, 443	TCP	Download repos and Morpheus

Table 12 – continued from previous page

Source	Destination	Port	Protocol	Description
Managed Machine	Application Tier	443	TCP	Morpheus Agent Communicati
Managed Machine	Application Tier	80, 443	TCP	Agent Installation. (Requires p
Managed Machine	Application Tier	N/A	N/A	Agent Installation Clout-init (L
Managed Machine	Application Tier	N/A	N/A	Agent Installation Cloudbase-i
Managed Machine	Application Tier	N/A	N/A	Agent Installation VMtools
Managed Machine	Application Tier	N/A	N/A	Static IP Assignment & IP Poo
Managed Machine	Docker Image Repo	443	TCP	Applicable if using docker
Managed Machine	Application Tier	69	TCP/UDP	PXE Boot (Forwarded to inter
Application Tier	Managed Machine	5985	TCP	Agent Installation WinRM (W
Application Tier	Managed Machine	22	TCP	Agent Installation SSH (Linux
Application Tier	Managed Machine	22, 3389, 443	TCP	Remote Console (SSH, RDP, F
Application Tier	AWS S3	443	TCP	Morpheus Catalog Image Dow
Application Tier	Hypervisor	443	TCP	Hypervisor hostname resolvab
Application Tier	Non- Transactional Database Tier	443	TCP	Applicable if using Amazon E
Application Tier	Docker CE Repo	443	TCP	Applicable only when integrat
Application Tier	Rubygems	443	TCP	
Application Tier	Morpheus Hub	443	TCP	(Optional) Telemetry data (Dis
Application Tier	Mail Server	25 or 465	SMTP	Send email from Morpheus
Application Tier	postmarkapp	2525	TCP	Send email from Morpheus (su
Application Tier	Messaging Tier	5672	TCP	AMQP non-TLS connections
Application Tier	Messaging Tier	5671	TCP	AMQPS TLS enabled connect
Application Tier	Messaging Tier	61613	TCP	STOMP Plugin connections (R
Application Tier	Messaging Tier	61614	TCP	STOMP Plugin TLS enabled c
Messaging Tier	Messaging Tier	25672	TCP	Inter-node and CLI tool comm
Administrator Web Browser	RabbitMQ Server Management	15672	TCP	Management plugin
Administrator Web Browser	RabbitMQ Server Management	15671	TCP	Management plugin SSL
Messaging Tier Cluster Node	Messaging Tier Cluster Node	4369	TCP	erlang (epmd) peer discovery s
Application Tier	Non-Transactional Database Tier	9200	TCP	Elasticsearch requests (Used in
Application Tier	Non-Transactional Database Tier	443	TCP	Elasticsearch requests (Used in
Non-Transactional Database Tier	Non-Transactional Database Tier	9300	TCP	Elasticsearch Cluster
Transactional Database Tier	Transactional Database Tier	4567	TCP/UDP	Write-set replication traffic (ov
Transactional Database Tier	Transactional Database Tier	4568	TCP	Incremental State Transfer (IS
Application Tier	Transactional Database Tier	3306	TCP	MySQL client connections
Backup Solution	Transactional Database Tier	4444	TCP	State Snapshot Transfer (SST)
Application Tier	Integrated Technology	Varies	TCP	Integrations (Uses the port of t

VMware Support Statement

Morpheus Data will support customers who run Morpheus products on supported operating systems, irrespective of whether they are running in VMware environments or not. Morpheus Data supports operating systems, not specific hardware configurations. Accordingly, VMware operates as a hardware abstraction layer.

VMware supports a set of certified operating systems and hardware. The customer and VMware will be responsible for any interactions or issues that arise at the hardware or operating system layer as a result of their use of VMware.

Morpheus Data will not require clients to recreate and troubleshoot every issue in a non-VMware environment; however, Morpheus Data does reserve the right to request our customers to diagnose certain issues in a native certified operating system environment, operating without the virtual environment. Morpheus Data will only make this request when there is reason to believe that the virtual environment is a contributing factor to the issue.

Any time spent on investigation of problems that may, in the sole opinion of Morpheus Data be related to VMware,

will be handled in the following fashion:

1. Morpheus Data will provide standard support to all Morpheus products.
2. If a problem is encountered while Morpheus is/are running in a VMware environment, the client may be required to recreate the problem on a non-VMware server unit, at which time Morpheus Data will provide regular support.
3. The client can authorize Morpheus Data to investigate the VMware-related items at normal time and materials rates. If such investigation shows that the problem is VMware related, the client may contract Morpheus Data to provide a software change to resolve the issue if such a resolution is possible.
4. Regardless of the problem type or source, if the problem is determined to be a non VMware-related issue, time spent on investigation and resolution will be covered as part of regular maintenance and support will be provided as usual.

5.2 Operations

5.2.1 Dashboard

The Dashboard is a single, high-level view into your environment which includes easy-to-read performance and configuration information. In many cases other areas within Morpheus UI will allow you to drill deeper into the information presented in the dashboard.

Note: Elasticsearch 7.16+ is required for the Log Trends Dashboard panel to work. When using versions below 7.16 you will see error messages in logs due to this panel missing needed dependencies to work correctly but these errors may be safely ignored.



Environment

- **Groups:** Total number of Groups currently created
- **Clouds:** Total number of Clouds currently integrated
- **Clusters:** Total number of clusters currently provisioned
- **Apps:** Total number of Apps currently provisioned
- **Instances:** Total number of Instances currently provisioned
- **Users:** Total number of Morpheus user accounts

System Status

System status gives a high-level view of the health of the appliance. More detailed information can be viewed on the appliance health detail page (Administration > Health) and a more detailed breakdown of the meaning of each status indicator is in Morpheus [health documentation](#).

Favorites

Any Instances you've "favorited" will appear here along with the Instance name, type, and IP address

Alarms

The most recent five Alarms are displayed here and the user may link through to the resource which triggered the Alarm. For the complete list of Alarms and more information on each Alarm navigate to Operations > Activity > Alarms

Instance Status

The total number of Instances is listed along with a pie chart showing the statuses of each. Hover over each section of the pie chart to see the total number and percentage of Instances in that state. States may include running, stopped, provisioning, and more

Instances By Cloud

The total number of Clouds which currently have an Instance provisioned is shown. Hover over each section of the pie chart to see the total number and percentage of Instances provisioned to each Cloud

Daily Cloud Instances

The number of Instances that have existed at any point in the day with additional breakdown to show the number provisioned to each Cloud. This number will include any pre-existing Instances which have carried over from previous days along with any new Instances that were provisioned and existed on that day even for a short time

Group Workloads

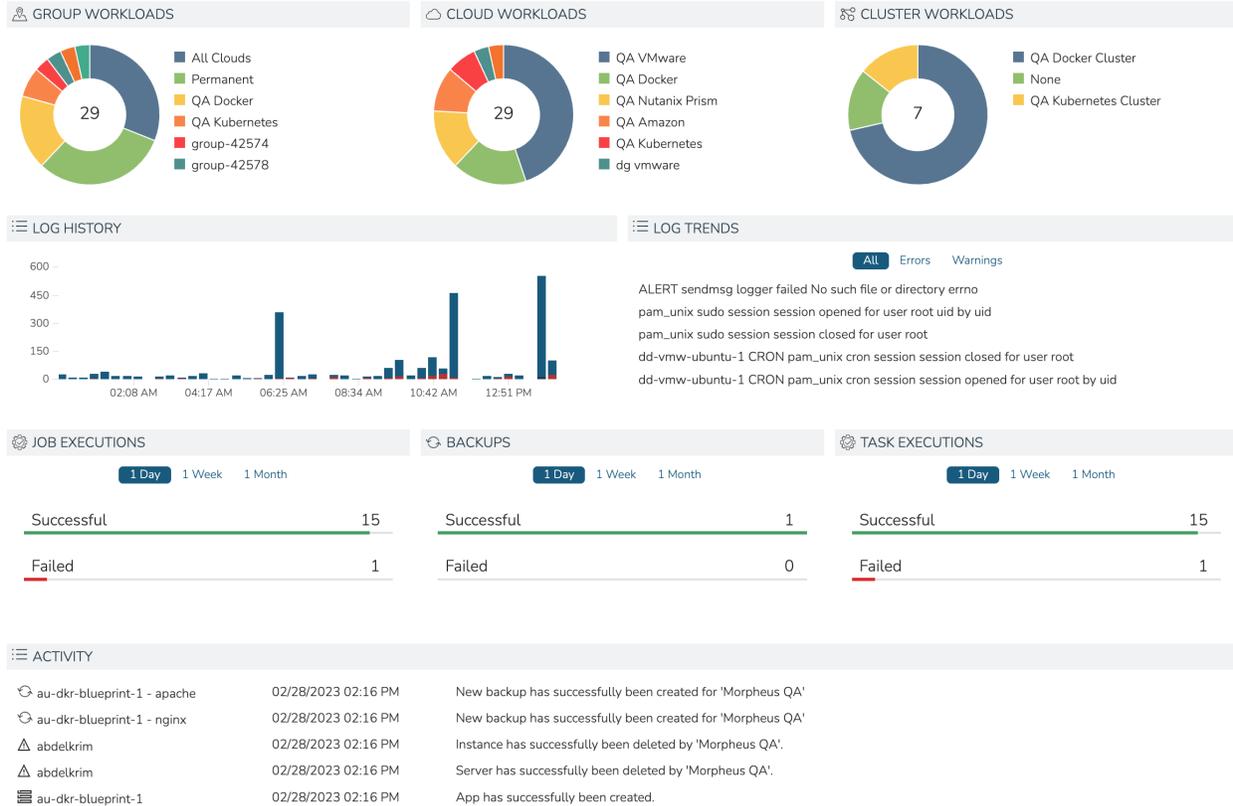
The instantaneous count of host or container records broken down by Group association

Cloud Workloads

The instantaneous count of host or container records broken down by Cloud association

Cluster Workloads

The instantaneous count of managed containers broken down by Cluster association



Log History

Log History shows a snapshot of various time segments throughout the current day and the number of logs generated during each segment. Hover over any bar on the graph to see a breakdown in severity of the logs within each segment

Log Trends

Log Trends identifies and lists repeated messages found in recent logs. If desired, this view can be filtered to show only Error or Warning-level log messages

Job Executions

Displays the number of successful and unsuccessful Job executions over the last day, week or month

Backups

Displays the number of successful and unsuccessful backups over the last day, week or month

Task Executions

Displays the number of successful and unsuccessful Task executions over the last day, week or month

Activity

The activity list displays the last few events that have taken place in Morpheus by any user. This could be new provisioned workloads, deleted workloads, backups, or a number of other things. A more complete list of recent activities can be viewed in Operations > Activity

5.2.2 Reports

Overview

Morpheus offers 28 different report types which are designed to slice up costing and usage across Clouds, Tenants, and more. Reports can be run on-demand as needed or can be scheduled to run on certain intervals to be viewed at a later time. The list of available report types can be viewed at Operations > Reports.

Tip: While Morpheus offers 28 built-in report types, users may also develop custom reports which display any information that may be needed for your business purposes. Take a look at our [guide to getting started with custom reports](#) for tips on how to begin the process

The screenshot shows the Morpheus Reports page. At the top, there is a navigation bar with the Morpheus logo, a search bar, and user information for Jaden Jabro. Below the navigation bar, there are tabs for various sections: Operations, Provisioning, Library, Infrastructure, Backups, Monitoring, Tools, and Administration. Under the Operations tab, there are sub-tabs for Dashboard, Reports, Analytics, Guidance, Wiki, Costing, Approvals, and Activity. The Reports section is active, showing a summary of 35 Report Types, 1162 Reports Run, and 2 Reports Scheduled. Below this summary, there are tabs for Report Types, Results, and Scheduled. The Report Types tab is selected, showing a search bar and a dropdown menu. A list of report types is displayed, each with an icon, a title, a category, a description, and two buttons: RUN NOW and SCHEDULE.

Report Type	Category	Description	Run Now	Schedule
TENANT INVENTORY SUMMARY	Account Inventory	View an inventory of VMs and containers by tenant.	Run Now	Schedule
CLOUD USAGE	Cloud Usage	View a cloud's usage of storage, memory, and compute resources.	Run Now	Schedule
CLOUD USAGE APP SUMMARY	Cloud Usage	View a cloud's usage of storage, memory, and compute resources by app.	Run Now	Schedule
CLOUD USAGE INSTANCE TYPE SUMMARY	Cloud Usage	View a cloud's usage of storage, memory, and compute resources by instance type.	Run Now	Schedule
TENANT USAGE	Cloud Usage	View a cloud's usage of storage, memory, and compute resources by tenant.	Run Now	Schedule
AMAZON RESERVATION COVERAGE	Cloud Cost	View how Amazon EC2 instance types are covered by Reserved Instances.	Run Now	Schedule
AMAZON RESERVATION UTILIZATION	Cloud Cost		Run Now	Schedule

Report Types

ACCOUNT INVENTORY

- Tenant Inventory Summary

CLOUD USAGE

- Cloud Usage
- Cloud Usage App Summary
- Cloud Usage Instance Type Summary
- [Tenant Usage](#)

CLOUD COST

- Amazon Reservation Coverage
- Amazon Reservation Utilization
- Amazon Savings Inventory Summary
- Amazon Savings Plan Coverage
- Amazon Savings Plan Utilization
- Application Cost
- Cloud Cost
- Group Cost
- Instance Cost
- Invoice Details
- Tenant Cost
- Time Series Cost

INFRASTRUCTURE INVENTORY

- Cloud Inventory Summary
- Container Host Inventory Summary
- Group Inventory Summary
- Guidance
- Hypervisor Inventory Summary
- Migration Planning
- Tenant Resource Allocation

PROVISIONING INVENTORY

- Instance Inventory Summary
- Software Inventory
- Software Inventory By Server
- Tag Compliance
- Virtual Machine Inventory Summary
- Workload Summary

By clicking into a report type, users can see any previous runs and active report schedules. New on-demand runs and new schedules of the selected report type can be created, edited, or deleted from here. The next few sections go into creating, editing, scheduling, viewing, and deleting reports in greater detail.

Create Reports

To create a new report, navigate to the report type list page (Operations > Reports). Click *RUN NOW* to the right of the specific report type to bring up the wizard to run that particular report. The required and optional fields to run the selected report type will appear, for example, the configuration panel for the Instance Cost report is shown below:

RUN INSTANCE COST REPORT ×

START END

GROUP

CLOUD

TAGS +

EXCLUDE TAGS +

TENANT

In this case, we can choose to scope the report by start and end dates, Groups, Clouds, Tenants, and can specifically include or omit Instances based on tags. Once the report is run, it will be visible in the list of Instance Cost reports and all reports until deleted.

Schedule Reports

In addition to running on-demand reports, Morpheus also allows reports to be scheduled. This allows you to save report configuration and have access to refreshed information on the schedule you need.

The process of scheduling a report is nearly identical to running on on-demand. From the report type list page (Operations > Reports) click *SCHEDULE* to the right of the report type you wish to schedule. The required and optional fields to schedule the selected report type will appear, for example, the configuration panel for the Instance Cost report is shown below:

SCHEDULE INSTANCE COST REPORT ✕

NAME

START END

GROUP

CLOUD

TAGS +

EXCLUDE TAGS +

TENANT

SCHEDULE

- Select
- Date And Time
- Daily
- Daily At Midnight
- Hourly
- Weekly On Sunday At Midnight

In this case, we can choose to scope the report by start and end dates, Groups, Clouds, Tenants, and can specifically include or omit Instances based on tags. Additionally, we select the time schedule on which this report should automatically run.

Note: Morpheus includes three schedules by default: Date and Time (run once at the specified time), Daily at Midnight, and Weekly on Sunday at Midnight. Any other listed scheduling periods are user-configured execution schedules (Library > Automation > Execute Scheduling). Create a new execution schedule if none of the existing schedules work for your reporting needs.

Viewing Results

A list of all report runs is viewable on the Results tab of the report types list page (Operations > Reports). To view the report itself, click on the hyperlinked report filters. Only reports that are ready for viewing will have an active hyperlink on their filters. In addition to report filters, the run date, report type, creating user, and run status are shown. Click on any of these headers to filter the report list by that column in either ascending or descending order. Any report can be deleted by clicking on the trash can icon at the end of its row.

MORPHEUS Search Support Jaden Jabro

Operations Provisioning Library Infrastructure Backups Monitoring Tools Administration

Dashboard Reports Analytics Guidance Wiki Costing Approvals Activity

REPORTS

35 REPORT TYPES 1162 REPORTS RUN 2 REPORTS SCHEDULED

Report Types **Results** Scheduled

Search All Types

FILTERS	DATE RUN	REPORT TYPE	CREATED BY	STATUS
Automation Analytics Feb 22, 2022 21:29:37	02/22/2022 02:29 PM	Automation Analytics		Ready
Automation Analytics Feb 22, 2022 21:17:59	02/22/2022 02:17 PM	Automation Analytics		Ready
Automation Analytics Feb 22, 2022 19:41:56	02/22/2022 12:41 PM	Automation Analytics		Ready
Automation Analytics Feb 22, 2022 19:28:00	02/22/2022 12:28 PM	Automation Analytics		Ready
Automation Analytics Feb 22, 2022 19:20:08	02/22/2022 12:20 PM	Automation Analytics		Ready
Automation Analytics Feb 22, 2022 17:49:16	02/22/2022 10:49 AM	Automation Analytics		Ready
Automation Overview Feb 21, 2022 22:07:43	02/21/2022 03:07 PM	Automation Overview		Ready
Automation Overview Feb 21, 2022 21:35:16	02/21/2022 02:35 PM	Automation Overview		Ready
Automation Overview Feb 21, 2022 21:31:53	02/21/2022 02:31 PM	Automation Overview		Ready
Automation Overview Feb 21, 2022 21:13:13	02/21/2022 02:13 PM	Automation Overview		Ready

Viewing Schedules

A list of all scheduled report runs can be viewed in the Scheduled tab of the report types list page (Operations > Reports). The friendly name of the report schedule is displayed along with the report type, last run time, next run time, and success status of the previous run. Schedules can be edited or deleted by clicking on the pencil or trash can icon, respectively. We can also view the most recent run of a given schedule (if it was successful) by clicking on the hyperlinked “last run” value.

5.2.3 Analytics

Overview

The Morpheus Analytics engine gives administrators the tools to break down costs and usage, then filter the results by relevant delineations including Groups, Clouds, Tenants or even tag values. Analytics dashboards can be organized into three primary categories based on their measurement intentions: costing, utilization, and workloads. Each dashboard type is discussed in further detail below.

Costing: Cloud Costing

CLOUD COST ANALYSIS

Search [input] Select Group [dropdown] Select Cloud [dropdown] MORE

31 CLOUDS \$4,808 MONTH TO DATE \$7,822 PROJECTED 16 CLOUD TYPES 8 REGIONS

TOTAL COST BY CLOUD TYPE

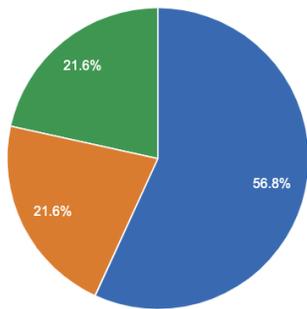


Table with 3 columns: Cloud Type, Name, and Cost. Amazon: \$2,732.64, Azure (Public): \$1,038.98, private: \$1,036.69.

TOTAL COST BY CLOUD

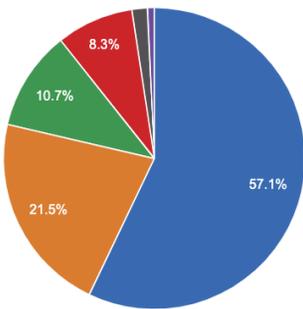


Table with 3 columns: Cloud Name, Name, and Cost. AWS MorpheusDemo: \$2,732.60, Morpheus Azure: \$1,030.92, Nutanix Labs 2: \$511.96, VMware Demo Cluster: \$395.17, etc.

TOTAL COST BY REGION

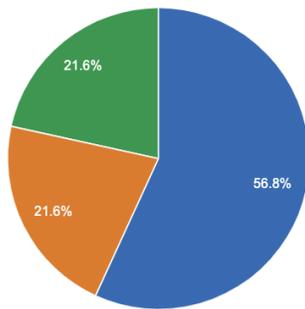


Table with 3 columns: Region Name, Name, and Cost. ec2.us-west-1.amazonaws.com: \$2,732.60, westus: \$1,038.41, unassigned: \$1,036.69.

Clouds

Table with 6 columns: TYPE, NAME, LOCATION, REGION, MONTH TO DATE, PROJECTED TOTAL. Includes rows for AWS MorpheusDemo, Morpheus Azure, and Nutanix Labs 2.

The Cloud Costing dashboard breaks down total costs by cloud type, cloud region, and individual cloud. This includes reporting the total number of clouds that meet your filters, the month-to-date running total, the projected monthly spend, among other useful metrics.

Filters

Filter the Clouds pulled into the dashboard by one or more of the following fields:

- Cloud (all matched by search)
• Group

- Cloud (selected from dropdown)
- Tenant
- Tag (Key)
- Value (tag value)

Data Displayed

The following aggregate totals are compiled for all Clouds that meet set filters:

- **CLOUDS:** The total number of Clouds that meet set filters
- **MONTH TO DATE:** The total spend in the current month for all Clouds meeting dashboard filters
- **PROJECTED:** The projected total spend for the current month for all Clouds meeting dashboard filters
- **CLOUD TYPES:** The number of distinct cloud types that meet the dashboard filters, such as Amazon AWS, Microsoft Azure, VMware, or any other Morpheus-supported Cloud types
- **REGIONS:** The total number of regions represented by Clouds meeting the dashboard filters

In addition to the totals described above, graphs visualize the percentage of these totals accounted for by specific Clouds, Cloud types, and Cloud regions.

Cloud List

Each Cloud that meets set filters is listed at the bottom of the dashboard, the following data points are revealed for each individual Cloud:

- **TYPE:** The Cloud type, such as Amazon AWS, Microsoft Azure, VMware, or any other Morpheus-supported Cloud type
- **NAME:** The name given to the Cloud in Morpheus at the time of integration
- **LOCATION:** The Cloud location (if available)
- **REGION:** The Cloud region (if available)
- **MONTH TO DATE:** The current month-to-date spend for the individual Cloud listed
- **PROJECTED TOTAL:** The projected total spend for the current month for the individual Cloud listed

Costing: Group Costing

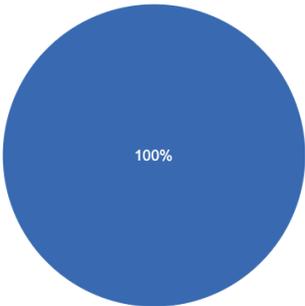
GROUP COST ANALYSIS

42
GROUPS

\$25,024
MONTH TO DATE

\$25,124
PROJECTED

TOTAL COST BY GROUP



ServiceNow \$24,865.05

TOP GROUPS

Groups

NAME	LOCATION	DATACENTER	MONTH TO DATE	PROJECTED TOTAL
[blurred]			\$24,865.05	\$24,865.05
[blurred]			\$61.86	\$104.31
[blurred]			\$30.49	\$30.49
[blurred]			\$26.04	\$43.21
[blurred]	Denver		\$11.72	\$11.72
[blurred]	Denver		\$9.16	\$9.16

The Group Costing dashboard aggregates cost totals for all Groups that meet filters set on the dashboard. This allows administrators to sort Groups by their total costs and anticipate monthly total costs by Group.

Filters

Filter the Groups pulled into the dashboard by one or more of the following fields:

- Group (all matched by search)
- Group (selected from dropdown)
- Cloud
- Tenant

Data Displayed

The following aggregate totals are compiled for all Groups that meet set filters:

- **GROUPS:** The total number of Groups that meet set filters
- **MONTH TO DATE:** The total spend in the current month for all Groups meeting dashboard filters
- **PROJECTED:** The projected total spend for the current month for all Groups meeting dashboard filters

Group List

Each Group that meets set filters is listed at the bottom of the dashboard, the following data points are revealed for each individual Group:

- **NAME:** The name given to the Group in Morpheus at the time of creation
- **LOCATION:** The Group location (if available)
- **DATACENTER:** The Group datacenter (if available)
- **MONTH TO DATE:** The current month-to-date spend for the individual Group listed
- **PROJECTED TOTAL:** The projected total spend for the current month for the individual Group listed

Costing: Tenant Costing

TENANT COST ANALYSIS

Search Select Cloud

▼ MORE

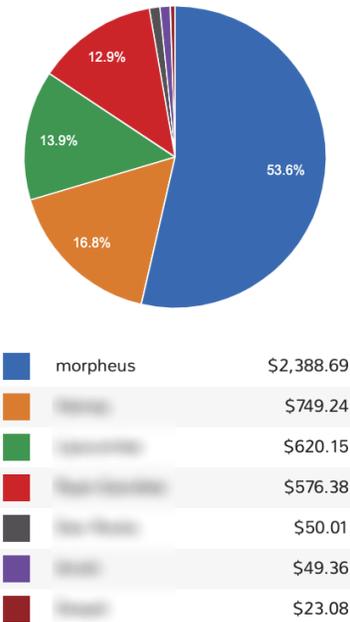
58
TENANTS

\$4,476
MONTH TO DATE

\$7,776
PROJECTED

\$7,848
LAST MONTH

TOTAL COST BY TENANT



Tenants

NAME	DESCRIPTION	MONTH TO DATE	PROJECTED TOTAL
morpheus		\$2,388.69	\$4,227.03
[Redacted]	[Redacted]	\$749.24	\$1,254.15
[Redacted]	[Redacted]	\$620.15	\$1,082.77
[Redacted]	[Redacted]	\$576.38	\$962.72

The Tenant Costing dashboard aggregates costing totals across all Tenants that meet the filters set on the dashboard. This information helps administrators track the current spend of each Tenant for the current monthly period. It also helps identify the costliest Tenants and to anticipate month-end costs for each individual Tenant.

Filters

Filter the Tenants pulled into the dashboard by one or more of the following fields:

- Tenant (all matched by search)

- Cloud
- Tenant (selected from dropdown)

Data Displayed

The following aggregate totals are compiled for all Tenants that meet set filters:

- **TENANTS:** The total number of Tenants that meet set filters
- **MONTH TO DATE:** The total spend in the current month for all Tenants meeting dashboard filters
- **PROJECTED:** The projected total spend for the current month for all Tenants meeting dashboard filters
- **LAST MONTH:** The total spend in the prior month for all Tenants meeting dashboard filters

Tenant List

Each Tenant that meets set filters is listed at the bottom of the dashboard, the following data points are revealed for each individual Tenant:

- **NAME:** The name given to the Tenant in Morpheus at the time of creation
- **DESCRIPTION:** The Tenant description (if available)
- **MONTH TO DATE:** The current month-to-date spend for the individual Tenant listed
- **PROJECTED TOTAL:** The projected total spend for the current month for the individual Tenant listed

Costing: User Costing

USER COST ANALYSIS

Search [magnifying glass icon] Select Group [dropdown] Select Cloud [dropdown] Current [dropdown] [dropdown arrow] MORE

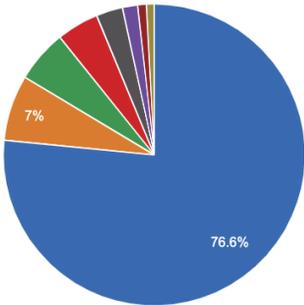
Select tenant [dropdown]

66
USERS

\$893
MONTH TO DATE

\$1,446
PROJECTED

TOTAL COST BY USER



discovered	\$673.01
[blurred]	\$61.86
[blurred]	\$49.52
[blurred]	\$39.72
[blurred]	\$25.00
[blurred]	\$14.22
[blurred]	\$8.16
[blurred]	\$7.47

Users

USERNAME	MONTH TO DATE	PROJECTED TOTAL
discovered	\$673	\$1,092
[blurred]	\$62	\$104

The User Costing dashboard allows administrators to analyze costs for a group of Users that meet specific filters. Once the group is selected, total costs by User for the current month and projected totals are displayed. Administrators can identify their costliest Users and anticipate the total cost by User for budgeting purposes.

Filters

Filter the Groups pulled into the dashboard by one or more of the following fields:

- User (all matched by search)
- Group
- Cloud
- Period (Current month, last three months, last six months, or last 12 months)
- Tenant

Data Displayed

The following aggregate totals are compiled for all Users that meet set filters:

- **USERS:** The total number of Users that meet set filters
- **MONTH TO DATE:** The total spend in the current month for all Users meeting dashboard filters
- **PROJECTED:** The projected total spend for the current month for all Users meeting dashboard filters

User List

Each User that meets set filters is listed at the bottom of the dashboard, the following data points are revealed for each individual User:

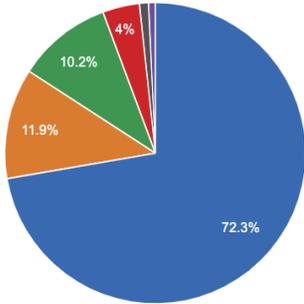
- **USERNAME:** The username given to the User in Morpheus at the time of creation
- **MONTH TO DATE:** The current month-to-date spend for the individual User listed
- **PROJECTED TOTAL:** The projected total spend for the current month for the individual User listed

Costing: Workload Costing

WORKLOAD COST ANALYSIS

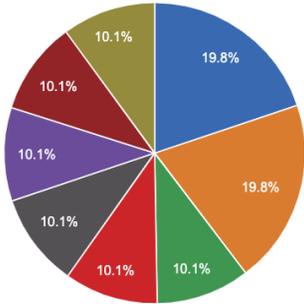
33 WORKLOADS
 \$330 MONTH TO DATE
 \$603 PROJECTED
 11 TYPES
 20 SIZES

TOP WORKLOAD TYPES BY COST



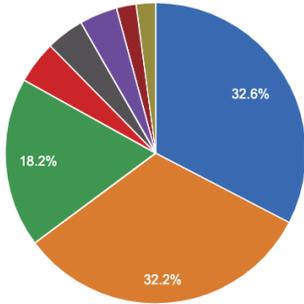
CentOS	\$238.60
Nutanix	\$39.17
Ubuntu	\$33.58
Others	\$13.10
Others	\$3.18
Others	\$2.44

TOP WORKLOADS BY COST



Workload 1	\$51.55
Workload 2	\$51.52
Workload 3	\$26.15
Workload 4	\$26.15
Workload 5	\$26.12
Workload 6	\$26.12
Workload 7	\$26.12
Workload 8	\$26.12

TOP WORKLOAD SIZES BY COST



2 vCPU, 8GB Memory	5	\$104.46
2 CPU, 16GB Memory	2	\$103.07
2 CPU, 8GB Memory	4	\$58.37
m1.large	1	\$14.65
1 CPU, 4GB Memory	1	\$13.10
1 vCPU, 4GB Memory	1	\$13.06
2GB Memory, 20GB Storage	1	\$6.73
1 CPU, 2GB Memory	1	\$6.72

The Workload Costing dashboard allows administrators to look at all or a subset of Morpheus-managed workloads to analyze their cost impact. Filters can be set to isolate a specific group of workloads and cost breakdowns are shown. Graphs are generated to reveal cost breakdowns of individual workloads or certain groups of workloads.

Filters

Filter the workloads pulled into the dashboard by one or more of the following fields:

- Workload name (all matched by search)
- Group
- Cloud
- Tenant

- Tag (Key). This is a required field and the top key in the list will be pre-selected by default
- Value (tag value)

Data Displayed

The following aggregate totals are compiled for all workloads that meet set filters:

- **WORKLOADS:** The total number of workloads that meet set filters
- **MONTH TO DATE:** The total spend in the current month for all workloads meeting dashboard filters
- **PROJECTED:** The projected total spend for the current month for all workloads meeting dashboard filters
- **TYPES:** The total number of workload types represented among workloads meeting set filters
- **SIZES:** The total number of unique workload sizes represented among workloads meeting set filters

Costing: Budget Analysis

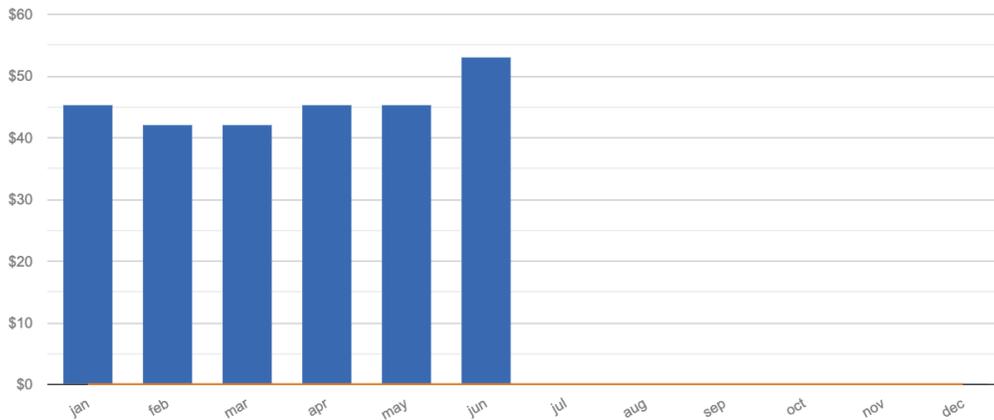
BUDGET COST ANALYSIS

Search User

APPLIED FILTERS Budget Scope: user

1 BUDGETS \$14 MONTH TO DATE \$0 BUDGETS TO DATE \$273 COST TO DATE \$273 TO BUDGET

BUDGET VS ACTUAL -



	JANUARY	FEBRUARY	MARCH	APRIL	MAY	JUNE	JULY	AUGUST	SEPTEMBER	OCTOBER	NOVEMBER	DECEMBER
Budget	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
Actual	\$45	\$42	\$42	\$45	\$45	\$53						

The Budget Analysis dashboard allows administrators to filter and view budgets in one place in order to keep track of progress against budget over time. Budgets in Morpheus (Operations > Budgets) are tied to a specific scope (Account, Tenant, Cloud, Group, or User) and budgets of the same scope are viewed together in this dashboard. A scope filter

must be set in order for data to be populated into the dashboard. Once a scope is selected, the search bar can be utilized to return only budgets within the selected scope whose “Name” meets the search terms.

Filters

Filter the budgets pulled into the dashboard by one or more of the following fields:

- Budget name (all matched by search)
- Scope (This is a required field, data is not populated into the dashboard until a scope is specified)

Data Displayed

The following aggregate totals are compiled for all budgets that meet set filters:

- **BUDGETS:** The total number of budgets that meet set filters
- **MONTH TO DATE:** The total spend in the current month against the selected budgets
- **BUDGETS TO DATE:** The total amount budgeted to date among budgets selected by the dashboard filters (from the start of the year to the end of the current interval)
- **TO BUDGET:** The difference between the COST TO DATE and BUDGETS TO DATE value, should be close to \$0 if the costs are appropriately tracking against the budgeted amounts

Budget List

Each budget with its own graph and breakdown is displayed going down the page. The format of the information presented depends on the interval that the specific budget is configured for.

Costing: Tag Costing

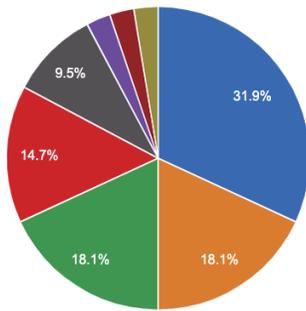
TAG COST ANALYSIS

20
TAGS

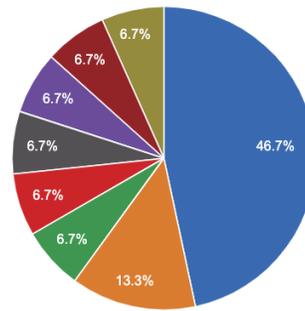
\$1,384
MONTH TO DATE

\$2,024
PROJECTED

TOP USED TAGS



TOP TAG VALUES



Morpheus Server Id	37
Morpheus Id	21
costCenter	21
Morpheus Instance Id	17
Morpheus Environment	11
Others	3

Others	21
Others	6
Others	3

Tags

TAG	WORKLOADS	MEMORY	STORAGE	CPU	PRICE	VALUES
Morpheus Server Id	37	4.6TiB	73.8TiB	1	\$299.32	▼ MORE
Morpheus Id	21	2.4TiB	25.4TiB	1	\$172.56	▼ MORE
costCenter	21	2.0TiB	50.5TiB	1	\$105.35	▼ MORE

The Tag Analysis dashboard creates groups of workloads based on the presence of specific tags and meeting other filters. This workload group can be analyzed for total cost and projected costs.

Filters

Filter the workloads pulled into the dashboard by one or more of the following fields:

- Tag key (all matched by search)
- Metric (apply to see the top tag values by workload count, price, memory, storage, or CPU cores)

- Group
- Cloud
- Tenant
- Tag (Key)

Data Displayed

The following aggregate totals (by tag) are compiled for workloads that meet set filters:

- **TAGS:** The total number of unique tag keys for workloads meeting dashboard filters
- **MONTH TO DATE:** The total spend in the current month for selected workloads
- **PROJECTED:** The total projected current-month spend for selected workloads

Tags List

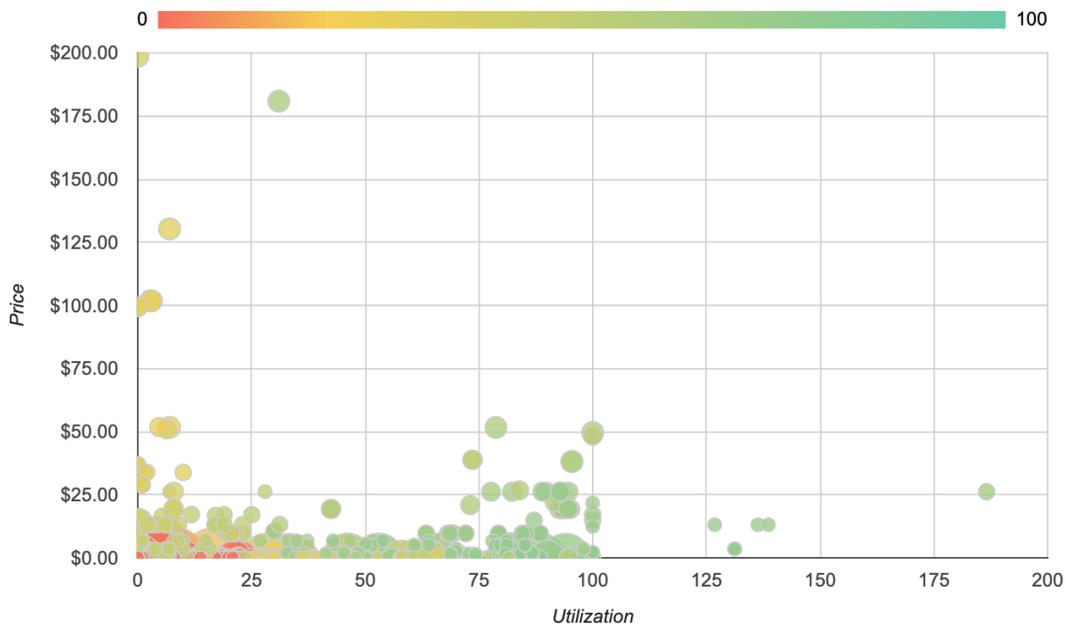
A list of each tag (key) represented on selected workloads is displayed in a list below the dashboard graphs. We also see the total number of workloads associated with the tag, the total memory, total storage, total CPU cores, and total price. If we click the “MORE” link at the end of each row, we can see a list of all tag values associated with the key.

Utilization: Utilization vs Cost

UTILIZATION VS COST

Summary

733 COUNT
 34 CLOUD COUNT
 \$3,823 MONTH TO DATE
 \$6,452 PROJECTED
 27% AVERAGE UTILIZATION



Lowest Utilization per Cost

POWER	OS	NAME	TYPE	CLOUD	PRICE	CPU UTILIZATION
		aa-sb-morpheus-centos-4110	Managed VM	VMWare vCenter	\$51.66	<div style="display: flex; gap: 10px;"> <div> 3 CPU</div> <div> 6 MEMORY</div> <div> 25 SCORE</div> </div>
		labs_west-chef-web-01a	Discovered	AWS MorpheusDemo	\$51.80	<div style="display: flex; gap: 10px;"> <div></div> <div></div> <div></div> </div>

The Utilization vs Cost dashboard is designed to reveal workloads which are underutilized (expensive and seldom-used) and which are very cost-efficient (inexpensive and frequently-used). Administrators can filter the workloads considered by the dashboard through the use of filters and potentially identify areas of cost savings by decommissioning seldom-used machines.

Filters

Filter the workloads pulled into the dashboard by one or more of the following fields:

- Workload name (all matching search terms)
- Time period (Current, one-day average, one-week average, one-month average, three-month average, six-month average, or one-year average)
- Type (virtual machines, hosts, or bare metal)
- Tenant
- Tag (Key)
- Value (Tag value)

Data Displayed

The following aggregate totals are compiled for workloads that meet set filters:

- **COUNT:** The total number of workloads that meet dashboard filters
- **CLOUD COUNT:** The total number of Clouds represented by the selected workloads
- **MONTH TO DATE:** The total spend in the current month for selected workloads
- **PROJECTED:** The total projected current-month spend for selected workloads
- **AVERAGE UTILIZATION:** The computed average utilization figure for all workloads selected by dashboard filters

Utilization List

In addition to the totals and graph displayed, two workload lists are given showing the least utilized workloads by cost (lowest utilization per cost dollar) and the least utilized workloads overall (lowest utilization overall). These workloads are listed with links to the Instance or server detail pages, along with other details related to price and resource utilization.

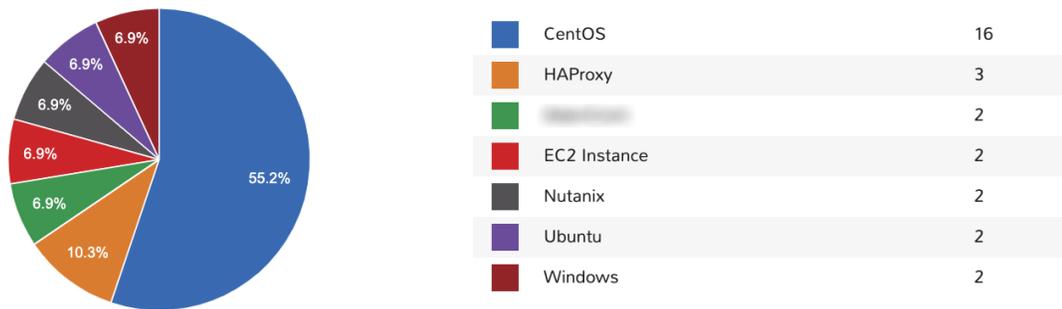
Workloads: Instance Type Usage

INSTANCE TYPE USAGE

Search Count Select Group Select Cloud

13 TYPES 35 INSTANCES \$346 MONTH TO DATE \$556 PROJECTED 165.3GB MEMORY 1.9TB STORAGE

TOP INSTANCE TYPES BY COUNT



Instance Type Usage

TYPE	NAME	GROUPS	CLOUDS	COUNT
	CentOS	11	5	16
	HAProxy	3	1	3
		1	1	2
	EC2 Instance	1	1	2
	Nutanix	2	2	2

The Instance Type Usage dashboard organizes workloads meeting dashboard filters by their Instance type. In addition to counts, administrators can view resource consumption and cost figures by Instance type groupings as well.

Filters

Filter the workloads pulled into the dashboard by one or more of the following fields:

- Instance type name (all matching search terms, Morpheus-default Instance types are not included when using the search filter)
- Metric (apply to see the top Instance types by workload count, price, memory, storage, or CPU cores)
- Group

- Cloud
- Tenant
- Tag (Key)
- Value (Tag value)

Data Displayed

The following aggregate totals are compiled for workloads that meet set filters:

- **TYPES:** The total number of Instance types represented among workloads meeting the dashboard filters
- **INSTANCES:** The total number of Instances represented in the data
- **MONTH TO DATE:** The total spend in the current month for selected workloads
- **PROJECTED:** The total projected current-month spend for selected workloads
- **MEMORY:** The total memory allotted to selected workloads
- **STORAGE:** The total storage allotted to selected workloads

Instance Type Usage List

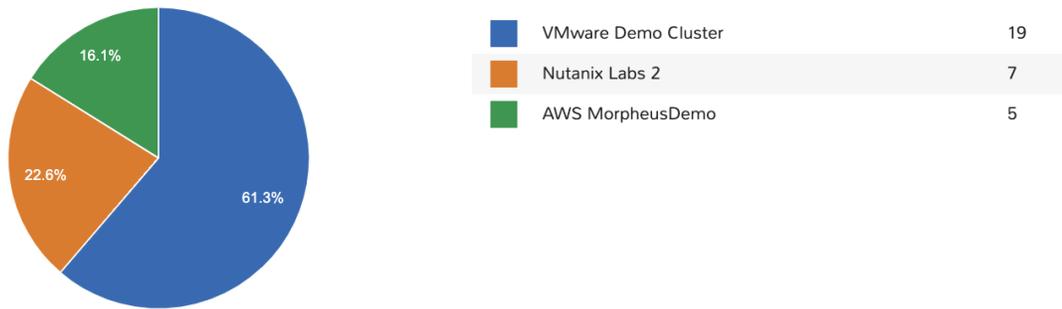
Each Instance type represented in the dashboard is listed below the graph. For each Instance type shown, we see the number of Groups the Instance type is represented in, the number of Clouds the Instance type has been provisioned into, and the total amount of memory allotted to workloads of each Instance type.

Workloads: Instance Usage

INSTANCE USAGE

8 CLOUDS
36 INSTANCES
\$346 MONTH TO DATE
\$556 PROJECTED
167.3GB MEMORY
1.9TB STORAGE

INSTANCE USAGE BY CLOUD



Instance Usage

CLOUD	COUNT	MEMORY	STORAGE	CPU	PRICE
VMware Demo Cluster	19	72.4GiB	842.0GiB	12	\$143.94
Nutanix Labs 2	7	52.5GiB	720.0GiB	8	\$91.82
AWS MorpheusDemo	5	7.0GiB	60.0GiB	1	\$17.41
...	1	7.5GiB	90.0GiB	4	\$15.08
...	1	2.0GiB	30.0GiB	1	\$0.00
...	1	2.0GiB	10.0GiB	1	\$0.00
...	1	8.0GiB	120.0GiB	1	\$26.23
...	1	16.0GiB	160.0GiB	1	\$51.78

The Instance Usage dashboard shows Instance counts, resource utilization, and cost breakdowns by Cloud. Administrators can set filters to limit the workloads that are considered for dashboard analysis and then see the results given by Cloud groupings.

Filters

Filter the workloads pulled into the dashboard by one or more of the following fields:

- Instance name (all matching search terms)
- Metric (apply to see the top Clouds by workload count, price, memory, storage, or CPU cores)
- Group

- Cloud
- Tenant
- Tag (Key)
- Value (Tag value)

Data Displayed

The following aggregate totals are compiled for workloads that meet set filters:

- **CLOUDS:** The total number of Clouds represented among workloads meeting the dashboard filters
- **INSTANCES:** The total number of Instances represented in the data
- **MONTH TO DATE:** The total spend in the current month for selected workloads
- **PROJECTED:** The total projected current-month spend for selected workloads
- **MEMORY:** The total memory allotted to selected workloads
- **STORAGE:** The total storage allotted to selected workloads

Instance Usage List

All Clouds represented in the dashboard are listed here. For each Cloud, we see the total Instance count, total memory allotted, total storage allotted, total CPU cores, and the total price.

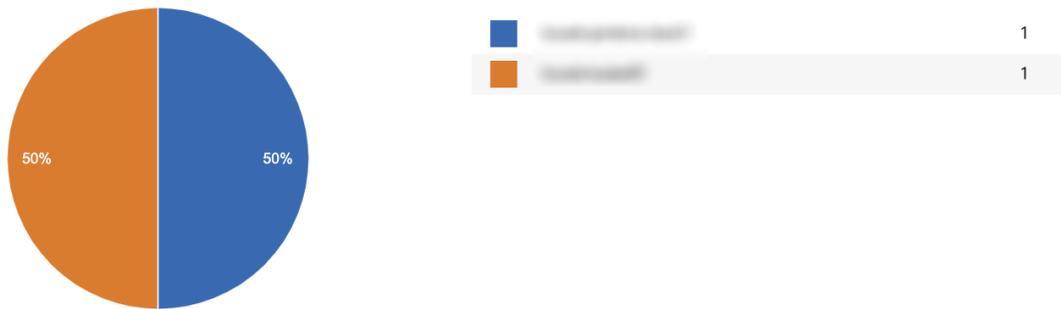
Workloads: Blueprint Usage

BLUEPRINT USAGE

Search Count Select Group Select Cloud

2 TYPES 2 APPS 2 INSTANCES \$0 MONTH TO DATE 5.8GB MEMORY 22GB STORAGE

TOP BLUEPRINTS BY COUNT



Blueprint Usage

TYPE	NAME	INSTANCES	GROUPS	CLOUDS	COUNT
MISSING LOGO	[Redacted]	1	1	1	1
MISSING LOGO	[Redacted]	1	1	1	1

The Blueprint Usage dashboard lists all provisioned Apps that meet filters set on the dashboard. Once the desired group of Apps is filtered into the dashboard, administrators will see the total provisioned from each Blueprint, total number of Instances created from the Apps, and costing details.

Filters

Filter the Apps pulled into the dashboard by one or more of the following fields:

- App name (all matching search terms)
- Metric (apply to see the top Clouds by workload count, price, memory, storage, or CPU cores)
- Group
- Cloud
- Tenant
- Tag (Key)
- Value (Tag value)

Data Displayed

The following aggregate totals are compiled for Apps that meet set filters:

- **TYPES:** The total number of App types represented among Apps meeting the dashboard filters
- **APPS:** The total number of Apps represented in the dashboard
- **INSTANCES:** The total number of Instances contained in all Apps meeting dashboard filters
- **MONTH TO DATE:** The total month-to-date spend for all Apps shown in the dashboard
- **MEMORY:** The total memory allotted to selected Apps
- **STORAGE:** The total storage allotted to selected Apps

Blueprint Usage List

All Blueprints which have a currently-existing App provisioned from them and selected in the dashboard filters are listed here. The name and type of the Blueprint is listed along with the total number of Instances across all provisionings, total Groups, total Clouds, and the total count of all Apps from that Blueprint.

Workloads: Apps Usage

APPS USAGE

Count
▼

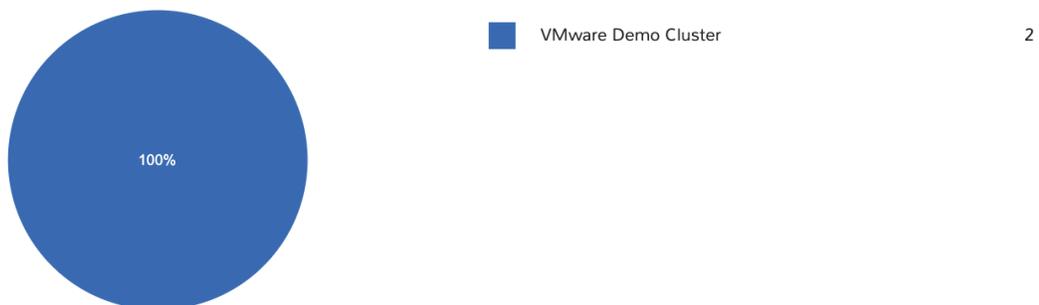
Select Group
▼

Select Cloud
▼

▼ MORE

1	2	3	\$0	5.8GB	22GB
CLOUDS	APPS	INSTANCES	TOTAL COST	MEMORY	STORAGE

APP USAGE BY CLOUD



Instance Usage

CLOUD	COUNT	MEMORY	STORAGE	CPU	PRICE
VMware	2	5.9GiB	22.0GiB	1	\$0.07

The Apps Usage dashboard lists all provisioned Apps that meet a set of filters and organizes them by Cloud. Totals for cost and resource usage of all relevant Apps can be viewed with a per-Cloud breakdown.

Filters

Filter the Apps pulled into the dashboard by one or more of the following fields:

- App name (all matching search terms)
- Metric (apply to see the top Clouds by workload count, price, memory, storage, or CPU cores)
- Group
- Cloud
- Tenant
- Tag (Key)
- Value (Tag value)

Data Displayed

The following aggregate totals are compiled for Apps that meet set filters:

- **CLOUDS:** The total number of Clouds represented among Apps meeting the dashboard filters
- **APPS:** The total number of Apps represented in the dashboard
- **INSTANCES:** The total number of Instances contained in all Apps meeting dashboard filters
- **TOTAL COST:** The total cost of all selected Apps
- **MEMORY:** The total memory allotted to selected Apps
- **STORAGE:** The total storage allotted to selected Apps

Instance Usage List

All Clouds with a currently-provisioned App which is selected in the dashboard filters are listed here. The name of the Cloud is listed along with its App count, the total memory, total storage, total CPU cores and price of the Apps provisioned in that Cloud are also listed.

5.2.4 Guidance

Overview

The **Operations > Guidance** section shows recommendations for resource and cost optimization. By analyzing the CPU, RAM, and Storage activity of Instances and Hosts, Morpheus can recommend actions for sizing and power state. Guidance is customizable to show recommendations based on 30, 60, or 90 day periods, this dropdown toggle is visible on the Guidance list page (**Operations > Guidance**). Guidance thresholds are also customizable, they can be edited in **Administration > Settings > Guidance**.

Configuration

Guidance is configured per Cloud and is turned off by default.

To turn on Morpheus Guidance for a Cloud:

1. Navigate to *Infrastructure > Clouds*
2. Click *EDIT*
3. Expand the *Advanced Options* section in the Edit Cloud modal
4. In the *Guidance* dropdown, select *Manual*

5. Click *SAVE CHANGES*

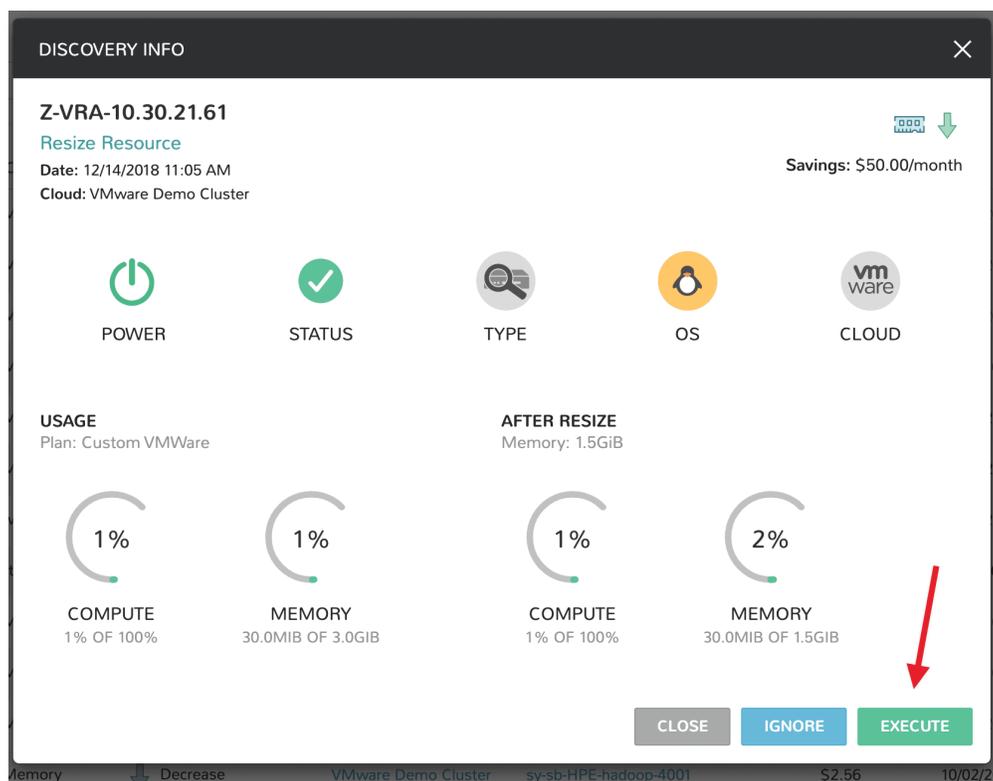
Guidance recommendations will begin to appear in the guidance section when generated.

Note: It will take approximately seven days before Morpheus gathers sufficient data to make guidance recommendations. [morpheus] Guidance is significantly improved when the Agent is installed on provisioned workloads. When the Agent is not installed, Morpheus still makes a best effort to provide relevant guidance recommendations.

Once Guidance has been turned on for a Cloud, Morpheus will determine if a guidance recommendation should be made once every 30 minutes. In the event that no recommendations can be made, no entry will be added to the list of suggested guidances. As the guidance list continues to grow, sorting and filtering may become necessary to focus on the recommendations that are relevant to the user at the time.

It's important to note that acting on recommendations is entirely manual at this time. In many cases, Morpheus provides one-click action to take the recommended steps but Guidance recommendations cannot be taken automatically. This is a feature that is being explored for a future update but has not been tagged for any specific release version at this time. In addition, it's recommended that Morpheus Agent be installed to maximize the benefits of Guidance. While Guidance will still work without installing the Agent, the greatly-enhanced statistics provided by the Agent will significantly improve Guidance recommendations.

To see more detail on a Guidance recommendation in your list, click on the (i) button at the far right side of each list row. This will open a detail modal that gives additional information on the Guidance entry. In some cases, such as with sizing and shutdown recommendations, the user can simply click the "EXECUTE" button to take the recommended action as shown below.



Other types of Guidance recommendations, such as reserve compute recommendations, must be taken in the cloud and Morpheus does not offer the execute button.

The screenshot shows a 'DISCOVERY INFO' window for 'Morpheus Azure'. It features a 'Reserve Compute' recommendation with a red arrow pointing to it. The date is '10/11/2019 06:05 PM' and the savings is '\$34.40/month'. Below this, there are icons for 'STATUS' (a green checkmark) and 'CLOUD' (the Azure logo). The main section compares 'CURRENT' costs (\$109.63) with 'AFTER RESERVATIONS' costs (\$75.23), resulting in a 'MONTHLY SAVINGS' of \$34.40 and a 'SAVINGS PERCENT' of 31.38%. At the bottom, a table lists the reservation details for 'Standard_D2' in the 'westus' region with a 'P1Y' term.

NAME	REGION	TERM	CURRENT COST	QUANTITY	PROPOSED COST	SAVINGS
Standard_D2	westus	P1Y	\$109.63	1	\$75.23	\$34.40

Note: The IGNORE button will remove the recommendation from the UI. Subsequent recommendations of the same type will NOT display for the same object (VM, Cloud etc) again unless the original recommendation is resolved.

Recommendations

To view and act on Guidance recommendations, navigate to *Operations > Guidance*.

The Guidance list contains the following details:

Severity Icon Indicates the severity of the recommended action.

Type Recommended action Type

Metric Guidance Metric used for recommended action.

Action Recommended Action for the Instance or Host, such as “Reduce Host memory” or “Shutdown Instance”

RESOURCE The Instance or Host targeted

SAVINGS Shows projected Monthly Costs savings if recommended action is taken.

DATE Date and Time stamp the recommended action was generated.

Information Link Click to view details on the recommendation.

Note: Guidance Actions are not automatically triggered at this time.

Filters

Search Search for Guidance recommendations

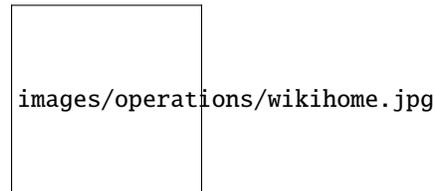
Type Filter by Sizing or Shutdown Guidance Types.

Severity Filter by Guidance Severity of All, Info, Warning, or Critical.

Metric Filter by All, Memory, CPU, or Power Guidance Metrics.

5.2.5 Wiki

The Morpheus Wiki is a tenant-wide, RBAC-controlled, auditable Wiki that allows easy UI, API and CLI access to information, notes, configurations or any other data needed to be referenced or shared with others. Wiki pages can be created directly from the Wiki tab of the detail page for various resource types, including Clouds, Groups, Servers, Instances, Clusters, and Self-Service Persona Catalog Items. Wiki pages created this way are automatically categorized under the appropriate resource type. Additional Wiki pages and custom categories can be created when viewing the whole Wiki at [Operations > Wiki](#). Here you will also see the complete Wiki, including pages created on various object detail pages which are categorized appropriately.



Highlights

- Main Wiki section is at [Operations > Wiki](#)
- Wiki tabs are on Clouds, Groups, Instances, Hosts, VMs, Bare Metal, and Clusters detail pages
- Additional Wiki Pages and Categories can be created from [Operations > Wiki](#)
- When a Wiki tab is populated, a Page is automatically added and accessible at [Operations > Wiki](#)
- One Wiki is created per Tenant. There is no multi-tenant access to Wikis
- The Wiki is accessible from the UI, CLI and API.
- Wiki access is RBAC-controlled via the Operations: Wiki permission in User and Tenant Roles (None, Read and Full)
- Page updates are stamped with the “Last Updated By” user and the time the edit was made
- Wiki pages can be searched from `/operations/wiki` or navigated from `/operations/wiki-page/page-index`
- All wiki pages are encrypted using AES 256-bit encryption
- Wiki pages use Flexmark for Markdown. Annotate your Wiki pages with headers, text styling, code blocks, hyperlinks, and more as needed
- Create a new page with title “Home” to replace the default Wiki landing page that ships with Morpheus

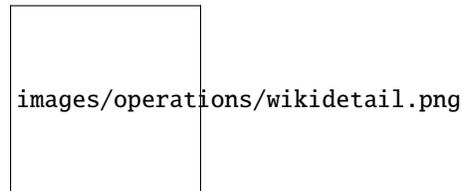
Note: The Wiki replaces Notes. Notes are automatically migrated to corresponding Wiki pages when upgrading Morpheus to 4.0.0+.

Creating Wikis

The Wiki service ties into assets throughout the environment. Create pages for Instances, hosts, groups, Clouds, and even clusters directly on their detail pages (see the Wiki tab). Users may also just create general notes pages in the centralized Wiki section (Operations > Wiki) in Markdown format.

Creating your first page is as simple as clicking the *Create Page* button from the Wiki home page (Operations > Wiki). Write down some content, give the page a title, and click *SAVE*. The Wiki will also keep track of who last edited a page and when. The beauty of this Wiki is that it's clean and easy to write down notes related to various parts of your application deployment or infrastructure without going to an external tool. Many Morpheus constructs, such as Instances, hosts, and more, also have their own Wiki page. Navigate to the detail page for the selected construct, open the Wiki tab, and click *EDIT* to add content.

Important: All wiki pages are encrypted using AES-256 bit encryption. Though we don't advise storing passwords in a Wiki document (services like Cypher are for that), role-based access control also can properly restrict access to content related to Instances or hosts which the user may not have access to.



Hosting Images

It's possible to add images to your Wiki pages and images can be sourced from the Internet, a Cloud storage bucket (like an AWS S3 Bucket), or even from files stored to the Morpheus appliance's local file system. Within your Wiki page markdown, add your image using the following syntax:

```
![my alt text](https://myimage.com/image.jpg)
```

The text within the square brackets [] sets the HTML "alt" description for the image and the URL within parentheses () is the "src" URL for the image. The Morpheus [Archives](#) feature is a great resource for hosting images for use in Wiki. Archives can target cloud storage buckets or even file shares on the Morpheus appliance local storage. Morpheus generates an access URL for each file stored in Archives, simply target this URL in your markdown to show an image stored in Morpheus Archives within your Wiki pages.

5.2.6 Costing

Budgets

Budgets provide insight into spending across their designated scope, allowing users to create and plan a budget targeted to their account, clouds, tenants, users, or groups.

Creating A Budget

1. Navigate to **Operations > Costing > Budgets**
2. Start a new budget by clicking + *ADD*
 1. **Name**
 2. **Description**
 3. **Scope:** Here you can choose which construct this budget is tied to (Account, Tenant, Cloud, Group, or User)
 4. **Period:** Currently “Year” is the only option
 5. **Year:** Select a year to set budgets for future years. Alternatively, select “custom” to create a multi-year budget or input a custom fiscal year if required by your organization
 6. **Interval:** Choose Month, Quarter, Year then fill in the budgeted amount for that interval (for quarter and year interval Budgets the entered amount is evenly split across the months in the given interval)
3. Click *SAVE CHANGES*

CREATE BUDGET
×

NAME

DESCRIPTION

ENABLED

SCOPE

PERIOD

YEAR

START DATE

END DATE

INTERVAL

JANUARY

FEBRUARY

Multi-year Budgets

Some organizations may need to create multi-year budgets or may need to set a fiscal year period. By default, annual budgets are tied to a calendar year (January - December) but Morpheus offers the flexibility of setting a fiscal year period when needed. When selecting a value from the YEAR dropdown in the add/edit budget modal, select “Custom” rather than one of the discreet years from the list. After selecting Custom, START DATE and END DATE fields allow the user to input any desired fiscal year period. Users can enter a period of up to three years using the start and end date bookends. The entered period must be one, two or three full years, partial years are not permitted.

In the example below, I’ve created a three-year budget:

CREATE BUDGET ×

NAME

DESCRIPTION

ENABLED

SCOPE

PERIOD

YEAR

START DATE

END DATE

INTERVAL

	2020	2021	2022
YEAR	<input type="text" value="10000.0"/>	<input type="text" value="15000.0"/>	<input type="text" value="8000.0"/>

Budget Monitoring

As the year (or years) goes on, existing Budgets can be reviewed to compare actual spend against the budgeted amount. To access the Budget detail, navigate to Operations > Costing > Budgets and select the desired Budget. The reported actual amount for a given month will be the same as the total cost reported for the month on the Invoice with the same scoping (for the current month, projected cost is used). Depending on the Cloud type, this figure can be pulled from a public clouds live costing API (such as with AWS, Azure, or GCP Clouds) or from the Morpheus in-built cost metering for private clouds (like VMware).

Example Budget, Cloud-scoped:

AWS Annual Budget

EDIT DELETE

Name: AWS Annual Budget
Description:
Interval: Year

Created By: Alex Harker
Scope: Morpheus AWS - aharker
Start Date: 01/01/2023

Enabled: Yes
Period: Year
End Date: 12/31/2023

BUDGET SUMMARY

Table with 2 rows: Budget (\$1,000), Actual (\$115) for year 2023.

Monthly budget summary table with columns for JANUARY through DECEMBER. Actual for January is \$115.



Example Cloud Invoice for the same month:

Invoices > 1559

Morpheus AWS - aharker

\$115

MTD PRICE

\$115

ESTIMATED PRICE



PRICE HISTORY

\$48 ↓ 30%

SAVINGS

Summary History

INFO

PRICES

Summary table with columns for COMPUTE (\$102), MEMORY (\$0), STORAGE (\$12), NETWORK (\$0.01), EXTRA (\$0), MTD PRICE (\$115), and TOTAL (PROJECTED) (\$115). Includes a detailed cost breakdown table below.

Cloud Budgets

If you scope a budget to a Cloud, visit the Cloud summary tab in **Infrastructure > Clouds > Select Cloud** to see a cost-to-budget breakdown for that Cloud.

Summary Clusters Hosts VMs Containers Load Balancers Networks Data Stores Resources Policies Profiles Wiki

Microsoft Azure
 Appliance URL: default
 Datacenter ID:
 Guidance: manual
 Security Server: off
 Time Zone: default
 Agent Install Mode: cloudinit
 Network Mode: standard
 Domain:
 Storage Mode: standard
 Local Firewall: off

▼ AZURE RESOURCES

Instances : 30
 Running Instances : 14
 Volumes : 199
 Load Balancers : 7
 Subnets : 1
 Security Groups : 3

▶ GUIDANCE

2	1	0	0	0	\$158.93
TOTAL	MEMORY	CPU	POWER	STORAGE	SAVINGS

▼ COSTING

\$1,273	\$1,545	\$1,596	-\$51
MONTH TO DATE	ESTIMATED SPEND	LAST MONTH	CHANGE FROM LAST MONTH

RESERVATION COVERAGE

INSTANCE TYPE	ON DEMAND HOURS	RESERVED HOURS	TOTAL HOURS	COVERAGE
DV2/DSV2 SERIES	384.15	0.0	384.15	0.0%
A SERIES WINDOWS	1,366.82	0.0	1,366.82	0.0%
A SERIES BASIC	1,556.2	0.0	1,556.2	0.0%
AV2 SERIES	178.33	0.0	178.33	0.0%
BS SERIES	1,534.95	0.0	1,534.95	0.0%
AV2 SERIES WINDOWS	3.07	0.0	3.07	0.0%
BS SERIES WINDOWS	381.92	0.0	381.92	0.0%
A SERIES	383.35	0.0	383.35	0.0%
DV2/DSV2 SERIES WINDOWS	2.47	0.0	2.47	0.0%

Budget Analytics

In **Operations > Analytics > Budget Analysis** select scope (Account, Tenant, Cloud, Group, User) to view the budget analysis. If a budget exists for the selected scope, a cost breakdown against budgeted amounts will be shown.

BUDGET COST ANALYSIS

Search Account

APPLIED FILTERS Budget Scope: account

1 BUDGETS \$16,713 MONTH TO DATE \$300,000 BUDGETS TO DATE \$309,948 COST TO DATE \$9,948 TO BUDGET

BUDGET VS ACTUAL - DEMO BUDGET

	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
Budget	\$25,000	\$25,000	\$25,000	\$25,000	\$25,000	\$25,000	\$25,000	\$25,000	\$25,000	\$25,000	\$25,000	\$25,000
Actual	\$17,634	\$25,149	\$27,313	\$30,827	\$34,149	\$33,066	\$19,850	\$19,509	\$34,380	\$30,328	\$20,517	\$17,226

Invoices

Morpheus offers highly-granular costing data through invoices. Invoice views are highly customizable, allowing for nearly unlimited combinations of output columns and filtering. Invoices are based on monthly periods and allow you to slice costs in a highly granular way, as well as predict final monthly costs before the end of the period. As with other Morpheus UI pages that support advanced table customization, these views can be saved to provide easy access to costing views specific to varying business needs. By default, the invoice list page will show the last three months' invoices across all clouds and invoice types but the list can be modified to target differing time periods or resource groupings. Read on in this section to discover the meaning of various invoice types and how this tool can be used to create targeted costing views.

Invoices for Public vs On-Prem Clouds

Morpheus supports live cost syncing for many public clouds, including all of the most commonly-used public clouds like Amazon AWS, Microsoft Azure, and Google Cloud Platform. For these clouds, costing sync is enabled through the **COSTING** field on the Add/Edit Cloud modal (Infrastructure > Clouds > selected Cloud > EDIT button). When live costs are synced from public clouds, invoices will be generated for this activity including month-to-date costs, projected monthly totals, historical cost data, and a lot more which is discussed in the following sections.

Additionally, Morpheus has implemented a costing service for on-prem clouds to closely mirror the live costing experience through a public cloud. As with public clouds, this costing service is enabled for on-prem clouds through the **COSTING** field on the Add/Edit Cloud modal (Infrastructure > Clouds > selected Cloud > EDIT button). Change the setting to “Sync Costing” and save the changes to the cloud integration. From this point, costs will be aggregated into invoices for on-prem clouds in the same way as public clouds including complete line item listings, current costs for the month, month-end projections, and much more as discussed in the next sections.

Invoice List Page

The invoices list page shows high-level aggregate data on a group of invoices and allows you to locate a specific invoice for a deeper look. By default, invoices from the last three months across all clouds and types are shown here. Depending on settings, this page can automatically refresh itself so the list of displayed invoices and the aggregate information on them is up to date.

MORPHEUS Search Support Jaden Jabro

Operations Provisioning Library Infrastructure Backups Monitoring Tools Administration

Dashboard Reports Analytics Guidance Wiki Costing Approvals Activity

Budgets **Invoices** Usage

Search All Types All Clouds Start Date End Date

287,991

INVOICES

\$1,083,213

RUNNING PRICE

\$2,068,552

TOTAL PRICE

INVOICE ID	TYPE	REFERENCE	CLOUD	PERIOD	START DATE	END DATE	MTD PRICE	TOTAL PRICE
282029	Resource		Azure US	February 2022	2/1/22, 12:00 AM	2/28/22, 11:59 PM	\$0.12	\$0.15
292790	Resource		Azure US	February 2022	2/1/22, 12:00 AM	2/28/22, 11:59 PM	\$0.00	\$0.00
292809	Resource		Azure US	February 2022	2/1/22, 12:00 AM	2/28/22, 11:59 PM	\$0.00	\$0.00
282143	Resource		Azure US	February 2022	2/1/22, 12:00 AM	2/28/22, 11:59 PM	\$0.26	\$0.32
287435	Resource		Azure US	February 2022	2/1/22, 12:00 AM	2/28/22, 11:59 PM	\$0.00	\$0.00
292704	Resource		Azure US	February 2022	2/1/22, 12:00 AM	2/28/22, 11:59 PM	\$0.12	\$0.48
282114	Resource		Azure US	February 2022	2/1/22, 12:00 AM	2/28/22, 11:59 PM	\$0.24	\$0.29
292239	Resource		Azure US	February 2022	2/1/22, 12:00 AM	2/28/22, 11:59 PM	\$0.00	\$0.00
285619	Resource		Azure US	February	2/1/22, 12:00	2/28/22, 11:59	\$0.00	\$0.00

Aggregated Invoice Data

The following aggregate totals are displayed for all invoices picked up by set filters:

- **Periods:** The total number of months in the period determined by your start and end date filters. If no start and end dates are set, a three month period will be shown. If a one-month period is selected, the name of that month (ex. Aug 2020) is shown
- **Total Invoices:** The total number of invoices captured by current filters
- **MTD Cost:** The combined month-to-date cost for all invoices captured by current filters
- **Total Cost:** The expected total month-end cost for all invoices captured by current filters

Creating Views

Invoice list views are highly-customizable allowing for virtually limitless combinations of output columns and filtering. A common set of output columns is provided in a default view but users can add and remove columns from a large list of data points and even save multiple views for easy access to different data sets. Any of your stored views can be set as the default to be displayed each time the invoices list page is active.

To create a new invoices view:

1. Click the gear icon ()
2. Click on one or more columns from the “Columns” list to add or remove an output field
3. Click “+ add view”
4. Provide a name and a description value for your new view
5. If desired, mark the box to set the new view as your default view so it appears automatically each time the invoices list page is accessed
6. Click *SAVE*

The screenshot shows the Morpheus web interface. At the top, there's a navigation bar with the Morpheus logo, a search bar, and user information for Jaden Jabro. Below this is a secondary navigation bar with tabs for Operations, Provisioning, Library, Infrastructure, Backups, Monitoring, Tools, and Administration. Under 'Operations', there are sub-tabs for Dashboard, Reports, Analytics, Guidance, Wiki, Costing, Approvals, and Activity. The main content area is titled 'Invoices' and shows a summary of 287,991 invoices, a total running price of \$1,083,213, and a total of \$2,060,000. Below the summary is a table of invoice details with columns for Invoice ID, Type, Reference, Cloud, Period, Start Date, and End Date. A settings menu is open over the table, showing options for auto refresh, advanced filters, and a list of columns. The 'Columns' list includes 'invoice id', 'type', 'ref uuid', 'ref id', 'reference', 'cloud id', 'cloud', 'instance id', 'instance', and 'server id'. The 'invoice id', 'type', 'reference', and 'cloud' items are checked. The 'Columns' section of the menu is highlighted with a red box.

INVOICE ID	TYPE	REFERENCE	CLOUD	PERIOD	START DATE	END DATE
282029	Resource		Azure US	February 2022	2/1/22, 12:00 AM	2/28/22, 11:59 PM
292790	Resource		Azure US	February 2022	2/1/22, 12:00 AM	2/28/22, 11:59 PM
292809	Resource		Azure US	February 2022	2/1/22, 12:00 AM	2/28/22, 11:59 PM
282143	Resource		Azure US	February 2022	2/1/22, 12:00 AM	2/28/22, 11:59 PM
287435	Resource		Azure US	February	2/1/22, 12:00	2/28/22, 11:59

Available Output Columns

When creating an invoices view, there are many output columns available to select. Consult the list below for more details on each of the available columns:

Available Output Columns: **Expand for Complete List**

- **Invoice ID:** The unique ID in Morpheus for the invoice
- **Type:** The invoice type; Cloud, Container, Group, Server, Instance, Resource, User, or Volume
- **Ref ID:** An ID for the reference object tied to the invoice (server, instance, cloud, etc.). Reference IDs are reused across invoice types so invoices referring to identical Ref IDs may not necessarily refer to the same reference object
- **Reference:** The name of the reference object (server, cloud, user, group, etc.) tied to the invoice
- **Cloud ID:** The internal ID for a Cloud integration in Morpheus. This field will be blank unless the invoice references a Cloud
- **Cloud:** The name for a Cloud integration in Morpheus. This field will be blank unless the invoice references a Cloud
- **Instance ID:** The internal ID for an Instance in Morpheus. This field will be blank unless the invoice references an Instance
- **Instance:** The name for an Instance in Morpheus. This field will be blank unless the invoice references an Instance
- **Server ID:** The internal ID for a server in Morpheus. This field will be blank unless the invoice references a server
- **Server:** The name for a server in Morpheus. This field will be blank unless the invoice references a server
- **Cluster ID:** The internal ID for a cluster in Morpheus. This field will be blank unless the invoice references a cluster
- **Cluster:** The name for a cluster in Morpheus. This field will be blank unless the invoice references a cluster
- **Plan ID:** The internal ID for a service plan in Morpheus. This field will be populated only for invoices that reference an object which would be associated with a service plan (server, Instance, container, etc.).
- **Plan:** The name for a service plan in Morpheus. This field will be populated only for invoices that reference an object which would be associated with a service plan (server, Instance, container, etc.).
- **Group ID:** The internal ID for a Group in Morpheus. This field will be blank unless the invoice references a Group
- **Group:** The name for a Group in Morpheus. This field will be blank unless the invoice references a Group
- **User ID:** The internal ID for a User in Morpheus. This field will be blank unless the invoice references a User.
- **User:** The name for a User in Morpheus. This field will be blank unless the invoice references a User.
- **Tenant ID:** The internal ID for the Morpheus Tenant which owns the reference object
- **Tenant:** The name of the Morpheus Tenant which owns the reference object
- **Period:** The monthly period during which the invoice was generated
- **Interval:** The length of the invoice billing period, currently all invoices are generated at a one-month interval
- **Start Date:** The start date and time for the invoice period, typically the first of the month at midnight
- **End Date:** The end date and time for the invoice period, typically the last day of the month at midnight

- **Ref Start:** The date and time the reference object is created or the start of the invoicing period if the reference object existed prior to the start of the invoicing period
- **Ref End:** The date and time the reference object is decommissioned or the end of the invoicing period if the reference object still existed at the end of the period
- **Compute Cost:** The actual compute costs for the invoice (from public cloud costing API when available or from an on-prem cloud with “Sync Costing” enabled)
- **Storage Cost:** The actual storage costs for the invoice (from public cloud costing API when available or from an on-prem cloud with “Sync Costing” enabled)
- **Network Cost:** The actual network costs for the invoice (from public cloud costing API when available or from an on-prem cloud with “Sync Costing” enabled)
- **Extra Cost:** The actual additional costs for the invoice (from public cloud costing API when available or from an on-prem cloud with “Sync Costing” enabled)
- **MTD Cost:** The actual month-to-date costs for the invoice (from public cloud costing API when available or from an on-prem cloud with “Sync Costing” enabled)
- **Total Cost:** The actual total costs for the invoice (from public cloud costing API when available or from an on-prem cloud with “Sync Costing” enabled)
- **Metered Compute Cost:** Compute costs determined by Morpheus usage and pricing data (when live pricing data is not available, such as with an on-prem cloud without “Sync Costing” enabled)
- **Metered Storage Cost:** Storage costs determined by Morpheus usage and pricing data (when live pricing data is not available, such as with an on-prem cloud without “Sync Costing” enabled)
- **Metered Network Cost:** Network costs determined by Morpheus usage and pricing data (when live pricing data is not available, such as with an on-prem cloud without “Sync Costing” enabled)
- **Metered Extra Cost:** Additional costs determined by Morpheus usage and pricing data (when live pricing data is not available, such as with an on-prem cloud without “Sync Costing” enabled)
- **Metered MTD Cost:** Month-to-date costs determined by Morpheus usage and pricing data (when live pricing data is not available, such as with an on-prem cloud without “Sync Costing” enabled)
- **Metered Total Cost:** Total costs determined by Morpheus usage and pricing data (when live pricing data is not available, such as with an on-prem cloud without “Sync Costing” enabled)
- **Compute Price:** The actual compute price (cost plus markup) for the invoice (from public cloud costing API when available or from an on-prem cloud with “Sync Costing” enabled)
- **Storage Price:** The actual storage price (cost plus markup) for the invoice (from public cloud costing API when available or from an on-prem cloud with “Sync Costing” enabled)
- **Network Price::** The actual network price (cost plus markup) for the invoice (from public cloud costing API when available or from an on-prem cloud with “Sync Costing” enabled)
- **Extra Price:** The actual additional price (cost plus markup) for the invoice (from public cloud costing API when available or from an on-prem cloud with “Sync Costing” enabled)
- **MTD Price:** The actual month-to-date price (cost plus markup) for the invoice (from public cloud costing API when available or from an on-prem cloud with “Sync Costing” enabled)
- **Total Price:** The actual total price (cost plus markup) for the invoice (from public cloud costing API when available or from an on-prem cloud with “Sync Costing” enabled)
- **Metered Compute Price:** Compute price (cost plus markup) determined by Morpheus usage and pricing data (when live pricing data is not available, such as with an on-prem cloud without “Sync Costing” enabled)

- **Metered Storage Price:** Storage price (cost plus markup) determined by Morpheus usage and pricing data (when live pricing data is not available, such as with an on-prem cloud without “Sync Costing” enabled)
- **Metered Network Price:** Network price (cost plus markup) determined by Morpheus usage and pricing data (when live pricing data is not available, such as with an on-prem cloud without “Sync Costing” enabled)
- **Metered Extra Price:** Additional price (cost plus markup) determined by Morpheus usage and pricing data (when live pricing data is not available, such as with an on-prem cloud without “Sync Costing” enabled)
- **Metered MTD Price:** Month-to-date price (cost plus markup) determined by Morpheus usage and pricing data (when live pricing data is not available, such as with an on-prem cloud without “Sync Costing” enabled)
- **Metered Total Price:** Total price (cost plus markup) determined by Morpheus usage and pricing data (when live pricing data is not available, such as with an on-prem cloud without “Sync Costing” enabled)
- **Active:** Indicates whether or not the reference object is currently existing and active
- **Date Created:** The date and time the invoice is created
- **Last Updated:** The date and time the invoice was last updated

Invoice Types

Invoices can reference any of the Morpheus workload element types or resource reference types in the list below. Some invoice types are broader and may account for resource costs calculated in other narrower invoice types. For instance, a container-type invoice returns costs for a single node of an Instance while an Instance-type invoice for the same period may be including costs for that same single node. The invoices list view can be filtered to show just one type or all types. Complete descriptions of each invoice type are included below:

- **Cloud:** In Morpheus, a Cloud is any connection into a public cloud, private cloud, hybrid cloud, or bare metal server
- **Container:** A single node of a service, in other words, a single node of a Morpheus Instance. This could be a virtual machine or Docker container which is part of a Morpheus-managed Instance
- **Group:** In Morpheus, Groups define which resources a user has access to through their role. Clouds are added to Groups and users access Clouds to which their roles give access
- **Server:** A server refers to any individual host, virtual machine, or bare metal server that is inventoried or managed by Morpheus. This can include servers which are parts of Morpheus-managed Instances or inventoried servers from integrated Clouds
- **Instance:** A set of containers or virtual machines which correlate to a single horizontally-scalable entity. This could be a single VM or it could be many VMs operating as a service
- **Resource:** Resource-type invoices are generated when Morpheus cannot determine that the referenced costs belong to any of the other resource reference types in this list
- **User:** User-type invoices aggregate the costs of resources owned by a specific Morpheus user during the invoicing period
- **Volume:** When possible, costs will be tied to known volumes and a volume-type invoice is generated as a result

Invoice Detail Page

Summary

The summary tab of the invoice detail page displays a great deal of reference information about the resource identified by the invoice. This will vary depending on the type of resource. In addition, total and projected costs are displayed along with cost breakdowns for compute, storage, network, and other categories. Month-to-date totals and final month projections are given.

M MORPHEUS
 Search
Support
Jaden Jabro

Operations Provisioning Library Infrastructure Backups Monitoring Tools Administration

Groups Clouds Clusters Compute Network Load Balancers Storage Trust Boot

Clouds > AWS

✓
AWS

EDIT
REFRESH
DELETE

Location: US West (N. California) Last Sync: 02/24/2022 11:44 AM Sync Duration: 14 seconds

\$4,762
COST THIS MONTH

8112.98
COST HISTORY


0%
MAX CPU

0%
MEMORY

1%
STORAGE

2
CONTAINER HOSTS

0
HYPERVISORS

0
BARE METAL

351
VIRTUAL MACHINES

36
DISCOVERED

Summary
Clusters
Hosts
VMs
Containers
Load Balancers
Networks
Resources
Policies
Profiles
Wiki



Appliance URL: default

Datacenter ID:

Guidance: manual

Security Server: off

Time Zone: default

Agent Install Mode: cloudinit

Network Mode: AWS

Domain: bw.bertramlabs.com

Storage Mode: lvm

Local Firewall: off

▶ AMAZON RESOURCES

▶ CURRENT ALARMS 4 OPEN 0 ERROR 4 WARNING

▶ GUIDANCE 2 TOTAL 2 MEMORY 0 CPU 0 POWER 0 STORAGE -\$0.00 SAVINGS

▼ COSTING

\$4,762
MONTH TO DATE

\$8,113
ESTIMATED SPEND

\$8,482
LAST MONTH

-\$369
CHANGE FROM LAST MONTH

SERVICES BREAKDOWN

	CLOUDTRAIL	\$0.12	i
	CLOUDWATCH	\$0.00	i
	CONFIG	\$16.16	i
	EC2	\$3,592.63	i
	EFS	\$0.00	i
	LAMBDA	\$0.00	i
	OTHER	\$552.09	i
	QUEUESERVICE	\$0.00	i

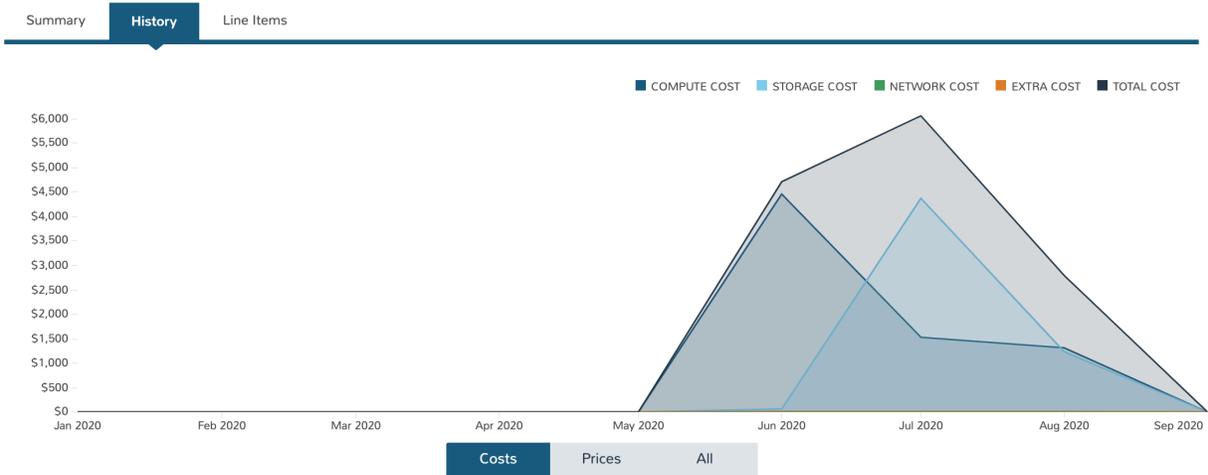


5.2. Operations

161

History

The history tab displays the costs and prices for the associated resource over time. This tab is especially valuable for resources that have existed through at least a few invoicing periods to show changes over time. In addition, cost breakdowns for compute, storage, network, and other categories are shown for each invoicing period. These costs can be displayed visually through graphs.



PERIODS

INVOICE ID	PERIOD	COMPUTE	STORAGE	NETWORK	EXTRA	MTD	TOTAL
1326425	September 2020	\$0	\$0	\$0	\$0	\$43.84	\$1,944.02 (Projected)
1309811	August 2020	\$1,309.98	\$1,227.51	\$9.56	\$0	\$2,783.84	\$2,783.84
1300026	July 2020	\$1,524.81	\$4,361.13	\$9.64	\$0	\$6,044.27	\$6,044.27
1279589	June 2020	\$4,448.53	\$61.58	\$8.24	\$0.00	\$4,518.35	\$4,700.00
404134	May 2020	\$0	\$0	\$0	\$0	\$105.37	\$105.37
394365	April 2020	\$0	\$0	\$0	\$0	\$66.99	\$66.99
401131	March 2020	\$0	\$0	\$0	\$0	\$31.58	\$31.58
387577	February 2020	\$0	\$0	\$0	\$0	\$40.62	\$40.62
382300	January 2020	\$0	\$0	\$0	\$0	\$22.31	\$22.31

Line Items

For supported resource types, Morpheus includes a tab to display all costing line items. This provides the user with a complete list of line items that make up the costing totals on the invoice.

ID	START DATE	END DATE	USAGE TYPE	USAGE	RATE	UNIT	COST	ITEM NAME
1612407	8/1/20 12:00 AM	8/30/20 1:00 PM	USW1-CW:Requests	1.0078647E7	1.0E-5	Requests	\$100.79	QA Amazon
1614136	8/1/20 6:00 PM	8/1/20 7:00 PM	USW1-DataTransfer-Out-Bytes	9.4892E-6	0.0897631338	GB	\$0.00	QA Amazon
1614137	8/1/20 6:00 PM	8/1/20 7:00 PM	USW1-DataTransfer-Out-Bytes	7.7188E-6	0.0897631338	GB	\$0.00	QA Amazon
1614138	8/1/20 6:00 PM	8/1/20 7:00 PM	USW1-LCUUsage	1.833333E-4	0.00799995	LCU-Hrs	\$0.00	QA Amazon
1614139	8/1/20 6:00 PM	8/1/20 7:00 PM	USW1-LoadBalancerUsage	1.0	0.028	Hrs	\$0.03	QA Amazon
1614140	8/1/20 6:00 PM	8/1/20 7:00 PM	USW1-LoadBalancerUsage	1.0	0.0252	Hrs	\$0.03	QA Amazon
1619670	8/1/20 7:00 PM	8/24/20 10:00 PM	us-west-1-KMS-Requests	39.0	3.0E-6	Requests	\$0.00	QA Amazon
1617343	8/3/20 3:00 PM	8/30/20 1:00 PM	USW1-LoadBalancerUsage	646.0	0.0252	Hrs	\$16.28	QA Amazon
1620003	8/5/20 4:00 PM	8/5/20 5:00 PM	USW1-DataProcessing-Bytes	6.1E-7	0.008	GB	\$0.00	QA Amazon
1620006	8/5/20 4:00 PM	8/5/20 5:00 PM	USW1-LCUUsage	2.055556E-4	0.00799995	LCU-Hrs	\$0.00	QA Amazon
1620008	8/5/20 4:00 PM	8/5/20 5:00 PM	USW1-LoadBalancerUsage	1.0	0.0252	Hrs	\$0.03	QA Amazon
1620005	8/5/20 4:00 PM	8/5/20 6:00 PM	USW1-DataTransfer-Out-Bytes	6.9551000000000005E-6	0.0897631338	GB	\$0.00	QA Amazon
1620004	8/5/20 4:00 PM	8/6/20 7:00 AM	USW1-DataTransfer-Out-Bytes	3.4650800000000004E-5	0.0897631338	GB	\$0.00	QA Amazon
1620007	8/5/20 4:00 PM	8/6/20 7:00 AM	USW1-LoadBalancerUsage	2.0	0.028	Hrs	\$0.06	QA Amazon
1620726	8/6/20 6:00 AM	8/6/20 7:00 AM	USW1-DataTransfer-Out-Bytes	7.5875E-6	0.0897631338	GB	\$0.00	QA Amazon
1620727	8/6/20 6:00 AM	8/6/20 7:00 AM	USW1-LCUUsage	2.055556E-4	0.00799995	LCU-Hrs	\$0.00	QA Amazon
1620728	8/6/20 6:00 AM	8/6/20 7:00 AM	USW1-LoadBalancerUsage	1.0	0.0252	Hrs	\$0.03	QA Amazon
1621067	8/6/20 3:00 PM	8/6/20 4:00 PM	USW1-DataTransfer-Out-Bytes	2.2820000000000002E-7	0.0897631338	GB	\$0.00	QA Amazon
1623759	8/11/20 8:00 AM	8/11/20 9:00 AM	USW1-DataTransfer-Out-Bytes	1.08387E-5	0.0897631338	GB	\$0.00	QA Amazon
1623762	8/11/20 8:00 AM	8/11/20 9:00 AM	USW1-LoadBalancerUsage	1.0	0.028	Hrs	\$0.03	QA Amazon

Usage

The *Operations > Costing > Usage* section shows billing information for Instances and hosts that have pricing configured on their Service Plan.

Important: Pricing must be enabled in Administration > Settings > Provisioning and Service Plans configured with price sets in Administration > Plans & Pricing for pricing to show in the Usage section.

View Usage

All Instances, discovered resources, virtual images, snapshots are listed by default, with the most recent usage information showing first. When additional details are available, usage records will display a small arrow on the left side of the row. Click this arrow to expand the details for that usage record. Details can include more granular cost breakdowns, such as specific CPU, memory, and/or storage costs for containers (Instances).

Usage records can be filtered by Cloud, type and date:

Cloud Default view is for all Clouds. Select a Cloud to show Instance, host and other usage for only one Cloud.

Date Default view shows most current Usage. Select the Date filter to scope to a different date range.

Type All usage record types are shown by default, select a specific type to filter the list to just one

Example: Below is an example of a discovered instance from AWS, which shows different usage periods and the pricing that would be related. In this case, the compute and storage are used to calculate the Usage Price when the instance is **Running** but only the storage when the instance is **Stopped**:

CLOUD	TYPE	NAME	PLAN	START DATE	END DATE	USAGE STATUS	USAGE PRICE
▼ AWS-US-EAST-2	Discovered	bastion	T2 Medium - 2 Core, 4GB Memory	11/10/2022 09:07 AM		Stopped	\$1.4006
	Price Type		Quantity				
	Storage	gp2	35 GB				\$1.4006
▼ AWS-US-EAST-2	Discovered	bastion	T2 Medium - 2 Core, 4GB Memory	11/10/2022 08:29 AM	11/10/2022 09:07 AM	Running	\$0.0324
	Price Type		Quantity				
	Compute		N/A				\$0.0293
	Storage	gp2	35 GB				\$0.003
▼ AWS-US-EAST-2	Discovered	bastion	T2 Medium - 2 Core, 4GB Memory	08/27/2022 07:36 PM	11/10/2022 08:29 AM	Stopped	\$8.5582
	Price Type		Quantity				
	Storage	gp2	35 GB				\$8.5582
▶ AWS-US-EAST-2	Discovered	Bastion-1	T2 Large - 2 Core, 8GB Memory	06/21/2022 08:03	08/27/2022 07:39	Stopped	\$17.5605

API & CLI

Usage information can also be extracted via the Morpheus API and CLI, including the ability to extract usage per Tenant.

Note: Appropriate Role permissions for *Operations: Usage* are required to view the Usage section.

5.2.7 Approvals

Morpheus and Service Now Approvals

Overview

Policies can be created for Groups and Clouds to require approvals for actions with the built-in Morpheus approvals engine, or via a ServiceNow integration. Approvals can be configured for Provisioning and Lifecycle extensions.

Configuring Approvals

Configuring Morpheus for Approvals

To configure Morpheus for approvals:

1. Configure Roles for Approval access
2. Optionally configure a ServiceNow Integration for ServiceNow approvals.
 - Please note ServiceNow integration is not required for Internal Approvals.
3. Create approvals policies for:
 - Internal Approvals
 - SNOW Approvals

Configure Roles

Configure User Role access settings in Administration > Roles > (Role) > Operations: Approvals.

- All Users with a Role applied containing Operations: Approvals set to Full will have approval authority, and be able to Approve, Deny or Cancel approval requests.
- All Users with a Role applied that has Operations: Approvals set to Read will be able to view Approval requests and history, but will not be able to Approve, Deny or Cancel approval requests.
- All Users with a Role applied that has Operations: Approvals set to None will not have access to the Operations: Approvals section, and such will not be able to see or act on approval requests.
- Regardless of Role settings, any instance or app provisioned by any user to a group or cloud with an active Approval policy applied will require approval before the instance or app will provision.

ServiceNow Approvals

Configure ServiceNow integration for SNOW Approvals

1. Navigate to Admin > Integrations
2. Select + **NEW INTEGRATION**
3. Select **ServiceNow** from the Type dropdown in the Integration modal and enter:
 - **Name** Name of the integration in Morpheus
 - **Enabled** Leave checked to enable the integration.
 - **Host** URL of the ServiceNow host (ex: <https://ven0000.service-now.com>)
 - **User** A User in ServiceNow that is able to access the REST interface and create/update/delete incidents, requests, requested items, item options, catalog items, workflows, etc.
 - **Password** Password for User above
4. Save Changes

Morpheus then configures the integration with ServiceNow, syncs ServiceNow workflows which are available when creating approvals policies. (This process can take up to 5 minutes depending on the size of the workflow table in ServiceNow.)

Create Approval Policies

- Policies applied to a Group are created in Infrastructure > Groups > (group) > Policies tab.
- Policies applied to a Cloud are created in Infrastructure > Clouds > (cloud) > Policies tab.

To create an Approval policy:

1. Navigate to the Policies tab in the Group or Cloud to which the policy will apply.
2. Select + **ADD POLICY** to open the New Policy wizard
3. Select Provision Approval from the Type dropdown
4. Add an optional description
5. **Leave Enabled selected for this Policy to be active once saved.**
 - Enabled can be deselected to disable to policy.

6. In the config section, select either Internal Approvals or ServiceNow Approvals:
 - **Internal Approvals** Approval requests will be managed within Morpheus via the Operations: Approvals section.
 - **ServiceNow Approvals** Approval requests will be managed with ServiceNow (SNOW). Please note a ServiceNow integration (Admin: Integrations) must be configured prior to SNOW Approval policy generation.
 - For ServiceNow Approvals, select the appropriate ServiceNow workflow for this policy. Please note the workflows presented are created in ServiceNow and synced with Morpheus .
7. Add the Morpheus Accounts to which this policy will apply, or leave the Accounts field blank to apply to all accounts.
8. Save

Upon saving, a new policy is created in the Group or Cloud Policies tab.

Note: SNOW Approvals will take a few moments to save as the policy is generated.

Managing Approval Requests

Once Instance approval policies are added to a Group or Cloud, any Instance or App provisioned into that Group or Cloud will create an approval request entry in the *Operations > Approvals* section.

Note: User Role permission *Operations: Approvals > FULL* is required to manage Approvals.

- To Approve, Deny, or Cancel an internal approval request, select the request and use the Actions dropdown.
- To Cancel a ServiceNow Approval request, select the request and use the Actions dropdown. ServiceNow approvals are managed in ServiceNow.

Note: Instances requiring provisioning approval will have a PENDING status until approved.

Each Approval Request will have:

- Name: A name for the approval in Morpheus
- Monthly Price (Est.): The estimated monthly price of the Instance to be provisioned
- Request Type: What is being requested, such as Instance approval
- External Name: For ServiceNow integrations, the name of the approval in the ServiceNow console
- Type: Internal or ServiceNow
- Status
- Date Created
- Requested By: The Morpheus user making the request
- **Actions dropdown (for internal approval requests)**
 - Approve
 - Deny

- Cancel
- **Actions dropdown (for ServiceNow requests)**
 - Cancel

Note: The Approvals list view can be sorted by NAME, REQUEST TYPE, EXTERNAL NAME, DATE CREATED, and REQUESTED BY

Internal approval requests

To Approve, Deny or Cancel an Internal approval request:

1. Navigate to *Operations > Approvals*
2. Select the Name of the Approval request
3. Select Actions on the far right of the request
4. Select Approve, Deny, or Cancel from the Actions dropdown
5. Select OK on the confirmation modal
 - When an Internal request is approved, the related instance will begin to provision immediately and the request will show approved.
 - When an Internal request is denied, the related instances status will change to Denied and the request will show Rejected in the Approvals section.
 - When an Internal request is canceled, the related related instances status will change to Cancelled and the request will be canceled.

ServiceNow Approval requests

ServiceNow approval request are managed in ServiceNow. The process of approving or rejecting requests is determined by the ServiceNow Workflow selected when configuring the SNOW Approval policy. These Workflows are configured in ServiceNow.

Important: Morpheus syncs with ServiceNow every 5 minutes. Once an Approval Request is Approved or Rejected in Service Now, it will take up to 5 minutes for the instance to respond accordingly, and the status for the approval request in the Approvals section in Morpheus to update.

5.2.8 Activity

The Activity section displays a recent activity report for Auditing. Morpheus defines an activity as any major action performed on an instance or server, such as, but not limited to adding a server, deleting a server, provisioning an instance, deleting an instance, creating a backup, etc... This view can be searched and filtered by type, user, and date range.

Activity

There are four types of activities that are displayed in the Activity Reports:

- Backup
- Provisioning
- Alert
- Permissions

To View a Recent Activity report:

1. Select the **Operations** link in the navigation bar.
2. Select **Activity** in the sub navigation bar.

Recent activity is displayed in order from recent to oldest. This view can be searched and filtered by type, user, and date range.

Review

To review the item the activity occurred on, click the name of the activity and it will go to a new page and display that item.

Note: Deleted activities are displayed as an alert and do not contain a link to the event item. If the activity is not a deletion event we provide a link on the activity name to go to the item the activity occurred on. Delete activity alerts are shown for Instances, servers, Clouds, Groups, and Monitoring Checks.

To Filter:

1. Click the filter drop down of type of filter you want to apply.
2. Select the appropriate filter.

Alarms

The *ALARMS* section shows Operation notifications from Cloud and other Service Integrations. Cloud and other Service Integration Alarms are not generated by Morpheus but synced and displayed for visibility in Morpheus.

History

The *HISTORY* section shows Process History from Instances and Apps processes. This is an aggregate view of the History tab in Instance and App details pages.

Processes can be expanded to view all process steps and process history detail including output and errors.

Access to *HISTORY* is given by the `Operations:Activity` Role permission.

The Logs displayed in **Administration - Health - Morpheus Logs** are from `/var/log/morpheus/morpheus-ui/current`. These logs show all ui activity and are useful for troubleshooting and auditing.

Note: Stack traces in **Administration - Health - Morpheus Logs** are filtered for Morpheus services. Complete stack traces can be found in `/var/log/morpheus/morpheus-ui/current`.

5.3 Provisioning

There are several capabilities in the Morpheus provisioning engine. Things ranging from application / service deployments via containers, virtual machines, and even bare metal. Deployment management and app template construction are also core aspects of the provisioning engine. Take advantage of custom tasks and workflows within any environment by building tasks and workflows from those tasks. There is a lot of information to cover with regards to provisioning but Morpheus makes it intuitive and smooth.

5.3.1 Requirements

Provisioning Instances and Apps typically involves many steps beyond starting a workload. Morpheus is centered around automating everything desired for your application to be fully operational, including networking, storage, hostnames, domains, dns, licenses, scripts/automation, scaling, load balancers, security, accessibility, governance, auditing, monitoring, backups, costs, sizing and on and on. Point being there is a lot that goes on when spinning up an instance or app, and to make the magic happen a few requirements need to be met.

Important: By default, Agent Installation is enabled when provisioning unless deselected on the Virtual Images or *SKIP AGENT INSTALL* is selected when provisioning.

VM Provision Steps

While an infinite number of steps can happen when provisioning an Instance or App using VM(s) in Morpheus, the basic order is:

- Look for Virtual Image

Morpheus will check if the Virtual Image set on the Node Type or selected during provisioning is already available in the source Cloud. If not and it is an Uploaded/Local Image, Morpheus will attempt to upload the Image to the target Cloud.

Upload Image

For Uploaded/Local Images that do not exist in the target cloud, Morpheus will need to upload the Image. Ensure the Virtual Image is valid for the target Cloud, the Image meets the target cloud upload requirements, and Morpheus has network access and permissions to upload the image.

Note: When uploading an image to a VMware Cloud, the Virtual Image is copied directly to the target ESXi host, NOT through the vCenter server. Ensure the Morpheus Appliance(s) can resolve target ESXi hostnames and connect on port 443 for successful vmdk/ova uploads.

Clone Image

Once the Image is confirmed available in the target cloud, Morpheus will clone the Image to the target Datastore.

Note: The target host must have access to the target Datastore of the Image

- Reconfigure Image

Once cloned, Morpheus will resize the Image based off provisioning parameters

- Cloud-init (if enabled)

Attached cloud-init iso When using cloud-init, Morpheus will attach a tiny metadata iso to the new VM. Network, Machine, User and any other cloud-init metadata will be sourced from this iso.

VM Tools Morpheus will run Guest Customizations via VMware VM Tools, including network config when assigning static IP's.

- Wait for Power On status and Network info

Morpheus will wait to hear back from the target cloud/hypervisor that the VM has successfully started and has an IP address.

Note: If VM TOOLS INSTALLED? is NOT checked on the source Virtual Image configuration, Morpheus will skip waiting for network.

- Finalize

By default this will include Agent Installation and any post-provision scripts or workflows or integration automation steps.

Important: If the VM is stuck in finalize for long periods of time, this typically means the Agent cannot be installed or has not been heard back from. This will result in a ! warning Instance status upon provisioning completion.

If agent installation is not possible or desired, uncheck “Install Agent” on the source Virtual Image configuration or select “Skip Agent Install” during provisioning to speed up provisioning completion.

Virtual Images

While containers are the future, the most common provisioning method involves Virtual Machines, and the most important part of Provisioning a VM is the Virtual Image. When provisioning a VM, Morpheus will need to do a few things depending on the location of the Virtual Image and if agent install, console access, and script execution is desired.

Synced Images need to be properly configured Morpheus gathers as much metadata for synced images as possible, but depending on the cloud, os, image configuration, agent install settings, by default the synced Virtual Images may not be ready to provision until configured. The Virtual Image is already at the target Cloud, but datastore selection, credentials, cloud-init settings, and networks and security settings on the Virtual Image can cause provisioning issues.

Local/Uploaded Virtual Images Images uploaded to Morpheus are configured during the *Add Virtual Image* process, however Morpheus in most scenarios will still need to copy the Virtual Image to the target Hypervisor/Cloud upon the first provision to the target Cloud. In addition to the requirements for provisioning a synced Virtual Image, copying an uploaded Virtual Image to the target Cloud is required and network and image configurations can cause upload failures, resulting in provisioning issues.

Marketplace Images AWS and Azure marketplace Images can be provisioned using the generic Amazon or Azure Instance Types, or added as Virtual Images as scoped to Node Types for custom Instance Types. Marketplace items provisioned/added to Morpheus still fall upon the requirements of the target Cloud, such as matching the region with the Image and licensing.

Synced Images

When a Cloud is added to Morpheus, all available Images/Templates records from that Cloud will be synced in regardless of Inventory settings on the Cloud. These Image records will be available in the Virtual Images section and can be provisioned by using the target clouds generic Instance Type, ie VMware, Amazon, Azure, Openstack etc Instance Types, or by creating custom Instance Types and selecting the Image on a Node Type.

Note: Synced Virtual Images are just meta-data records in Morpheus pointing to the Image in the target Cloud. The actual Image files are not copied/imported to Morpheus.

Before provisioning a synced Virtual Images, ensure the image is configured properly:

Name Name of the Virtual Image in Morpheus. This can be changed from the name of the Image, but editing will not change the name of the actual Image.

Operating System Specifies the Platform and OS of the image. All Windows images will need to have Operating System specified on the Virtual Image, as Morpheus will assign Linux as the Platform for all Images without Operating System specified.

Minimum Memory The Minimum Memory setting will filter available Service Plans options during provisioning. Service Plans that do not meet the Minimum Memory value set on the Virtual Image will not be provided as Service Plan choices.

Cloud Init Enabled? On by default, uncheck for any Image that does not have Cloud-Init or Cloudbase-Init installed.

Important: Provisioning a Virtual Image that has *Cloud Init Enabled?* checked on the Virtual Record in Morpheus but does not have cloud-init installed will result in immediate provisioning failure.

Install Agent On by default, uncheck to skip Agent install. Note this will result in the loss of utilization statistics, logs, script execution, and monitoring. (Some utilization stats are collected for agent-less hosts and vm's from VMware and AWS clouds).

Username Existing Username on the Image. This is required for authentication, unless Morpheus is able to add user data, Cloud-Init, Cloudbase-Init or Guest Customizations. If Cloud-Init, Cloudbase-Init Guest Customizations or Nutanix Sysprep are used, credentials are defined in Administration > Settings > Provisioning and User Settings. If credentials are defined on the Image and Cloud-Init is enabled, Morpheus will add that user during provisioning, so ensure that user does not already exist in the image (aka root). For Windows Guest Customizations, Morpheus will set the Administrator password to what is defined on the image if Administrator user is defined. Do not define any other user than Administrator for Windows Images unless using Cloudbase-init. Morpheus recommends running Guest Customizations for all Windows Images, which is required when joining Domains as the SID will change.

Password Password for the Existing User on the image if Username is populated.

Storage Provider Location where the Virtual Image will be stored. Default Virtual Image Storage location is /var/opt/morpheus/morpheus-ui/VMs. Additional Storage Providers can be configured in Infrastructure > Storage.

Cloud-Init User Data Accepts what would go in runcmd and can assume bash syntax. Example use: Script to configure satellite registration at provision time.

Permissions

Set Tenant permissions in a multi-tenant Morpheus environment. No impact on single-tenant environments.

Visibility

Private Image is only available in the specified Tenants below.

Public Image is available to all Tenants.

Tenant If Visibility is set to Private, specify Tenants the Image will be available for.

Auto Join Domain? Enable to have Instances provisioned with this image auto-join configured domains (Windows only, domain controller must be configured in Infrastructure > Network and the configured domain set on the provisioned to Cloud or Network).

VirtIO Drivers Loaded? Enable if VirtIO Drivers are installed on the image for provisioning to KVM based Hypervisors.

VM Tools Installed? On by default, uncheck if VMware Tools (including OpenVMTools) are not installed on the Virtual Image. Morpheus will skip network wait during provisioning when deselected.

Force Guest Customization? VMware only, forces guest customizations to run during provisioning, typically when provisioning to a DHCP network where guest customizations would not run by default. This is required for host/computer name definitions, domain joining, licenses and user definitions when using DHCP.

Trial Version Enable to automatically re-arm the expiration on Windows Trial Images during provisioning.

Enabled Sysprep? Applicable to Nutanix Only. Enable if the Windows Image has been sysprepped. If enabled, Morpheus will inject Unattend.xml through the Nutanix API (v3+ only)

Important: Provisioning a Virtual Image that has *Cloud Init Enabled?* checked on the Virtual Record in Morpheus but does not have cloud-init installed will result in immediate provisioning failure.

Important: For Linux images without cloud-Init, an existing username and password must be defined on the Virtual Image record for Agent Install, Domain joining, licensing, script execution and other automation, and SSH or RDP Console access.

Local Virtual Images

A Local Virtual Image means it has been uploaded to Morpheus. To provision, Morpheus will need to upload the Image to the target Cloud upon first provision.

- Ensure the Virtual Image is valid for the target Cloud, the Image meets the target cloud upload requirements, and Morpheus has network access and permissions to upload the image.

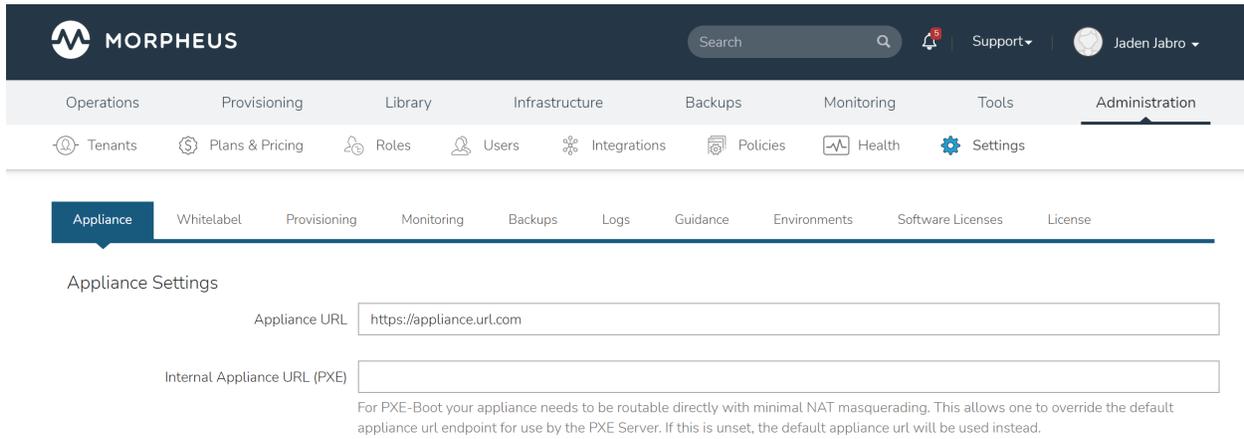
Note: When uploading an image to a VMware Cloud, the Virtual Image is copied directly to the target ESXi host, NOT through the vCenter server. Ensure the Morpheus Appliance(s) can resolve target ESXi hostnames and connect on port 443 for successful vmdk/ova uploads.

Once a Local Virtual Image has been uploaded to a Cloud, subsequent provisions will use the Image local to the cloud instead of uploading again as long as the copied image is still available in the source Cloud.

Agent Install

When provisioning an instance, there are some network and configuration requirements to successfully install the morpheus agent. Typically when a vm instance is still in the provisioning phase long after the vm is up, the instance is unable to reach Morpheus, or depending on agent install mode, Morpheus is unable to reach the instance.

The most common reason an agent install fails is the provisioned instance cannot reach the Morpheus Appliance via the `appliance_url` set in Administration > Settings over both 443 and 80. When an instance is provisioned from Morpheus, it must be able to reach the Morpheus appliance via the `appliance_url` or the Agent will not be installed.



Appliance Settings

Appliance URL

Internal Appliance URL (PXE)

For PXE-Boot your appliance needs to be routable directly with minimal NAT masquerading. This allows one to override the default appliance url endpoint for use by the PXE Server. If this is unset, the default appliance url will be used instead.

In addition to the main `appliance_url` in Administration > Settings, additional `appliance_urls` can be set per cloud in the Advanced options of the cloud configuration pane when creating or editing a cloud. When this field is populated, it will override the main appliance url for anything provisioned into that cloud.

Tip: The Morpheus UI current log, located at `/var/log/morpheus/morpheus-ui/current`, is very helpful when troubleshooting agent installations.

Agent Install Modes

There are 3 Agent install modes:

- ssh/winrm
- VMware Tools
- cloud-init

For All Agent Install modes

When an instance is provisioned and the agent does not install, verify the following for any agent install mode:

- The Morpheus `appliance_url` (Administration > Settings) is both reachable and resolvable from the provisioned node.
- The `appliance_url` begins with `https://`, not `http://`.

Note: Be sure to use `https://` even when using an IP address for the appliance.

- Inbound connectivity access to the Morpheus Appliance from provisioned VM's and container hosts on port 443 (needed for agent communication)
- Private (non-morpheus provided) vm images/templates must have their credentials entered. These can be entered/edited in the Library > Virtual Images section but clicking the Actions dropdown of an image and selecting Edit.

Note: Administrator user is required for Windows agent install.

- The instance does not have an IP address assigned. For scenarios without a DHCP server, static IP information must be entered by selecting the Network Type: Static in the Advanced section during provisioning. IP Pools can also be created in the Infrastructure > Networks > IP Pools section and added to clouds network sections for IPAM.
- DNS is not configured and the node cannot resolve the appliance. If DNS cannot be configured, the IP address of the Morpheus appliance can be used as the main or cloud appliance.

SSH/Winrm

Linux Agent

- Port 22 is open for Linux images, and SSH is enabled
- Credentials have been entered on the image if using custom or synced image. Credentials can be entered on images in the Library > Virtual Images section.

Windows Agent

- Port 5985 must be open and WinRM enabled for Windows images.
- Credentials have been entered on the image if using custom or synced image. Credentials can be entered on images in the Library > Virtual Images section.

Note: Administrator user is required for Windows agent install.

VMware tools (vmtools) rpc mode

- VMware tools is installed on the template(s)
- Credentials have been entered on the Image if using uploaded or synced image when Cloud-init or Guest Customizations or Sysprep for Windows are not used. Credentials can be entered on Images in the Library > Virtual Images section.

Cloud-Init agent install mode

- Cloud-Init is configured in Administration > Settings > Provisioning section
- Provisioned image/blueprint has Cloud-Init (linux) or Cloudbase-Init (windows) installed

5.3.2 Provisioning Concepts

Morpheus is a powerful infrastructure-agnostic Cloud Application Management Platform. Compared to other CMP platforms in the space, some terminology and concepts may differ. These concepts are documented in this section along with places where terminology may be slightly different compared with other platforms or with common industry parlance.

Morpheus refers to itself as a CAMP (Cloud Application Management Platform) as opposed to a (Cloud Management Platform). While that may seem minor, it actually is a big deal. Many CMP applications start at the IaaS layer and work up to the application layer (often needing additional PaaS architectures to fill out the model). Morpheus was designed from a middle-ground perspective. As such, some concepts are a bit different. This provides a more complete platform that allows for greater capabilities out of the box as will be seen when these concepts are covered.

Instances

Morpheus starts with provisioning Instances. In some platforms, an Instance is representative of a singular object like a virtual machine in Amazon AWS. In Morpheus, this concept was rethought. An Instance is more of a representation of a resource or service. This service may involve several virtual machines or several Docker containers.

For example, in the Morpheus Instance wizard, MongoDB is an option and contains several Instance configurations. One of these configurations is a full MongoDB cluster consisting of either seven virtual machines or seven Docker containers. Rather than representing these directly as seven individual “instances”, Morpheus groups them together into a singular Instance of a service that contains multiple containers or virtual machines within it. This even allows for Instance actions that can be performed to expand capacity on an Instance (either horizontally or vertically). In the past, a database server may have been representative of a singular server, but this model has drastically changed in a big data world. This same concept also can apply to something like a simple Apache web server where there are 10 copies of a web server which are horizontally scaled out to handle traffic.

When viewing an Instance detail page, one is able to look at details and statistics specific to a virtual machine or container. Morpheus simply helps simplify the management model for tracking these services.

Containers / Nodes / Virtual Machines

In relation to Instances, an Instance can have many nodes. A node is a generic representation of a container or a virtual machine. In most cases, Morpheus will represent a node as a Container or Virtual Machine depending on the provisioning engine used for the Instance. Node is just a generic naming representation when referring to these types of items. The public Morpheus developer API, however, often refers to both virtual machines and Docker containers as “containers”. The UI was since updated to better delineate this concept for easier understanding but, in essence, the name is valid for both concepts of containerized environments as well as Virtual Machines. In fact, one can even think of a Docker Host as a Hypervisor (which we do).

Hosts / Servers

This concept is mostly tailored to users of Morpheus who are responsible for managing and maintaining the underlying infrastructure integrations. A Host typically refers to a Docker Host in which a container (within an Instance) is running, or a hypervisor that virtual machines can be provisioned onto. A server is the underlying general representation of a physical or virtual server. It could be a Host representation, a Virtual Machine, or even a Bare Metal delineation.

When a user provisions a VM-based Instance, a corresponding server record is created to represent the link to the actual resource via the underlying provisioning engine. This may seem a bit odd but provides an aspect of Morpheus that is quite powerful. This singular concept is what allows Morpheus to ingest “brownfield” environments. We do not need to start clean. Morpheus can be integrated into existing environments and manage existing virtual machines. The way Morpheus does this is by periodically syncing existing VMs from the added cloud integrations. A server record will be created and periodically updated (every five minutes, by default) with realtime information and changes. This, in essence, provides CMDB-like capabilities as well. When a server is discovered, the user (given the appropriate access) can convert the virtual machine to a managed Instance. When this is done, a corresponding Instance is made in the provisioning section of Morpheus and the Morpheus Agent can optionally be installed to provide more refined guest operating system-level statistics and logging.

Apps

On top of all the previous concepts, Morpheus provides an Apps layer. An App is a collection of Instances linked together via application tiers. Tiers allow the user to define segregated sections of connectivity between the various elements (Instances) within an application. Once these Instances are all linked together in an application concept, this may affect Instance environments and provide service discovery capabilities for them to cross connect. There are several service discovery aspects within Morpheus as well as integrations with services.

App Blueprints

An App Blueprint allows a user to define an application structure for easy reproducibility and deployment into various environments. They can be used to mix and match various Instance types to provision an application dependent on multiple layers of services.

5.3.3 Catalog

The Catalog presents a simplified self-service view where users can select and deploy Instances, Blueprints or Workflows with pre-defined configuration in just a few clicks and without presenting an overwhelming list of options. Selections are presented as tiles and users can add items to a cart as if shopping on an eCommerce website. For users who tend to provision regularly from a small selection of Instance Types and configurations, the catalog is an easier option compared with the much more detailed and option-rich Instance provisioning wizard.

Configuring Catalog Item Access

Within the Catalog, users are presented with selections based on their User Role. Additionally, Catalog Item access can be set on the Tenant Role to restrict certain items from all users in the Tenant. By default, User Roles have no access to any catalog items. Thus, administrators will need to enable access to some Catalog Items in order for users to make use of this view.

Configuring Global Access:

- **Full:** Gives access to all Catalog Items
- **Custom:** Gives access to individually-selected items from the list below

- **None:** No access is given to any Catalog Items

Tip: When giving Custom access, be sure to set access on some of the individual catalog items to Full in order to reveal those items to the Role group.

Catalog

The catalog shows the complete list of predefined items available to the user for provisioning. Catalog items are not created here, however. For more on creating catalog items, see the Catalog Items tab in the Morpheus Library section (Library > Blueprints > Catalog Items).

The screenshot shows the Morpheus user interface. At the top is a dark navigation bar with the Morpheus logo and the name 'MORPHEUS'. On the right side of the bar, there is a shopping cart icon and a user profile icon for 'Alex Harker'. Below the navigation bar is a horizontal menu with icons and labels for 'Dashboard', 'Catalog', 'Instances', 'Apps', 'Executions', and 'Order History'. The main content area is titled 'CATALOG' and contains a grid of four catalog items, each with a star icon in the top right corner:

- Docker Simple Blueprint:** Includes a 'simple' tag and 'don't delete' text. An 'ORDER' button is at the bottom.
- NGINX Simple Instance:** Includes a 'simple' tag and 'don't delete' text. An 'ORDER' button is at the bottom.
- Apache Amazon Options:** Includes the Apache logo and 'api options' text. An 'ORDER' button is at the bottom.
- Resource:** Includes a globe icon and 'ap-required-field' text. An 'ORDER' button is at the bottom.

Below the catalog grid is an 'ORDER HISTORY' section containing a table with the following data:

ORDER PLACED	ITEMS ORDERED	ORDER STATUS
09/22/2020	1	Complete

Below the table, there is a row showing the Apache logo and the text 'tm-persona-apache'.

Placing Orders

From the Catalog page, select the tile for your chosen item to see any custom options that may need to be set prior to provisioning. The catalog shows a complete list of items but the list can be filtered by entering search terms or by selecting a category. When adding or editing catalog items, an optional category may be set which aids in filtering for environments which have a lot of catalog items to select from.

The screenshot displays the Morpheus application's 'Amazon Options' catalog page. At the top, a dark navigation bar contains the Morpheus logo and the name 'MORPHEUS' on the left, and a shopping cart icon and the user name 'Alex Harker' on the right. Below this is a horizontal menu with icons and labels for 'Dashboard', 'Catalog', 'Instances', 'Apps', 'Executions', and 'Order History'. The main content area has a breadcrumb 'Catalog > Order' and a header for the 'Amazon Options' catalog item, which includes the Apache HTTP Server logo and the text 'Amazon Options' and 'api options'. The item description, 'Amazon Apache', states: 'This is a sample catalog item that provisions a single amazon apache VM. Enjoy!'. To the right of the description is a configuration panel with several dropdown menus: 'Group' (All Clouds), 'Environment' (Dev), 'Cloud' (QA Amazon), 'Resource Pool' (labs), 'Plan' (T2 Micro - 1 Core, 1GB Memory), and 'Security Group' (default). Below these is a text input field for 'Name' containing 'My Workload'. The price is shown as '\$11.3016 / Month'. At the bottom of the configuration panel are two buttons: a dark blue 'ADD TO ORDER' button and a light blue 'ORDER NOW' button.

Once the item is in the cart, make any additional selections to complete the order. Once finished, proceed to the cart by clicking on the cart icon at the top of the application window. Each selected item is listed along with its estimated cost. The total estimated cost for the entire order is also computed.

The screenshot shows the 'Review Order' page in the Morpheus interface. At the top, there is a dark navigation bar with the Morpheus logo, a shopping cart icon with a '1' notification, and the user name 'Alex Harker'. Below this is a secondary navigation bar with icons for Dashboard, Catalog, Instances, Apps, Executions, and Order History. The main content area is titled 'Review Order' and features a card for 'Amazon Options' with the Apache HTTP Server logo. The card contains configuration details: Group: All Clouds, Environment: Dev, Cloud: QA Amazon, Resource Pool: labs, Plan: T2 Micro - 1 Core, 1GB Memory, Network: 172.31.0.0/20 (subnet-22ba157b), Security Group: default, Name: My Workload, and Price: \$11.3016 / Month. A 'Total: \$11.3016 / Month' is displayed at the bottom right of the card. A blue 'PLACE ORDER' button is located at the bottom right of the page.

Once *PLACE ORDER* is clicked, the provisioning process will begin and the user is redirected back to the catalog page. Any new orders can now be viewed from their respective list pages. Depending on whether you've ordered an Instance, App, or Workflow execution, navigate to the appropriate list page for your item to view it.

Order Detail

The screenshot shows the 'Order History' page in the Morpheus interface. It features the same top navigation bar as the previous screenshot. The main content area is titled 'ORDER HISTORY' and contains a table with the following data:

ORDER PLACED	ITEMS ORDERED	ORDER STATUS
09/22/2020	1	Complete

Below the table, there is a row for the order item: 'tm-persona-apache' with the Apache HTTP Server logo. A grey tooltip message 'This item has been deleted' is visible over the 'Complete' status.

The Order Detail page includes a complete list of orders and some basic details about them. If the item still exists, you can link through to the detail page for the item (whether that be Instance detail, App detail, or Execution detail). When the item name is not hyperlinked, the item has been deleted but the record of the order remains in the history.

5.3.4 Instances

Instances are a great starting point for taking advantage of self service features and spinning up both VMs and containers. In Morpheus it may be advisable to cover the definition of a few terms used within the application so as to reduce confusion. These concepts are also covered in greater detail in the [Provisioning Concepts](#) section.

Instance A set of containers or virtual machines that can correlate to a single horizontally-scalable entity or a service suite, like a database. It is important to note that an Instance can contain one or more containers/VMs depending on the Instance type and configuration.

Container Typically a docker container provisioned via a Morpheus Docker host.

Virtual Machine A virtualized compute server provisioned onto various hypervisor hosts.

The top of the main Instances page shows overall statistics for the listed Instances, including count, status, and resource utilization. You can search for Instances by name, or filter by group, instance type, or category.

Note: Instances listed are determined by group access and role permissions. When filtered to show Instances of “All Statuses”, any Instances which are in a state of pending removal due to a delayed delete policy in place are not shown under this filter. Instead you must filter for “Pending Removal” to see these Instances and prevent deletion, if desired.

The Instance list contains important information about each instance, including the instance name, environment tag, instance type icon, IP address and port info, Instance version, the number of virtual machines or containers in the instance, the group the instance is in, and the cloud or clouds the instance is in.

Creating Instances

The Instance catalog is the one-stop shop for selecting items to be provisioned and pieced together. It contains not only basic container and VM options but also tailored services for SQL databases, NoSQL databases, cache stores, message busses, web servers, and even full-fledged apps. The list contains a lot of items to choose from and they are represented to the user based on what provisioning engines are enabled and integrated in the Morpheus environment.

To get started, simply click the + *Add* button in the upper right of the Provisioning > Instances section. A modal will display allowing the catalog to be searched. Once an item is selected it is just a matter of following the steps through the wizard.

Tip: The Instance catalog can be customized via role-based access control (RBAC) thereby restricting access to non-sanctioned catalog items, as well as added to via the Library section. It is completely customizable.

The next step will ask for a Group and Cloud to be selected. The Group is an abstract representation that can contain multiple cloud integrations. Clouds can be in multiple Groups and Groups are also useful for using RBAC to restrict provisioning access and set retention policies. If the environment is new and these do not yet exist, it may be advisable to refer to one of our starter guides, such as the guide on getting started with Morpheus and [VMware](#). The wizard continues by allowing us to choose a name for the Instance as well as an environment.

Note: Currently the Environment option is mostly useful for presenting the user with informative metadata around the Instance when coming back to it later.

Moving on, it is now time to configure the Instance. Depending on the Instance Configuration that is chosen, fields will change. This can include cloud-specific fields (i.e. Datastore for VMware or Network). There will also be options like setting an initial user account. Some of these fields are optional and will be represented as such.

Configuration options provided in this screen are very powerful. An example is MySQL where a Master/Slave or Master/Master layout can be selected. These configurations will automatically deploy two MySQL VMs or containers and link them together to provide replication. These types of configurations exist for a wide range of Instance types and are optimized for high performance and scale. It is even possible to provision entire sharded MongoDB clusters.

One last step before the Instance can be provisioned is the Automation step. This wizard step may or may not appear depending on the capabilities of the Instance type or previous configurations in the account. It is here one can easily select a post-provisioning workflow to run (see more on Tasks and Workflows elsewhere in Morpheus documentation), assign a load balancer, or even configure the backup job that gets created.

Now that the steps are completed for provisioning the selected Instance type, simply review your selections and complete. The Instance will automatically show up in the Instances list and its provisioning state will be represented. Depending on what was provisioned this step can range from seconds to minutes (typically a container configuration will be rather quick if the Instance type has previously been provisioned before).

Converting Discovered Resources to Managed Instances

When creating new cloud integrations (or updating existing ones), users may opt for Morpheus to onboard any existing resources that currently reside in the Cloud. For example, these may be virtual machines that exist on vCenter hosts prior to integration with Morpheus, EC2 instances pre-existing on an Amazon AWS account, or virtual machines that are running on a KVM host. With the Add/Edit Cloud modal open, mark INVENTORY EXISTING INSTANCES for Morpheus to automatically onboard these resources. Not only will Morpheus inventory these instances at the time the cloud is integrated (or updated), it will also continue to poll the target cloud every five minutes (by default) for newly added or removed servers. Users can see these discovered servers by looking in Infrastructure > Compute. Depending on the type of resource, it may appear on the Virtual Machines tab, the Containers tab, or another tab. Additionally, we can see a list of discovered servers on Cloud detail pages (Infrastructure > Clouds > Selected Cloud). Just click on the tabs for VMs, Containers or Hosts tab. Discovered resources will be indicated as such whereas containers which are associated with a managed Instance will be marked as a “Managed”.

Additionally, Morpheus allows users to convert discovered resources into managed Instances. Begin from the server detail page (Infrastructure > Compute > Virtual Machines > selected machine) and from the ACTIONS menu select “Convert to Managed”. At this point, we must make a number of selections:

- Assign to the primary Tenant or one of the Subtenants
- Select a Group (this dropdown contains a filtered list of Groups which the associated Cloud is in)
- Username and password for a seeded account
- Opt to install Morpheus Agent or not (for more on Morpheus Agent, click [here](#))
- Select the Instance Type which should be associated with the new Instance containing this VM
- Select a version number for the Instance (such as 20.04 for a basic Ubuntu Instance)
- Select a Layout, Instance Types often have multiple Layout configurations
- Identify the operating system
- Select a Plan (this dropdown contains a filtered list of plans which correlate to the size of the VM)

Finally, click *EXECUTE*. Once this process is completed, the server will be indicated as “Managed” in the servers list. Additionally, a new Instance will appear on the Instances List page (Provisioning > Instances). We can now work with it in the same way we can work with any other Instance, such as by adding it to an App or expanding the Instance horizontally with added nodes.

Instance Details

The instance detail page is where you can view and fully manage an instance. To get to an instance detail page, navigate to Provisioning > Instances, and click on an Instance. Please note Instance details and actions will differ between Instance types and user permissions.

There are several sections within an Instance page that provide useful capabilities to the user.

The screenshot shows the Morpheus web interface. At the top, there's a navigation bar with 'MORPHEUS' logo, a search bar, and user profile 'Alex Harker'. Below is a menu with 'Provisioning' selected, showing sub-items like 'Instances', 'Apps', 'Catalog', 'Jobs', 'Executions', and 'Code'. The main content area shows the instance 'qa-hosts-kube' with a status of 'Running'. It includes a summary section with details like Group (Permanent), Owner (Tommy Marks), Cores (2), Price (\$464.00 / Month), Cloud (QA VMware), Layout (VMware VM), Memory (8.0GiB), Source Image (Morpheus Ubuntu 20.04.1 v1), Date Created (02/16/2022 01:23 PM), Version (20.04), Total Storage (160.0GiB), and Provision Time (2 minutes 3 seconds). Below the summary is a table of instance resources.

HEALTH	LAST BACKUP	AVAILABILITY	RESPONSE TIME	MAX CPU	MEMORY	STORAGE
?	-	100.00%	N/A	73%	23%	0%

Summary	Resources	Runtime	Storage	Network	Monitoring	Backups	History	Costing	Console	Wiki
Group: Permanent Owner: Tommy Marks Cores: 2 Price: \$464.00 / Month			Cloud: QA VMware Layout: VMware VM Memory: 8.0GiB Source Image: Morpheus Ubuntu 20.04.1 v1			Date Created: 02/16/2022 01:23 PM Version: 20.04 Total Storage: 160.0GiB Provision Time: 2 minutes 3 seconds				

STATUS	NAME	ADDRESS(ES)
▶	qa-hosts-kube-1	10.███
▶	qa-hosts-kube-2	10.███
▶	qa-hosts-kube-3	10.███
▶	qa-hosts-kube-4	10.███

Summary Basic information, stats and status information

Deploy Track deployment history for instance types that support deployments or manually kick off a deployment (only visible for Instance Types that support deployments)

Settings Some Instance Types support custom configuration settings (for example, MySQL presents the my.ini)

Resources VMs, containers, or other resources associated with the Instance are listed here. Some Instance Types, such as XaaS Instances, will not have resources and the tab is not displayed

Runtime View the environment variables presented to the Instances or exported to the Instances via Apps (more on this in the Apps section). Users may also see Imported environment variables that may be referenced by the running Instance.

For Instances that support load balancing and auto scaling, configure auto scaling thresholds and load balancer settings in the Scale subsection that pertain to a particular Instance.

The software subsection will show any tracked software which is Installed as part of the provisioning process and is being tracked.

Storage See storage information associated with the Instance including the list of volumes and controllers which are associated with each machine that makes up the Instance.

Network Useful for configuring network interfaces for your VMs or security groups which allow access to the Instance.

Monitoring Quick summary of the monitoring system and all checks that were configured to test the state of the Instance. Stats views (memory, cpu, etc.) can be zoomed out to a 90-day view if desired (in global settings, ensure your stats retention setting will support this). Logs and guidance for the individual Instance are also shown in their respective subtabs.

Backups Quick backup dashboard. Useful for viewing historical backups and snapshots as well as adding new backup jobs.

History See historical information related to automation which has been run against the Instance. This is useful for examining automation which was run as part of a phase of a Provisioning Workflow. Users can also drill into the Workflows to examine individual Tasks, including viewing the output from these Tasks to confirm success or troubleshoot issues.

Costing Invoices pertaining to the Instance are displayed here. See the Instance level or host level invoices along with individual line items. In the History subtab view historical pricing data to monitor trends. In the Prices subtab view any prices which have been created and used to build a metered costing profile for the workload.

Console Access the Instance or container via a client-less Console supporting SSH, RDP, VNC, or even hypervisor-level remote consoles.

Wiki View the Wiki page for this Instance or edit the existing Wiki page (which may currently be blank). The content field supports markdown formatting, see the main Wiki section of Morpheus documentation for additional details.

Managing Instances

Instance actions allow you to perform numerous management tasks on instances. The actions available depend on the instance type, hypervisor, roles permissions, and instance state.

Edit Edit the Name, Description, Environment, Group, Metadata, Tags, and Owner for the Instance.

Delete Deletes the Instance.

Important: Deleting an Instance will delete the actual VM's or Containers and cannot be undone, unless a Delayed Removal policy has been applied prior to the Deletion. To delete Instances without deleting associated VM's, delete the Instances VM record(s) from the Infrastructure section with "Remove Infrastructure" deselected and select "Remove Associated Instances" in the VM delete modal options. This will delete the records in Morpheus but leave the infrastructure in place.

Tip: You can change the owner of an instance easily by selecting the edit button and entering a new owner in the corresponding field.

Actions

Available options in the Actions dropdown can include:

Suspend Puts the VM in a suspended state without shutting down the OS.

Stop/Start/Restart Service Stops, Starts or Restarts the service associated with the Instance Type.

Stop/Start/Restart Server Stops, Starts or Restarts the Virtual Machine.

Import as Image Clones and exports VM in its current state to target Storage provider and adds Virtual Image Record with metadata matching the source Instance's configuration.

Clone to Image Clones and converts VM in its current state to image in the source Cloud and adds Virtual Image Record with metadata matching the source Instance's configuration.

Lock/Unlock Instance A locked instance cannot be deleted until it is unlocked.

Reconfigure The Reconfigure action allows service plan, disk, cpu, ram, networks and storage controller changes. Available options depend on the instance type and service plan configuration. Some resize actions require an instance restart.

Clone Creates a new Instance from the Instance at its current state.

Backup Immediately executes a backup of the Instance. Only available for Instances with backups enabled.

Run Workflow Presents workflow options and then immediately runs selected Workflow on the Instance. Workflows can be created in the Library > Automation section.

Run Script Presents Script options and immediately executes selected Script on the Instance. Scripts can be created in the Library section.

Apply Template Presents Template options and immediately applies selected Template to the Instance. Templates can be created in the Library section.

Add Node Adds an additional node to the configuration. Additional options and configurations are required in the add node wizard depending on instance configuration and type.

Eject Disk Ejects attached disk/iso.

Add Slave Adds a database slave in the Instance.

Change Master Changes the database Master node in an Instance.

Clone to Template (VMware) Creates a new VMware Template from the Instance with corresponding Morpheus Virtual Image record.

Tip: Scrolling down in the Actions dropdown may be necessary to see all options.

Performing Instance Actions

1. Select the Provisioning link in the navigation bar.
2. Click the Instance from the list of instances you wish to perform an action on.
3. Click the Actions drop down button and select an Action.

Notes

Every Instance has a Wiki section for adding useful information about the Instance. Wiki can be added by selecting the Wiki tab button on the bottom of Instance Detail pages. Instances with associated VMware VM's will bi-directionally sync Morpheus Instance Wiki content and VMware VM Notes. See the [Wiki](#) Section for more details.

Tip: Markdown Syntax is supported in Wikis.

5.3.5 Apps

Apps allow instances having general relationships to be grouped in a clean and organized manner. App functionality enables full control of which instances belong in an app as well setting Firewall and Access Control List (ACL) rules. Use Apps to structure all necessary components into a single place. Add checks and groups for web servers, database nodes, etc.

Apps can be created from Blueprints, which are made in Library > Blueprints > App Blueprints or from Existing Apps.

Creating Apps

New Apps can be created from Blueprints or using existing Instances.

Creating Apps from Blueprints

1. Click **+ADD** on the right side of the main Apps section in Provisioning.
2. Select an existing App Blueprint and click **NEXT**.

Note: App Blueprints must be created in Library > Blueprints > App Blueprints to appear as options when provisioning an App.

3. Enter a Name for the App and select a Group. Default Cloud and Env can also be selected.
4. Click **NEXT**. Blueprint configurations matching the Group, Cloud and Environment selections will auto-populate the configurations of the Instances in the App. If no Blueprint Configuration matched the Group, Cloud or Env selections, the Instances will have default configurations.
5. Configure your Instances. Depending on the Blueprint Configurations settings, instances may already be fully configured. Fields that are locked in a Blueprint cannot be edited when creating an App.

Note: Once an Instance is fully configured, a green checkmark will appear next to the Instance. Instances that have required fields that need populated will have a red X and must be completed. If your Blueprint is already fully configured, you can simply select **COMPLETE**!

6. Select **COMPLETE** and the App will be created, and the Instances will begin provisioning.

NEW APP ✕

TEMPLATE
SETUP
CONFIGURE

Builder
Raw
Preview

STRUCTURE

- Spud Marketing
 - Web
 - Tomcat ✓
 - Database
 - MySQL ✓

CONFIGURATION

▼ Instance Info

NAME

DESCRIPTION

▼ Configuration Options

CLOUD

VERSION

INSTANCE CONFIGURATION

PLAN
Memory: 1 GB Price: \$5.13 / Month

VOLUMES GB

HOST

▶ Advanced Options

▶ Automation

▶ Deployment

▶ Load Balancer

▶ Backups

▶ Lifecycle

PREV
COMPLETE

Creating Apps from Existing Instances

1. Click **+ADD** on the right side of the main Apps section in Provisioning.
2. Select **APP FROM EXISTING INSTANCES** from the Blueprints list and click **NEXT**.
3. Enter a Name for the App and select a Group. Default Cloud and Env can also be selected.

Note: Only instances within the selected Group and Cloud will be available to be added to the App.

4. In the **STRUCTURE** section, select **+** to add a Tier

5. Select or enter a Tier Name.
6. Select the Tier to set Boot Order, rename, or once multiple Tiers are added, connect the Tier to other Tiers.
7. In the STRUCTURE section, select + in a Tier to add an Instance
8. Select the Instance Type of the Existing Instance to be added to the App.
9. In the STRUCTURE section, select the Instance.
10. In the CONFIGURATION section, select the Cloud the Existing Instance is in. Existing INSTANCES that match the Group, Cloud and Instance Types set will populate.
11. Select the desired Instance from the INSTANCES list. Selected instance will show in the SELECTED INSTANCE section.

Note: Only one existing Instance can be added per Instance. To add multiple Existing Instances, repeat the step above including adding an Instance for each Existing Instance to be added to the App.

12. Once all Existing Instances have been selected, click *COMPLETE*.
13. A new App will be created out of the Existing Instances.

Builder
Raw
Preview

STRUCTURE

- Morpheus HA01 +
 - App + ✕
 - CentOS ✓
 - CentOS
 - Database + ✕
 - PERCONA
 - Messaging + ✕
 - RabbitMQ
 - ES + ✕
 - elastic

CONFIGURATION

USE EXISTING

INSTANCE

CLOUD Nutanix

SEARCH search for instances

SELECTED INSTANCE

MORPHEUS-HA-AP1

SSH: 10.30.20.181:22 SSH: 10.30.20.62:22

VERSION: 7.3

CONTAINERS:2

STATUS: ▶

INSTANCES

JW-CENTOS-AHV	VERSION: 7.3	CONTAINERS:3
SSH: 10.30.20.165:22 SSH: 10.30.20.91:22 SSH: 10.30.20.149:22		
	STATUS: ▶	
JW-CENTOS-AHV-1	VERSION: 7.3	CONTAINERS:3
SSH: 10.30.20.135:22 SSH: 10.30.20.37:22 SSH: 10.30.20.52:22		
	STATUS: ●	
JW-CENTOS-AHV-2	VERSION: 7.3	CONTAINERS:1
SSH: 10.30.20.104:22		
	STATUS: ▶	
MORPHEUS-HA-DB	VERSION: 7.3	CONTAINERS:3
SSH: 10.30.20.153:22 SSH: 10.30.20.139:22 SSH: 10.30.20.196:22		
	STATUS: ▶	
MORPHEUS-HA-MQ	VERSION: 7.3	CONTAINERS:2
SSH: 10.30.20.129:22 SSH: 10.30.20.58:22		
	STATUS: ▶	
MORPHEUS-HA-MQ-LB-1	VERSION: 7.3	CONTAINERS:1
SSH: 10.30.20.53:22		
	STATUS: ▶	

PREV
COMPLETE

Managing Apps

App Status

App Status is determined by the status of the Instances within the App or by the DELETE action. All Instances in an App must be in 'Running' status for the App status to equal 'Running'.

Table 13: **App Statuses**

Status Icon	App Status	Status Reason
	Running	All Instance Statuses = Running
	Failed	Any Instance Status = Failed
	Pending	Any Instance Status = Pending
	Pending Removal	Any Instance Status = Pending Removal
	Provisioning	Any Instance Status = Provisioning
	Removing	The DELETE action was trigger on the App
	Unknown	Any Instance Status = Unknown, or the App is empty

Editing an App

The *EDIT* action allows permissioned users to update an Apps metadata, Environment, Group and Owner.

1. Navigate to Provisioning > Apps
2. Select an App from the list to view the App detail page
3. Select *EDIT*
4. Update the following
 - NAME** App Name
 - DESCRIPTION** App Description
 - ENVIRONMENT** App Environment
 - GROUP** App Group assignment
 - OWNER** User assigned as Owner of the App
5. Select *SAVE CHANGES*

Deleting an App

The *DELETE* action allows permissioned users to delete an App and, by default, all Instances within the App.

1. Navigate to Provisioning > Apps
2. Select an App from the list to view the App detail page
3. Select DELETE
4. The DELETE APP? confirmation modal will be displayed:
 - Remove Instances** Deletes all Instances associated with the App - Enabled by Default
 - Preserve Backups** Preserves Backups of the Instances associated with the App - Disabled by Default
 - Preserve Volumes** Preserves Storage Volumes of the Instances associated with the App - Disabled by Default

Force Delete Forces deletion of the App - Required when any Instances associated with the App are in “provisioning” state - WARNING Force deleting may cause orphaned infrastructure or resources. - Disabled by Default

5. Select *DELETE* to proceed with deleting the App, or *CANCEL* to abort the delete action.

Exporting Configuration JSON

To export an App Blueprint as JSON:

1. Navigate to Provisioning > Apps
2. Select an App from the list to view the App detail page
3. Select *ACTIONS* > Export
4. The App export file will be downloaded to your computer as `{app_name}.json`

5.3.6 Jobs

Jobs are for scheduled execution of Automation Tasks and Workflows. Jobs can be set to execute on a schedule, at one specific point in time, and/or execute manually (on-demand). Jobs are linked to existing Tasks or Workflows, and allow for custom configuration options. Jobs can be associated with Instances, Servers, or have no association, such as a job for an SSH task.

Jobs allow for scheduled execution of nearly anything as Tasks Types include Bash, Powershell, HTTP/API, Ansible, Chef, Puppet, Groovy, Python, jRuby, Javascript, and library scripts and templates, which can be configured for resource, remote, or local execution targets. If you need something to execute on a schedule, Morpheus Jobs can deliver.

Jobs are configured in the JOBS tab, and the JOB EXECUTIONS tab contains Job execution history with result output.

Jobs

Required Role Permissions **Click to Expand/Hide**

Provisioning: Jobs

- **None:** Cannot access Provisioning > Jobs > Jobs tab
- **Read:** Can access Provisioning > Jobs > Jobs tab but cannot create, edit, or delete Jobs
- **Full:** Full permissions to create, view, edit, and delete Jobs

Provisioning: Job Executions

- **None:** Cannot access Provisioning > Jobs > Job Executions tab
- **Read:** Can access and view Provisioning > Jobs > Job Executions tab including job execution history, status, and Job output

Creating Jobs

Note: Jobs require existing Tasks or Workflows. See the appropriate section of Morpheus docs for more on creating [Tasks](#) and [Workflows](#).

To create a new job:

1. Navigate to Provisioning > Jobs
2. Select + *ADD*
3. Enter the following
 - **NAME:** Name of the Job in Morpheus
 - **JOB TYPE:** A Task Job will execute a selected Task, a Workflow Job will execute a selected Workflow
 - **ENABLED:** When checked, the Job will run as scheduled
4. Select *NEXT*
5. Configure the Job

Task Jobs

TASK: Select target Task. If relevant to the Task, Input fields will be presented

SCHEDULE:

Manual: Job is not scheduled but can be executed from Provisioning > Jobs and selecting Actions > Execute

Date And Time: Job will be executed at one specific point in time and not again (unless rescheduled or executed manually)

Schedule: Select a configured Execution Schedule. Execution Schedules are created in Library > Automation > Execute Scheduling

Note: Morpheus provides two default execution schedules, *Daily at Midnight* and *Weekly on Sunday at Midnight*. Any additional schedules were created by a User. Additional schedules can be added in Library > Automation > Execute Scheduling

CONTEXT TYPE: Server or Instance

CONTEXT SERVER/INSTANCE: Select the Server or Instance you wish to target with the Job

RUN NOW: When checked, the Job will execute on save regardless of SCHEDULE setting.

Workflow Jobs WORKFLOW: Select target Workflow. If relevant to the Workflow, Input fields will be presented

SCHEDULE: Manual: Job is not scheduled but can be executed from Provisioning > Jobs and selecting Actions > Execute

Date And Time: Job will be executed at one specific point in time and not again (unless rescheduled or executed manually)

Schedule: Select a configured Execution Schedule. Execution Schedules are created in Library > Automation > Execute Scheduling

Note: Morpheus provides two default execution schedules, **Daily at Midnight** and **Weekly on Sunday at Midnight**. Any additional schedules were created by a User. Additional schedules can be added in **Library > Automation > Execute Scheduling**

CONTEXT TYPE: Server or Instance

CONTEXT SERVER/INSTANCE: Select the Server or Instance you wish to target with the Job

RUN NOW: When checked, the Job will execute on save regardless of SCHEDULE setting.

6. Select *NEXT*
7. Select *COMPLETE*

Creating and Running Security Scan Jobs

Security Scan Jobs allow users to create and schedule SCAP program (Security Content Automation Program) scans for groups of managed systems. These Jobs can call in existing SCAP packages and checklists, which are used to scan the targeted systems on-demand or on a scheduled basis. Historical data for these scans is saved in the Job Execution list and in the software section of server detail pages. Detailed scan reports can also be viewed for each system as needed once the scan is complete. See the [SCAP documentation](#) on the NIST website for information on developing your own scanning procedures.

Note: Creating and editing Security Scan Jobs requires the “Security: Scanning” Role permission set to Full. Viewing Security Scan Jobs and seeing the results for scanned servers requires at least a Read-level permission.

Add a new Security Scan Job

Note: New security scan packages are added in Morpheus Library rather than here in the Jobs section. Ensure you have uploaded the desired security package in **Library > Templates > Security Packages** before proceeding with new security Job creation.

1. Navigate to **Provisioning > Jobs > Jobs Tab**
2. Click **+ADD**
3. Set the Job type to “Security Scan Job” and provide a friendly name for the Job
4. Click *NEXT*

NEW JOB [X]

SETUP > CONFIGURE > REVIEW

Job Summary

NAME: Scan Job

JOB TYPE: Security Scan Job

ENABLED

PREVIOUS NEXT

5. Select a security package, see the previous section to add a new one
6. Enter your Scan Checklist (XML document) and Security Profile (XCCDF document), more information on these can be found in the SCAP documentation linked above
7. Set a schedule or leave as Manual to only run this scan on-demand (new execution schedules can be created in Library > Automation if needed)
8. Set the context, can be Instance or Server. Select as many Instances or Servers as needed for this scanning run
9. Click *NEXT*
10. After final review, click *COMPLETE*

EDIT JOB [X]

SETUP **CONFIGURE** REVIEW

Job Configuration

SECURITY PACKAGE: scap security bundle 0.151

SCAN CHECKLIST: /scap-security-guide-0.151/ssg-ubuntu1804-ds.xml

SECURITY PROFILE: xccdf_org.ssgproject.content_profile_cis

Execution Config

SCHEDULE: Daily

RUN NOW

CONTEXT TYPE: Server

CONTEXT SERVER: Search

de-ubuntu-1 [X]

PREVIOUS NEXT

Running Security Scan Jobs

Once created, Security Scan Jobs will run based on the configured schedule. They can also be run on-demand when needed:

1. Navigate to Provisioning > Jobs > Jobs Tab
2. Click *MORE*
3. Click “Execute”

The screenshot shows the Morpheus dashboard with the following data:

- JOBS: 290
- EXECUTIONS TODAY: 9
- DAILY EXECUTIONS: 18
- DAILY JOB STATUS: Completed (9), Failed (0)

The 'Job Executions' table is as follows:

LAST RESULT	TYPE	NAME	DETAILS	LAST RUN	NEXT RUN
✓	Workflow Job	[REDACTED]	[REDACTED]	11/14/2019 02:42 PM	[REDACTED]
✓	Security Scan Job	Centos 8 SCAP Scan Job	[REDACTED]	01/23/2022 12:43 PM	[REDACTED]
✗	Security Scan Job	[REDACTED]	[REDACTED]	04/22/2021 09:15 AM	[REDACTED]

Viewing Completed Security Scan Jobs

To view a list of completed Security Scan Jobs (and Jobs of other types):

1. Navigate to Provisioning > Jobs > Job Executions Tab
2. Additional details can be viewed by clicking (i)

The screenshot shows the 'Job Executions' tab selected. The statistics are:

- JOBS: 290
- EXECUTIONS TODAY: 10
- DAILY EXECUTIONS: 19
- DAILY JOB STATUS: Completed (9), Failed (1)

The 'Job Executions' table shows a specific job:

✓	Ubuntu check - ah-sb-morpheus-ubuntu-4001	Start Date: 02/22/2022 06:29 AM	Duration: 3m 21s	User: [REDACTED]
---	---	---------------------------------	------------------	------------------

To view scan results for specific servers:

1. Navigate to the server detail page (Infrastructure > Hosts > Virtual Machines tab > Selected server)
2. Click on the Software tab part way down the page, then click on the Security subtab
3. High level details on previous scans is viewable here

The screenshot shows the Morpheus Security interface. At the top, there are navigation tabs: Summary, Wiki, Storage, Network, Logs, **Software**, History, and Console. Below these are sub-tabs for Software and Security. A search bar is present with the text 'Search' and a magnifying glass icon. A table displays scan results:

STATUS	NAME	TYPE	SCAN DATE	SCORE	RESULTS
✓	scap security bundle 0.1.51	SCAP Package	10/15/2020 06:54 PM	28.12	29 passed 39 failed 3
✓	scap security bundle 0.1.51	SCAP Package	10/15/2020 06:48 PM	28.12	29 passed 39 failed 3

4. To view the full report, click (i)

Compliance and Scoring

The target system did not satisfy the conditions of 39 rules! Please review rule results and consider applying remediation.

Rule results



Severity of failed rules



Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	28.125000	100.000000	28.13%

Security Drift

In addition to tracking the scan results over time as described in the previous section, Morpheus also provides detail into the change from the most recent scan to the one prior. This information is displayed in the Software tab (and Security subtab) of the detail page for the virtual machine (accessed from the associated Instance detail page or at Infrastructure > Hosts > Virtual Machines). The information surfaced by this view is listed below. If there is no change, you'll simply see a "No Drift" message.

- **Title:** The criteria for the test that has newly passed or failed
- **Severity:** The severity level for the indicated security requirement
- **Result:** The indicator for whether this test has newly passed or failed
- **New Pass:** The number of tests that have newly passed compared to the prior scan
- **New Fail:** The number of tests that have newly failed compared to the prior scan
- **Status:** An indicator of the change in security posture since the prior scan. A net gain in test failures will yield a negative status indicator while net gains in passed tests (or no change) will yield a positive status indicator

SECURITY DRIFT

TITLE	SEVERITY	RESULT	2	0	
			NEW PASS	NEW FAIL	STATUS
▶ Windows Server 2019 users must be prompted to authenticate when the system wakes from sleep (plugged in).	medium	PASS			
▶ Windows Server 2019 users must be prompted to authenticate when the system wakes from sleep (on battery).	medium	PASS			

Job Executions

Required Role Permissions [Click to Expand/Hide](#)

Access and capabilities for the **Job Executions** section is determined by the following role permissions:

Role: Feature Access: Provisioning: Job Executions

- None: Cannot access Provisioning: Job Executions section
- Read: Can access Provisioning: Job Executions section

The Job Executions tab contains execution history of completed Jobs, including any process outputs and error messages. Information included in the Job Executions list includes:

- Execution Status Icon
- **Job Name - Task Name - Result Error Message:** The title of each execution includes the Job Name, Task or Workflow name (for Task and Workflow job types), and execution result error messages (when applicable) separated by hyphens (-). The title also links to the Job Execution detail page
- **Start Date:** The date and time the Job Execution kicked off. When expanded, the start date and time of each individual Task are also shown
- **Duration:** The time taken for the Job to complete. When expanded, the time to complete each individual Task is also shown
- **User:** The user who executed or scheduled the Job

Additional details and actions are available per execution:

- Select an execution name to go to the Job Execution Detail page
- Select the  icon at the end of the row to expand the execution and view additional details, including task process output
- Select the  icon to copy process output to local clipboard
- Select the  icon at the end of an expanded to collapse additional execution details

5.3.7 Executions

Executions contains the execution status and history from Task and Operational Workflow Executions run from Library > Automation > Tasks and Library > Automation > Workflows. Executions for Tasks and Workflows are split out to their own labeled tabs.

Note: Tasks and Workflows executed from a Job or from Instance or Host Actions do not populate in the Executions Section, and can be referenced from the History tab on the target resource. All task and Workflow executions can be referenced in Operations > Activity > History

Execution results in the ui display:

NAME Name of the Task or Workflow Executed

TYPE Type of execution (Task or Workflow)

START DATE Date and time of execution

ETA/DURATION Estimate time of completion for executions in progress, or the total execution time for completed executions.

RESULTS Result status of execution (Succeeded, Failed, In-Progress or Pending)

Execution Detail (i) Click on the **i** to view process output results

Note: Job and automation executions can be expanded to show process details by clicking on the arrow icon immediately to the right of the *NAME* column.

5.3.8 Code

Note: In v5.3.2+, provisioning/deployments is moved to provisioning/code.

Provisioning > Code contains the Repositories, Deployments and Code Integrations sections.

Required Role Permissions **Click to Expand/Hide**

Access and capabilities for the **Code** section is determined by the following role permissions:

Role: Feature Access: Infrastructure: Groups

- None: Cannot access Provisioning: Code section
- Read or Full: Can access Provisioning: Code section

Role: Feature Access: Provisioning: Code Repositories

- None: Cannot access Provisioning: Code Repositories
- List Files: Can browse repo folder and file names, select branch, refresh Repositories. Cannot access/view file contents.
- Read or Full: Can browse repo folder and file names, select branch, refresh Repositories and access/view file contents.

Role: Feature Access: Provisioning: Code Deployments

- None: Cannot access Provisioning: Code Deployments.

- Read: Can view Code Deployments. Cannot create, delete or edit Code Deployments.
- Full: Can create, delete and edit Code Deployments.

Role: Feature Access: Provisioning: Code Integrations

- None: Cannot access Provisioning: Code Integrations.
- Read: Can view Code Integrations. Cannot create, delete or edit Code Integrations.
- Full: Can create, delete and edit Code Integrations

Repositories

The *Provisioning* → *Code* → *Repositories* section contains the repositories integrated with Morpheus allowing users to browse repository folders and files, view file contents from any branch, trigger a refresh, and create tasks, scripts and templates directly from the repos. This section also handles Import/Export functionality which allows users to backup Morpheus items (Tasks, Library Items, etc.) as code in a repository which can also be imported into another appliance, if desired.

- Browse integrated repositories
- View repo files
- Switch branches
- Trigger repo refreshes
- Filter by Integration, Organization or Text search
- Create Custom Views
- Create Tasks from repo files
- Create Spec Templates from repo files

Role Permissions

Access and capabilities for the **Repositories** section is determined by the following role permissions:

Role: Feature Access: Infrastructure: Groups

- None: Cannot access Provisioning: Code section
- Read or Full: Can access Provisioning: Code section

Role: Feature Access: Administration: Users

- None: Can view repo list but cannot browse repo folder and file names, select branch, refresh repositories or access/view file contents
- Read or Full: Can view repo list, browse repo folder and file names, select branch, refresh repositories. Cannot access/view file contents without Read or Full level permission on Provisioning: Code Repositories

Role: Feature Access: Provisioning: Code Repositories

- None: Cannot access Provisioning: Code Repositories

- List Files: Can browse repo folder and file names, select branch, refresh Repositories. Cannot access/view file contents
- Read or Full: Can browse repo folder and file names, select branch, refresh Repositories and access/view file contents

Role: Feature Access: Provisioning: Tasks

- None: Cannot create Tasks from repo files in repository browser
- Read or Full: Can create Tasks from repo files in repository browser

Role: Feature Access: Provisioning: Library

- None: Cannot create Spec Templates from files in repository browser
- Read or Full: Can create Spec Templates from files in repository browser

List Repositories

1. Select the **Provisioning** link in the navigation bar
2. Select the **Code** link in the sub-navigation bar
3. Users with sufficient permissions will see a list view of all integrated code repositories
4. Use the Search, Integrations or Organizations filter to filter listed repositories

Tip: Select the gear icon in the top right of the repos list view to create and save custom list views.

Refresh Repository

1. Select the **Provisioning** link in the navigation bar
2. Select the **Code** link in the sub-navigation bar
3. Select name of target repository
4. Select **ACTIONS** > **Refresh**

Browse Repositories

1. Select the **Provisioning** link in the navigation bar
2. Select the **Code** link in the sub-navigation bar
3. Select name of target repository
4. Users with sufficient permissions can browse repo folder and file names, select branches, and refresh repositories. Users can access/view file contents with Read or Full level permission on **Provisioning: Code Repositories**
5. Select target folder icon to drill into the folder

View Repository File

1. Select the Provisioning link in the navigation bar
2. Select the Code link in the sub-navigation bar
3. Select name of target repository
4. Select icon to right of target file name

Note: Users can access/view file contents only with Read or Full level permission on Provisioning: Code Repositories. File contents displayed is from last repo sync. Refresh repo to ensure current version for recent commits.

Create Task from Repository File

1. Select the Provisioning link in the navigation bar
2. Select the Code link in the sub-navigation bar
3. Select name of target repository
4. Select gear icon to right of compatible target file name
5. Select target task type from available actions
6. Complete the NEW TASK wizard to create a new Task. The TYPE, SOURCE, REPOSITORY and FILE PATH fields will be automatically configured

Note: Shell and Powershell Tasks types can be created from the code repo browser in v7.0.0. Ensure file compatibility with target Task type.

Note: Users can create tasks from Repositories only with Read or Full level permission on Library: Tasks.

Create Spec Template from Repository File

1. Select the Provisioning link in the navigation bar
2. Select the Code link in the sub-navigation bar
3. Select name of target repository
4. Select gear icon to right of target file name
5. Select target spec template type from available actions
6. Complete the NEW SPEC TEMPLATE wizard to create a new Spec Template. The TYPE, SOURCE, REPOSITORY and FILE PATH fields will be automatically configured

Note: Terraform spec template types can be created from the code repo browser in v7.0.0. Other spec template types can be created from repo files by changing the TYPE field in the NEW SPEC TEMPLATE wizard.

Note: Users can create tasks from Repositories only with Read or Full level permission on Library: Templates.

Import and Export

Onboarded Git repositories can be configured as either import or export targets for a Morpheus appliance. This means that many created Morpheus constructs, such as Tasks, Library Items, and others, can be backed up to an integrated Git repository as code. This backup can take place on an automatic schedule (syncs every four hours) or can be triggered manually after changes are made. Users can back up all supported constructs within the appliance to a single repository or use Labels to back up only selected items. Exported constructs can also be imported into target appliances. This is useful for sharing items between two Morpheus environments, such as from a development appliance to a production appliance.

Note: The use of this feature requires an integrated Git repository. Please see our [integration guide](#) for Github or other Git integrations if you've not yet integrated your code repositories. While Morpheus can onboard and work with both public and private code repositories, it is strongly recommended that you use private repositories for exporting your Morpheus constructs as code.

Supported Constructs:

- Tasks
- Workflows
- Spec Templates
- Library Items
- Forms

Note: When exporting all supported constructs Morpheus will export more than just the above types, however, only the above types are exportable as individual resources. This list may expand in the future as additional constructs become supported by future updates.

Role Permissions

Access and capabilities for the Import/Export feature set is determined by the following role permissions:

Role: Feature Access: Admin: Export/Import

- **None:** Cannot access the edit button on the Code List Page (Provisioning > Code) to set Import/Export settings or see Import/Export actions on the Code Repository Detail Page
- **Full:** Code repositories can be edited to set Import/Export configurations and Import/Export actions can be viewed and used on Code Repository Detail Pages

Import/Export Settings

Code repositories are set to allow import, export or both from the Code Repositories List Page (Provisioning > Code). Click the edit button () to the right of the selected repository to edit its import/export settings. When editing a code repository for import and export, set the following configurations:

- **ENABLED:** When marked, routine syncs will take place between Morpheus and this repository. This includes all file syncs and not just actions related to import and export
- **IMPORT/EXPORT:** Set “Auto export all” to automatically export once every four hours, “Manual export” to enable this repository for manual exports on demand, “Manual import” to enable this repository for manual imports on demand, “Import/Export” to enable both manual imports and exports on demand
- **PATH:** The path within the repository where Morpheus should import from or export to
- **EXPORT LABEL FILTER:** Enter a Label and Morpheus will export or import only constructs which include the Label into the repository. This must be a single Label, it cannot be a list of multiple Labels

Once you’ve configured the code repository, click *SAVE CHANGES*

Note: Code repositories must have at least one file in them in order to export Morpheus constructs as code. This can be as simple as a README file, they just cannot be empty.

EDIT REPOSITORY ×

NAME

DESCRIPTION

ENABLED

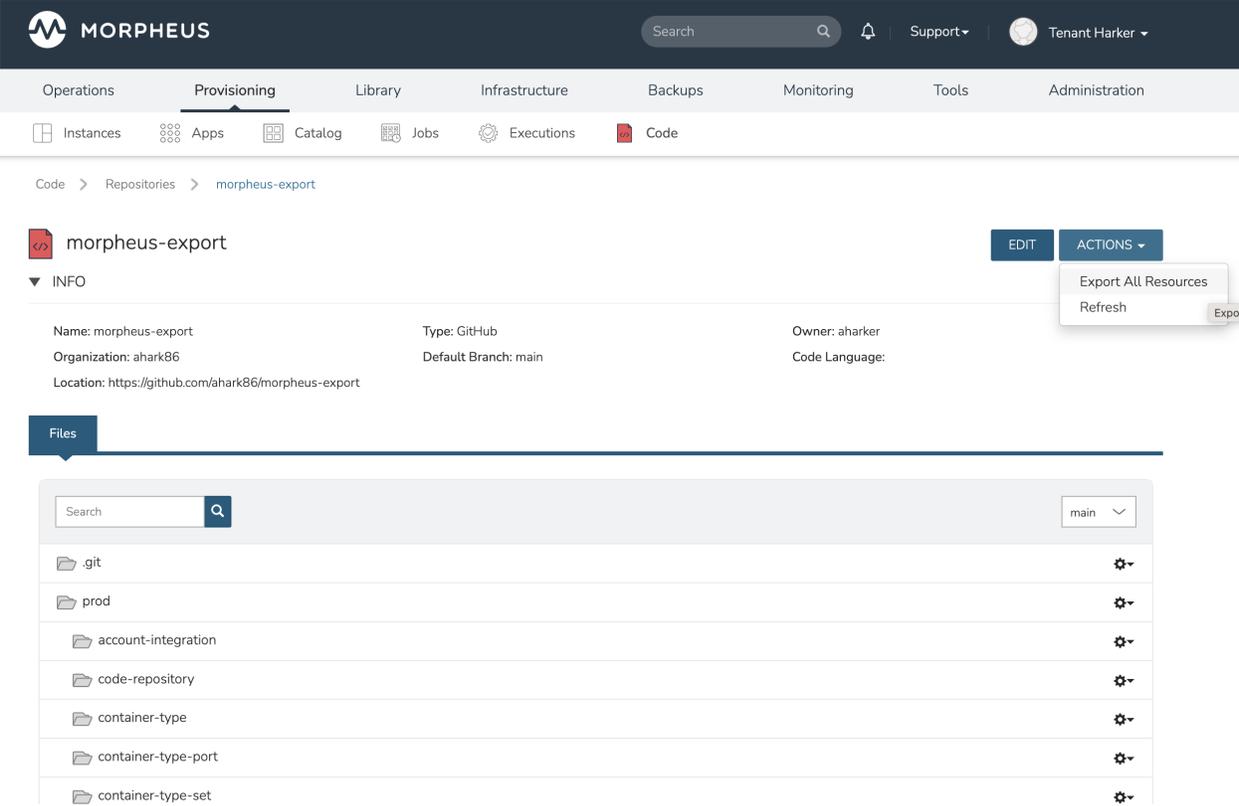
IMPORT/EXPORT ▾

PATH

EXPORT LABEL FILTER

Exporting

After a repository is configured to allow export (see previous section), it may perform periodic automatic refreshes or the user may need to refresh the repository on demand (depending on settings). To manually initiate an export, drill into the Repository Detail page, click *ACTIONS*, and click “Export All Resources”.



Any new or updated constructs will be refreshed within the repository at the path your repository is configured to export into. Bear in mind that, even if you've configured Morpheus to export only constructs categorized with a specific Label, any required dependencies would also be exported. For example, if you've labeled a Workflow to be exported, Morpheus will also export the dependency Tasks so the Workflow will be functional. A similar behavior applies for exported Library Items which may have a number of dependencies. In the screenshot below, files can be seen populating the targeted Github repository.

morpheus-export / prod /

Add file

Name	Last commit message	Last commit date
..		
account-integration	scribe [Python Task (Git-sourced).scribe, a1f5c6ba-791c-49e1-975c-0f...]	15 minutes ago
code-repository	scribe [Python Task (Git-sourced).scribe, a1f5c6ba-791c-49e1-975c-0f...]	15 minutes ago
container-type-port	scribe [aws.scribe, tr.scribe, manual csv.scribe, bafe06ca-fbf2-439d...]	8 minutes ago
container-type-set	scribe [aws.scribe, tr.scribe, manual csv.scribe, bafe06ca-fbf2-439d...]	8 minutes ago
container-type	scribe [aws.scribe, tr.scribe, manual csv.scribe, bafe06ca-fbf2-439d...]	8 minutes ago
file-content	scribe [Test Operational Workflow.scribe, 43d7a6cc-e6ef-49d6-9667-c4...]	8 minutes ago
instance-type-layout	scribe [aws.scribe, tr.scribe, manual csv.scribe, bafe06ca-fbf2-439d...]	8 minutes ago
instance-type	scribe [aws.scribe, tr.scribe, manual csv.scribe, bafe06ca-fbf2-439d...]	8 minutes ago
option-type-list	scribe [aws.scribe, tr.scribe, manual csv.scribe, bafe06ca-fbf2-439d...]	8 minutes ago
option-type	scribe [aws.scribe, tr.scribe, manual csv.scribe, bafe06ca-fbf2-439d...]	8 minutes ago
task-set-task	scribe [Test Operational Workflow.scribe, 43d7a6cc-e6ef-49d6-9667-c4...]	8 minutes ago
task-set	scribe [Test Operational Workflow.scribe, 43d7a6cc-e6ef-49d6-9667-c4...]	8 minutes ago
task	scribe [Test Operational Workflow.scribe, 43d7a6cc-e6ef-49d6-9667-c4...]	8 minutes ago

Morpheus items are exported as scribe files. These are HCL-formatted representations of the construct as code. They include static attributes representative of the attributes set on the construct itself or they may use UUIDs to refer to other constructs or integrations. It shouldn't be necessary to view or edit them unless you're curious.

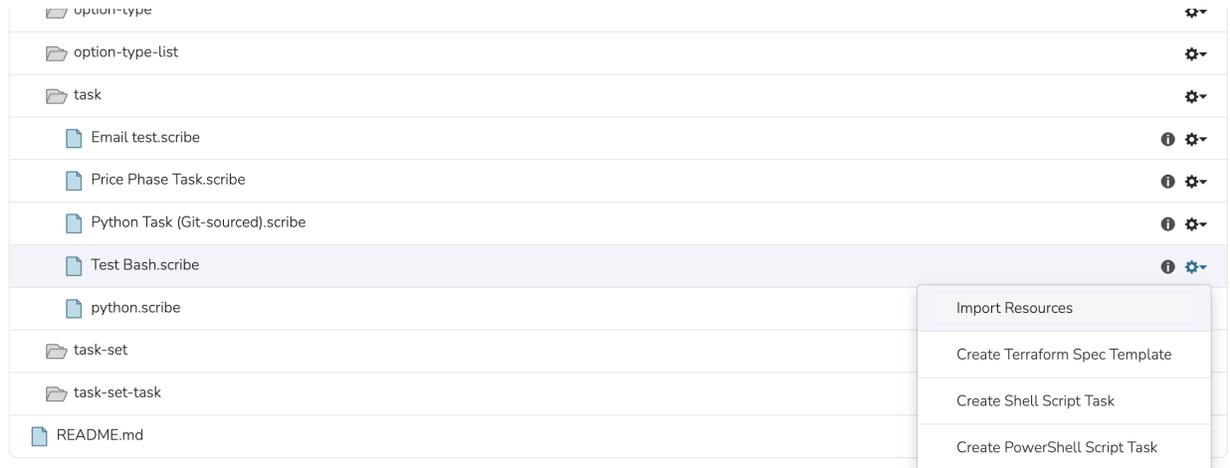
```

Code Blame Executable File · 15 lines (15 loc) · 427 Bytes
1 resource "task" "Price Phase Task" {
2   name = "Price Phase Task"
3   uuid = "77790589-fce9-41cb-8224-6ec2cb71f4c8"
4   dateCreated = "2022-10-26T15:34:01.000Z"
5   executeTarget = "local"
6   lastUpdated = "2022-10-26T15:57:36.000Z"
7   options = [
8     {
9       content = { uuid = "68043802-43a4-4564-81a1-db349314097d" }
10      optionType = { code = "groovyScript" }
11    }
12  ]
13  resultType = "value"
14  taskType = { code = "groovyTask" }
15 }

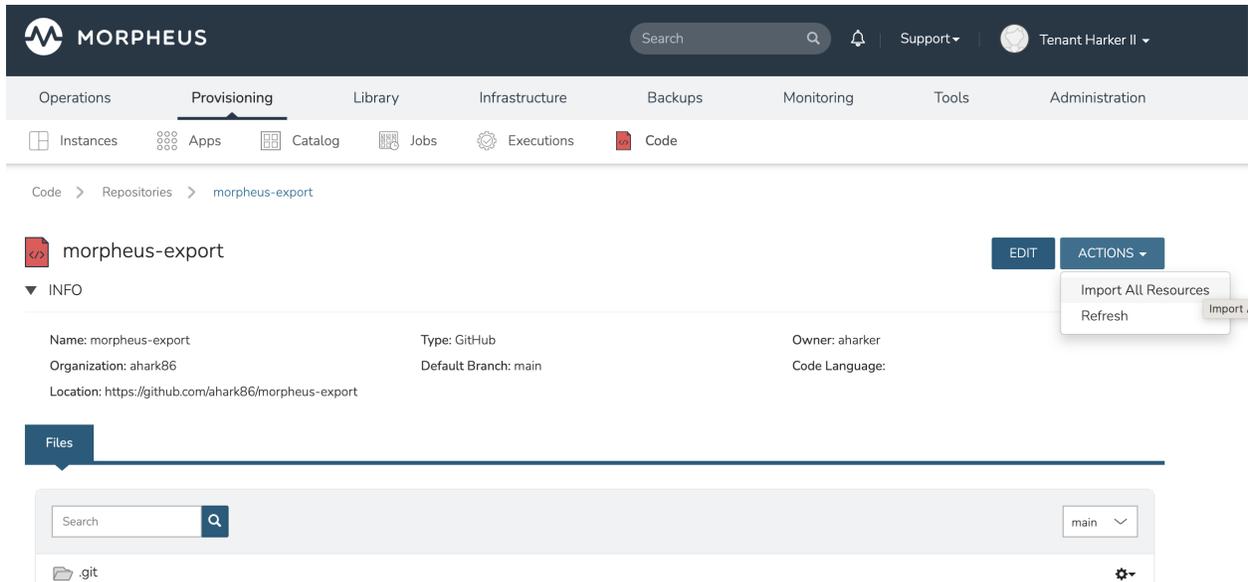
```

Importing

After Morpheus constructs have been exported as code, they can be pulled down into other appliances which have the same repositories integrated. Items can be imported ad-hoc from the file browser within a Code Repository Detail page. Click on the gear () dropdown to the right of any scribe file that may exist in your repository. For example, in the screenshot below, an individual Task is being imported into the destination appliance.



In addition to importing one-off items, code repositories may also be configured for import so that many items can be imported at once. The same configurations can be made for import as export, including a specific path within the repository to import from and whether only specific Label groups should be targeted. Triggering a larger scale import for a whole repository is done from the Code Repository Detail Page. Click the *ACTIONS* button and then click “Import All Resources”.



Once the import has been initiated, Morpheus will check to see if a new item will be created, if an existing item would be updated, if no action could be taken due to a conflict, or if no changes would result. This information is presented to the user who may then decide if the import action should go ahead.

IMPORT RESOURCES
✕

REPOSITORY

GIT REF

PATH

Resources

7	1	0	0
CREATE	UPDATE	CONFLICT	UNCHANGED

Create	account-integration ahark86 Github
Create	file-content d96b3462-0ee0-48e5-a354- XXXXXXXXXX
Create	file-content a1f5c6ba-791c-49e1-975c-0 XXXXXXXXXX
Create	file-content 20429be8-6c53-42da-aa61- XXXXXXXXXX
Create	task python
Create	task Python Task (Git-sourced)
Create	task Email test
Update	code-repository XXXXXXXXXX

IMPORT

Important: Care has been taken with Morpheus Import/Export to ensure that existing work cannot be wiped out by an import action. For example, if a Task with a name identical to a pre-existing Task would try to be imported, Morpheus will require the naming conflict be resolved before the import can take place. In fact, Morpheus Import can only create new items or overwrite items created by import. It cannot overwrite anything user-created or anything pre-seeded with the system at install. Additionally, import cannot be scheduled automatically. Users must initiate all imports.

Important: Importing resources which rely on existing integrations or sensitive values will require some additional configuration on the destination appliance. For example, importing a Task which references code in a separate repository which is not integrated on the target appliance will be imported with reference to the missing integration. This integration would need to be added before the Task could be used. Another example scenario would be a Task which references a secret value held in Morpheus Cypher. Once again, the Task would be imported and the required Cypher references would need to be made within the destination appliance before the Task could be used.

Checking Imported Items Prior to Use

As mentioned in the previous section, not every imported item will immediately be available for use. This is primarily due to the target appliance lacking an integration (for example, an Ansible Tower integration required for an imported Task) or a securely stored secret string (for example, a Task that calls a secret string stored in Morpheus Cypher). Morpheus will still import these resources but this section discusses some manual changes that may have to be made to imported items in order for them to be usable.

Protected Fields and User Credentials

Protected information and secret values are not exported. When importing any resources which rely on protected

information, the user will need to manually update the resource on the target appliance. Examples include anything authenticated using a stored credential set (from the Infrastructure > Trust > Credentials section), anything with a password field (such as HTTP Tasks or REST-based Option Lists accessing password-protected information), Chef Task Data Bag Keys, and more. Anything configured with a password or other protected field will need to be edited on the importing appliance and updated to have the protected information needed to function.

Integrations

When an imported Task requires an integration that does not exist on the appliance, an integration will be created with the same name. The integration credentials will not be established, nor will any other configuration fields outside of the name. Users will need to update the integration configuration to make it functional in order for the imported Task to function.

SSH Keys

SSH Keys are not set on imported resources, such as Tasks configured to be executed on a remote host. Users will need to add a valid key to the importing appliance and update the Task to use that SSH Key.

Ansible Tower Tasks

Following import, “Inventory” and “Job Template” fields will need to be configured on Ansible Tower Tasks.

vRealize Orchestrator Tasks

Following import, “Inventory” and “Job Template” fields will need to be configured on vRO Tasks.

Option Lists

Option lists need to be edited and saved prior to use in order to load any initial data sets that have been configured on the Option List.

Forms

Input defaults that are ID-based will require manual correction after import. For example, if you have a Group Input which defaults to `GroupName:A(ID:2)`, when imported it will still be based off an ID value of 2. It's highly likely that in the importing environment, a different Group will have an ID value of 2. A manual change to the correct ID value is required in the destination environment. Bear in mind that if you re-import a Form, you will also undo any manual changes you've made in the importing environment. Thus, if you update all defaults after importing, and then import once again, these manual default changes will be wiped out.

Additionally, when Forms are using existing Inputs, those Inputs will be exported and imported for use in the destination environment. On import, Morpheus will add the needed Inputs unless the destination environment already has an Input with the same CODE value. This could give the appearance of creating duplicate Inputs if the destination environment happens to have an Input with the same name but a different code value. Users may wish to rename Inputs in the event they end up with more than one having the same name.

Tasks

The user must ensure the presence of any dependency required by a Task. One example would be any Cypher keys being called by the Task. Add any required Cypher entries with identical keys or add compatible Cypher keys and update the Task configuration to utilize the updated Cypher mountpoints. Cypher is just one example, any other dependency required by the Task must be present. Morpheus does not parse the Task configuration to determine which dependencies are or are not present or required.

Icons and Images

Icons which are set up on imported Instance Types will need to be manually uploaded to the appliance after import. They are not brought over as part of the import/export process.

Virtual Images

When importing Node Types, the associated Virtual Image is mapped by its “Code” value. It is expected that a Virtual Image with the correct “Code” value already exists on the target appliance. If not, the Node Type will be imported

without any Virtual Image attached. The user will then need to edit the Node Type and associate it with a compatible Virtual Image in order to become usable.

Deployments

Note: In v5.3.2+, *Provisioning* → *Deployments* has been moved to *Provisioning* → *Code* → *Deployments*

The deployments section provides very useful PaaS like capabilities when it comes to deploying applications into the newly provisioned environment. These can be uploaded directly from the UI, pulled from a build server, pulled from a public or private Git repository or even via the API and the various plugins created, such as Jenkins, and Gradle to support continuous build / integration workflows.

A deployment can be considered a set of versions that relate to a particular project or application being deployed. This allows one to keep track of a history of versions and easily reuse these deployment versions across instances that may exist in different environments. An example might be to deploy a version from a deployment to a staging instance and (once approved) also deployed into production.

Role Permissions

Access and capabilities for the **Deployments** section is determined by the following role permissions:

Role: Feature Access: Provisioning: Code Deployments

- None: Cannot access Provisioning: Code Deployments.
- Read: Can view Code Deployments. Cannot create, delete or edit Code Deployments.
- Full: Can create, delete and edit Code Deployments.

Getting Started

Getting started with deployments is easy. They can vary slightly for the application stack being deployed but the simplest phase of a deployment is adding a version and adding the appropriate files to the deployment archive that are needed for the application to run. This could be a single file like a WAR file for Tomcat, or it could be hundreds of files for stacks like *Ruby on Rails*.

There are a few ways to create a deployment. The first is to use the *Provisioning* → *Code* → *Deployments* section of the application to create them. Simply add a new deployment and give it a name representing the application that is being deployed. Once a deployment is created select the deployment to view its versions (which will be empty to start). Next, it is time to add a version.

When adding a version there are several options. There are 3 types represented by the UI. These include File, Fetch, and Git respectively. A File deployment allows the user to simply drag their files into the file explorer presented by the dialog. This file explorer can take single files or entire file trees (If files exist in subfolders then only the Chrome browser is supported due to browser limitations at the time of this writing). This is also the common type that is represented when files are uploaded via the CLI, or available build tool integration plugins. Once the files have completed their upload simply save the version for use.

Git

For performing git based deploys Morpheus supports both public and private repositories. To utilize a private git repository the add version dialog will display a public keypair that can be added to the git service for authentication purposes. Currently this keypair is shared across the account and not specifically scoped to the user so it may be advisable to connect this integration to a deployment account in git. From here either a *ssh* or *https* git url can be entered along with a git branch or tag name. Once the version is saved, this repository will be copied down into the deployment archive for use.

Fetch

Fetch based deployments are pretty straightforward. Simply enter a url to a file representing the deployment. This can be a single file (in which case it will just be added to the deployment archive singularly) or it can be a zip file (which will automatically be expanded into the archive). HTTP Authentication options can also be entered if the url requires some form of basic authentication scheme for access by the appliance.

Deploying to an Instance

Now that a version has been added to a deployment, it is easy to push that deployment out to any instance provisioned within Morpheus by navigating to the specific Instance that it needs deployed to. On the Instance detail page there is a tab called *Deploy*. From here simply add a deploy. The dialog will ask firstly from which deployment the deploy is from (or allow you to create a new one on the spot), and secondly which version to deploy (also with the option to add one on the fly). The next step of the wizard will display any configuration options that might be specific to the instance type being deployed to (i.e. *CATALINA_OPTS* for Tomcat or *Java Command* for java) as well as the file explorer and deployment type selections for review (or use when creating a new version on the fly). Fill in the required items then simply hit complete. The deploy will now be asynchronously sent off to all of the virtual machines or containers within the instance in a rolling restart and the deployment status will be represented.

Tip: When deploying to an instance, the custom configuration options that were entered during the previous deployment are automatically carried forward allowing one to edit them or leave them as is.

Rolling Backwards and Forwards

Because of the tracked history of deployments kept within Morpheus, the deploy tab of instance detail makes it easy to choose a previously run deployment and jump back to it in the event of a failed deployment. The history will automatically be updated and the configuration, as well as data from the previous deployment state of the instance will be restored.

Offloading Storage

Since a full history of the backup builds are kept in Morpheus, as the appliance grows it becomes necessary to change where these are stored. On a fresh install these are stored on the local appliance in `/var/opt/morpheus` or wherever the master account may have changed the configuration to point to. It is also possible to adjust the deployment archive store by creating a *Storage Provider* tied to an S3 compatible object store, Openstack Swift object store, or any other type of mountpoint provided. This option can be adjusted in *Administration* → *Settings* → *Provisioning* once a storage provider is created within the account.

Add Deployment

1. Select the **Provisioning** link in the navigation bar.
2. Select the **Code** link in the sub-navigation bar.
3. Select the **Deployments** tab.
4. Click the **+ Add** button.
5. Enter a **Name** for the deployment and a description (optional)
6. Click the **Save Changes** button to save.

Add Version

1. Select the **Provisioning** link in the navigation bar.
2. Select the **Code** link in the sub-navigation bar.
3. Select the **Deployments** tab.
4. Click the **Name** of the deployment you would like to add a version to.
5. Click the **Add Version** button.
6. From the **Add Version Wizard** select the deployment type.
 - Files** Drag files into the file explorer presented by the dialog. This file explorer can take single files or entire file trees.
 - Fetch** Enter a url to a file representing the deployment.
 - Git** The add version dialog will display a public key pair that can be added to the git service for authentication purposes. Either a ssh or https git url can be entered along with a git branch or tag name.
9. Click the **Save Changes** button to save.

Edit Deployment

1. Select the **Provisioning** link in the navigation bar.
2. Select the **Code** link in the sub-navigation bar.
3. Select the **Deployments** tab.
4. Click the  icon on the row of the deployment you wish to edit, or click the **Name** of the deployment and then the **Edit** button from the deployment detail page.
5. Modify information as needed
6. Click the **Save Changes** button to save.

Delete Deployment

1. Select the Provisioning link in the navigation bar.
2. Select the Code link in the sub-navigation bar.
3. Select the Deployments tab.
4. Click the  icon on the row of the deployment you wish to delete, or click the Name of the deployment and then the DELETE button.

Code Integrations

The *Provisioning* → *Code* → *Integrations* section is where Code Integrations, such as Git and Github Repository Integrations, and Jenkins Build Service Integrations, can be created and managed.

Role Permissions

Access and capabilities for the **Integrations** section is determined by the following role permissions:

Role: Feature Access: Infrastructure: Groups

- None: Cannot access Provisioning: Code section
- Read or Full: Can access Provisioning: Code section

Role: Feature Access: Provisioning: Code Integrations

- None: Cannot access Provisioning: Code Integrations.
- Read: Can view Code Integrations. Cannot create, delete or edit Code Integrations.
- Full: Can create, delete and edit Code Integrations

5.4 Library

5.4.1 Labels

- Labels Video Demo

Labels are a categorization feature designed to allow easier filtering of list views in the Morpheus Library. The following library constructs can be labeled:

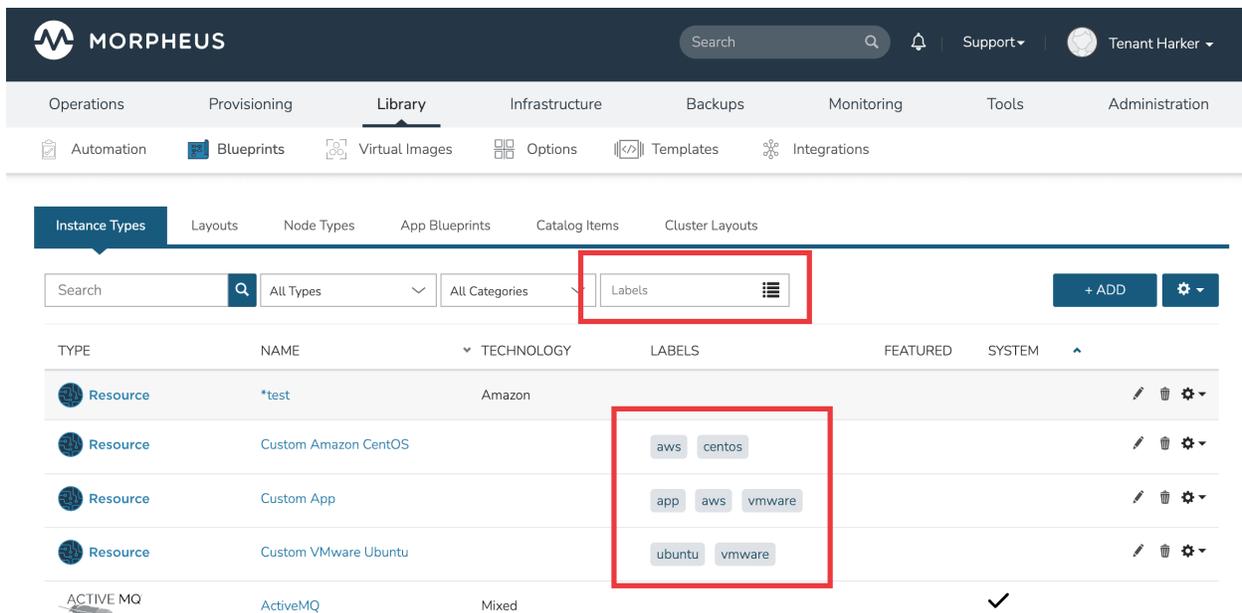
- Clouds
- Groups
- Tasks
- Workflows
- Jobs
- Instance Types

- Instances
- Layouts
- Node Types
- Servers
- Virtual Images
- Inputs
- Option Lists

Labels are visible from the list views of any constructs listed above. By default, labels are turned on in the list view but if they aren't showing, click the gear icon () and then click Labels to enable them.

The list view contains a row of filters above the list, one of which is Labels. Enter a search string to find an existing label or click the dropdown button within the field to select an existing label. This will filter the list to show only constructs which have the selected label.

Note: Any list may be filtered by any label which exists anywhere in the current Tenant. When a label is removed from a construct and no other constructs also have that label, Morpheus will remove the label from the list during its nightly sync. It is normal for a label to continue to exist in this list, even if it's not currently applied to any constructs, until the next nightly sync has taken place.



The screenshot shows the Morpheus UI interface. At the top, there is a navigation bar with the Morpheus logo, a search bar, and user information. Below this is a secondary navigation bar with tabs for Operations, Provisioning, Library (selected), Infrastructure, Backups, Monitoring, Tools, and Administration. Under the Library tab, there are sub-tabs for Automation, Blueprints, Virtual Images, Options, Templates, and Integrations.

The main content area shows the 'Instance Types' view. It has a search bar, a dropdown for 'All Types', a dropdown for 'All Categories', and a 'Labels' filter field with a dropdown arrow. To the right of the filter are '+ ADD' and a gear icon. Below the filter is a table with columns: TYPE, NAME, TECHNOLOGY, LABELS, FEATURED, and SYSTEM. The table contains several rows of instance types, each with a 'Resource' icon and a gear icon for editing. The 'LABELS' column shows comma-separated labels for each instance type.

TYPE	NAME	TECHNOLOGY	LABELS	FEATURED	SYSTEM
Resource	*test	Amazon			
Resource	Custom Amazon CentOS		aws, centos		
Resource	Custom App		app, aws, vmware		
Resource	Custom VMware Ubuntu		ubuntu, vmware		
ACTIVE MQ	ActiveMQ	Mixed			✓

Adding and Removing Labels

Labels can be created when adding or editing any of the supported constructs listed above. When adding or editing the object, enter or edit the comma-separated list of labels you wish to apply.

EDIT INSTANCE TYPE
✕

NAME

CODE
Useful shortcode for provisioning naming schemes and export reference.

DESCRIPTION

CATEGORY

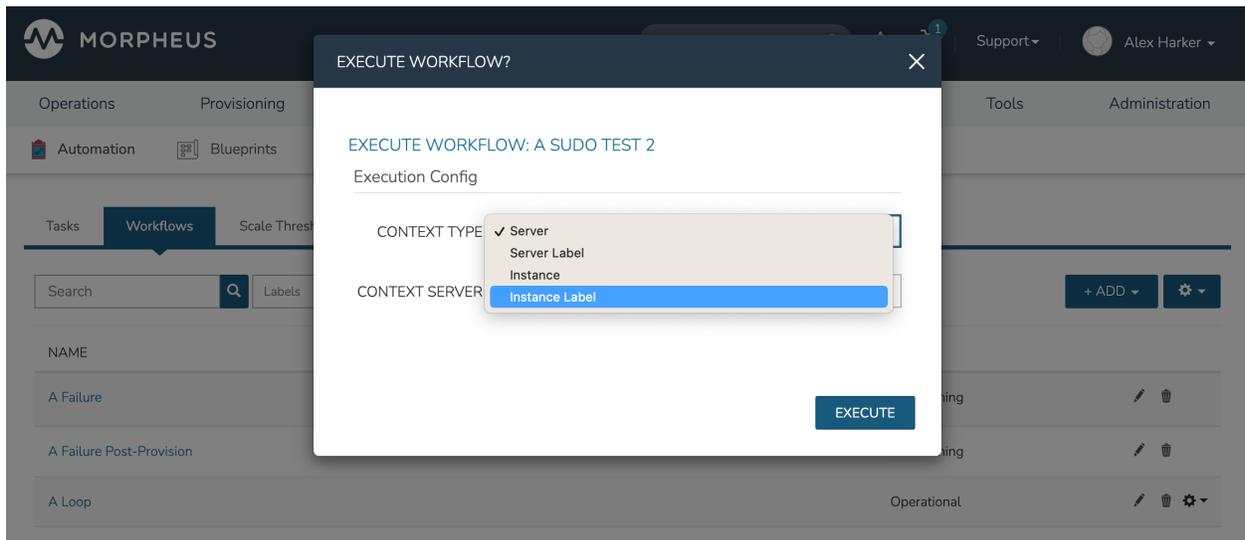
LABELS
A comma separated list of labels that can be used to group/organize items.

ICON

Running Automation Against Label Targets

The Morpheus automation constructs Jobs, Tasks, and Workflows can be run against Instance Labels or Server Labels. When creating the Job or executing the Task or Workflow, select either Server Label or Instance Label. After specifying the Label, the automation will be run against all Instances or Servers which have the indicated Label. Currently, only one Label may be selected and users cannot enter multiple Labels in the field. If a non-existent Label is entered, the automation simply will not run against any Workloads since the Label does not match any.

Note: Instance and server Labels are separate. Even if some Instances or servers have the same Label, the automation is only run against the selected construct (Instance Labels or Server Labels).



5.4.2 Automation

Library > Automation

The Automation section is composed of Tasks and Workflows. Tasks can be scripts added directly, scripts and blueprints from the Library section, recipes, playbooks, salt states, puppet agent installs, or http (api) calls. These Tasks are combined into workflows, which can be selected to run at provision time or executed on existing instances via **Actions > Run Workflow**.

Tasks

Overview

There are many Task Types available, including scripts added directly, scripts and templates from the Library section, recipes, playbooks, salt states, puppet agent installs, and http (api) calls. Tasks are primarily created for use in Workflows, but a single Task can be executed on an existing instance via **Actions > Run Task**.

Role Permissions

The User Role Permission 'Provisioning: Tasks FULL' is required to create, edit and delete tasks.

Tasks Types that can execute locally against the Morpheus Appliance have an additional Role Permission: **Tasks - Script Engines**. Script Engine Task Types will be hidden for users without **Tasks - Script Engines** role permissions.

Common Options

When creating a Task, the required and optional inputs will vary significantly by the Task type. However, there are options which are common to Tasks of all types.

Target Options

When creating a Task, users can select a target to perform the execution. Some Task types allow for any of the three execution targets listed below and some will limit the user to two or just one. The table in the next section lists the available execution targets for each Task type.

- **Resource:** A Morpheus-managed Instance or server is selected to execute the Task
- **Local:** The Task is executed by the Morpheus appliance node
- **Remote:** The user specifies a remote box which will execute the Task

Execute Options

- **Continue on Error:** When marked, Workflows containing this Task will continue and will remain in a successful state if this Task fails
- **Retryable:** When marked, this Task can be configured to be retried in the event of failure
- **Retry Count:** The maximum number of times the Task will be retried when there is a failure
- **Retry Delay:** The length of time (in seconds) Morpheus will wait to retry the Task
- **Allow Custom Config:** When marked, a text area is provided at Task execution time to allow the user to pass extra variables or specify extra configuration. See the next section for an example.

Allow Custom Config

When “Allow Custom Config” is marked on a Task, the user is shown a text area for custom configuration when the Task is executed manually from the Tasks List Page. If the Task is to be part of an Operational Workflow, mark the same box on the Workflow rather than on the Task to see the text area at execution time. This text area is inside the “Advanced Options” section, which must be expanded in order to reveal the text area. Within the text area, add a JSON map of key-value pairs which can be resolved within your automation scripts. This could be used to pass extra variables that aren’t always needed in the script or for specifying extra configuration.

Example JSON Map:

```
{"key1": "value1",  
"key2": "value2",  
"os": "linux",  
"foo": "bar"}
```

When the Task is executed, these extra variables would be resolved where called into the script such as in the following simple BASH script example:

```
echo "<%=customOptions.os%>"  
echo "<%=customOptions.foo%>"
```

The above example would result in the following output:

```
linux  
bar
```


Task Types

Table 14: Available Task Types

	Task Type	Task Description	Source Options	Execute Target Options	Configuration Requirements	Role Permissions Requirements
	Ansible	Runs an Ansible playbook. Ansible Integration required	Ansible Repo (Git)	Local, Resource	Existing Ansible Integration	Provisioning: Tasks
	Ansible Tower	Relays Ansible calls to Ansible Tower	Tower Integration	Local, Remote, Resource	Existing Ansible Tower Integration	Provisioning: Tasks
	Chef bootstrap	Executes Chef bootstrap and run list. Chef Integration required	Chef Server	Resource	Existing Chef Integration	Provisioning: Tasks
	Conditional Workflow	Allows the user to set JavaScript logic. If it resolves to <code>true</code> , Morpheus will run the Operational Workflow set as the “IF OPERATIONAL WORKFLOW” and if it resolves to <code>false</code> , Morpheus will run the “ELSE OPERATIONAL WORKFLOW”	N/A (JavaScript logic must be locally sourced, Tasks housed within the associated Workflows may have different sourcing options depending on their types.)	Local	Existing Operational Workflows	Provisioning: Tasks
	Email	Send an email from a Workflow	Task Content	Local	SMTP Configured	Provisioning: Tasks
	Groovy script	Executes Groovy Script locally (on Morpheus app node)	Local, Repository, Url	Local	None	Provisioning: Tasks, Tasks - Script Engines
	HTTP	Executes REST call for targeting external API's.	Local	Local	None	Provisioning: Tasks
	Javascript	Executes Javascript locally (on Morpheus app node)	Local	Local	None	Provisioning: Tasks - Script Engines
5.4. Library						Tasks - Script Engines

Task Configuration

- **Ansible Playbook**



- **NAME:** Name of the Task
- **CODE:** Unique code name for API, CLI, and variable references
- **ANSIBLE REPO:** Select existing Ansible Integration
- **GIT REF:** Specify tag or branch (Option, blank assumes default)
- **PLAYBOOK:** Name of playbook to execute, both `playbook` and `playbook.yml` format supported
- **TAGS:** Enter comma separated tags to filter executed tasks by (ie `--tags`)
- **SKIP TAGS:** Enter comma separated tags to run the playbook without matching tagged tasks (ie `--skip-tags`)

Important: Using different Git Refs for multiple Ansible Tasks in same Workflow is not supported. Git Refs can vary between Workflows, but Tasks in each Workflow must use the same Git Ref.

- **Ansible Tower Job**



- **NAME:** Name of the Task
- **CODE:** Unique code name for API, CLI, and variable references
- **TOWER INTEGRATION:** Select an existing Ansible Tower integration
- **INVENTORY:** Select an existing Inventory, when bootstrapping an Instance, Morpheus will add the Instance to the Inventory
- **GROUP:** Enter a group name, when bootstrapping an Instance, Morpheus will add the Instance to the Group if it exists. If it does not exist, Morpheus will create the Group
- **JOB TEMPLATE:** Select an existing job template to associate with the Task
- **SCM OVERRIDE:** If needed, specify an SCM branch other than that specified on the template
- **EXECUTE MODE:** Select Limit to Instance (template is executed only on Instance provisioned), Limit to Group (template is executed on all hosts in the Group), Run for all (template is executed on all hosts in the Inventory), or Skip Execution (to skip execution of the template on the Instance provisioned)

- **Chef bootstrap**



- **NAME:** Name of the Task
- **CODE:** Unique code name for API, CLI, and variable references
- **CHEF SERVER:** Select existing Chef integration
- **ENVIRONMENT:** Populate Chef environment, or leave as `_default`
- **RUN LIST:** Enter Run List, eg `role[web]`

- **DATA BAG KEY:** Enter data bag key (will be masked upon save)
- **DATA BAG KEY PATH:** Enter data bag key path, eg `/etc/chef/databag_secret`
- **NODE NAME:** Defaults to Instance name, configurable
- **NODE ATTRIBUTES:** Specify attributes inside the `{}`

- **Conditional Workflow**

- **NAME:** Name of the Task
- **CODE:** Unique code name for API, CLI, and variable references
- **LABELS:** A comma separated list of Labels for organizational purposes. See elsewhere in Morpheus docs for additional details on utilizing Labels
- **CONDITIONAL (JS):** JavaScript logic which determines the Operational Workflow which is ultimately run. If it resolves to `true`, the “If” Workflow is run and if it resolves to `false` the “Else” Workflow is run
- **IF OPERATIONAL WORKFLOW:** Set the Operational Workflow which should be run if the JavaScript conditional resolves to `true`
- **ELSE OPERATIONAL WORKFLOW:** Set the Operational Workflow which should be run if the JavaScript conditional resolves to `false`

- **Groovy script**



- **NAME:** Name of the Task
- **CODE:** Unique code name for API, CLI, and variable references
- **RESULT TYPE:** Single Value, Key/Value Pairs, or JSON
- **CONTENT:** Contents of the Groovy script if not sourcing it from a repository

- **Email**



Email

- **NAME:** Name of the Task
- **CODE:** Unique code name for API, CLI, and variable references
- **SOURCE:** Choose local to draft or paste the email directly into the Task. Choose Repository or URL to bring in a template from a Git repository or another outside source
- **EMAIL ADDRESS:** Email addresses can be entered literally or Morpheus automation variables can be injected, such as `<%=instance.createdByEmail%>`
- **SUBJECT:** The subject line of the email, Morpheus automation variables can be injected into the subject field
- **CONTENT:** The body of the email is HTML. Morpheus automation variables can be injected into the email body when needed
- **SKIP WRAPPED EMAIL TEMPLATE:** The Morpheus-styled email template is ignored and only HTML in the Content field is used

Tip: To whitelabel email sent from Tasks, select SKIP WRAPPED EMAIL TEMPLATE and use an HTML template with your own CSS styling

- **HTTP (API)**



- **NAME:** Name of the Task
- **CODE:** Unique code name for API, CLI, and variable references
- **RESULT TYPE:** Single Value, Key/Value Pairs, or JSON
- **URL:** An HTTP or HTTPS URL as the HTTP Task target
- **HTTP METHOD:** GET (default), POST, PUT, PATCH, HEAD, or DELETE
- **AUTH USER:** Username for username/password authentication
- **PASSWORD:** Password for username/password authentication
- **BODY:** Request Body
- **HTTP HEADERS:** Enter requests headers, examples below:

Authorization	Bearer <i>token</i>
Content-Type	application/json

- **IGNORE SSL ERRORS:** Mark when making REST calls to systems without a trusted SSL certificate

- **Javascript**



- **NAME:** Name of the Task
- **CODE:** Unique code name for API, CLI, and variable references
- **RESULT TYPE:** Single Value, Key/Value Pairs, or JSON
- **SCRIPT:** Javascript contents to execute

- **jRuby Script**



- **NAME:** Name of the Task
- **CODE:** Unique code name for API, CLI, and variable references
- **RESULT TYPE:** Single Value, Key/Value Pairs, or JSON
- **CONTENT:** Contents of the jRuby script is entered here if it's not being called in from an outside source

- **Library Script**



- **NAME:** Name of the Task
- **CODE:** Unique code name for API, CLI, and variable references

- **RESULT TYPE:** Single Value, Key/Value Pairs, or JSON
- **SCRIPT:** Search for an existing script in the typeahead field

- **Library Template**



- **NAME:** Name of the Task
- **CODE:** Unique code name for API, CLI, and variable references
- **TEMPLATE:** Search for an existing template in the typeahead field

- **Nested Workflow**

- **NAME:** Name of the Task
- **CODE:** Unique code name for API, CLI, and variable references
- **OPERATIONAL WORKFLOW:** The Workflow to be embedded as a Task for reference inside other Workflows

- **Powershell Script**



- **NAME:** Name of the Task
- **CODE:** Unique code name for API, CLI, and variable references
- **RESULT TYPE:** Single Value, Key/Value Pairs, or JSON
- **ELEVATED SHELL:** Run script with administrator privileges
- **IP ADDRESS:** IP address of the PowerShell Task target
- **PORT:** SSH port for PowerShell Task target (5985 default)
- **USERNAME:** Username for PowerShell Task target
- **PASSWORD:** Password for PowerShell Task target
- **Content:** Enter script to execute if not calling the script in from an outside source

Note: Setting the execution target to local requires Powershell to be installed on the Morpheus appliance box(es). [Microsoft Documentation](#) contains installation instructions for all major Linux distributions and versions.

- **Puppet Agent Install**



- **NAME:** Name of the Task
- **CODE:** Unique code name for API, CLI, and variable references
- **PUPPET MASTER:** Select Puppet Master from an existing Puppet integration
- **PUPPET NODE NAME:** Enter Puppet node name. Variables supported eg. `<%= instance.name %>`
- **PUPPET ENVIRONMENT:** Enter Puppet environment, eg. `production`

- **Python Script**



Important: Beginning with Morpheus version 4.2.1, Python Tasks use virtual environments. For this reason, `virtualenv` must be installed on your appliances in order to work with Python Tasks. See the information below for more detailed steps to install `virtualenv` on your Morpheus appliance node(s).

- **NAME:** Name of the Task
- **CODE:** Unique code name for API, CLI, and variable references
- **RESULT TYPE:** Single Value, Key/Value Pairs, or JSON
- **CONTENT:** Python script to execute is entered here if not pulled in from an outside repository
- **COMMAND ARGUMENTS:** Optional arguments passed into the Python script. Variables supported eg. `<%= instance.name %>`
- **ADDITIONAL PACKAGES:** Additional packages to be installed after `requirements.txt` (if detected). Expected format for additional packages: `'packageName==x.x.x packageName2==x.x.x'`, the version must be specified
- **PYTHON BINARY:** Optional binary to override the default Python binary

Enterprise Proxy Considerations

Additional considerations must be made in enterprise proxy environments where Python Tasks are run with additional package download requirements. These additional packages are downloaded using `pip` and may not obey global Morpheus proxy rules. To deal with this, create or edit the `pip` configuration file at `/etc/pip.conf`. Your configuration should include something like the following:

```
[global]
proxy = http://some-proxy-ip.com:8087
```

For more information, review the `Pip` documentation on using proxy servers [here](#).

CentOS 7 / Python 2.7 (RHEL system Python)

With a fresh install of Morpheus on a default build of CentOS 7, Python Tasks will not function due to the missing requirement of `virtualenv`.

If you attempt to run a python task, you will get an error similar to the following:

```
Task Execution Failed on Attempt 1
sudo: /tmp/py-8ae51ebf-749c-4354-b6e4-11ce541afad5/bin/python: command not found
```

In order to run Morpheus Python Tasks in CentOS 7, install `virtualenv`: `yum install python-virtualenv`

If you require `python3`, you can specify the binary to be used while building the virtual environment. In a default install, do the following: `yum install python3`. Then, in your Morpheus Python Task, specify the binary in the `PYTHON BINARY` field as `"/bin/python3"`. This will build a virtual environment in `/tmp` using the `python3` binary, which is equivalent to making a virtual environment like so: `virtualenv ~/venv -p /bin/python3`.

If you wish to install additional Python packages into the virtual environment, put them in `pip` format and space-separated into the `ADDITIONAL PACKAGES` field on the Python Task. Use the help text below the field to ensure correct formatting.

CentOS 8 and Python

In CentOS 8, Python is not installed by default. There is a `platform-python` but that should not be used for anything in userland. The error message with a default install of CentOS 8 will be similar to this:

```
Task Execution Failed on Attempt 1
sudo: /tmp/py-cffc9a8f-c40d-451d-956e-d6e9185ade33/bin/python: command not found
```

The default `virtualenv` for CentOS 8 is the `python3` variety, for Morpheus to use Python Tasks, do the following: `yum install python3-virtualenv`

If Python2 is required, do the following: `yum install python2` and specify `/bin/python2` as the `PYTHON BINARY` in your Morpheus Task.

This will build a `virtualenv` in `/tmp` using the `python2` binary, which is equivalent to making a `virtualenv` like so: `virtualenv ~/venv -p /bin/python2`

If you wish to install additional Python packages into the virtual environment, put them in `pip` format and space-separated into the `ADDITIONAL PACKAGES` field on the Python Task. Use the help text below the field to ensure correct formatting.

- **Restart**



- **NAME:** Name of the Task
- **CODE:** Unique code name for API, CLI, and variable references

- **Shell Script**



- **NAME:** Name of the Task
- **CODE:** Unique code name for API, CLI, and variable references
- **RESULT TYPE:** Single Value, Key/Value Pairs, or JSON
- **SUDO:** Mark the box to run the script as `sudo`
- **CONTENT:** Script to execute is entered here if not pulled in from an outside repository

Tip: When the `EXECUTE TARGET` option is set to “Local” (in other words, the Task is run on the appliance itself), two additional fields are revealed: `GIT REPO` and `GIT REF`. Use `GIT REPO` to set the `PWD` shell variable (identifies the current working directory) to the locally cached repository (ex. `/var/opt/morpheus-node/morpheus-local/repo/git/76fecffdf1fe96516e90becdab9de`) and `GIT REF` to identify the Git branch the Task should be run from if the default (typically `main` or `master`) shouldn’t be used. If these options are not set, the working folder will be `/opt/morpheus/lib/tomcat/temp` which would not allow scripts to reference file paths relative to the repository (if needed).

- **vRealize Orchestrator Workflow**



- **NAME:** Name of the Task
- **CODE:** Unique code name for API, CLI, and variable references
- **RESULT TYPE:** Single Value, Key/Value Pairs, or JSON
- **vRO INTEGRATION:** Select an existing vRO integration
- **WORKFLOW:** Select a vRO workflow from the list synced from the selected integration
- **PARAMETER BODY (JSON):**

- **Write Attributes**



- **NAME:** Name of the Task
- **CODE:** Unique code name for API, CLI, and variable references
- **ATTRIBUTES:** A JSON map of arbitrary values to write to the attributes property of the target resource

Tip: This is often useful for storing values from one phase of a Provisioning Workflow for access in another phase. See the video demo below for a complete example.

There are a number of ways that a JSON payload can be statically drafted within a Write Attributes Task or called into the Task as a result from a prior Task. Consider the following examples:

To pass in a static JSON map with static values, use the format shown below.

```
{
  "my_key1": "my_value1",
  "my_key2": "my_value2"
}
```

To pass in a static JSON map with dynamic values seeded from prior Task results, ensure the RESULT TYPE value of one or more of the prior Tasks in the Workflow phase is set to “Single Value” and refer to the values within the JSON map as shown in the next example. Note that “taskCode1” and “taskCode2” refer to the CODE field value for the Task whose output you wish to reference.

```
{
  "my_key1": "<%=results.taskCode1%>",
  "my_key2": "<%=results.taskCode2%>"
}
```

To pass in a dynamic JSON map returned from a prior Task, format your Write Attributes Task as shown in the next example. Ensure that the RESULT TYPE value for the Task returning a JSON map is set to “JSON”. Note that “taskCode” in the example refers to the CODE field value for the Task being referenced. In order for the JSON map to be set correctly and able to be referenced from future Tasks, you must set the “instances” key and call the `encodeAsJSON()` Groovy method as shown in the example.

```
{  
  "instances": <%=results.taskCode?.encodeAsJSON()%>  
}
```

Task Management

Adding Tasks

1. Select Automation from within the Library menu
2. On the Tasks tab, click the *Add* button
3. From the New Task Wizard input a name for the task
4. Select the type of task from from the type dropdown
5. Input the appropriate configuration details. These will vary significantly based on the selected Task type. More details on each Task type are contained in the preceding sections.
6. Once done, click *SAVE CHANGES*

Tip: When writing a Task config, it's often necessary to reference Morpheus variables which pertain to the specific Instance the Task is being run against. Morpheus includes a pop-out column along the right side of the Add/Edit Task modal which lists available variables. Click and drag the relevant variable into the config area and Morpheus will automatically fill in the variable call formatted for the currently chosen Task type. See the screenshot below.

NEW TASK
✕

NAME

CODE

LABELS
A comma separated list of labels that can be used to group/organize items.

TYPE

RESULT TYPE

SOURCE

CONTENT

```
morpheus['cypher']['read']('')
```

COMMAND ARGUMENTS

ADDITIONAL PACKAGES

SCRIPT VARIABLES

instance

morpheus

- getApiAccessToken()
- formatMemory(size, unit)
- applianceUrl
- applianceHost
- appliancePort
- applianceScheme
- applianceSsl
- morpheusHome
- morpheusUser
- publicKey
- privateKey
- cloudConfig
- cypher
 - read(key)
 - write(key, value)
 - delete(key)
 - readUuid(key)
 - readEncryptionKey(key)
 - readPassword(key)
- archives
 - link(bucketName, filePath)
- account
 - accountId
 - accountType
- apps - []
 - appContext
 - description
 - id
 - name
 - cloudConfig

Editing Tasks

1. Select Automation from within the Library menu
2. Click the pencil icon () on the row of the task you wish to edit
3. Modify Task as needed
4. Once done, click *SAVE CHANGES*

Deleting Tasks

1. Select Automation from within the Library menu
2. Click the trash icon () on the row of the Task you wish to delete

Task Results

Overview

Using Task results, the output from any preceding Tasks within the same Workflow phase is available to be called into additional Tasks. The results are stored on the `results` variable. Since results are available to all Tasks, we can use results from any or all prior Tasks so long as they are executed within the same provision phase.

In script type tasks, if a `RESULT TYPE` is set, Morpheus will store the output on the `results` variable. It's important to understand that the result type indicates the format of the Task output Morpheus should expect. Morpheus will parse that output into a Groovy map which can be retrieved and further parsed by resolving the `results` variable. If the

RESULT TYPE is incorrectly set, Morpheus may not be able to store the Task results correctly. Jump to the section on [Script Config Examples](#) to see how script results are processed in various example cases.

Results Types

- **Single Value** Entire task output is stored in `<%=results.taskCode%>` or `<%=results["Task Name"]%>` variable.
- **Key/Value pairs** Expects `key=value,key=value` output. Entire task output is available with `<%=results.taskCode%>` or `<%=results["Task Name"]%>` variable (output inside []). Individual Values are available with `<%=results.taskCode.key%>` variables.
- **JSON** Expects `key:value,key:value` json formatted output. Entire task output is available with `<%=results.taskCode%>` or `<%=results["Task Name"]%>` variable (output inside []). Individual Values are available with `<%=results.taskCode.key%>` variables.

Important: The entire output of a script is treated as results, not just the last line. Ensure formatting is correct for the appropriate result type. For example, if Results Type is `json` and the output is not fully json compatible, the result would not return properly.

Important: Task results are not supported for Library Script task types

Script Config Examples

Single Value using Task Code

Source Task Config

- NAME: Var Code (single)
- CODE: singleExample
- RESULT TYPE: Single Value
- SCRIPT: echo "string value"

Source Task Output: string value

Results Task Config (using task code in variable)

- NAME: N/A
- CODE: N/A
- RESULT TYPE: N/A
- SCRIPT: echo "single: <%=results.singleExample%>"

Results Task Output: single: string value

Single Value using Task Name

Source Task Config

NAME Var Code

CODE none

RESULT TYPE Single Value

SCRIPT echo "string value"

Source Task Output string value

Results Task using task name in variable

Results Task Script echo "task name: <%=results["Var Code"]%>"

Results Task Output task name: test value

Key/Value Pairs

Source Task Config

NAME Var Code (keyval)

CODE keyvalExample

RESULT TYPE Key/Value pairs

SCRIPT echo "flash=bang,ping=pong"

Source Task Output flash=bang,ping=pong

Results Task for all results

Results Task Script echo "keyval: <%=results.keyvalExample%>"

Results Task Output keyval: [flash:bang, ping:pong]

Results Task for a single value)

Results Task Script echo "keyval value: <%=results.keyvalExample.flash%>"

Results Task Output keyval value: bang

JSON

Source Task Config

NAME Var Code (json)

CODE jsonExample

RESULT TYPE JSON

SCRIPT echo "{\"ping\":\"pong\",\"flash\":\"bang\"}"

Source Task Output {"ping":"pong","flash":"bang"}

Results Task for all results

Results Task Script echo "json: <%=results.jsonExample%>"

Results Task Output json: [ping:pong, flash:bang]

Results Task for a single value

Results Task Script echo "json value: <%=results.jsonExample.ping%>"

Results Task Output json value: pong

Multiple Task Results

Results Task Script

```
echo "single: <%=results.singleExample%>"
echo "task name: <%=results["Var Code"]%>"
echo "keyval: <%=results.keyvalExample%>"
echo "keyval value: <%=results.keyval.flash%>"
echo "json: <%=results.jsonExample%>"
echo "json value: <%=results.jsonExample.ping%>"
```

Results Task Output

```
single: string value
task name: string value
keyval: [flash:bang, ping:pong]
keyval value: bang
json: [ping:pong, flash:bang]
json value: pong
```

Workflow Config

Add one or multiple tasks with Results Type configured to a workflow, and the results will be available to all tasks in the same phase of the workflow via the `<%=results.variables%>` during the workflow execution.

- Task Results are only available to tasks in the same workflow phase
- Task Results are only available during workflow execution

Workflows

Workflows are groups of Tasks, which are described in detail in the preceding section. Operational Workflows can be run on-demand against an existing Instance or server from the Actions menu on the Instance or server detail page. Additionally, they can be scheduled to run on a recurring basis through Morpheus Jobs (Provisioning > Jobs).

Provisioning Workflows are associated with Instances at provision time (in the Automation tab of the Add Instance wizard) or after provisioning through the Actions menu on the Instance detail page. Provisioning Workflows assign Tasks to various stages of the Instance lifecycle, such as Provision, Post Provision, and Teardown. When the Instance reaches a given stage, the appropriate Tasks are run. Task results and output can be viewed from the History tab of the Instance or server detail page.

Provisioning Workflow Execution Phases

Phase	Description	Usage Example	Notes
Configuration	Tasks are run prior to initial calls to the specified cloud API to initiate provisioning	Call to an external platform to dynamically generate a hostname prior to kicking off provisioning or dynamically altering configuration of a Catalog Item prior to provisioning	
Price	Price Phase Tasks are only invoked when the Workflow is tied to a Layout. Like the Configuration Phase, these Tasks are run prior to any calls made to the target Cloud API and allow pricing data to be overridden for the Workload being provisioned. A “spec” variable containing Instance config is passed into the Task and a specific return payload is expected in order to work properly. Any other pricing (such as on the Service Plan) is overridden. See the section below for a detailed example of this Phase being used.	An MSP customer calling out to a custom pricing API to deliver Instance pricing to their own customers.	See the section below on Price Phase implementation for a detailed setup example.
Pre Provision	For VMs, Tasks are run after the VM is running and prior to any Tasks in the Provision phase. For containers, Tasks in this phase are run on the Docker host and prior to <code>docker run</code>	Prepare a Docker host to run containers	Pre Provision can be used for a Blueprint so it is added before a script which is set at the Provision phase executes. Pre Provision for scripts is mainly for Docker as you can execute on the host before the container is running.
Provision	Like pre-provision, Tasks for VMs are run after the VM is running. For containers, these Tasks are run on the containers once they are running on the host. For many users, this is the most commonly-used phase.	Join the server to a domain	Tasks included with in the Provision phase are considered to be vital to the health of the Instance. If a Task in the Provision phase fails, the Workflow will fail and the Instance provisioning will also fail. Tasks not considered to be vital to the existence of the Instance should go in the Post Provision phase where their failure will not constitute failure of the Instance.
Post Provision	Tasks are run after the entire provisioning process has completed	Disable UAC or Windows Firewall on a Windows box or join Active Directory	When adding a node to an Instance, Tasks in this phase will be run on all nodes in the Instance after the new node is provisioned. This is because Post Provision operations may need to affect all nodes, such as when joining a new node to a cluster. Tasks in Pre Provision and Provision phases would only be run on the new node in this scenario.
5.4. Library			
Start Service	Tasks in this phase are intended to start the service associated with the Instance type.	Include a script to start the service associated	Start services is manually run from the Instance detail page and is designed to refer to the service the In-

Add Workflow

1. Select the Library link in the navigation bar
2. Select Automation from the sub-navigation menu
3. Click the Workflows tab to show the Workflows tab panel
4. Click the + *Add* dropdown and select a Workflow type (Operational or Provisioning, see the section above for more on Workflow type differences)
5. From the New Workflow Wizard input a name for the workflow
6. Optionally input a description and a target platform
7. Add Tasks and Inputs using the typeahead fields, Tasks must be added to the appropriate phases for Provisioning Workflows
8. If multiple tasks are added to the same execution phase, their execution order can be changed by selecting the grip icon and dragging the task to the desired execution order
9. For multi-Tenant environments, select Public or Private visibility for the Workflow
10. For Operational Workflows, optionally mark “Allow Custom Config” from the Advanced Options section if needed. See the next section for more on this selection
11. Click the *SAVE CHANGES* button to save

Note: When setting Workflow visibility to Public in a multi-Tenant environment, Tenants will be able to see the Workflow and also execute it directly from the Workflows list (if it’s an Operational Workflow). They will not be able to edit or delete the Workflow.

Allow Custom Config

When “Allow Custom Config” is marked on Operational Workflows, the user is shown a text area for custom configuration at execution time. This text area is inside the “Advanced Options” section, which must be expanded in order to reveal the text area. Within the text area, add a JSON map of key-value pairs which can be resolved within your automation scripts. This could be used to pass extra variables that aren’t always needed in the script or for specifying extra configuration.

Example JSON Map:

```
{ "key1": "value1",  
  "key2": "value2",  
  "os": "linux",  
  "foo": "bar" }
```

When the Workflow is executed, these extra variables would be resolved where called into the script such as in the following simple BASH script example:

```
echo "<%=customOptions.os%>"  
echo "<%=customOptions.foo%>"
```

The above example would result in the following output:

```
linux  
bar
```

Retrying Workflows

When a Workflow fails, Morpheus allows users to retry from the failed Task. Access the Workflow execution from the executions list page (Provisioning > Executions), from the Executions tab of the Workflow detail page (Library > Automation > Workflows > Selected Workflow), or from the History tab on the Instance detail page (Provisioning > Instances > Selected Instance). From the execution, select the Retry button which looks like a clockwise circular arrow and is highlighted in the screenshot below. This can be very useful as it allows you to resume what could potentially be a very long running Workflow from a point of failure without needing to start from the beginning. Similarly, if a provisioned Instance is in a failed state due to a failure in an attached Workflow (such as a failed Task in the Provision phase of an attached Provisioning Workflow), the user can opt to resume the Tasks from the failure point after making a correction and restore the Instance to a successfully-provisioned state.

- [Retry Workflow Tasks Video Demo](#)

The screenshot displays the Morpheus web interface. At the top, there is a navigation bar with the Morpheus logo, a search bar, and user information for Alex Harker. Below this is a secondary navigation bar with tabs for Operations, Provisioning, Library, Infrastructure, Backups, Monitoring, Tools, and Administration. The main content area shows a list of workflow tasks under the 'Provisioning' section. The tasks are as follows:

- Running handlers:
Running handlers complete
Chef Client finished, 57/69 resources updated in 10 seconds
morpheus-node Reconfigured!
- Retry Test - Write Pre-Provision Templates (Status: Success)
- Retry Test - Write Provision Templates (Status: Success)
- Retry Test - Run Task: *Retry Test 1 in *Retry Test Provisioning - Provision (Status: Success)
- Task 1
- Retry Test - Run Task: *Retry Test 2 in *Retry Test Provisioning - Provision (Status: Failure). The 'Retry' button (a circular arrow icon) is highlighted with a red box.
- Task 2
- Retry Test - Run Task: *Retry Test 3 in *Retry Test Provisioning - Provision (Status: Pending)
- Retry Test - Complete Provisioning (Status: Pending)

Cancelling Workflow Tasks

When a Workflow is running, Morpheus offers the capability of cancelling a Task and stopping any subsequent Tasks from starting. When viewed later in History or Executions, this leaves the Task and Workflow in a cancelled state. This is useful if you have a very long-running Task that you know will fail and wish to cancel or if you want to prevent a “retryable” Task from running again.

To cancel a Workflow, open the execution. Within the running Task will be a cancel button, click the button to interrupt the Workflow. Once cancelled, see that the Workflow and Task are now considered to be in a cancelled state which is shown in the UI. At this point the cancel button becomes a retry button and the Task could be resumed if desired.

Note: Cancelling a Task doesn’t actually interrupt the process already running on the workloads themselves. It simply interrupts the Workflow and stops it from continuing. Behind the scenes, Morpheus allows the running process to complete or time out rather than risk corrupting data with a non-graceful interrupt.

The screenshot displays the Morpheus web interface. At the top, there is a navigation bar with the Morpheus logo, a search bar, and user information for 'Tenant Harker'. Below this is a secondary navigation bar with tabs for Operations, Provisioning, Library (selected), Infrastructure, Backups, Monitoring, Tools, and Administration. Under the Library tab, there are sub-tabs for Automation, Blueprints, Virtual Images, Options, Templates, and Integrations. The main content area shows a timeline of executions from 05:25 AM to 12:34 PM. A search bar is present with the text 'All'. The execution list shows a workflow task 'Local - Run Workflow: Workflow kill test' with a red 'X' icon, indicating it is cancelled. Below this, there are three task entries: 1) 'Local - Run Task: Task kill test in Workflow kill test - Operation' with a green checkmark, indicating success. 2) 'Local - Run Task: Task kill test in Workflow kill test - Operation' with a red 'X' icon, indicating failure, and a 'Retry' button. 3) 'Local - Complete Job Execution'.

Price Phase Task Utilization

- Price Phase Tasks Video Demo

Price Phase Tasks allow computed pricing for any workload in any Cloud (even public Clouds) to be overridden based on custom logic designed by the user. The variable “spec” is fed into the Task which represents the Instance configuration. The Task can be designed to use the Instance config data and compute an appropriate price for the Instance. Morpheus expects a return payload in the format below for the price override to work correctly. If used, pricing computed via Task replaces any other costing data which would have been applied to the workload (such as pricing based on the Service Plan). The user will see price estimates based on the Price Phase Task in the Instance provisioning wizard where the Service Plan pricing would otherwise be shown. Additionally, since Workflows which invoke Price Phase Tasks are tied to the Layout, the user can see different pricing depending on which Instance Type Layout is selected.

Note: Price Phase Tasks are only invoked if the Workflow is tied to a Layout.

The return payload should be a JSON array of “priceData” objects. priceData objects should contain values for each of the keys in the table below:

Key	Description	Data Type	Possible Values
in-cur-Charges	Indicates the Instance state when this charge should be applied	String	Running: Charge is incurred while the workload is in a running state, Stopped: Charge is incurred while the workload is in a stopped or shutdown state, Always: This charge is always applied. Some charges may apply simultaneously, for example, “Always” and “Running” states will apply while the workload is running.
currency	Indicates the currency in which the charge will be applied	String	Enter any three-letter currency code which Morpheus supports for its pricing, such as “USD”, “CAD”, or “GBP”
unit	Indicates the time interval at which the charge is applied	String	Enter “minute”, “hour”, “day”, “month”, or “year”
cost	Indicates the amount applied as cost for each configured time unit interval that passes. This is the cost to you, not the price with markup which the customer would see.	Number	A numerical amount such as “3.00” or “34.23”
price	Indicates the amount applied as price for each configured time unit interval that passes. This is the price to the customer with any built-in markup you need to apply.	Number	A numerical amount such as “3.00” or “34.23”

A number of different Task types could be used in this phase. As long as the Task is returning the required JSON array, the Task will work correctly. Below is an example using a Groovy Task. This is simply outputting a static payload though in a real world scenario you’d likely use Task logic to output a dynamic array based on the Instance configuration.

```
def rtn = [
  priceData: [
    [
      incurCharges: 'always',
      currency: 'USD',
      unit: 'hour',
      cost: 2.0,
      price: 2.0
    ],
    [
      incurCharges: 'running',
      currency: 'USD',
      unit: 'hour',
      cost: 3.0,
      price: 3.0
    ],
    [
      incurCharges: 'stopped',
      currency: 'USD',
      unit: 'hour',
      cost: 1.0,
      price: 1.0
    ]
  ]
]
return rtn
```

Nesting Workflows

Morpheus allows Workflows to be nested for easier Workflow creation when many Workflows are used in an environment which have only slight differences or which are made up of common pieces. Nestable Workflows are created like any other Operational Workflow. Once the Workflow is saved, it can be embedded into a special Task type called “Nested Workflow.” A Nested Workflow-type Task simply references an Operational Workflow which may need to be used within other Workflows. Once Nested Workflow Tasks are created they can be used as part of any new Operational or Provisioning Workflows that are created thereafter (or may be added to existing Workflows too). For more on creating Tasks, see Morpheus [Task documentation](#).

Note: Results from prior Tasks are still accessed using the same syntax even when a prior Task is embedded in a Nested Workflow. Additional syntax to reference the Nested Workflow Task or the Workflow itself are not needed. See Morpheus [Task documentation](#) for more on chaining Task results.

- [Nested Workflows Video Demo](#)

Edit Workflow

1. Select the Library link in the navigation bar.
2. Select Automation from the sub-navigation menu.
3. Click the Workflows tab to show the workflows tab panel.
4. Click the Edit icon on the row of the workflow you wish to edit.
5. Modify information as needed.
6. Click the *Save Changes* button to save.

Delete Workflow

1. Select the Library link in the navigation bar.
2. Select Automation from the sub-navigation menu.
3. Click the Workflows tab to show the workflows tab panel.
4. Click the Delete icon on the row of the workflow you wish to delete.

Scale Thresholds

Scale Thresholds are pre-configured settings for auto-scaling Instances. When adding auto-scaling to an instance, existing Scale Thresholds can be selected to determine auto-scaling rules.

Creating Scale Thresholds

1. Navigate to Library > Automation > Scale Thresholds
2. Select + *ADD*
3. Populate the following:

NAME Name of the Scale Threshold

AUTO UPSCALE Enable to automatically upscale per Scale Threshold specifications

AUTO DOWNSCALE Enable to automatically downscale per Scale Threshold specifications

MIN COUNT Minimum node count for Instance. Auto-scaling will not downscale below MIN COUNT, and will auto upscale if the MIN COUNT is not met)

MAX COUNT Maximum node count for Instance. Auto-scaling will not upscale past MAX COUNT, and will auto downscale if MAX COUNT is exceeded.

ENABLE MEMORY THRESHOLD Check to set auto-scaling by specified memory utilization threshold (%)

MIN MEMORY Enter MIN MEMORY % for triggering downscaling.

MAX MEMORY Enter MAX MEMORY % for triggering upscaling.

ENABLE DISK THRESHOLD Check to set auto-scaling by specified disk utilization threshold (%)

MIN DISK Enter MIN DISK % for triggering downscaling.

MAX DISK Enter MAX DISK % for triggering upscaling.

ENABLE CPU THRESHOLD Check to set auto-scaling by specified overall CPU utilization threshold (%)

MIN CPU Enter MIN CPU % for triggering downscaling.

MAX CPU Enter MAX CPU % for triggering upscaling.

Power Scheduling

Set weekly schedules for shutdown and startup times for Instances and VM's, apply Power Schedules to Instances pre or post-provisioning, apply Power Schedule policies on Group or Clouds, or use Guidance to automatically recommend and apply optimized Power Schedules.

Create Power schedules

1. Navigate to Library > Automation > Power Scheduling
2. Select + *ADD*
3. Configure the following options:

NAME Name of the Power Schedule

DESCRIPTION Description for the Power Schedule

TIME ZONE Time Zone the Power Schedule times correlate to.

TYPE

Power On Power Up and then Down at scheduled times

Power off Power Down then Up at scheduled times

Enabled Check for Power Schedule to be Active. Uncheck to disable Power Schedule.

DAYS Slide the start and end time controls for each day to configure each days Schedule. Green sections indicate Power on, red sections indicate Power Off. Time indicated applies to selected Time Zone. The sliders can be used to set time in 15-minute steps, for single-minute granularity click the pencil icon and enter a specific time down to the minute.

EDIT POWER SCHEDULE✕

NAME	<input type="text" value="Nighty Night"/>
DESCRIPTION	<input type="text" value="Down at 6pm, up at at 6am weekdays. Off Weekends"/>
TIME ZONE	<input type="text" value="America/Denver"/> ▾
TYPE	<input type="text" value="Power On"/> ▾

Enabled

MONDAY	 6:00 18:00
TUESDAY	 6:00 18:00
WEDNESDAY	 6:00 18:00
THURSDAY	 6:00 18:00
FRIDAY	 6:00 18:00
SATURDAY	 0:00
SUNDAY	 0:00

4. Select *SAVE CHANGES*

Tip: To view the Instances a power schedule is currently set on, select the name of a Power Schedule to go to the Power Schedule Detail Page.

Add Power Schedule to Instance

1. Navigate to Provisioning > Instances
2. Select an Instance
3. Select *EDIT*
4. In the POWER SCHEDULE dropdown, select a Power Schedule.
5. Select *SAVE CHANGES*

Add Power Schedule to Virtual Machine

1. Navigate to Infrastructure > Compute > Virtual Machines
2. Select a Virtual Machine
3. Select *EDIT*
4. Expand the Advanced Options section
5. In the POWER SCHEDULE dropdown, select a Power Schedule.
6. Select *SAVE CHANGES*

Add Power Schedule Policy

Note: Power Schedule Policies apply to Instances created after the Policy is enabled.

1. Navigate to Administration > Policies
2. Select + *ADD*
3. Select TYPE *Power Schedule*
4. Configure the Power Schedule Policy:

NAME Name of the Policy

DESCRIPTION Add details about your Policy for reference in the Policies tab.

Enabled Policies can be edited and disabled or enabled at any time. Disabling a Power Schedule Policy will prevent the Power Schedule from running on the Clouds Instances until re-enabled.

ENFORCEMENT TYPE

- User Configurable: Power Schedule choice is editable by User during provisioning.
- Fixed Schedule: User cannot change Power Schedule setting during provisioning.

POWER SCHEDULE Select Power Schedule to use in the Policy. Power schedule can be added in Library > Automation > Power Scheduling

SCOPE

Global Applies to all Instances created while the Policy is enabled

Group Applies to all Instances created in or moved into specified Group while the Policy is enabled

Cloud Applies to all Instances created in specified Cloud while the Policy is enabled

User Applies to all Instances created by specified User while the Policy is enabled

Role Applies to all Instances created by Users with specified Role while the Policy is enabled

Permissions- TENANTS Leave blank to apply to all Tenants, or search for and select Tenants to enforce the Policy on specific Tenants.

5. Select *SAVE CHANGES*

Execute Scheduling

Execute Scheduling creates time schedules for Jobs, including Task, Workflow and Backup Jobs. Jobs, which are discussed in greater detail in [another section](#) of Morpheus docs, combine either a Task or Workflow with an Execute Schedule to run the selected Task or Workflow at the needed time. Backup Jobs are a special type of Job configured in the Backups section which also use Execute Schedules to time backup runs as needed.

Schedules use CRON expressions, such as `0 23 * * 2` equalling **Executes every week on Tuesday at 23:00**. CRON expressions can easily be created by clicking the corresponding translation in the create or edit Execution Schedule modal below the Schedule field and selecting a new value.

Note: Execute Schedules CRON expressions should not include seconds or years. The days of the week should be numbered 1-7, beginning with Monday and ending with Sunday. SUN-SAT notation may also be used. For more on writing CRON expressions, many guides are hosted on the Internet including [this one](#). Morpheus execution schedules support most cron syntax but certain more complex expressions may fail to evaluate and the execute schedule will not save. Additionally, some complex expressions may save and work correctly while the friendly written evaluation below the SCHEDULE field is not interpreted correctly. This is due to an issue with the underlying library used to build this feature and cannot easily be resolved at this time.

Create Execution Schedules

NAME Name of the Execution Schedule

Note: When assigning Execution Schedules, the name value will appear in the selection drop-down. Using a name that references the time interval is often helpful

DESCRIPTION Description of the Execution Schedule for reference in the Execution Schedules list

VISIBILITY Master Tenant administrators may share Execute Schedules with Subtenants by setting the visibility to Public

TIME ZONE The time zone for execution

Enabled Check to enable the schedule. Uncheck to disable all associated executions and remove the schedule as an option for Jobs in the future

SCHEDULE Enter CRON expression for the Execution Schedule, for example `0 0 * * *` equals **Every day at 00:00**

SCHEDULE TRANSLATION The entered CRON schedule is translated below the SCHEDULE field. Highlighted values can be updated by selecting the value, and relevant options will be presented. The CRON expression will automatically be updated

5.4.3 Blueprints

Overview

The Blueprints section is used to compose provisioning catalogs. The Blueprints section is composed of:

- Instance Types
- Layouts
- Node Types
- App Blueprints
- Catalog Items
- Cluster Layouts

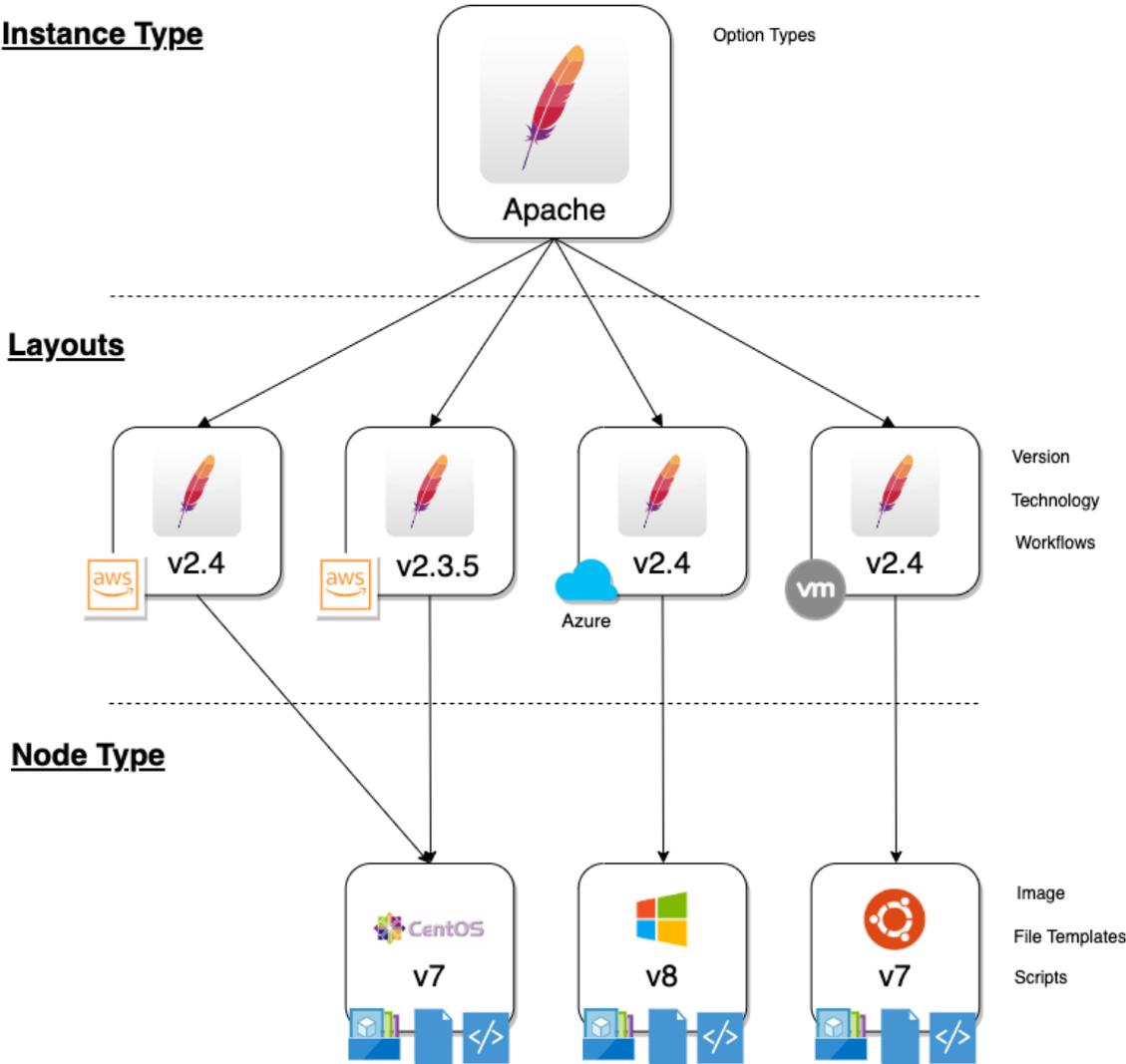
Additionally, items from Options and Templates sections are incorporated to call in custom options for users, IaaS spec files, scripts, and more. See Options and Templates within Morpheus Library for more information on creating or sourcing the items below from version control. In some cases, they may need to be pre-existing for the most efficient creation of Blueprints.

- Inputs
- Option Lists
- File Templates
- Scripts
- Spec Templates

Blueprint Development Overview

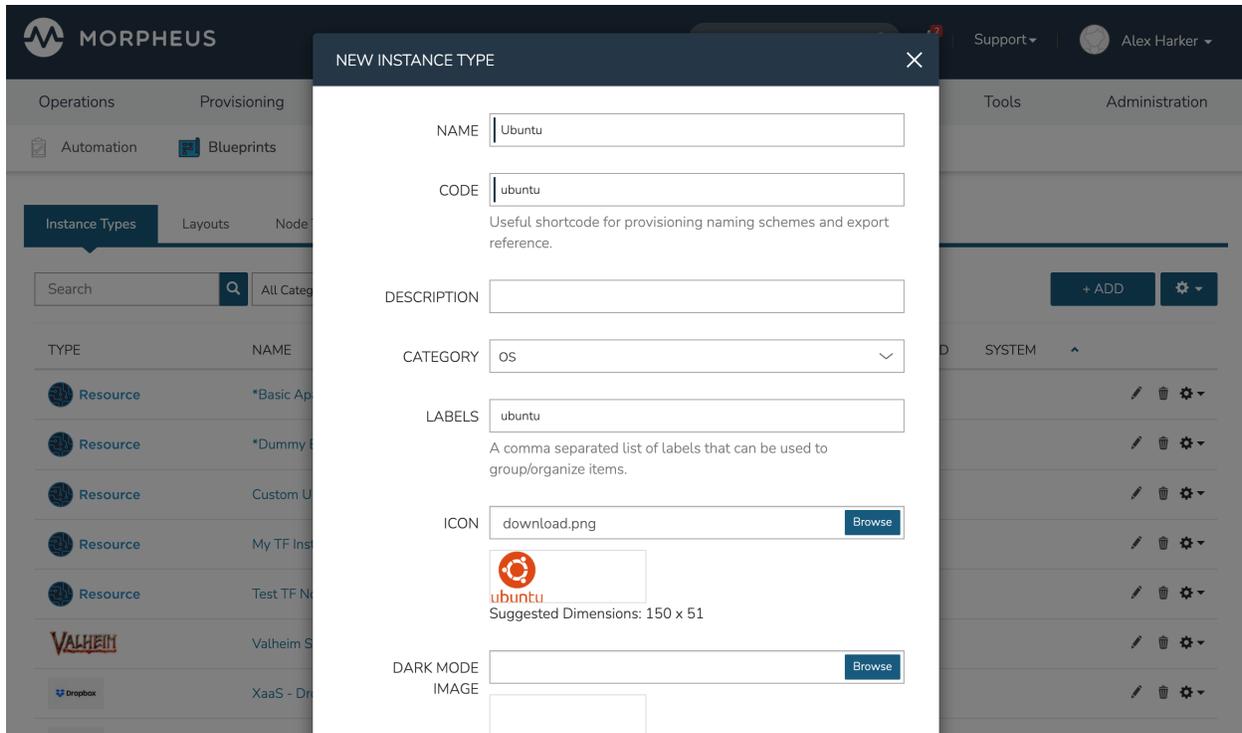
When provisioning, the User selects an `INSTANCE TYPE` from the provisioning wizard. At this stage, we can present custom `INPUTS` to the User which alter deployment in ways the administrator predetermines. Based on the selected Cloud technology and Version, the User is presented with applicable `LAYOUTS` selections. `LAYOUTS` can take advantage of Workflows which automate Tasks and can utilize a wide range of DevOps automation technologies. One or more `NODE TYPES` is associated with the `LAYOUT`. `NODE TYPES` are the bridge between `LAYOUTS` and Images. `NODE TYPES` can also take advantage of File Templates for custom configuration and Scripts which can be queued to run at any stage of the Instance lifecycle.

Understanding Library Items



Instance Types

Adding an Instance Type creates a new Library item category. Multiple Layouts can be added to an Instance Type and these Layouts can have different Nodes attached. The Instance provisioning wizard will present the Layout options compatible with the selected Cloud. If Cloud selection is turned off, all Layouts will be presented for all Cloud types accessible by the User.



Name Name of the Instance Type in the provisioning Library

Code A useful shortcode for provisioning naming schemes and export reference

Description The description of the Instance Type shown in the Provisioning Library. (255 characters max)

Category For filtering in Instance sections and provisioning wizard

- Web
- SQL
- NoSQL
- Apps
- Network
- Messaging
- Cache
- OS
- Cloud
- Utility

Icon An identifiable icon to display in-line with your Instance Type in the provisioning wizard (Suggested dimensions: 150 x 51)

Visibility

- Private: Only accessible by assigned Accounts/Tenants
- Public: Accessible by all Accounts/Tenants

Inputs Custom options presented to the user at provision time, Inputs are also created and stored in Morpheus Library

Price Sets Associate a Price Set with the Instance Type, Price sets are created in Administration > Plans & Pricing > Price Sets. Price Sets which are added to Instance Types become additive with any pricing which may apply on the Service Plan. For example, a “fixed” Price Set of \$1000/month has been associated with the Instance Type. If this Instance Type is provisioned to an Amazon AWS Cloud, the additional fixed price would be computed along with any Price which is pre-existing on the AWS Service Plan

- **Instance Type Price Sets Demo**

Environment Prefix Used for exportable environment variables when tying Instance Types together in App contexts. If not specified, a name will be generated

Environment Variables Name and value pairs for environment variables to be loaded on initialization

Enable Settings Allows for attachment of modifiable file templates to Node Types in a settings tab of the Instance after deployment

Enable Scaling (Horizontal) Enables load balancer assignment and auto-scaling features

Support Deployments Enables deployment features (Requires a data volume be configured on each version. Files will be copied into this location)

Upon saving, this Instance Type will be available in the provisioning catalog, per User Role access. However, we still need to add Layouts to the Instance Type, and prior to creating a Layout, we will add a Node Type.

Layouts

Layouts are attached to Instance Types. A Layout can only be attached to a single Instance Type and a single Technology. An Instance Type can have one or many Layouts attached to it, allowing for a single Instance Type to work with any Technology type. Node Types are added to Layouts. A Layout can have one or many node types attached to it. Node types can be shared across Layouts of matching Technology types.

Important: Once an Instance Type is defined on a Layout and saved, the Instance Type setting and Technology selections on the Layout cannot be changed.

Layout List View

The default page for Layouts is the Layout list view. Select + *ADD* to create a new Layout. Layouts can also be created from an Instance Type detail page.

The following fields are displayed for each Layout:

- **INSTANCE TYPE:** Select the Instance Type to associate with the Layout. This attribute is automatically set and not shown when creating the Layout from an Instance Type detail page
- **NAME:** Links to the Layout detail page
- **VERSION**
- **DISPLAY ORDER:** Determines the Layout display order in the provisioning wizard when adding an Instance of the associated Instance Type
- **DESCRIPTION**

Note: The order of Layouts given when an Instance Type is being provisioned is in high-to-low order of the Layout Display Order property. Bear in mind this is reversed from how Service Plans are ordered, which is in low-to-high order based on the Display Order value.

The Actions menu in each row reveals the following options:

- **Permissions:** Scope the Layout to Group(s) to narrow the list of available groups for a chosen Instance Type at provision time
- **Edit:** Edit the Layout
- **Delete:** Delete the Layout

Note: A Layout that is in use cannot be deleted.

Available Filters:

- **Technology:** Display Layouts by selected Cloud technology
- **Instance Type:** Display Layouts by the associated Instance Type

Layout Detail View

The Layout Detail view shows details on the Layout including Name, Short Name, Version, and Category. It also lists all associated Node Types.

- Select a Layout Name from the list page or Instance Type detail page to get to a Layout detail page.

Layout Configuration Options

Instance Type Select the Instance Type to add to the new Layout. Custom Instance Types must already be created and one Layout cannot be added to multiple instance types. The Instance Type also cannot be changed after creation.

Note: Layouts cannot be added to Morpheus pre-defined Instance Types

Name The name the Layout presents as in the Configuration Options dropdown of the provisioning wizard

Version The version number or name for the Layout. Layouts in an Instance Type with the same version will all show under the Configuration Options dropdown when that version is selected while provisioning

Description Description of the Layout, viewable on the Layout list view

Creatable When checked, this Layout will be selectable at provision time for the associated Instance Type (assuming the Layout is otherwise compatible with provisioning conditions). Instance Types with no Creatable Layouts will not be selectable from the provisioning wizard

Technology Technology determines which Cloud this layout will be available for and which Node Types can be added to it

Minimum Memory Defines the minimum amount of memory required for this Layout. Only Service Plans that meet the defined memory minimum will be available during provisioning when this Layout is selected. Custom memory values must also meet this minimum. Entering a minimum memory value of zero (the default value) indicates no minimum. This minimum memory value will override any Virtual Image minimum memory requirements

Workflow Select a Workflow which will be associated as the Provisioning Workflow for Instances provisioned using this Layout. If a Workflow is defined, it is not shown to the user at provision time and will be run in addition to any Workflows set on the Instance Type, in Workflows Policies, defined in App Blueprints, or selected manually at provision time.

Supports Convert to Managed Enabled to allow users to select this layout when converting a Discovered workload to Managed

Enable Scaling (Horizontal) Enables Instances with this layout to use scaling features

Environment Variables Custom environment variables to be added to the Instance when provisioned

Inputs Search for and select one or multiple Inputs to add to the Layout. Inputs (except for Hidden Inputs) will appear in Provisioning, App, Blueprint, and Cloning wizards when this layout is selected

Nodes Single or multiple nodes can be added to a Layout by searching for and selecting the Node(s)

Price Sets Associate a Price Set with the Layout, Price Sets are created in Administration > Plans & Pricing > Price Sets. Price Sets which are added to Layouts become additive with any pricing which may apply on the Service Plan. For example, a “fixed” Price Set of \$1000/month has been associated with the Layout. If this Layout is provisioned to an Amazon AWS Cloud, the additional fixed price would be computed along with any Price which is pre-existing on the AWS Service Plan

- **Layout Price Sets Demo**

Node Types

Node Types are the link between Images and Layouts.

Node Type Configuration Options

The following fields are for all Technology types:

Name Name of the Node Type in Morpheus

Short Name The short name is a lowercase name with no spaces used for display in your container list

Version Version for the Node Type. Examples: 7.5, 2012 R2, latest

Technology Select associated Technology. This will filter the available configuration options, images and which Layouts the Node Type can be added to

Environment Variables Add pre-set environment variables to the Node Type

Category Node Types of differing categories within the same Layout can have differing sizing

Technology-Specific Options

The Options fields will change depending on the Technology option selected. For VM provisioning technology options, select an image from the VM Image dropdown. This list is populated from the Morpheus Virtual Images section and will include images uploaded into Morpheus as well as synced images from added clouds.

Note: Amazon and Azure Marketplace Images can be added in the Virtual Images section for use as Node Types in custom library items.

Docker Options

For Docker, type in the name and version of the Docker Image, then select the integrated registry.

Service Ports To open ports on the node, enter the name and port to expose. The Load Balancer HTTP, HTTPS, or TCP setting is required when attaching to Load Balancers.

Defining an Exposed port will also create a hyperlink(s) on the container location (IP) in the VM or Container section of the associated Instance detail page.

Scripts Search for and select one or multiple scripts to be executed when the Node Type is provisioned

File Templates Search for and select one or multiple File Templates to be written when the Node Type is provisioned

Example port configuration:

Expose Ports			Add Port
NAME	PORT	LB	
<input type="text" value="HTTP"/>	<input type="text" value="8000"/>	<input type="text" value="HTTP"/>	
<input type="text" value="Collector"/>	<input type="text" value="8088"/>	<input type="text" value="None"/>	
<input type="text" value="Forwarder"/>	<input type="text" value="9997"/>	<input type="text" value="None"/>	
<input type="text" value="KVstore"/>	<input type="text" value="8191"/>	<input type="text" value="None"/>	
<input type="text" value="TCP"/>	<input type="text" value="1514"/>	<input type="text" value="None"/>	
<input type="text" value="Custom"/>	<input type="text" value="1515"/>	<input type="text" value="None"/>	

VMware Options

Morpheus supports VMware-specific configurations related to Node Types targeted for VMware vCenter Clouds. Setting the TECHNOLOGY field on the Node Type to “VMware” reveals these fields.

vApp Properties

Some VMware images may expect the user to provide values for vApp properties related to server configuration. Morpheus allows the user to set values for vApp properties on the Node Type, which can be static values or even Morpheus variables such as if we wanted to provide the next available IP address from a Morpheus IP pool or source a password from Morpheus Cypher. Consider the following example workflow for examining an OVA image, uploading it to Morpheus as a Virtual Image, and setting vApp properties on the Node Type.

If you have an OVA that doesn't have the properties laid out in a visible format, you can unzip it and inspect the OVF file to help set the vApp properties in Morpheus. For example, take a look at the screenshot below from an OVF file associated with an F5 image. There are four vApp properties I wish to set related to network and user configuration. The userConfigurable property should be toggled to true for any that may be set to false. The key is identified by the key property and, if desired, default values can be set via the value property. Save any changes to the OVF file.

```
<Category>Network properties</Category>
  <Property ovf:key="net.mgmt.addr" ovf:type="string" ovf:value="" ovf:userConfigurable="true">
    <Label>mgmt-addr</Label>
    <Description>F5 BIG-IP VE's management address in the format of "IP/prefix"</Description>
  </Property>
  <Property ovf:key="net.mgmt.gw" ovf:type="string" ovf:value="" ovf:userConfigurable="true">
    <Label>mgmt-gw</Label>
    <Description>F5 BIG-IP VE's management default gateway</Description>
  </Property>
<Category>User properties</Category>
  <Property ovf:key="user.root.pwd" ovf:password="true" ovf:type="string" ovf:value="" ovf:userConfigurable="true">
    <Label>root-pwd</Label>
    <Description>F5 BIG-IP VE's SHA-512 shadow or plain-text password for "root" user</Description>
  </Property>
  <Property ovf:key="user.admin.pwd" ovf:password="true" ovf:type="string" ovf:value="" ovf:userConfigurable="true">
    <Label>admin-pwd</Label>
    <Description>F5 BIG-IP VE's SHA-512 shadow or plain-text password for "admin" user</Description>
  </Property>
```

With changes saved, the image can be uploaded to Morpheus as a Virtual Image from which we can create and configure a Node Type. Below you can see the Virtual Image uploaded and revealed on the Virtual Images list page.

The screenshot shows the Morpheus web interface. The top navigation bar includes the Morpheus logo, a search bar, and user information for Alex Harker. Below the navigation bar, there are tabs for Operations, Provisioning, Library (selected), Infrastructure, Backups, Monitoring, Tools, and Administration. Under the Library tab, there are sub-tabs for Automation, Blueprints, Virtual Images (selected), Options, Templates, and Integrations. The main content area shows a search bar with 'f5' entered, and filters for 'All Types', 'User', and 'Labels'. A table lists the virtual images:

TYPE	NAME	PLATFORM	LABELS	SIZE	VISIBILITY	TENANTS	SOURCE	STATUS
VMXK	F5New	linux 64-bit		2.2GiB	Private	Morpheus QA	UPLADED	Active

Next, create a new Node Type. After setting the TECHNOLOGY value to “VMware” the fields related to vApp Property configuration will be revealed. Select the uploaded Virtual Image as the “VM IMAGE” and set the key/value pairs in VAPP PROPERTIES. In this case, I've dynamically loaded the values using Morpheus variables.

VMware VM Options

VM IMAGE	F5New	
LOG FOLDER		
CONFIG FOLDER		
DEPLOY FOLDER		
(Optional) If using deployment services, this mount point will be replaced with the contents of said deployments.		
EXTRA OPTIONS	Name	Value
VAPP PROPERTIES	net.mgmt.addr	<%=networkConfig.prim
	user.admin.pwd	<%=networkConfig.prim
	user.root.pwd	
	net.mgmt.gw	<%=networkConfig.prim

The rest of the process is the same as building out any other Morpheus library item. House the Node Type within a Layout and the Layout within an Instance Type. It should then be provisionable as any other Instance Type.

Extra Options

When VMware Technology Type is selected, EXTRA OPTIONS will be available in the VMware VM Options section. These allow defining Advance vmx-file parameters during provisioning.

Some Example include:

```
tools.setinfo.sizeLimit : 1048576
vmci0.unrestricted : FALSE
isolation.tools.diskWiper.disable : TRUE
```

Note: Not all parameters can be set using extra config parameters. A sample reference list can be found at <http://www.sanbarrow.com/vmx/vmx-advanced.html#vmx>

Important: Use caution when setting Extra Options. Malformed config files can break provisioning. Troubleshooting issues related to Extra Options defined are beyond the scope of Morpheus product support.

App Blueprints

App Blueprints support a vast array of providers and configurations with programmatic markup or Infrastructure as Code capabilities. Blueprints configs can be manually added or scoped to a git repo. Morpheus blueprints allows for full automation configuration, locked fields, tiered boots, and linked tiers with exported envs. All blueprints have permission settings for controlling group and tenant access.

App Blueprint Types

- Morpheus
- Terraform
- ARM (Azure)

- CloudFormation (AWS)
- Kubernetes
- Helm

ARM Blueprints

ARM Blueprints provide a simple and repeatable way of deploying infrastructure-as-code to Azure Clouds. Objects and properties are defined in a JSON file and are provisionable on-demand in *[ProApp]*

To create a new ARM Blueprint, navigate to Library > Blueprints > App Blueprints. Click + *ADD*.

On the Name tab of the New Blueprint modal, enter a name for your new Blueprint. In the Type dropdown menu, select ARM. *NEXT*

In the Blueprint Summary section, complete the following fields as needed:

- **NAME:** Enter a name for this Blueprint as it will appear in the Morpheus Blueprints list
- **DESCRIPTION:** An optional description field for your Blueprint
- **CATEGORY:** An optional category tag for your Blueprint, such as web, utility, or app
- **IMAGE:** An optional image icon to more easily identify your Blueprint from a list. If no image is uploaded, a default image will be used

The ARM template itself is defined in the ARM Configuration section. Using the Config Type dropdown menu, we can opt to write or paste JSON configuration directly into this modal, or we can choose to bring in a JSON which we're keeping under version control in a Git repository.

Depending on whether we need the Morpheus Agent installed and/or cloud-init enabled, mark the following boxes in the next section:

- **INSTALL AGENT**
- **CLOUD INIT ENABLED**

If writing or pasting your configuration JSON directly into the modal, fill out the following fields:

- **OS TYPE:** Identify the resources to be created as Linux or Windows
- **CONFIG TYPE:** ARM Template JSON (.json)
- **CONFIG:** Your JSON configuration template

If bringing in a template from a Git repository, fill out the following fields:

- **OS TYPE:** Identify the resources to be created as Linux or Windows
- **CONFIG TYPE:** "Git Repository"
- **SCM INTEGRATION:** If a pre-existing SCM integration is selected here, the available selections in the "Repository" dropdown menu will be filtered to show only those associated with the chosen SCM integration
- **REPOSITORY:** Select the repository in which your configuration resides
- **BRANCH OR TAG:** The branch in which your configuration resides
- **WORKING PATH:** The path to your configuration files
- **CONFIG:** Your selected config file

Once finished, click *COMPLETE*.

Your new ARM Blueprint is now saved and should be visible in the list of Blueprints. Blueprints are deployed in the Provisioning > Apps section of Morpheus. See the Apps section of Morpheus docs for more information on that process.

CloudFormation Blueprints

CloudFormation Blueprints consume new or existing CloudFormation templates to create easily-deployable application stacks. CloudFormation templates in Morpheus are JSON or YAML-formatted text documents that declare all relevant AWS resources needed for the provisioned application. They can be created directly in the New Blueprint modal or pulled in from existing Git repositories.

If needed, Amazon has educational resources available for getting started with CloudFormation. They can be found in the [AWS CloudFormation documentation](#).

To create a new CloudFormation Blueprint, navigate to Library > Blueprints > App Blueprints. Click + *ADD*.

On the Name tab of the New Blueprint modal, enter a name for your new Blueprint. In the Type dropdown menu, select CloudFormation. Click *NEXT*

In the Blueprint Summary section, complete the following fields as needed:

- **NAME:** Enter a name for this Blueprint as it will appear in the Morpheus Blueprints list
- **DESCRIPTION:** An optional description field for your Blueprint
- **CATEGORY:** An optional category tag for your Blueprint, such as web, utility, or app
- **IMAGE:** An optional image icon to more easily identify your Blueprint from a list. If no image is uploaded, a default image will be used

Depending on whether we need the Morpheus Agent installed and/or cloud-init enabled, mark the following boxes in the next section:

- **INSTALL AGENT**
- **CLOUD INIT ENABLED**

In some cases, you must explicitly acknowledge that your template contains certain capabilities in order for the application to successfully be deployed. There is more information on this in Amazon's documentation [here](#). If any of the following capabilities are contained in your application, acknowledge them by marking any of the following boxes that apply:

- **CAPABILITY_IAM**
- **CAPABILITY_NAMED_IAM**
- **CAPABILITY_AUTO_EXPAND**

Continuing on with the CloudFormation Configuration section, complete the following fields as needed when entering your configuration directly into the new Blueprint:

- **CONFIG TYPE:** "CloudFormation Template JSON (.json)"
- **CONFIG TYPE:** "CloudFormation Template YAML (.yaml)"
- **CONFIG:** Enter your configuration here

In the CloudFormation Configuration section, complete the following fields as needed when syncing in configuration from a Git repository:

- **CONFIG TYPE:** "Git Repository"

- **SCMINTEGRATION:** If a pre-existing SCM integration is selected here, the available selections in the “Repository” dropdown menu will be filtered to show only those associated with the chosen SCM integration
- **REPOSITORY:** Select the repository in which your configuration resides
- **BRANCH OR TAG:** The branch in which your configuration resides
- **WORKING PATH:** The path to your configuration files
- **CONFIG:** Your selected config file

Once finished, click *COMPLETE*.

Your new CloudFormation Blueprint is now saved and should be visible in the list of Blueprints. Blueprints are deployed in the Provisioning > Apps section of Morpheus. See the Apps section of Morpheus docs for more information on that process.

Helm Blueprints

If you’re using Helm Charts to manage Kubernetes applications, Morpheus allows you to bring them in from a Git repository as a Blueprint. The selected repository must be integrated with Morpheus before creating the Blueprint.

To create a new Helm Blueprint, navigate to Library > Blueprints > App Blueprints. Click + *ADD*.

On the Name tab of the New Blueprint modal, enter a name for your new Blueprint. In the Type dropdown menu, select Helm. Click *guilabel:NEXT*.

In the Blueprint Summary section, complete the following fields as needed:

- **NAME:** Enter a name for this Blueprint as it will appear in the Morpheus Blueprints list
- **DESCRIPTION:** An optional description field for your Blueprint
- **CATEGORY:** An optional category tag for your Blueprint, such as web, utility, or app
- **IMAGE:** An optional image icon to more easily identify your Blueprint from a list. If no image is uploaded, a default image will be used

In the Helm Configuration section, complete the following fields as needed to sync in configuration from a Git repository:

- **CONFIG TYPE:** “Git Repository”
- **SCMINTEGRATION:** If a pre-existing SCM integration is selected here, the available selections in the “Repository” dropdown menu will be filtered to show only those associated with the chosen SCM integration
- **REPOSITORY:** Select the repository in which your configuration resides
- **BRANCH OR TAG:** The branch in which your configuration resides
- **CHART PATH:** The path to the folder within the repository containing your configuration files, enter “./” if this is the top level folder within the repository
- **CONFIG:** Config files within your selected folder are displayed here for confirmation

Once finished, click *COMPLETE*.

Your new Helm Blueprint is now saved and should be visible in the list of Blueprints. Blueprints are deployed in the Provisioning > Apps section of Morpheus. See the Apps section of Morpheus docs for more information on that process.

Kubernetes Blueprints

Morpheus allows you to store Kubernetes configuration YAML files for easy deployment on-demand. Kubernetes Blueprints can be built by pulling in Kubernetes spec stored as a Morpheus Spec Template object, those tracked under version control in a Git repository, or you can write them directly in the New Blueprint modal.

To create a new Kubernetes Blueprint, navigate to Library > Blueprints > App Blueprints. Click + *ADD*.

On the Name tab of the New Blueprint modal, enter a name for your new Blueprint. In the Type dropdown menu, select Kubernetes. *NEXT*

In the Cluster Summary section, complete the following fields as needed:

- **NAME:** Enter a name for this Blueprint as it will appear in the Morpheus Blueprints list
- **DESCRIPTION:** An optional description field for your Blueprint
- **CATEGORY:** An optional category tag for your Blueprint, such as web, utility, or app
- **IMAGE:** An optional image icon to more easily identify your Blueprint from a list. If no image is uploaded, a default image will be used

Complete the Kubernetes Configuration section as follows depending on your Config Type selection.

To consume a Morpheus Spec Template containing Kubernetes spec:

- **CONFIG TYPE:** “Kubernetes Spec”
- **SPEC TEMPLATE:** Use the typeahead field to locate the desired Spec Template

To draft or paste configuration directly in the New Blueprint modal:

- **CONFIG TYPE:** “Kubernetes Yaml Spec”
- **CONFIG:** Enter your YAML configuration template here

To consume YAML configuration files tracked in a Git repository:

- **CONFIG TYPE:** “Git Repository”
- **SCM INTEGRATION:** If a pre-existing SCM integration is selected here, the available selections in the “Repository” dropdown menu will be filtered to show only those associated with the chosen SCM integration
- **REPOSITORY:** Select the repository in which your configuration resides
- **BRANCH OR TAG:** The branch in which your configuration resides
- **WORKING PATH:** The path to your configuration files
- **CONFIG:** Your selected config file

Once finished, click *COMPLETE*.

Your new Kubernetes Blueprint is now saved and should be visible in the list of Blueprints. Blueprints are deployed in the Provisioning > Apps section of Morpheus. See the Apps section of Morpheus docs for more information on that process.

Morpheus Blueprints

Morpheus App Blueprints allow pre-configured full multi-tier application deployments for multiple environments. Blueprints can be provisioned from the Provisioning > Apps section and can be fully configured for one click provisioning. Blueprints can be built within the **Builder** section or by code in the **Raw** section. Blueprints can also be exported as YAML or JSON and created with the Morpheus API and CLI.

A unique capability of the YAML/JSON based Morpheus blueprint structure is the ability to have multiple configurations per instance being provisioned within the app blueprint. This can be a scoped configuration that acts as overrides based on selected cloud, group, and/or environment the app is being provisioned in as a target. For example, maybe the “development” environment doesn’t need as many horizontally scaled nodes as the “production” environment. Another great aspect of this configuration markup is a blueprint can be defined as a hybrid cloud blueprint. This makes the app blueprint structure very powerful and in some ways better than alternative infrastructure as code orchestrators. For Example, ARM is locked into Azure, while CloudFormation is locked into AWS. Even Terraform does not allow a tf file to expand its bounds beyond a specific provider type.

Basic Blueprint Structure

In a Morpheus App Blueprint there are a few structural concepts to be aware of. Firstly there is a concept of a *Tier*. A *Tier* is a grouping of instances within an app blueprint. Tiers can be used for a variety of things including sequenced booting of instances or even properly creating endpoint groups and security group contexts in network security tools like Cisco ACI. An example of a Tier structure might be a *Web* tier and a *Database* tier. These tiers can also be marked as connected such that network communication rules can appropriately be defined. A basic 2 Tier blueprint skeleton might look something like this:

```
name: Tier Example
type: morpheus
tiers:
  Web:
    linkedTiers:
      - Database
    tier:
      bootOrder: 1
    instances:
  Database:
    tier:
      bootOrder: 0
    instances:
```

This example has defined 2 tiers as yaml properties under the *tiers* object. They are called *Web* and *Database*. A Tier can optionally define its connected tiers which are bi-directional even though only one tier has to define them. This is the *linkedTiers* array and simply lists the connected tiers by tier name. A Boot Order can also optionally be defined under a nested {"tier": {"bootOrder": 1}} object structure.

Configuration Scopes

Another capability of Morpheus App Blueprint structure is its configuration scoping. This allows properties to be overridden based on the apps target environment or even target group and cloud. For example. Maybe we want to use a larger plan size in production vs. development

An example of that can be done using “environments” overrides.

```
name: Simple Nginx
type: morpheus
tiers:
  Web:
    instances:
      - instance:
          type: docker
          name: Sample Nginx
        clouds:
          AWS Cali:
            instance:
              layout:
                code: docker-1.7-single
            config:
              dockerImageVersion: latest
              dockerRegistryId: ''
              dockerImage: nginx
            plan:
              code: container-128
          environments:
            Production:
              groups:
                All Clouds Demo:
                  clouds:
                    AWS Cali:
                      plan:
                        code: container-256
```

Note the new environments object. The object graph of the morpheus blueprint structure gets merged and flattened at provision time based on the scope of the configurations provided as well as the users target cloud, group, and environment selection. In the Above example, a selective override was done for the *AWS Cali* cloud when using a Production Environment and deploying to the group *All Clouds Demo*. This specific example changes the plan to a larger size. Scoped configurations have various levels of precedence. Cloud is the lowest level of precedence. a cloud configuration in a group is the next level higher and finally an environment configuration in a group in a cloud is the highest level of scoped precedence.

Getting Started

To get started, it may be best to look at a simple App Blueprint configuration. Docker templates are less complex than virtual machine based templates so lets look at a Blueprint that deploys a single Nginx container to a target cloud:

```
name: Simple Nginx
type: morpheus
tiers:
  Web:
    linkedTiers: []
    instances:
      - instance:
          type: docker
          name: Sample Nginx
    clouds:
      AWS Cali:
        instance:
          layout:
            code: docker-1.7-single
            id: 206
        volumes:
          - rootVolume: true
            name: root
            size: 1
        backup:
          createBackup: false
        config:
          dockerImageVersion: latest
          dockerRegistryId: ''
          dockerImage: nginx
        plan:
          id: 68
          code: container-128
        ports:
          - name: HTTP
            port: 80
            lb: HTTP
```

There are some useful things to look at in the above docker example. One is there are different objects based on the different available configuration options for the target provision type. These options are actually data driven and can be extracted from the Inputs api in the morpheus api doc. That is a useful resource to look at while building morpheus blueprints or by using the *morpheus-cli* which provides prompts for helping build custom morpheus app blueprints.

EDIT APP TEMPLATE
✕

Builder
Raw
Preview

STRUCTURE

- Spud Marketing
+
- Web
+
✕
- Tomcat
+
✕
- Group: All Clouds Demo,
Cloud: Appliance KVM &
Docker
✕
- Group: All Clouds Demo,
Cloud: Labs UCS
✕
- Database
+
✕
- MySQL
+
✕
- Group: All Clouds Demo,
Cloud: Appliance KVM &
Docker
✕
- Group: All Clouds Demo,
Cloud: Labs UCS
✕

CONFIGURATION

Template Summary

NAME

DESCRIPTION

CATEGORY

IMAGE BROWSE

SAVE

Creating Morpheus App Blueprints

1. Navigate to Library > Blueprints > App Blueprints
2. Select + *ADD*
3. Enter a NAME for the Blueprint and select *NEXT*
4. Optionally add a Description, Category, and Image for the Blueprint.

Add Tiers

1. In the STRUCTURE section, select + to add a Tier
2. Select or enter a Tier Name.
3. Select the Tier to set Boot Order, rename, or once multiple Tiers are added, connect the Tier to other Tiers.

Add Instances to Tiers

1. In the STRUCTURE section, select + in a Tier to add an Instance
2. Select an Instance Type
3. Optionally add a name for the Instance. Instances with blank names will automatically be named based off the App name.

Tip: You can use the variable `${app.name}` in your instance naming convention to reference the name of the application you're deploying.

Add Configurations to Instances

1. In the STRUCTURE section, select + in an Instance to add a Configuration
2. Select at least one option from Group, Cloud or Environment.
3. Select **ADD CONFIG** to create the configuration
4. Populate the Configuration
 - Configurations can be fully partially or populated
 - Fields can be locked or hidden by selecting the Lock icon next to the Field. Locking prevents the field from being editable when provisioning an App using the Blueprint while hidden fields are not revealed to the user at all
 - **ALLOW EXISTING INSTANCE** will allow users to add existing Instances to the App when using the blueprint

Save

Once all desired Tiers, Instances and Configurations are added, select Save. The Blueprint will be created, can be edited after saving, and will available in the Apps section for provisioning.

Note: Blueprints are not provisioned when created. To provision a Blueprint, use Provisioning > Apps.

RAW

Blueprints can be create, edited or Exported in the RAW section when creating or editing a blueprint.

EDIT APP TEMPLATE

Builder Raw Preview

EXPORT YAML

RAW

```
image: >-
  https://demo.morpheusdata.com/storage/logos/uploads/AppTemplate/1/templateImage/6377893_original.png
tiers:
  Web:
    linkedTiers:
      - Database
    tier:
      bootOrder: 1
      lockedFields:
        - bootOrder
    instances:
      - instance:
          type: tomcat
          groups:
            All Clouds Demo:
              clouds:
                Appliance KVM & Docker:
                  backup:
                    createBackup: true
                  instance:
                    layout:
```

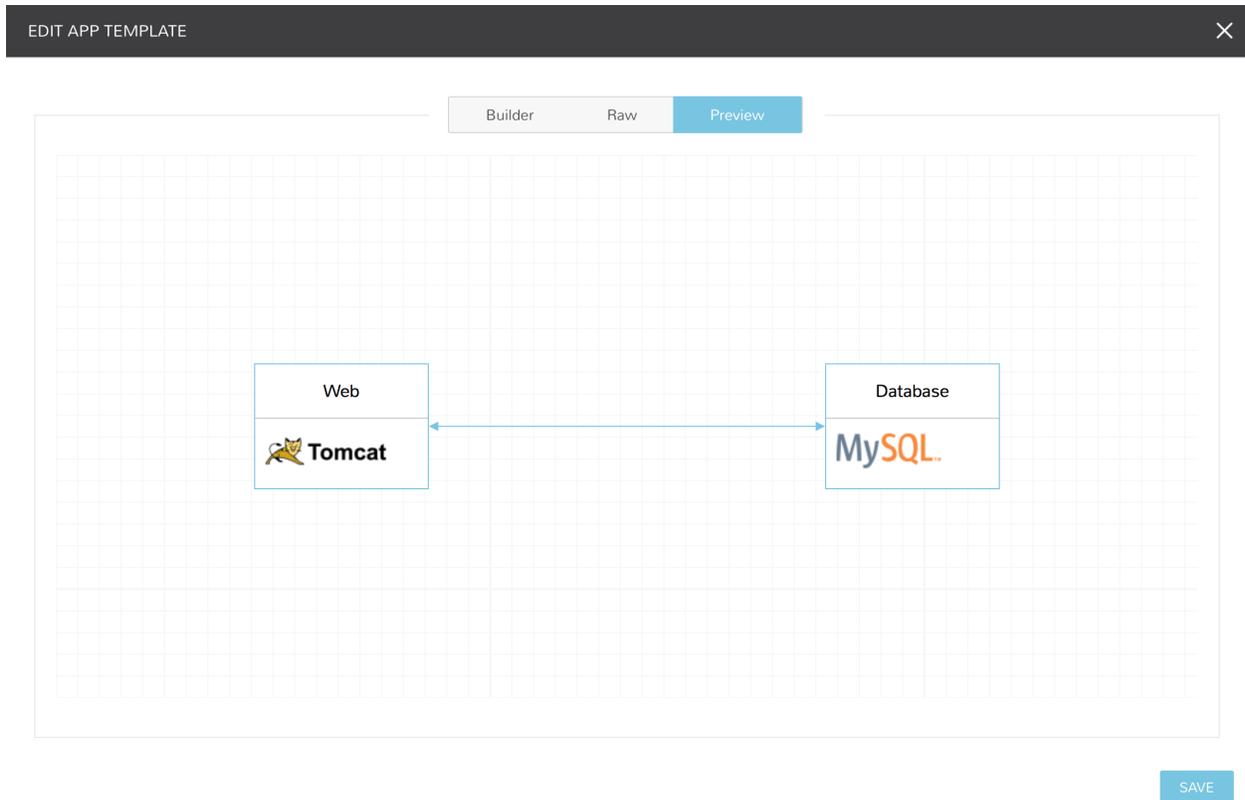
SAVE

To Export a Blueprint as JSON or YAML:

1. Navigate to Library > Blueprints > App Blueprints
2. Edit an existing App by clicking on the pencil icon
3. On the Edit Blueprint modal, select the Raw tab
4. Select YAML or JSON from the dropdown in the top right
5. Click the Export button
6. Select the configurations to include in the export by selecting or deselecting configurations as needed. Selected configurations will be highlighted
7. Click the DOWNLOAD CONFIGURATION button
8. The Blueprint export file will be downloaded to your computer as {app_name}-config.json or {app_name}-config.yaml

Preview

In the APP BLUEPRINT modal, select the Preview section to display a graphical representation of your Blueprint Tiers, Instances and Tier Connections.



Important: When Tiers are connected, the Instances in a Tier will import the envs from Instances in connected Tiers, and if Morpheus is managing the Instance Firewalls, communication between the Instances will be facilitated based on the Instances port configurations.

Provisioning

To provision a Blueprint, navigate to Provisioning > Apps and select the Blueprint when creating an App. See the [App section](#) of Morpheus docs for more on provisioning Apps.

Terraform Blueprints

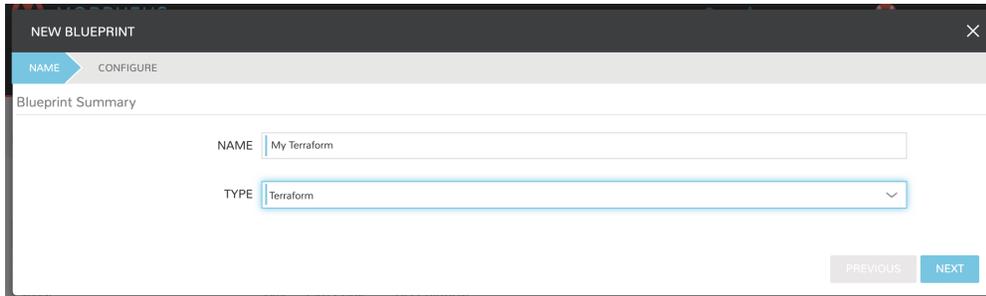
Terraform Blueprints are one way that Terraform can be integrated and leveraged with Morpheus, with the other being the Morpheus Terraform provider which is not discussed in this section. Morpheus and Terraform are complimentary technologies which together can increase efficiency and simplify automation across cloud environments. For more on this relationship, see our [whitepaper](#) on how Morpheus and Terraform are better together.

Currently, Morpheus supports provisioning Apps based on Terraform Blueprints to VMware, Amazon, Azure, and Oracle Clouds with additional Cloud support coming in future releases. On first attempt to provision a Terraform App, Morpheus will automatically install Terraform. It is possible in some operating system configurations for this automated

installation process to fail, requiring you to install Terraform manually. If needed, manual installation instructions and guidance are provided [here](#).

To create a new Terraform Blueprint, navigate to Library > Blueprints > App Blueprints. Click + *ADD*.

On the Name tab of the New Blueprint modal, enter a name for your new Blueprint. In the Type dropdown menu, select Terraform. *NEXT*



The screenshot shows a modal window titled "NEW BLUEPRINT" with a close button in the top right. It has two tabs: "NAME" (active) and "CONFIGURE". Under the "NAME" tab, there is a "Blueprint Summary" section. This section contains two input fields: "NAME" with the text "My Terraform" and "TYPE" with a dropdown menu showing "Terraform". At the bottom right of the modal, there are two buttons: "PREVIOUS" (disabled) and "NEXT" (active).

In the Blueprint Summary section, complete the following fields as needed:

- **NAME:** Enter a name for this Blueprint as it will appear in the Morpheus Blueprints list
- **DESCRIPTION:** An optional description field for your Blueprint
- **CATEGORY:** An optional category tag for your Blueprint, such as web, utility, or app
- **IMAGE:** An optional image icon to more easily identify your Blueprint from a list. If no image is uploaded, a default image will be used

The Terraform Configuration section is where the Terraform template file (.tf) is added or linked to the Blueprint. Using a Config Type of “Terraform (.tf)” or “Terraform JSON (.tf.json)”, you can write or paste your configuration directly into the new Blueprint. Alternatively, you can pull in config files from an integrated Git repository using the “Git Repository” Config type.

In the Terraform Configuration section, complete the following fields as needed when entering your configuration directly into the new Blueprint:

- **CONFIG TYPE:** “Terraform (.tf)” or “Terraform JSON (.tf.json)” to create or paste configuration directly into the new Blueprint
- **CONFIG:** Enter your configuration here
- **TFVAR SECRET:** Select an existing TFVar-formatted Cypher. See the Cyphers section or Morpheus docs for more information on Cyphers
- **OPTIONS:** Enter any additional options, such as a variable definition

In the Terraform Configuration section, complete the following fields as needed when syncing in configuration from a Git repository:

- **CONFIG TYPE:** “Git Repository”
- **SCM INTEGRATION:** If a pre-existing SCM integration is selected here, the available selections in the “Repository” dropdown menu will be filtered to show only those associated with the chosen SCM integration
- **REPOSITORY:** Select the repository in which your configuration resides
- **BRANCH OR TAG:** The branch in which your configuration resides
- **WORKING PATH:** The path to your configuration files
- **CONFIG:** Your selected config file

- **TFVAR SECRET:** Select an existing TFVar-formatted Cypher. See the Cyphers section of Morpheus docs for more information on Cyphers
- **OPTIONS:** Enter any additional options, such as a variable definition

Once finished, click *COMPLETE*.

Your new Terraform Blueprint is now saved and should be visible in the list of Blueprints. Blueprints are deployed in the Provisioning > Apps section of Morpheus. See the Apps section of Morpheus docs for more information on that process.

Catalog Items

The Self Service catalog (Library > Blueprints > Catalog Items) is where administrators can create easily-deployable items for consumption by users operating under the “Service Catalog” Persona in Morpheus. Catalog items can be fully-configured Morpheus Instances or Blueprints, complete with user input through Morpheus Inputs, automation Workflows, and more. The catalog items are presented in a simplified interface for ease of deployment without sacrificing configurability for administrators. All available catalog items are built in the Self Service area and users will see relevant items in their catalogs based on Role permissions.

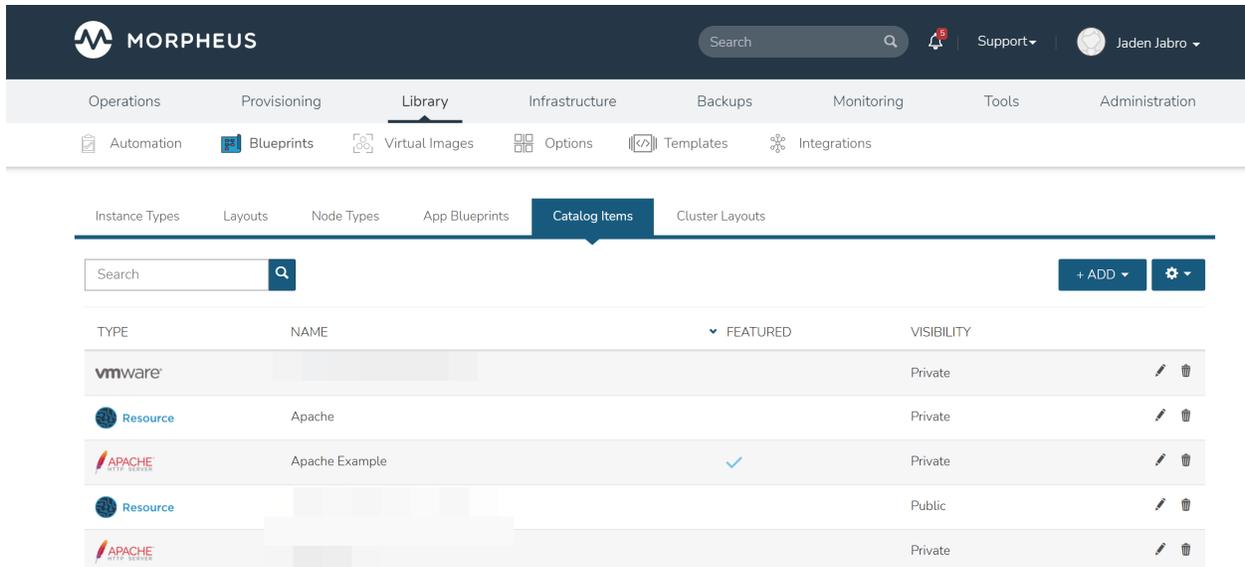
Note: For more on Personas and using the Service Catalog persona, see the Personas sections of our documentation.

Access is granted to the Self Service section through the Library: Catalog Items Role permission. Users with Read rights can view the catalog while users with full rights can create and edit catalog items. Users without any rights to Self Service will not be able to access the page at all.

Additionally, a user’s Role determines access to the standard and/or service catalog persona and which service catalog items are available for a user under the service catalog persona. See the Roles section of Morpheus documentation for more information on administering Roles.

Viewing the Self Service Catalog

The complete Self Service catalog can be viewed by clicking on Catalog from the Provisioning menu. The complete list of items available for the Self Service catalog are shown here but users working in the Service Catalog Persona will see only those allowed based on their user role. In addition to the name and type of each catalog item, we can also see a description and whether not the catalog item is featured or active. Featured items are given special visibility in the Service Catalog Persona and inactive items will not appear as provisioning options.



TYPE	NAME	FEATURED	VISIBILITY	
vmware			Private	 
Resource	Apache		Private	 
APACHE HTTP SERVER	Apache Example		Private	 
Resource			Public	 
APACHE HTTP SERVER			Private	 

Building Catalog Instances

Note: API usage for Catalog Items with Forms is not yet supported. If planning to use the API or CLI to order Catalog Items you should not use Forms on Catalog Items until it is supported.

An Instance in Morpheus is a set of one or more containers or virtual machines that correlate to a single, horizontally-scalable entity or service suite. From the Self Service section, we can pre-configure Morpheus Instances and present them to users viewing the Service Catalog Persona for one-click deployment.

From the Catalog Items List Page (Library > Blueprints > Catalog Items), click *ADD*. From the dropdown menu, select Instance. The modal window will appear to configure and add a new catalog Instance.

EDIT CATALOG ITEM

NAME DESCRIPTION ENABLED FEATUREDVISIBILITY

LOGO

CONFIG [CONFIGURATION WIZARD](#)

```
{
  "group": {
    "id": "<%= customOptions.fgroups %>"
  },
  "cloud": {
    "id": 12,
    "name": "AWS"
  },
  "type": "apache",
  "instance": {
    "userGroup": {
      "id": ""
    },
    "expireDays": "2",
    "shutdownDays": "1"
  }
}
```

CONTENT **# Amazon Apache VM**

This is a sample catalog item that provisions a single amazon apache service as a VM. Several configuration options are presented using option types and lists.

Configure the following:

- **NAME:** A friendly name for the catalog item in Morpheus
- **CODE:** An optional shortcode for coded naming schemes or programmatic reference
- **DESCRIPTION:** An optional description identifying the catalog item
- **CATEGORY:** Select an existing category or enter a new one. When provisioning from the catalog, items can be filtered by category
- **ENABLED:** When checked, this catalog item will be available for provisioning
- **LABELS:** A comma-separated list of descriptive strings which can be used to categorize the Catalog Item. The Catalog Items list view page can be filtered by labels to make them easier to find
- **ENABLED:** When marked, this Catalog Item will be available from the provisioning Catalog view

- **DISABLE AUTO PRICE:** When marked, price estimates are no longer refreshed with every field change at the time this Catalog Item is ordered. For some workloads with heavy Configuration-phase Tasks this may improve performance of the order page
- **FEATURED:** When checked, this catalog item will be given special visibility in the Service Catalog Persona view
- **ALLOW QUANTITY:** When checked, an additional field is added to the order page allowing users to order multiple copies. If this option is enabled, ensure your configuration supports that flexibility (such as when IP or load balancer details are specified)
- **VISIBILITY:** Set to private to keep the catalog item available only to users in the current Tenant. Master Tenant administrators may set catalog items to public to make them viewable and usable by Subtenant users
- **LOGO:** Select or upload a logo to be associated with this catalog item
- **DARK LOGO:** If desired, set an alternate logo for use when the dark theme is applied to the Morpheus appliance
- **CONFIG:** Enter, view, or edit Instance config here. Click *CONFIGURE* to build a base configuration through the Morpheus Instance wizard. After configuration through the Instance wizard, you may need to overwrite some static values in the Instance JSON body with calls to custom Input values. This allows your users to easily set the Instance Plan, Group, name, tags, or anything else they may need to control. Dynamic inputs are passed with the following syntax: “<%= customOptions.fieldName %>” where fieldName is the Field Name value set on the Input
- **CONTENT:** Optionally include documentation content for this Catalog Item. Markdown-formatted text is accepted and displayed appropriately when the item is ordered from the Service Catalog. A new Catalog Item-type Wiki entry will also be added containing this information
- **FORM TYPE:** Choose whether a Form or individual Inputs should be used to set the user options at provisioning time. See [Forms documentation](#) for complete details on creating Forms.
- **FORM:** If the Form Type of “Form” is selected, choose the pre-existing Form which should be associated with the Catalog Item
- **INPUTS:** If the Form Type of “Inputs” is selected, choose the pre-existing Inputs which should be associated with the Catalog Item

Using Forms with Catalog Items

Note: API usage for Catalog Items with Forms is not yet supported. If planning to use the API or CLI to order Catalog Items you should not use Forms on Catalog Items until it is supported.

Using Forms provides a number of advantages over using Inputs. Once the Form is selected, Morpheus helpfully provides a sidebar which contains all variables that can be consumed in the Instance config from the Form:

EDIT CATALOG ITEM ✕

NAME

CODE
Useful shortcode for provisioning naming schemes and export reference.

DESCRIPTION

CATEGORY ☰

LABELS
A comma separated list of labels that can be used to group/organize items.

ENABLED

FEATURED

ALLOW QUANTITY

VISIBILITY

LOGO

DARK LOGO

CONFIG CONFIGURE

1	{
---	---

FORM VARIABLES ?

Group

AUTO INJECT

Cloud

Pool

Layout

Plan

Plan id

Plan Max Cores

Plan Max Memory

Plan Max Cores / Socket

Network

Disk

Tags

CANCEL
SAVE

Many Form Inputs are designed to auto-inject themselves into the Instance config, you can see in the screenshot above that “AUTO INJECT” is checked. For variables that auto-inject, you do not need to override any static configuration with a variable call in order to consume that form value. In most cases, you should leave auto-inject turned on but the option is available to disable it for more advanced situations such as if you wanted to code custom logic into the variable call.

Other types of Form Inputs do not auto-inject and, for these, you must override any static configuration in the Instance config with a variable call. For these situations, Morpheus helpfully provides the results of all variables so you can be sure you’re injecting the proper call. Click on the question mark (?) button immediately to the right of the “FORM VARIABLES” header to see a list of available variables and an example resolved variable. Variable tiles may be dragged from the sidebar into the CONFIG text area and a properly-formatted variable call will be inserted.

×
OPTION TYPE HELP

INPUTS	KEY VALUE
<ul style="list-style-type: none"> Group (Group) Cloud (Cloud) Pool (Resource Pool) Layout (Layout) Plan (Plan) Network (Networks) Disk (Disks) <li style="background-color: #e6f2ff;">Tags (Key Value) 	<p>OUTPUT</p> <pre>[{ "key": "a", "value": "b" }, { "key": "c", "value": "d" }]</pre>

As an example, see the configuration for an Ubuntu server in the expandable section below. You'll notice in the configuration that a VMware Cloud, a specific Group, a specific Plan, and other static configurations are set. Since the Group, Cloud, Plan and other variables are able to be automatically injected, the user may select a different Group, Cloud, Plan, etc. from the form at provision time. The creator of the Catalog Item does not need to override those static configurations with variable calls.

- **Example Catalog Item Config**

```
{
  "hostName": "${userInitials}-${cloudCode}-${type}-${sequence}",
  "metadata": "<%=customOptions.targetTags%>",
  "backup": {
    "backupRepository": 40,
    "veeamManagedServer": "",
    "jobSchedule": 2,
    "createBackup": true,
    "jobAction": "new",
    "jobRetentionCount": "3",
    "providerBackupType": 12,
    "target": 37006
  },
  "instance": {
    "userGroup": {
      "id": ""
    },
    "tags": "Forms,Test"
  },
  "defaultExpandAdvanced": false,
  "volumes": [
    {
      "maxIOPS": null,
      "displayOrder": 0,
      "unitNumber": "0",
      "minStorage": 5368709120,
      "configurableIOPS": false,

```

(continues on next page)

(continued from previous page)

```

    "uuid": "a6781cc1-31ca-406b-aea0-e33ea1a18b7f",
    "controllerMountPoint": "2200223:0:4:0",
    "internalId": "[ESXi-DC2-QA-LUN01] Morpheus Ubuntu 22.04 20230307/Morpheus_
↪Ubuntu 22.04 20230307.vmdk",
    "id": 5255832,
    "datastoreId": "autoCluster",
    "maxStorage": 26843545600,
    "volumeCustomizable": true,
    "readonlyName": false,
    "controllerId": 2200223,
    "externalId": "2000",
    "virtualImageId": 1418543,
    "vId": 1418543,
    "size": 25,
    "name": "root",
    "planResizable": true,
    "rootVolume": true,
    "storageType": 1,
    "typeId": 1,
    "resizeable": true,
    "uniqueId": null
  }
],
"type": "ubuntu",
"ports": [
  {
    "code": "ubuntu.22",
    "visible": true,
    "internalPort": 22,
    "loadBalancePort": null,
    "loadBalanceProtocol": null,
    "sortOrder": 1,
    "name": "SSH",
    "id": 7,
    "shortName": "ssh",
    "externalPort": 22,
    "loadBalance": false
  }
],
"version": "22.04",
"hideLock": true,
"cloud": {
  "name": "QA VMware",
  "id": 26324
},
"layout": {
  "code": "vmware-ubuntu-22.04-single",
  "id": 2608414
},
"showScale": false,
"environment": "2",
"networkInterfaces": [

```

(continues on next page)

```
{
  "ipMode": "",
  "primaryInterface": true,
  "showNetworkPoolLabel": true,
  "showNetworkDhcpLabel": false,
  "network": {
    "idName": "VLAN0002 - Internal Server",
    "pool": {
      "name": "10.32.20.0 /22",
      "id": 18823
    },
    "id": "network-173431",
    "hasPool": false
  },
  "networkInterfaceTypeId": 4,
  "networkInterfaceTypeIdName": "VMXNET 3"
}
],
"copies": 1,
"loadBalancer": [],
"name": "${userInitials}-${cloudCode}-${type}-${sequence}",
"storageControllers": [
  {
    "editable": false,
    "typeName": "IDE",
    "maxDevices": 2,
    "displayOrder": 0,
    "active": true,
    "unitNumber": null,
    "reservedUnitNumber": -1,
    "busNumber": "0",
    "removable": false,
    "name": "IDE 0",
    "typeId": 2,
    "id": 1729031,
    "category": "ide"
  },
  {
    "editable": false,
    "typeName": "IDE",
    "maxDevices": 2,
    "displayOrder": 1,
    "active": true,
    "unitNumber": null,
    "reservedUnitNumber": -1,
    "busNumber": "1",
    "removable": false,
    "name": "IDE 1",
    "typeId": 2,
    "id": 1729032,
    "category": "ide"
  }
],
```

(continues on next page)

(continued from previous page)

```
{
  "editable": false,
  "typeName": "SCSI LSI Logic Parallel",
  "maxDevices": 15,
  "displayOrder": 2,
  "active": true,
  "unitNumber": null,
  "reservedUnitNumber": 7,
  "busNumber": "0",
  "removable": false,
  "name": "SCSI 0",
  "typeId": 4,
  "id": 1729030,
  "category": "scsi"
},
"config": {
  "poolProviderType": null,
  "isVpcSelectable": true,
  "smbiosAssetTag": null,
  "isEC2": false,
  "resourcePoolId": "pool-139625",
  "hostId": null,
  "createUser": true,
  "nestedVirtualization": null,
  "vmwareFolderId": "group-v80",
  "noAgent": false
},
"plan": {
  "code": "vm-8192",
  "id": 149
},
"group": {
  "name": "All Clouds",
  "id": "2"
}
}
```

Once done, click *SAVE CHANGES*

Building Catalog Blueprints

Note: API usage for Catalog Items with Forms is not yet supported. If planning to use the API or CLI to order Catalog Items you should not use Forms on Catalog Items until it is supported.

Morpheus Blueprints allow for full multi-tier application deployment. In the Self Service catalog, user can create catalog items based on pre-existing App Blueprints. If new Blueprints need to be created for the Service Catalog, see other sections of Morpheus docs on building App Blueprints of various supported types. Just like with catalog Instances, we can pre-configure Blueprints and present them to users viewing the Service Catalog Persona view for easy, one-click deployment.

From the Catalog Items List Page (Library > Blueprints > Catalog Items), click *ADD*. From the dropdown menu, select Blueprint. The modal window will appear to configure and add a new catalog Blueprint.

Configure the following:

- **NAME:** A friendly name for the catalog item in Morpheus
- **CODE:** An optional shortcode for coded naming schemes or programmatic reference
- **DESCRIPTION:** An optional description identifying the catalog item
- **CATEGORY:** Select an existing category or enter a new one. When provisioning from the catalog, items can be filtered by category
- **LABELS:** A comma-separated list of descriptive strings which can be used to categorize the Catalog Item. The Catalog Items list view page can be filtered by labels to make them easier to find
- **ENABLED:** When checked, this catalog item will be available for provisioning
- **DISABLE AUTO PRICE:** When marked, price estimates are no longer refreshed with every field change at the time this Catalog Item is ordered. For some workloads with heavy Configuration-phase Tasks this may improve performance of the order page
- **FEATURED:** When checked, this catalog item will be given special visibility in the Service Catalog Persona view
- **ALLOW QUANTITY:** When checked, an additional field is added to the order page allowing users to order multiple copies. If this option is enabled, ensure your configuration supports that flexibility (such as when IP or load balancer details are specified)
- **VISIBILITY:** Set to private to keep the catalog item available only to users in the current Tenant. Master Tenant administrators may set catalog items to public to make them viewable and usable by Subtenant users
- **LOGO:** Select or upload a logo to be associated with this catalog item
- **DARK LOGO:** If desired, set an alternate logo for use when the dark theme is applied to the Morpheus appliance
- **CONFIGURE:** Click *CONFIGURE* to use the familiar App provisioning wizard to tie Blueprint and App deployment configuration to the Catalog Item
- **APP SPEC:** Inject App spec here for any fields required to provision an App from your Blueprint. You may also inject any overrides to the existing Blueprint spec that are desired. App Spec configuration must be YAML, a simple example that names the App and sets the Group and Cloud is included below:

```
#Example App Spec

name: '<%= customOption.appName %>'
group:
  name: Dev Group
environment: Dev
tiers:
  Web:
    instances:
      - instance:
          type: nginx
          cloud: Dev AWS
  App:
    instances:
      - instance:
          type: apache
          cloud: Dev AWS
```

- **CONTENT:** Optionally include documentation content for this Catalog Item. Markdown-formatted text is accepted and displayed appropriately when the item is ordered from the Service Catalog. A new Catalog Item-type Wiki entry will also be added containing this information.

Note: App spec custom option variables should be single quoted in YAML: `ccloud: '<%= customOption.ccloud %>'`. Additionally, not all variables are available here as many are unknown until provisioning. Users may use any custom Input values (customOption) as well as name or hostname values which are resolved as part of naming policy evaluation.

- **FORM TYPE:** Choose whether a Form or individual Inputs should be used to set the user options at provisioning time. See [Forms documentation](#) for complete details on creating Forms.
- **FORM:** If the Form Type of “Form” is selected, choose the pre-existing Form which should be associated with the Catalog Item
- **INPUTS:** If the Form Type of “Inputs” is selected, choose the pre-existing Inputs which should be associated with the Catalog Item

Tip: There are a number of advantages to using Forms over Inputs. See the section above on using Forms with Catalog Items for a complete description on how they are used and the advantages to using them.

Once done, click *SAVE CHANGES*

Building Catalog Workflows

Note: API usage for Catalog Items with Forms is not yet supported. If planning to use the API or CLI to order Catalog Items you should not use Forms on Catalog Items until it is supported.

From the Catalog Items List Page (Library > Blueprints > Catalog Items), click *ADD*. From the dropdown menu, select Workflow. The modal window will appear to configure and add a new catalog Workflow.

Configure the following:

- **NAME:** A friendly name for the catalog item in Morpheus
- **CODE:** An optional shortcode for coded naming schemes or programmatic reference
- **DESCRIPTION:** An optional description identifying the catalog item
- **CATEGORY:** Select an existing category or enter a new one. When provisioning from the catalog, items can be filtered by category
- **LABELS:** A comma-separated list of descriptive strings which can be used to categorize the Catalog Item. The Catalog Items list view page can be filtered by labels to make them easier to find
- **ENABLED:** When checked, this Workflow item will be available for selection in the Service Catalog
- **DISABLE AUTO PRICE:** When marked, price estimates are no longer refreshed with every field change at the time this Catalog Item is ordered. For some workloads with heavy Configuration-phase Tasks this may improve performance of the order page
- **FEATURED:** When checked, this catalog item will be given special visibility in the Service Catalog Persona view

- **VISIBILITY:** Set to private to keep the catalog item available only to users in the current Tenant. Master Tenant administrators may set catalog items to public to make them viewable and usable by Subtenant users
- **LOGO:** Select or upload a logo to be associated with this catalog item
- **DARK LOGO:** If desired, set an alternate logo for use when the dark theme is applied to the Morpheus appliance
- **WORKFLOW:** Select an existing Workflow to be associated with this Catalog Item, new Workflows are created in Library > Automation
- **CONTEXT:** Optionally restrict users to a specific target context, Instance, Server, or None
- **CONFIG:** Enter an optional custom config JSON body. See [Workflows documentation](#) for a formatting example
- **CONTENT:** Optionally include documentation content for this Catalog Item. Markdown-formatted text is accepted and displayed appropriately when the item is ordered from the Service Catalog. A new Catalog Item-type Wiki entry will also be added containing this information.
- **FORM TYPE:** Choose whether a Form or individual Inputs should be used to set the user options at provisioning time. See [Forms documentation](#) for complete details on creating Forms.
- **FORM:** If the Form Type of “Form” is selected, choose the pre-existing Form which should be associated with the Catalog Item
- **INPUTS:** If the Form Type of “Inputs” is selected, choose the pre-existing Inputs which should be associated with the Catalog Item

Tip: There are a number of advantages to using Forms over Inputs. See the section above on using Forms with Catalog Items for a complete description on how they are used and the advantages to using them.

Once done, click *SAVE CHANGES*

Editing and Deleting from the Self Service Catalog

Once created, Service Catalog items can be edited or deleted from the Catalog Items list view (Library > Blueprints > Catalog Items). Click the pencil icon in the relevant row to edit the Service Catalog item or the trash can icon to delete it. Alternatively, Service Catalog items can be made inactive to remove them as provisioning options rather than deleting them.

Cluster Layouts

Morpheus includes many different types of Cluster Layouts out-of-the-box which support a number of provisioning technologies. Many of these Morpheus-provided Cluster Layouts may also be cloned for use in creating custom layouts. Edit or delete user-created Cluster Layouts using the pencil () or trash can () icons next to the desired layout on the Cluster Layouts list page. See the next section for an

TYPE	NAME	VERSION	DESCRIPTION
	Alibaba Docker Host	18.04	This will provision a single docker host vm in alibaba
	Amazon Docker Host	16.04	This will provision a single docker host vm in amazon
	Azure Docker Host	16.04	This will provision a single docker host vm in azure
	Copy of MKS Kubernetes 1.20 Cluster on Ubuntu 18.04	16.04	provision a kubernetes 1.20 cluster on ubuntu 18.04

Note: Morpheus now syncs available (non-preview) AKS k8s versions daily. Existing synced versions that are no longer supported by Azure are automatically disabled. The table below includes available AKS versions at time of v7.0.0 release.

- **List of Default System Cluster Layouts**

NAME	CODE
Kubernetes 1.20.7 Cluster AKS	kubernetes-azure-aks-1.20.7
Kubernetes 1.20.5 Cluster AKS	kubernetes-azure-aks-1.20.5
Kubernetes 1.19.11 Cluster AKS	kubernetes-azure-aks-1.19.11
Kubernetes 1.19.9 Cluster AKS	kubernetes-azure-aks-1.19.9
Kubernetes 1.18.19 Cluster AKS	kubernetes-azure-aks-1.18.19
Kubernetes 1.18.17 Cluster AKS	kubernetes-azure-aks-1.18.17
MKS Kubernetes 1.20 Cluster on Ubuntu 18.04	kubernetes-1.20.2-ubuntu-18.04.5-xen-amd64-single
MKS Kubernetes 1.20 Cluster on Ubuntu 18.04	kubernetes-1.20.2-ubuntu-18.04.5-scvm-amd64-single
MKS Kubernetes 1.20 Cluster on Ubuntu 18.04	kubernetes-1.20.2-ubuntu-18.04.5-opentelekom-amd64-single
MKS Kubernetes 1.20 Cluster on Ubuntu 18.04	kubernetes-1.20.2-ubuntu-18.04.5-nutanix-amd64-single
MKS Kubernetes 1.20 Cluster on Ubuntu 18.04	kubernetes-1.20.2-ubuntu-18.04.5-morpheus-amd64-single
External Kubernetes 1.20	kubernetes-external-1.20
MKS Kubernetes 1.20 Cluster on Ubuntu 18.04	kubernetes-1.20.2-ubuntu-18.04.5-hyperv-amd64-single
MKS Kubernetes 1.20 Cluster on Ubuntu 18.04	kubernetes-1.20.2-ubuntu-18.04.5-huawei-amd64-single
MKS Kubernetes 1.20 Cluster on Ubuntu 18.04	kubernetes-1.20.2-ubuntu-18.04.5-openstack-amd64-single
MKS Kubernetes 1.20 Cluster on Ubuntu 18.04	kubernetes-1.20.2-ubuntu-18.04.5-google-amd64-single
MKS Kubernetes 1.20 Cluster on Ubuntu 18.04	kubernetes-1.20.2-ubuntu-18.04.5-amazon-amd64-single
Copy of MKS Kubernetes 1.20 Cluster on Ubuntu 18.04	ffc21dc1-feb7-4c02-ae90-204fd080c23d
Docker Cluster on Ubuntu 18.04	docker-ubuntu-18.04.2-fusion-amd64-single
MKS Kubernetes 1.20 Cluster on Ubuntu 18.04	kubernetes-1.20.2-ubuntu-18.04.5-vmware-amd64-single
MKS Kubernetes 1.20 Cluster on Ubuntu 18.04	kubernetes-1.20.2-ubuntu-18.04.5-fusion-amd64-single
Kubernetes 1.17 Cluster EKS	kubernetes-amazon-eks-1.17
Kubernetes 1.17 HA Cluster on Ubuntu 16.04, Weave, OpenEBS	kubernetes-1.17-xen-ubuntu-16.04-cluster-weave-openebs
Kubernetes 1.17 Cluster on Ubuntu 16.04, Weave, OpenEBS	kubernetes-xen-1.17-ubuntu-16.04-single
Kubernetes 1.17 HA Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-scvm-ubuntu-18.04-cluster-weave-openebs

Table 15 – continued from previous page

NAME	CODE
Kubernetes 1.17 Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-scvm-ubuntu-18.04-single
Kubernetes 1.17 HA Cluster on Ubuntu 16.04, Weave, OpenEBS	kubernetes-openstack-ubuntu-18.04-cluster-weave-openebs
Kubernetes 1.17 Cluster on Ubuntu 16.04, Weave, OpenEBS	kubernetes-openstack-ubuntu-18.04-single
Kubernetes 1.17 Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-opentelekom-ubuntu-18.04-single
Kubernetes 1.17 HA Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-nutanix-ubuntu-18.04-cluster-weave-openebs
Kubernetes 1.17 HA Cluster on Linux, Weave, OpenEBS	kubernetes-1.17-morpheus-linux-cluster-weave-openebs
Kubernetes 1.17 Cluster on Linux, Weave, OpenEBS	kubernetes-1.17-morpheus-linux-single
Kubernetes 1.17 HA Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-hyperv-ubuntu-18.04-cluster-weave-openebs
Kubernetes 1.17 Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-hyperv-ubuntu-18.04-single
Kubernetes 1.17 HA Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-huawei-ubuntu-18.04-cluster-weave-openebs
Kubernetes 1.17 Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-huawei-ubuntu-18.04-single
Kubernetes 1.17 HA Cluster on Ubuntu 16.04, Weave, OpenEBS	kubernetes-google-ubuntu-18.04-cluster-weave-openebs
Kubernetes 1.17 Cluster on Ubuntu 16.04, Weave, OpenEBS	kubernetes-google-ubuntu-18.04-single
Kubernetes 1.17 HA Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-esxi-ubuntu-18.04-cluster-weave-openebs
Kubernetes 1.17 Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-esxi-ubuntu-18.04-single
Kubernetes 1.17 HA Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-digitalOcean-ubuntu-18.04-cluster-weave-openebs
Kubernetes 1.17 Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-digitalOcean-ubuntu-18.04-single
Kubernetes 1.17 HA Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-azure-ubuntu-18.04-cluster-weave-openebs
Kubernetes 1.17 Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-azure-ubuntu-18.04-single
Kubernetes 1.17 HA Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-amazon-ubuntu-18.04-cluster-weave-openebs
Kubernetes 1.17 Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-amazon-ubuntu-18.04-single
Kubernetes 1.17 Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-alibaba-ubuntu-18.04-single
Kubernetes 1.17 HA Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-vmware-ubuntu-18.04-cluster-weave-openebs
Kubernetes 1.17 Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-vmware-ubuntu-18.04-single
Copy of Kubernetes 1.14 Cluster on Ubuntu 16.04, Weave, OpenEBS	6441b891-a61d-4f0b-a7ff-19c81d2ffd51
Kubernetes 1.17 Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-1.17-fusion-ubuntu-18.04-single
Kubernetes 1.16 Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-1.16-fusion-ubuntu-18.04-single
Kubernetes 1.15 Cluster on Ubuntu 18.04, Weave, OpenEBS	kubernetes-1.15-fusion-ubuntu-18.04-single
External Kubernetes 1.17 Cluster	kubernetes-external-1.17
External Kubernetes 1.16 Cluster	kubernetes-external-1.16
External Kubernetes 1.15 Cluster	kubernetes-external-1.15
External Kubernetes 1.14 Cluster	kubernetes-external-1.14
KVM on Ubuntu 16.04	kvm-vmware-ubuntu-16.04-single
Morpheus KVM and Container Cluster	morpheus-kvm-combo-cluster
VMware Docker CentOS 7.5	docker-vmware-centos-7.5-single
Oracle Cloud Docker Host	docker-oraclecloud-ubuntu-16.04-single
Morpheus Kubernetes Manual Cluster	morpheus-kubernetes-manual-cluster
Alibaba Docker Host	docker-alibaba-ubuntu-16.04-single
SCVMM Docker Host	docker-scvm-ubuntu-16.04-single
KVM on Ubuntu 16.04	kvm-fusion-ubuntu-16.04-single
UpCloud Docker Host	docker-upcloud-ubuntu-16.04-single
Morpheus KVM Ubuntu Cluster	morpheus-kvm-ubuntu-cluster
Morpheus KVM CentOS Cluster	morpheus-kvm-centos-cluster
Azure Docker Host	docker-azure-ubuntu-16.04-single
KVM on CentOS 7.5	kvm-vmware-centos-7.5-single
KVM on CentOS 7.5	kvm-fusion-centos-7.5-single
Kubernetes 1.14 HA Cluster on Ubuntu 16.04, Weave, OpenEBS	kubernetes-blumix-ubuntu-16.04-cluster-weave-openebs
Kubernetes 1.14 Cluster on Ubuntu 16.04, Weave, OpenEBS	kubernetes-blumix-ubuntu-16.04-single
Kubernetes 1.17 HA Cluster on Ubuntu 16.04, Weave, OpenEBS	kubernetes-vcd-ubuntu-16.04-cluster-weave-openebs

Table 15 – continued from previous pag

NAME	CODE
Kubernetes 1.17 Cluster on Ubuntu 16.04, Weave, OpenEBS	kubernetes-vcd-ubuntu-16.04-single
VCD Docker Host	docker-vcd-ubuntu-16.04-single
Kubernetes 1.14 HA Cluster on Ubuntu 16.04, Weave, OpenEBS	kubernetes-softlayer-ubuntu-16.04-cluster-weave-openebs
Kubernetes 1.14 Cluster on Ubuntu 16.04, Weave, OpenEBS	kubernetes-softlayer-ubuntu-16.04-single
SoftLayer Docker Host	docker-softlayer-ubuntu-16.04-single
Open Telekom Docker Host	docker-opentelekom-ubuntu-16.04-single
Huawei Docker Host	docker-huawei-ubuntu-16.04-single
Google Docker Host	docker-google-ubuntu-16.04-single
ESXi Docker Host	docker-esxi-ubuntu-16.04-single
IBM Docker Host	docker-bluemix-ubuntu-16.04-single
Xen Docker Host	docker-xen-ubuntu-16.04-single
Digital Ocean Docker Host	docker-digitalOcean-ubuntu-16.04-single
Hyper-V Docker Host	docker-hyperv-ubuntu-16.04-single
Docker on Linux	manual-linux-docker-morpheus-single
Kubernetes Cluster 1.14 on Ubuntu 16.04, Weave, OpenEBS	kubernetes-morpheus-ubuntu-16.04-cluster-weave-openebs
Kubernetes 1.14 on Ubuntu 16.04, Weave, OpenEBS	kubernetes-morpheus-ubuntu-16.04-single
Docker on Bare Metal	docker-morpheus-metal-ubuntu-16.04-single
Docker on Ubuntu 16.04	docker-morpheus-ubuntu-16.04-single
Amazon Docker Host	docker-amazon-ubuntu-16.04-single
OpenStack Docker Host	docker-openstack-ubuntu-16.04-single
Nutanix Docker Ubuntu 16.04	docker-nutanix-ubuntu-16.04-single
VMware Docker Ubuntu 16.04	docker-vmware-ubuntu-16.04-single
Kubernetes 1.14 HA Cluster on Ubuntu 16.04, Weave, OpenEBS	kubernetes-fusion-ubuntu-16.04-cluster-weave-openebs
Kubernetes 1.14 Cluster on Ubuntu 16.04, Weave, OpenEBS	kubernetes-fusion-ubuntu-16.04-single
Docker on Ubuntu 16.04	docker-fusion-ubuntu-16.04-single

Creating Custom Cluster Layouts

Cluster Layouts can be built for an infinite number of use cases and targeting a wide range of configurations or provisioning technologies. However, it still may help to see an example Cluster Layout created. In this section, we'll create an example Cluster Layout which provisions a Kubernetes Cluster to a VMware vCenter Cloud containing a master node and three worker nodes. We'll build out two Node Types to embed in the Cluster, one master and one worker. Within the Node Types, we will embed Script Templates needed to configure the nodes as they are provisioned and allow the workers to join the cluster. Script Templates can be configured to execute at various phases of the lifecycle of the node, similar to a Provisioning Workflow, so Morpheus can orchestrate the whole process.

Creating Script Templates

We'll first start by creating the necessary Script Templates. In this example, I'll use a generic prep script that both the master and worker nodes will utilize and then I'll create three additional scripts (two for the master node and one for the worker nodes) to accomplish various Kubernetes cluster setup tasks (`kubeadm init`, creating Role Bindings, joining workers to the cluster, etc.). I'll briefly describe each here and step through the process of creating the Script Template objects in Morpheus.

To begin a new Script Template, navigate to Library > Templates > Script Templates and click + *ADD*. All of the scripts used in this example will be "Bash" type, run as user "root" and with SUDO marked.

The first script to add will be a prep script that both the master and worker nodes will use. This script turns swap off, installs `containerd`, runs updates, sets network config, and installs `kubelet`, `kubeadm`, and `kubect1`. Set this script to run in the PreProvision phase. The complete script, named "k8sprep" in my example can be viewed below:

• k8sprep Script Template

```

function wait_for_dpkg() {
    FRONT_END_LOCK_FILE=/var/lib/dpkg/lock-frontent
    DPKG_LOCK_FILE=/var/lib/apt/lists/lock

    echo "checking for dpkg lock existence..."
    if [ -e "$DPKG_LOCK_FILE" ]; then
        while fuser $DPKG_LOCK_FILE >/dev/null 2>&1 ; do
            echo "Waiting for lock $DPKG_LOCK_FILE..."
            sleep 1
        done
    fi
    echo "checking for dpkg lock-frontent existence..."
    if [ -e "$FRONT_END_LOCK_FILE" ]; then
        while fuser $FRONT_END_LOCK_FILE >/dev/null 2>&1 ; do
            echo "Waiting for lock $FRONT_END_LOCK_FILE..."
            sleep 1
        done
        echo "force removing lock $FRONT_END_LOCK_FILE"
        sudo rm $FRONT_END_LOCK_FILE
        sleep 5
    fi
}

# Swap must be turned off see https://kubernetes.io/docs/setup/production-
↪environment/tools/kubeadm/install-kubeadm/
sudo swapoff -a ; sudo sed -i '/ swap / s/^#/' /etc/fstab
# Install containerd packages from Debian sse: https://docs.docker.com/engine/
↪install/ubuntu/
wait_for_dpkg
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg lsb-release -y
sudo mkdir -m 0755 -p /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -
↪o /etc/apt/keyrings/docker.gpg
sudo mkdir -m 0755 -p /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -
↪o /etc/apt/keyrings/docker.gpgsudo
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.
↪gpg] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/
↪null
sudo apt-get update
# the Docker documentation advises to install the whole Docker runtime,
↪environment but containerd.io is sufficient
sudo apt-get install containerd.io -y
# Prepare the necessary network config see: https://kubernetes.io/docs/setup/
↪production-environment/container-runtimes/
sudo cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF
sudo modprobe overlay

```

(continues on next page)

(continued from previous page)

```

sudo modprobe br_netfilter
sudo cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1
EOF
# Apply sysctl params without reboot
sudo sysctl --system
# Install kubeadm following the K8s documentation: https://kubernetes.io/docs/
↳setup/production-environment/tools/kubeadm/install-kubeadm/
sudo apt-get install -y apt-transport-https ca-certificates curl
sudo curl -fsSL https://packages.cloud.google.com/apt/doc/apt-key.gpg|sudo gpg --
↳dearmor -o /etc/apt/trusted.gpg.d/k8s.gpg
sudo echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/
↳apt/sources.list.d/kubernetes.list
sudo apt-get update
sudo apt-get -y install -y kubelet=1.26.1-00 kubeadm=1.26.1-00 kubectl=1.26.1-00

sudo apt-mark hold kubelet kubeadm kubectl
# see https://github.com/etcd-io/etcd/issues/13670
cat << EOF | sudo tee /etc/containerd/config.toml
version = 2
[plugins]
[plugins."io.containerd.grpc.v1.cri"]
[plugins."io.containerd.grpc.v1.cri".containerd]
[plugins."io.containerd.grpc.v1.cri".containerd.runtimes]
[plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc]
runtime_type = "io.containerd.runc.v2"
[plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc.options]
SystemdCgroup = true
EOF
sudo systemctl restart containerd

```

Next we'll add a script called "kubeadm-init" in this example which will create and configure some working directories, set the kubeadm config yaml, and run kubeadm init. Set this script to run in the PreProvision phase as well. The complete script can be viewed below:

- **kubeadm-init Script Template**

```

mkdir -p <%=morpheus.morpheusHome%>/kube
mkdir -p <%=morpheus.morpheusHome%>/kube/working
mkdir -p <%=morpheus.morpheusHome%>/k8s
sudo chown <%=morpheus.morpheusUser%>:<%=morpheus.morpheusUser%> <%=morpheus.
↳morpheusHome%>/kube
sudo chown <%=morpheus.morpheusUser%>:<%=morpheus.morpheusUser%> <%=morpheus.
↳morpheusHome%>/kube/working
cat <<EOF | sudo tee <%=morpheus.morpheusHome%>/kube/working/kubeadm-config.yaml
# kubeadm-config.yaml
kind: ClusterConfiguration
apiVersion: kubeadm.k8s.io/v1beta3
kubernetesVersion: v1.26.1
networking:
  serviceSubnet: "10.96.0.0/16"

```

(continues on next page)

(continued from previous page)

```

    podSubnet: "10.244.0.0/24"
    dnsDomain: "cluster.local"
apiServer:
  extraArgs:
    authorization-mode: "Node,RBAC"
clusterName: "example-cluster"
---
kind: KubeletConfiguration
apiVersion: kubelet.config.k8s.io/v1beta1
cgroupDriver: systemd
EOF
sudo kubeadm init --config <%=morpheus.morpheusHome%>/kube/working/kubeadm-config.
↪yaml
sudo cp -i /etc/kubernetes/admin.conf <%=morpheus.morpheusHome%>/.kube/config &&
sudo chown <%=morpheus.morpheusUser%>:<%=morpheus.morpheusUser%> <%=morpheus.
↪morpheusHome%>/.kube/config

```

Lastly, we'll add a setup script for the Kubernetes master node called "k8s-master-setup" for this example. This script creates the service account and role bindings. Set this script to run in the PostProvision phase and view the complete script below:

- **k8s-master-setup Script Template**

```

#create a service account
cd <%=morpheus.morpheusHome%>
#kubectl -n kube-system create sa morpheus
kubectl create sa morpheus
cat <<EOF | tee <%=morpheus.morpheusHome%>/kube/morpheus-sa.yaml
apiVersion: v1
kind: Secret
metadata:
  name: morpheus-token
  annotations:
    kubernetes.io/service-account.name: morpheus
type: kubernetes.io/service-account-token
EOF
kubectl create clusterrolebinding serviceaccounts-cluster-admin --
↪clusterrole=cluster-admin --group=system:serviceaccounts
kubectl create -f <%=morpheus.morpheusHome%>/kube/morpheus-sa.yaml
kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/
↪manifests/calico.yaml

```

The three scripts we've just created will prepare the master node for our cluster. We now need an additional script that will prepare the worker nodes and join them to the cluster. The worker nodes will also use the "k8s-prep" script we added in the very first step which you'll see in the next step when we create the node types.

Add a fourth script which I've called "k8s-join" for this example. This script takes advantage of a special `joinCommand` variable as you'll see when you view the full script. If you want to see exactly what this does later, you can create a new Bash script Task that echoes out that variable and run it against an existing Kubernetes worker node VM (echo "`<%=morpheus.kubernetes.joinCommand%>`"). Set this script to run in the PostProvision phase and view the full script below:

- **k8s-join Script Template**

```
sudo <%=morpheus.kubernetes.joinCommand%>
```

This completes the needed Script Templates which we will set on two new Node Types in the next step. Continue on to the next section.

Creating Node Types

With the Script Templates created we now need to make two new Node Types, one for the master node and one for the worker nodes. In this example case, we don't need to add a new Virtual Image because we can use one of the preinstalled system images for Ubuntu on VMware vCenter which will work fine. Node Types are created in Library > Blueprints > Node Types. Once there, click + *ADD*. Within the modal for the new Node Type, configure the following:

- **NAME:** Provide a name for the Node Type, can be anything to denote this is the master node
- **SHORT NAME:** A shortened version of the name without any spaces
- **VERSION:** The version number you wish to apply for this particular Node Type which is useful if you iterate on your Node Types at any point
- **TECHNOLOGY:** For this example case, VMware. Once set, additional options will appear
- **VM IMAGE:** A pre-installed Ubuntu system image will work
- **SCRIPTS:** Using the typeahead field, set the “k8sprep”, “kubeadm-init” and “k8s-master-setup” Script Templates (you may have called them something different)

Finally, click *SAVE CHANGES*.

Repeat the process to create a second Node Type. The second time around, use the following configurations:

- **NAME:** Provide a name for the Node Type, can be anything to denote this is the worker node
- **SHORT NAME:** A shortened version of the name without any spaces
- **VERSION:** The version number you wish to apply for this particular Node Type which is useful if you iterate on your Node Types at any point
- **TECHNOLOGY:** For this example case, VMware. Once set, additional options will appear
- **VM IMAGE:** A pre-installed Ubuntu system image will work
- **SCRIPTS:** Using the typeahead field, set the “k8sprep” and “k8s-join” Script Templates (you may have called them something different)

Once done, click *SAVE CHANGES* to save the second Node Type. With the pieces in place, we are now ready to create the Cluster Layout object itself. Continue on to the next section.

Creating a Cluster Layout

At this point we can create the Cluster Layout object in Morpheus and attach the Node Types we've just created (which themselves have our Script Templates applied). Cluster Layouts are created in Library > Blueprints > Cluster Layouts. Click + *ADD* and configure the following:

- **NAME:** Provide a name for the Cluster Layout, can be anything to denote this is a Kubernetes cluster
- **VERSION:** The version number you wish to apply to this Cluster Layout which is useful if you later iterate on this Cluster Layout
- **CLUSTER TYPE:** Kubernetes Cluster
- **TECHNOLOGY:** VMware

- **MINIMUM MEMORY:** 4096 MB
- **INPUTS:** Use the typeahead field to add configured Inputs to the Cluster Layout
- **MASTER NODES:** Use the typeahead field to find the Kubernetes master Node Type we just created
- **WORKER NODES:** Use the typeahead field to find the Kubernetes worker Node Type we just created. Set the “Count” value to three (3) since we wish to have three worker nodes in this cluster
- **CLUSTER PACKAGES:** Use the typeahead field to add configured Cluster Packages to the Cluster Layout. Cluster Packages are created in the [Templates section](#)
- **SPEC TEMPLATES:** Use the typeahead field to add Spec Templates to the Cluster Layout

Click *SAVE CHANGES* to save the Cluster Layout.

Testing and Wrap-Up

At this point we are finished and we have a viable Kubernetes cluster that we can deploy with just one click. To add a new managed cluster to this Morpheus environment, navigate to Infrastructure > Clusters. To provision a cluster from the layout we just created, click + *ADD CLUSTER* and select “Kubernetes Cluster” from the dropdown menu. Make the appropriate selections to target the new cluster to an existing VMware vCenter Cloud and complete the wizard. Once complete, the new cluster will be visible on your clusters list page and from there you can drill into the detail page to view relevant details about the cluster including monitoring metrics, current workloads, and more.

5.4.4 Virtual Images

Overview

The Virtual Image section displays a list of all images, local and synced, that are available to deploy. Morpheus includes a rich catalog of pre-configured System Images available for every cloud type. User Images are automatically synced from Cloud Integrations and added to the Virtual Images section. Images can also be uploaded directly into Morpheus via local file or url. Amazon and Azure Marketplace images can also be added to the Virtual Images Section.

Understanding the process of prepping images for consumption in Morpheus is a very important step toward building an effective Morpheus environment. In addition to the information contained in this section on Virtual Images, it may be helpful to see a complete image prep example walkthrough. Our getting started guide for Morpheus and VMware includes a [section on preparing images](#) that may provide a helpful example.

Tip: Morpheus includes a wide catalog of system image types as examples to show how the product can be used and to give users a starting point for implementing their own library. The included images are not intended to be production-ready images. Morpheus always recommends its users create their own gold images which meet their required specifications.

Important: Invalid Image Settings cause provisioning failures. Morpheus syncs in as much meta-data as possible for synced images, but additional configuration may be needed to ensure successful provisioning.

Warning: Cloud-init is enabled by default for all Linux images. If your Linux image does not have Cloud-init installed, *Cloud-init Enabled* must be unchecked before provisioning the image or it will fail immediately.

Image Types

Morpheus provides a vast *System Image* repo with pre-configured images for every Cloud. All other images are *User Images*. User images can be added directly to Morpheus, or automatically synced from integrated clouds. It is important to configure synced User Images for metadata, including specifying the Platform and User Credentials, prior to provisioning. Provisioning a User Image that has not been configured may result in failed provisioning.

Important: Synced User Images need to be configured prior to provisioning.

Configuring Virtual Images

System Images

System Virtual Images are pre-configured with metadata and have Cloud-Init or Cloudbase-Init installed. These images are ready to be provisioned with no configuration necessary, however it is required to populate Administration > Settings > Provisioning, Cloud-Init section, with user data as well as User Profile(s) users data when creating additional users prior to provisioning, as the user data from these sections is required when provisioning System provided Virtual Images.

Note: System Images settings are not editable.

User Images

Typically Morpheus does not have sufficient metadata to successfully provision synced User Images. After integrating clouds and User Images have synced, it is highly recommended to configure the images prior to provisioning.

To edit and configure an existing Virtual Image:

1. Select the pencil icon at the right of any row on the Virtual Images list page, or click *EDIT* on a Virtual Image detail page.
2. Configure the following on the Image:

Name Name of the Virtual Image in Morpheus. This can be changed from the name of the image, but editing will not change the name of the actual image

Operating System Specifies the platform and OS of the image. All Windows images will need to have the operating system specified on the Virtual Image, as Morpheus will assign Linux as the platform for all images without an operating system specified

Minimum Memory The Minimum Memory setting will filter available Service Plan options during provisioning. Service Plans that do not meet the minimum value set on the Virtual Image will not be provided as Service Plan choices

Cloud Init Enabled? On by default, uncheck for any Image that does not have Cloud-Init or Cloudbase-Init installed

Install Agent? On by default, uncheck to skip Agent install. Note this will result in the loss of utilization statistics, logs, script execution, and monitoring. (Some utilization stats are still collected for Agent-less hosts and VMs depending on the cloud)

Username Existing username on the image. This is required for authentication, unless Morpheus is able to add user data, Cloud-Init, Cloudbase-Init or Guest Customizations. If Cloud-Init, Cloudbase-Init or Guest

Customizations are used, credentials are defined in Administration > Settings > Provisioning and User Settings. If credentials are defined on the image and Cloud-Init is enabled, Morpheus will add that user during provisioning, so ensure that user does not already exist on the image (such as `root`). For Windows Guest Customizations, Morpheus will set the Administrator password to what is defined on the image if Administrator user is defined. Do not define any other user than Administrator for Windows Images unless using Cloudbase-init. Morpheus recommends running Guest Customizations for all Windows Images, which is required when joining Domains as the SID will change

Password Password for the user on the image if username is populated

Bucket Location where the Virtual Image will be stored. Default Virtual Image Storage location is `/var/opt/morpheus/morpheus-ui/vms`. Additional Storage Providers can be configured in Infrastructure > Storage

Cloud-Init User Data Accepts what would go in `runcmd` and can assume Bash syntax. Example use: Script to configure satellite registration at provision time

Create Image ID Select FILE to browse locally for an image or drop an image file into the dropzone. Alternatively, select URL to download the image from an accessible URL. It is recommend to configure the rest of the settings below prior to uploading the source Image File(s)

Permissions Set Tenant permissions in a multi-tenant Morpheus environment. Select private visibility and select specific Tenants to which the Virtual Image will be made available. Select public visibility to share the Virtual Image with all Tenants

Auto Join Domain? Enable to have Instances provisioned with this image auto-join configured domains (Windows only, domain controller must be configured in Infrastructure > Network and the configured domain set on the provisioned to Cloud or Network)

VirtIO Drivers Loaded? Enable if VirtIO Drivers are installed on the image for provisioning to KVM-based hypervisors

FIPS Compliant Image? When selected, Morpheus will install the FIPS-compliant Morpheus Agent package

VM Tools Installed? On by default, uncheck if VMware Tools (including OpenVMTools) are not installed on the Virtual Image. Morpheus will skip network wait during provisioning when deselected

Force Guest Customization? VMware only, forces guest customizations to run during provisioning, typically when provisioning to a DHCP network where guest customizations would not run by default. This options requires that VMware Tools is installed on the image

Trial Version Enable to automatically re-arm the expiration on Windows Trial Images during provisioning

Enabled Sysprep? Applicable to multiple Clouds, including VMware vCenter, SCVMM, Nutanix, Hyper-V, KVM, and Google GCP. Enable if the Windows Image has been sysprepped. If enabled, Morpheus will inject `unattend.xml`

3. Click *Save Changes*

Note: Cloud-Init is enabled by default on all images. Images without Cloud-Init or Cloudbase-Init installed must have the `cloud-init` flag disabled on the Virtual Image setting or Provisioning may fail.

Provisioning Images

When provisioning a system image, Morpheus will stream the image from Amazon S3 to the target Cloud if the image is not local to the Cloud.

When using images that already exist in the destination Cloud, such as synced, marketplace, or previously copied images, no image stream from S3 through the Morpheus Appliance to the destination cloud will take place.

Note: The Morpheus Appliance must be able to download from Amazon S3 when provisioning system images.

Note: The Morpheus Appliance must be able reach and resolve the destination Host when provisioning System Images or uploaded Images for the first time. This included being able to resolve ESXi host names in VMware vCenter clouds, and reach the destination ESXi host over port 443.

Add Virtual Image

Virtual Images can be upload to Morpheus from local files or URL's. Amazon and Azure Marketplace metadata can also be added to the Virtual Images library, enabling the creation of custom catalog Instance Type from Marketplace images (no image is transferred to Morpheus when adding Marketplace images).

Warning: Be conscious of your Storage Provider selection. The default Storage Provider is the Morpheus Appliance at `/var/opt/morpheus/morpheus-ui/vms`. Uploading large images to the Morpheus Appliance when there is inadequate space will cause upload failures and impact Appliance functionality. Ensure there is adequate space on your selected Storage Provider. Additional Storage Provider can be added at *Infrastructure > Storage*, which can be configured as the default Virtual Image Store or selected when uploading Images.

Note: VMware-type OVF Virtual Images do not support mounted ISO uploads

To Add Virtual Image:

1. Select + *Add* in the Virtual Images page.
2. Select Image format:
 - Alibaba
 - Amazon AMI
 - Azure Marketplace
 - Digital Ocean
 - ISO
 - PXE Boot
 - QCOW2
 - RAW
 - VHD
 - VMware (vmdk/ovf/ova)

3. Configure the following on the Virtual Image:

Name Name of the Virtual Image in Morpheus. This can be changed from the name of the image, but editing will not change the name of the actual image

Operating System Specifies the platform and OS of the image. All Windows images will need to have the operating system specified on the Virtual Image, as Morpheus will assign Linux as the platform for all images without an operating system specified

Minimum Memory The Minimum Memory setting will filter available Service Plan options during provisioning. Service Plans that do not meet the minimum value set on the Virtual Image will not be provided as Service Plan choices

Cloud Init Enabled? On by default, uncheck for any Image that does not have Cloud-Init or Cloudbase-Init installed

Install Agent? On by default, uncheck to skip Agent install. Note this will result in the loss of utilization statistics, logs, script execution, and monitoring. (Some utilization stats are still collected for Agent-less hosts and VMs depending on the cloud)

Username Existing username on the image. This is required for authentication, unless Morpheus is able to add user data, Cloud-Init, Cloudbase-Init or Guest Customizations. If Cloud-Init, Cloudbase-Init or Guest Customizations are used, credentials are defined in Administration > Settings > Provisioning and User Settings. If credentials are defined on the image and Cloud-Init is enabled, Morpheus will add that user during provisioning, so ensure that user does not already exist on the image (such as root). For Windows Guest Customizations, Morpheus will set the Administrator password to what is defined on the image if Administrator user is defined. Do not define any other user than Administrator for Windows Images unless using Cloudbase-init. Morpheus recommends running Guest Customizations for all Windows Images, which is required when joining Domains as the SID will change

Password Password for the user on the image if username is populated

Bucket Location where the Virtual Image will be stored. Default Virtual Image Storage location is `/var/opt/morpheus/morpheus-ui/vms`. Additional Storage Providers can be configured in Infrastructure > Storage

Cloud-Init User Data Accepts what would go in `runcmd` and can assume Bash syntax. Example use: Script to configure satellite registration at provision time

Create Image ID Select FILE to browse locally for an image or drop an image file into the dropzone. Alternatively, select URL to download the image from an accessible URL. It is recommend to configure the rest of the settings below prior to uploading the source Image File(s)

Permissions Set Tenant permissions in a multi-tenant Morpheus environment. Select private visibility and select specific Tenants to which the Virtual Image will be made available. Select public visibility to share the Virtual Image with all Tenants

Auto Join Domain? Enable to have Instances provisioned with this image auto-join configured domains (Windows only, domain controller must be configured in Infrastructure > Network and the configured domain set on the provisioned to Cloud or Network)

VirtIO Drivers Loaded? Enable if VirtIO Drivers are installed on the image for provisioning to KVM-based hypervisors

FIPS Compliant Image? When selected, Morpheus will install the FIPS-compliant Morpheus Agent package

VM Tools Installed? On by default, uncheck if VMware Tools (including OpenVMTools) are not installed on the Virtual Image. Morpheus will skip network wait during provisioning when deselected

Force Guest Customization? VMware only, forces guest customizations to run during provisioning, typically when provisioning to a DHCP network where guest customizations would not run by default. This options requires that VMware Tools is installed on the image

Trial Version Enable to automatically re-arm the expiration on Windows Trial Images during provisioning

Enabled Sysprep? Applicable to multiple Clouds, including VMware vCenter, SCVMM, Nutanix, Hyper-V, KVM, and Google GCP. Enable if the Windows Image has been sysprepped. If enabled, Morpheus will inject `unattend.xml`

Note: Default Storage location is `/var/opt/morpheus/morpheus-ui/vms`. Additional Storage Providers can be configured in *Infrastructure > Storage*. Ensure local folders are owned by `morpheus-app.morpheus-app` if used.

Warning: Provisioning will fail if *Cloud init Enabled* is checked and Cloud-Init is not installed on the Image.

Note: Existing Image credentials are required for Linux Images that are not Cloud-Init enabled and for Windows Images when Guest Customizations are not used. Cloud-Init and Windows user settings need to be configured in *Administration > Settings > Provisioning* when using Cloud-Init or Guest Customizations and new credentials are not set on the Virtual Image.

4. Upload Image

Images can be uploaded by File or URL:

File Drag and Drop the image file, or select *Add File* to select the image file.

Url Select the URL radio button, and enter URL of the Image.

Note: The Virtual Image configuration can be saved when using a URL and the upload will finish in the background. When selecting/drag and dropping a file, the image files must upload completely before saving the Virtual Image record or the Image will not be valid.

5. Save Changes.

VMware - VM Templates Copies

In a VMware environment, you may have a single VM template that you use across different vCenters. Uploading an image to Morpheus, mentioned in the Add Virtual Image section, is one method to solve this. Alternatively, an organization may decide to create a VM template in one vCenter and then transfer it to other vCenters, which then could be sync'd into Morpheus.

If all the vCenters are added as Clouds into Morpheus and the templates are named the same in each vCenter, they will be aggregated under a single virtual image in Morpheus. This means that as you deploy to the various vCenter Clouds in Morpheus using this virtual image, it will choose the correct VM template to use based on the Cloud deployed to.

This eliminates the need for creating multiple Node Types for each virtual image if the templates were named differently in each vCenter. This can reduce the overhead of maintaining multiple Node Types and reduces user selections. As well, this can reduce the cloning time of VMs by avoiding network transfers of images between geographic locations, ensuring the closest VM template is selected.

Morpheus supports VMware Content Libraries storing VM templates and syncing into Morpheus, the same as a template in a folder. Additionally, the Content Library can be used to house the same template in multiple libraries. If they have the same name, these templates will be aggregated under a single virtual image. If the Content Library is stored on a datastore that the target host/cluster has access to, it will use that library first, to reduce the cloning time. If the Content Library is not stored in a datastore accessible by the cluster/host, a copy of the VM template will be performed to the target host/cluster instead.

Note: VM templates are a **Data Center** level object. The same process above applies to a single VMware cloud with multiple logical data centers. It will not apply to clusters, as a template is not associated with a cluster, only when it is converted to a VM.

5.4.5 Options

Options are user-defined custom inputs that are used throughout Morpheus. Depending on the type of Input being created, users may need to create only an Input, or it might need to be paired with an Option List. Option Lists are static or dynamic lists of selections that would be associated with an Input if the user must make choices from a select list or typeahead field. This section discusses how to create different types of Inputs depending on the situation and powerful tools the user can take advantage of, such as dynamically filtered cascading inputs, logically generated Option Lists called from local or remote sources, and more.

Forms

Forms are designed to be used with [Catalog Items](#). Using the provided tools, build forms to provide the appropriate customization fields for endusers in the Provisioning Catalog. Catalog Items can also be built using Morpheus Inputs but Forms provide unique capabilities that open up additional functionality and simplify the process. Once a Form is created it is then available to be associated with a Catalog Item. This section discusses the creation of Forms, the available options, and how they can be used effectively to create Catalog Items.

Creating a New Form

To begin a new Form, navigate to Library > Options > Forms and click + *ADD*.

NEW FORM

new form

+ ADD INPUT

+ ADD FIELD GROUP

DETAILS

NAME
new form

CODE
form code

DESCRIPTION
add a short description for the form

CONTEXT

CANCEL SAVE

Adding Field Groups

Field Groups are logical groupings of Input fields on your Form. Especially with larger forms, they make the Form easier to read and use. You can even collapse down sections of less commonly needed Inputs to keep your Forms clean. By default a new Form has no Field Groups but one can be added by clicking “+ ADD FIELD GROUP” in the center of the NEW FORM modal (which can be seen in the screenshot above). Field Groups can accept the following options:

- **NAME:** The name and label for the Field Group
- **DESCRIPTION:** An optional description for the Field Group
- **COLLAPSIBLE:** When marked, the user can collapse (hide) or show the Field Group
- **COLLAPSIBLE BY DEFAULT:** If the Field Group is marked collapsible, this option will appear to make the Field Group shown (unchecked) or hidden (checked) by default
- **VISIBILITY FIELD:** Enter the fieldName of another Input and this Field Group will appear only when that Input has a value

The screenshot shows a 'NEW FORM' interface. At the top, there is a dark header with 'NEW FORM' and a close button. Below the header, the main area is titled 'new form' and contains a '+ ADD INPUT' button. A field group is visible, titled 'NEW FIELDGROUP', with a list icon on the left and edit/delete icons on the right. Below the field group, there is another '+ ADD INPUT' button and a '+ ADD FIELD GROUP' button. On the right side, a 'DETAILS' sidebar is open, showing the following fields: 'NAME' (containing 'new fieldgroup'), 'DESCRIPTION' (a large empty text area), 'COLLAPSIBLE' (checkbox), and 'VISIBILITY FIELD' (text input). Below the sidebar, there are 'CANCEL' and 'SAVE' buttons.

Adding Inputs

Inputs are the Form options themselves and can take various forms, such as text fields, checkboxes, select lists or other custom forms. Inputs can be added outside of any created Field Groups or they may be added within them. Click + *ADD INPUT* within the desired area (inside a Field Group or outside of them).

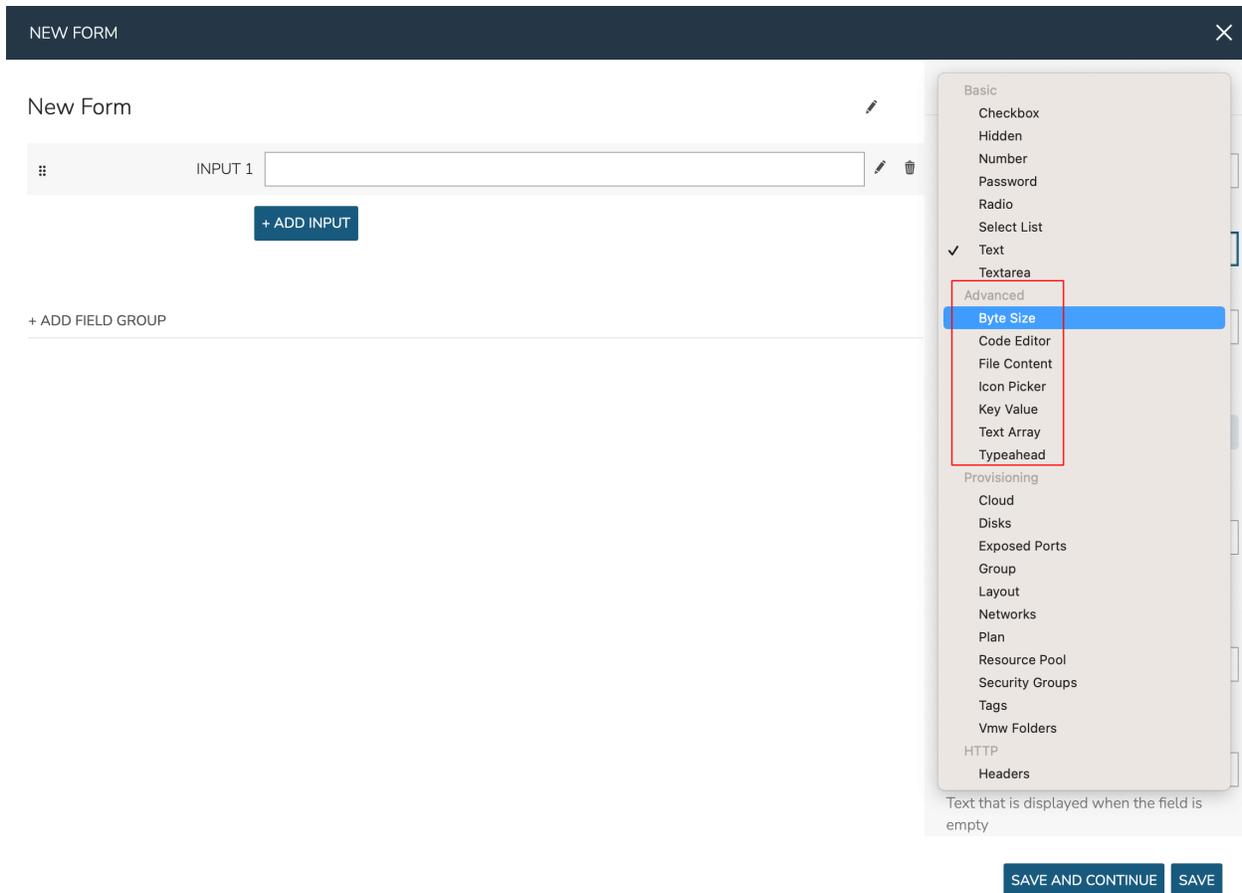
When adding Inputs, you may add any Input (Library > Options > Inputs) that is pre-existing to your form. Set “USE EXISTING” to “Yes” and make the selection from the “EXISTING OPTION TYPE” typeahead field. Once selected, you’ll notice that all other settings cannot be edited. If needed, edit this Input within the Inputs section (Library > Options > Inputs).

When adding Inputs, you may also create new Inputs within the Form builder. Some available types are no different than other Inputs you might make in the Input section (Library > Options > Inputs) but other types are unique to Forms. To create a new Input, set “USE EXISTING” to “No” and select the desired type.

Input types are organized by categories. Inputs in the basic category are primarily the same types of Inputs that can be made in the Inputs section (Library > Options > Inputs). These include text fields, select lists, checkbox arrays, numerical inputs, and more. You can read more about the basic Input types in Morpheus [Input documentation](#).

Inputs in the Advanced and Provisioning categories are unique to Forms and require additional explanation. Advanced inputs are similar to basic but provide some sort of data manipulation capability or have a specific targeted function. See the list of advanced Input types below:

- **Byte Size:** Allows numerical disk or storage size values to be given with selectable units (MB, GB, etc.). When a value is input and the unit is changed, the same value will be automatically computed into the new unit amount (1 GB > 1024 MB, for example). Users may select which unit is initially loaded by default.
- **Code Editor:** Gives the user a code editor field. Set a language or markdown format in the “HIGHLIGHTING” field to enable automatic syntax highlighting and spacing for the user. For example, the user could enter a provisioning shell script for their workload or provide a custom JSON payload at provision time (complete with syntax highlighting for easy entry).
- **File Content:** Access file content either locally entered or sourced from an integrated repository or outside URL
- **Icon Picker:** Allows the user to select an icon for their workload at provision time. The user may select from previously uploaded icons, upload their own, or use the built-in icon generator tool to create a unique icon right in the Form
- **Key Value:** Allows the user to enter as many key/value pairs as they’d like which can be onboarded into the workload config at provision time
- **Text Array:** Allows the user to enter multiple values separated by a delimiter of your choosing. Morpheus will parse out the entered values which can be individually deleted if desired before the form is submitted
- **Typeahead:** Similar to Select List, especially for very long lists. Search for the desired value by typing the first few letters as a search parameter. Users may also browse the complete list by clicking the dropdown icon. This Input type can also support multiple selections, if needed. Associate this type of Input with a pre-defined Option List or create a new Option List right inside the Form builder



Inputs in the provisioning category are specifically tied to some provisioning construct in Morpheus (Groups, Clouds, etc.). They're very useful for allowing users to select specific provisioning constructs, such as the Group, Cloud, Layout, or Network they wish to provision, and can automatically inject the user's selection into the Catalog Item. This makes for much simpler Catalog Item development as compared to setting up the configuration using just the Input construct outside of Forms.

Provisioning Inputs also include relevant reload and filtering behaviors by default. For example, your Cloud Input field will automatically reload after making a Group selection or your Layout Input field will automatically reload after setting a Group, Cloud, and Instance Type. This makes it very easy to create flexible Catalog Item forms that are useful across Clouds.

The following provisioning Input types are supported, each with their own automatic filtering behavior and auto-inject capability into the Catalog Item spec:

- Cloud
- Disks
- Exposed Ports
- Group
- Layout
- Networks
- Plan
- Resource Pool

- Security Groups
- Tags
- Vmw Folders

NEW FORM
✕

New Form ✎

⋮ INPUT 1 ✎ 🗑

+ ADD INPUT

+ ADD FIELD GROUP

Basic

- Checkbox
- Hidden
- Number
- Password
- Radio
- Select List
- ✓ Text
- Textarea

Advanced

- Byte Size
- Code Editor
- File Content
- Icon Picker
- Key Value
- Text Array
- Typeahead

Provisioning

- Cloud
- Disks
- Exposed Ports
- Group
- Layout
- Networks
- Plan
- Resource Pool
- Security Groups
- Tags
- Vmw Folders

HTTP

Headers

Text that is displayed when the field is empty

SAVE AND CONTINUE
SAVE

In order for provisioning Inputs to work properly, be sure to properly set the fields they should filter against. In the screenshot below you can see for a Resource Pool Input I've set the Group, Cloud, Layout, and Plan Inputs that it must be filtered against in order to work. Search for the Field Label of the target Input.

Once the type selected, the new Input will have many configuration options, most of which are the same options available when creating an Input from the Inputs section though some are new and some are presented in slightly different ways. The available options depend on the Input type selected but common options are shown in the expandable section below:

Common Input Configuration Options

FIELD LABEL The name and label of the Input

LOCALIZED LABEL If a localization code is selected, this field will have a translated label relative to the localization language selected for the appliance or user

FIELD NAME This is the Input fieldName property used to resolve the field value into code or to refer to this field for creating dynamic relationships with other Input fields

Note: Field names should only contain letters, numbers, and hyphen (-), underscore (_), or dot '.' for separation.

DEFAULT VALUE Pre-populates field with a default value

PLACEHOLDER Background text that populates inside a field for adding example values, does not set a value

HELP BLOCK Helpful text that will appear under your Input field to inform users about their selection

LOCALIZED HELP BLOCK If a localization code is selected, this field will have a translated help block relative to the localization language selected for the appliance or user

REQUIRED Prevents User from proceeding without setting value

EXPORT AS TAG Creates Tags for fieldName/value (key/value) on Instances

DISPLAY VALUE ON DETAILS When selected, the Input label and value (label: value) will be visible in a list of custom options on the Instance detail page

LOCKED The Input field is visible but locked from being edited by the user. Any configured default values will be seen and set on the Instance but the user may not change the value

HIDDEN Hides the field from view. The field is still active, however, and any configured default value would still be set

EXCLUDE FROM SEARCH For Select List and Typeahead Inputs, check to exclude the form data from being stored as variables (which can be leveraged from an API call when needed)

EDITABLE Allow the Input value to be updated when editing an Instance (This attribute is hidden if SHOW ON EDIT is not selected)

SHOW ON EDIT Display the Input name and value when editing an Instance

ALLOW MULTIPLE SELECTIONS For certain Input types which support multiple selections (Select List and Typeahead, for example), check to allow multiple items to be selected

DEPENDENT FIELD The Field Name value for a field that will reload this Option List to present a different set of selections. Take a look at the section below on Cascading Inputs as well as the [associated article](#) in our KnowledgeBase for documented examples of this feature

VISIBILITY FIELD A Field Name and selection value that will trigger this field to become visible. Currently, this only works when the Input is associated with a Service Catalog Item and viewed from the Service Catalog Persona perspective. See the section below on the Visibility Field for instructions on configuring this value

VERIFY PATTERN For Text and Text Area-type Inputs. If desired, enter a regex pattern string and user entries must match the string to be accepted

REQUIRE FIELD A fieldName that will trigger required attribute of this option

A complete example form making use of provisioning Inputs and Field Groups is shown below:

Using Localized Labels on Form Fields

In creating a Form Input, you may enter a custom static text label by setting the `FIELD LABEL` attribute on the Input. Using this will set a label for your Form field which will appear the same for all users. Alternatively, you can set a dynamic text label by selecting an entry from the `LOCALIZED LABEL` typeahead field. By setting a Localized Label, the label will appear differently depending on the user’s web browser language localization setting, the Morpheus appliance localization setting (set in the [global settings area](#)), or the user’s own localization setting from within the [User Settings section](#). More specific localization settings (such as a user’s setting over the appliance-wide setting) override less specific settings.

As an example, I’ll create a simple Form which has only a Group-type Input. Both the Field Label and Localized Label fields have been filled but the Localized Label takes precedence. Based on the localization settings for the user creating this Form, we see the label presented in US English as “GROUP.”

NEW FORM

Localization Example

GROUP Morpheus Clouds

+ ADD INPUT

+ ADD FIELD GROUP

DETAILS

USE EXISTING: No

TYPE: Group

FIELD LABEL: Choose Group

Label text for the input

LOCALIZED LABEL: Group

i18n code for the label

FIELD NAME: group

This field sets the name attribute for the input.

If we then navigate to provision a Catalog Item based on this Form, we still see the US English “GROUP” field label as this specific user has a default localization of “English (United States)” in its User Settings menu.

Resource

Localization Example

Localization Test

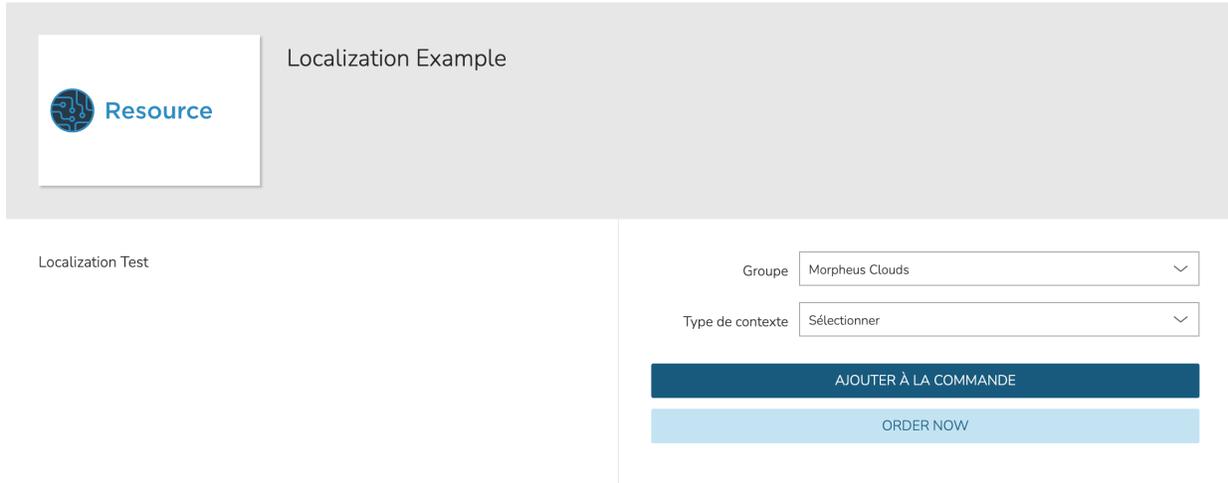
Group: Morpheus Clouds

Context Type: Select

ADD TO ORDER

ORDER NOW

However, if we select a different user which has a default localization of “French,” and go to provision the same Catalog Item, the field label is shown in French.



Resource

Localization Example

Localization Test

Groupe Morpheus Clouds

Type de contexte Sélectionner

AJOUTER À LA COMMANDE

ORDER NOW

Morpheus has several complete language packs and also allows users to contribute to new or in-progress language packs. If interested in creating or contributing to UI translations, take a look at [this YouTube video](#).

Turning Forms into Catalog Items

Once created, Forms can be associated with Catalog Items in the same way individual Inputs could before Forms were added. Add a new Catalog Items (or edit an existing one) in Library > Blueprints > Catalog Items. For complete details, refer to documentation on [creating Catalog Items](#).

Inputs

Inputs are custom input fields that can be added to Instance Types and Layouts, then presented in Instance, App, and Cloning wizards. The resulting value is available in the Instance config map as `<%=customOptions.fieldName%>` or `<%=customOptions['fieldName']`, the latter syntax being required in certain situations such as when the field name contains a hyphen “-” (ex. `<%=customOption['my-field-name']`). The `fieldName` and value can also be exported as Tags.

Create Input

Note: All possible fields listed. Displayed fields depend on **TYPE** selection

NAME Name of the Input

DESCRIPTION Description for reference in Input list view

FIELD NAME This is the input fieldName property that the value gets assigned to

Note: Field names should only contain letters, numbers, and hyphen (-), underscore (_), or dot'.' for separation.

EXPORT AS TAG Creates Tags for fieldName/value (key/value) on Instances

HIDDEN When marked, this Input will be hidden from the user but the value will still be accessible. This allows for alternate Input types to be used as hidden Inputs than what is possible with “Hidden” type Inputs (described further below). This is also different from Input “Visibility” settings as the values of non-visible Inputs is not accessible for consumption in automation routines

DEPENDENT FIELD The Field Name value for a field that will reload this Option List to present a different set of selections. Take a look at the section below on Cascading Inputs as well as the [associated article](#) in our KnowledgeBase for documented examples of this feature

VISIBILITY FIELD A Field Name and selection value that will trigger this field to become visible. Currently, this only works when the Input is associated with a Service Catalog Item and viewed from the Service Catalog Persona perspective. See the section below on the Visibility Field for instructions on configuring this value. It should also be noted that when an Input is not visible, its value is not available for consumption in automation routines. Use the “Hidden” flag (see above) to configure hidden yet accessible values

REQUIRE FIELD A fieldName that will trigger required attribute of this option

SHOW ON EDIT Display the Input name and value when editing an Instance

EDITABLE Allow the Input value to be updated when editing an Instance (This attribute is hidden if SHOW ON EDIT is not selected)

DISPLAY VALUE ON DETAILS When selected, the Input label and value (label: value) will be visible in a list of custom options on the Instance detail page

TYPE

- **Text:** Text Input Field
- **Text Area:** A text area input, when selected an additional option appears to allow the user to configure the default number of visible rows in the text area
- **Select List:** Populated by Option Lists, presents a manual or REST-populated dropdown list
- **Checkbox:** Checkbox for on or off values
- **Number:** Input field allowing only numbers
- **Typeahead:** Populated by Option Lists: Rather than presenting a potentially-large dropdown menu, the user can begin typing a selection into a text field and choose the desired option. Multiple selections can be allowed with this type by marking the 'ALLOW MULTIPLE SELECTIONS' box
- **Hidden:** No field will be displayed, but the field name and default value will be added to the Instance config map for reference
- **Password:** An input field with suitable encryption for accepting passwords
- **Radio List:** Populated by Option Lists, presents a selection of radio buttons for the provisioning user

LABEL This is the input label that typically shows to the left of a custom option

ROWS For Textarea type Option Lists, determines how many text rows will be given when the Input is presented

PLACEHOLDER Background text that populates inside a field for adding example values, does not set a value

DEFAULT VALUE Pre-populates field with a default value

HELP BLOCK Helpful text that will appear under your Input field to inform users about their selection

REQUIRED Prevents User from proceeding without setting value

REMOVE NO SELECTION For Select List-type Inputs. When marked, the Input will default to the first item in the list rather than to an empty selection. This is especially useful when only one choice is anticipated to be in the list as it saves the user from manually making a default selection

VERIFY PATTERN For Text and Text Area-type Inputs. If desired, enter a regex pattern string and user entries must match the string to be accepted

Verify Pattern Demo

DEFAULT CHECKED For Checkbox types, when marked the Checkbox will be checked by default

OPTION LIST For Select List types, select a pre-existing Option List to set dropdown values

Note: Select List and Typeahead Inputs require creation and association of an Option List

Cascading Inputs

One powerful facet of Morpheus Inputs is the ability to present users with different lists of input options based on their selections in other Inputs within the same wizard or modal. One common example, which is fully illustrated in this section, is to have a user select:

- The Group they wish to provision into...
- Then select the target Cloud from a list limited to Clouds which are in the selected Group...
- Then select the target network from a list limited to networks which are available to the selected Cloud *and* Group

To set this up, we will first configure our Inputs (custom option fields that can be applied to Instance Types and other Morpheus constructs) and Option Lists (dynamic lists of possible choices which can be associated with Inputs and presented in a dropdown or typeahead format). Once the custom options are configured, we will associate them with a new service catalog item and take a look at how the user would interact with them.

Group Custom Options

To begin, we will create a new [Option List](#). In this case, we will select type of “Morpheus Api” which will populate the list based on a call to the internal Morpheus API. Option Lists can also be populated by calls to external REST APIs or even from static lists that you enter manually. When dynamically populating Option Lists, whether via Morpheus API or an external API, translation and/or request scripts may be needed to prepare the request or translate the results. More on that as we build out the example.

I have called my Option List “Groups” and selected “Groups” from the OPTION LIST menu. This simply indicates that Groups are the construct we want to call into our list from Morpheus API. In this case, we want to present a list of all Groups to the user by their name and pass the Group database ID in the background. Since it is common to create Option Lists from Morpheus API where the construct name is displayed to the user and the ID is passed, we actually do not need to input any translation scripts in this case. However, I will include a translation script here which does the same thing simply to provide more clarity to the example. Morpheus [Option List documentation](#) includes additional details on available translation script inputs and which are available without translation as a convenience feature.

```
for (var x = 0; x < data.length; x++) {
  results.push({name: data[x].name, value:data[x].id});
}
```

After saving the Option List, create the Input that presents the list we just created. I gave my Input the name of “Selected Group”, field name of “selectedGroup”, and label of “Group”. For type, choose “Select List” and a new field will appear at the bottom of the modal where we can select the Option List we just created. With this configuration, the Input will present as a dropdown list containing the options called from our Option List.

Cloud Custom Options

Adding the Option List and Input for Clouds will be similar to the prior step with the exception that we will be including a request script which effectively filters the list of available Clouds to only those associated with the selected group. Follow the same process to start a new Option List, I have configured mine as follows:

- **NAME:** Parsed Clouds
- **TYPE:** Morpheus Api
- **OPTION LIST:** Clouds

We also need a request script that loads the `siteId` attribute of the `results` variable with the Group ID if the user has made a group selection. Essentially it appends this input as a query parameter to the API call, calling (for example)

./api/clouds?siteId=1 rather than ../api/clouds. It should be similar to the script below. Note that we are referencing the selectedGroup field name we created previously and that a “site” is the term for Groups in the Morpheus database.

```
if (input.selectedGroup) {
  results.siteId = input.selectedGroup
}
```

We also need a translation script which will be identical to the one used previously with the exception that if there is no input on the selectedGroups field, nothing will be displayed for the Clouds option.

```
if (input.selectedGroup) {
  for (var x = 0; x < data.length; x++) {
    results.push({name:data[x].name, value:data[x].id});
  }
}
```

We also need to create an Input to house this Option List. This process will be very similar to creating the previous Input except that we need to set selectedGroup as the Dependent Field. Setting a dependent field on an Input will trigger it to reload each time a selection is made in the indicated option. My configuration is as follows:

- **NAME:** Parsed Cloud
- **FIELD NAME:** parsedCloud
- **DEPENDENT FIELD:** selectedGroup
- **TYPE:** Select List
- **LABEL:** Cloud
- **OPTION LIST:** Parsed Clouds

Save your changes once done.

Network Custom Option

Finally, we will create an Option List/Input pair for network selection. In this case, it will be dependent on both the Group and Cloud selection. My Option List configuration is below:

- **NAME:** Parsed Networks
- **TYPE:** Morpheus Api
- **OPTION LIST:** Networks

Request Script:

```
if (input.parsedCloud && input.selectedGroup) {
  results.cloudId = input.parsedCloud
  results.groupId = input.selectedGroup
}
```

Translation Script:

```
if (input.parsedCloud && input.selectedGroup) {
  for (var x = 0; x < data.length; x++) {
    results.push({name:data[x].name, value:data[x].id});
  }
}
```

(continues on next page)

(continued from previous page)

```

}
}

```

The Input is configured as follows:

- **NAME:** Parsed Networks
- **FIELD NAME:** parsedNetwork
- **DEPENDENT FIELD:** parsedCloud
- **TYPE:** Select List
- **LABEL:** Network
- **OPTION LIST:** Parsed Networks

Setting Custom Options at Provision Time

At this point, our dependent options are ready to be applied to custom Instance Types, Workflows or Service Catalog items as needed. When creating them, we can select an unlimited number of Inputs from a typeahead field on the create modal and they will be presented when a user goes to provision that element or run that Workflow. As an example, I have created a Service Catalog item that incorporates the three Inputs we have created. You can see how the dependent fields reload and present different options based on my selections.

Visibility Field

The Input Visibility field allows users to set conditions under which the Input field is displayed. Visibility field accepts `fieldName:value` or `fieldName:(regex)`, where “fieldName” equals the fieldName of another Input which will determine the visibility of this Input, and “value” equals the target value of the other Input (or a regex pattern that matches to the values that meet your desired conditions). You can simply enter “fieldName” when visibility should be triggered when any value is entered. When the value of the target Input matches the “value” or “(regex)” set in the Visibility field, this Input will be displayed. When the value of the target Input does not match “value” or satisfy the “(regex)” set in the Visibility field, this Input will not be displayed.

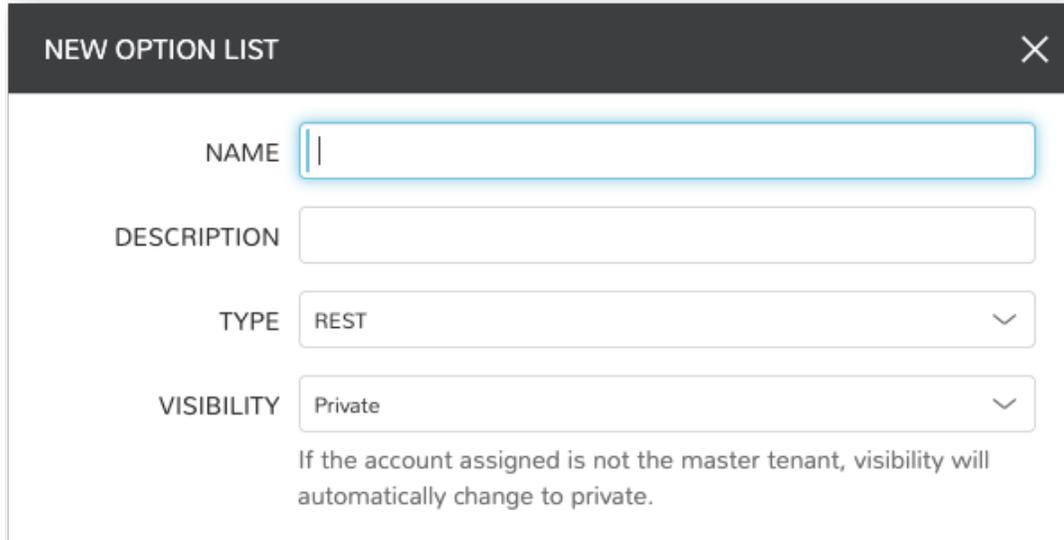
Expanding on the simplified example above, we could trigger visibility based on any one of multiple selections from the same Input by using a different regular expression, such as `color:(red|blue|yellow)`. Additionally, we are not restricted to the conditions of just one Input to determine visibility as the following would also be valid: `color:(red|blue|yellow), shape:(square)`. In the previous example, the Input “Color” would have to be set to red, blue, or yellow *OR* the Input “Shape” would have to be set to square in order to trigger visibility of the Input currently being configured. Prepend the previous example with `matchAll::` in order to require both conditions to be met rather than one or the other (ex. `matchAll::color:(red|blue|yellow), shape:(square)`).

Required Field

The Required field allows for Inputs to be conditionally required. In this field, enter the Field Name value for another Input and, if that Input is filled by the user, the current Input will become required. This feature could also be used in conjunction with the Visibility field described above in that you may want a field to be required when visible but not required when hidden. Below is a simple abstract example showing how the second displayed Input becomes required when the first displayed Input is filled.

Option Lists

Option Lists allow you to give the user more choices during provisioning to then be passed to scripts and/or automation. Option Lists, however, are pre-defined insofar as they are not free-form. They can be manually entered CSV or JSON, they can be dynamically compiled from REST calls via GET or POST requests, or populated by LDAP queries.



NEW OPTION LIST

NAME

DESCRIPTION

TYPE REST

VISIBILITY Private

If the account assigned is not the master tenant, visibility will automatically change to private.

Generic Option List Fields

The displayed fields in the create/edit Option List modal depend on the TYPE value selected.

NAME Name of the Option List

DESCRIPTION Description of the Option List for reference in Option List list view

TYPE

- **REST:** REST API call to populate Option List
- **Manual:** Manually entered dataset, CSV or JSON
- **Morpheus API:** Call to internal Morpheus API to populate the Option List
- **LDAP:** Searches and returns a list of Active Directory objects
- **Plugin:** Sourced by custom-coded DataSetProvider plugins. See [developer documentation](#) for additional details

VISIBILITY If the account currently signed in is not in the master tenant, visibility will automatically change to private

Manual Option List Fields

DATASET Appears only for manual Option Lists. Add your CSV or JSON list to this field

Note: JSON entries must be formatted like the following example: [{"name":"Test","value":1}, {"name":"Testing","value":2}]

Plugin Option List Fields

OPTION LIST Select an Option List made available by a currently-integrated plugins

REST Option List Fields

SOURCE URL A REST URL used to fetch list data which is cached in the appliance database

REAL TIME When checked, a REST call will be made to update the Option List at the time its presented to the User

IGNORE SSL ERRORS Do not fail API query for self-signed or invalid certs on REST call target

SOURCE METHOD GET or POST

SOURCE HEADERS Custom HTTP Headers to include in the source request

CREDENTIALS Use a stored credential set or manually enter credentials to access data requiring authentication. Currently, only basic auth is supported

INITIAL DATASET Create an initial JSON or CSV dataset to be used as the collection for this option list. It should be a list containing objects with properties 'name' and 'value'

TRANSLATION SCRIPT Create a JS script to translate the result data object into an array containing objects with properties 'name' and 'value'. The input data is provided as 'data' and the result should be put on the global variable 'results'.

Example:

```
for(var x=0;x < data.length; x++) {
  results.push({name: data[x].name,value:data[x].id});
}
```

REQUEST SCRIPT Create a JS script to prepare the request. Return a data object as the body for a POST request, and return an array containing properties 'name' and 'value' for a GET request. The input data is provided as 'data' and the result should be put on the global variable 'results'

Example:

```
results.push({name: 'userId', value : data.users})
```

In a GET request (SOURCE METHOD = GET), the value of the results variable is used to build out the request parameters. Thus, in the example results value:

```
results=[{name:"name1",value: "value1"}]
```

The request would be made to: `https://<someURL>?name1=value1`.

In a POST request (SOURCE METHOD = POST), the value of the results variable is used to build the body of the POST request. Thus, in the example results value:

```
results=[{name:"name1", value:"value1"}, {name:"name2", value:"value2"}]
```

The following JSON body would be posted to the target URL:

```
{name:"name1", value:"value1"}, {name:"name2", value:"value2"}
```

An alternative method to building the POST request (SOURCE METHOD = POST), can be seen below. As well, we can access other **Inputs** that are available on the same form, when provisioning an Instance or Catalog Item. As seen below, the other Inputs can be accessed using the data variable. We can access another Input by calling its **Field Name**, which can be configured when editing the Input in Library > Options > Inputs. This allows using data from other Inputs to be used in this Input's request.

In the example below the Input Field Name we'll access is myinputfieldname, which we can get either the name (visible value for lists) or value from the item:

Name variable: data.myinputfieldname Value variable: data.myinputfieldname_value

```
var postBody = {};  
postBody["number"] = data.myinputfieldname_value;  
postBody["env"] = "all";  
results = postBody;
```

The following JSON body would be posted to the target URL:

```
{ "number": "123456", "env": "all" }
```

Morpheus API Option List Fields

OPTION LIST A list of available object types to return

TRANSLATION SCRIPT Create a JS script to translate the result data object into an array containing objects with properties 'name' and 'value'. The input data is provided as 'data' and the result should be put on the global variable 'results'.

Example:

```
var i=0;  
results = [];  
for(i; i<data.length; i++) {  
  results.push({name: data[i].name, value: data[i].value});  
}
```

Translation script inputs:

Clouds

- id: <Number>
- value: <Number> // id, convenience
- name: <String>
- code: <String>
- description: <String>
- regionCode: <String>
- location: <String>

- zoneType: <Object>
 - id: <Number>
 - name: <String>
 - cloud: <String> // “public” or “private” value
 - code: <String>

Environments

- id: <Number>
- value: <Number> // id, convenience attribute to avoid requiring translation
- code: <String>
- name: <String>

Groups

- id: <Number>
- value: <Number> // id, convenience attribute to avoid requiring translation
- name: <String>
- code: <String>
- uuid: <String>
- location: <String>
- datacenterId: <Number>

Instances

- id: <Number>
- value: <Number> // id, convenience
- name: <String>
- displayName: <String>
- category: <String>
- description: <String>
- apiKey: <String>
- status: <String>
- hourlyPrice: <Number>
- hourlyCost: <Number>
- instanceType: <Object>
 - id: <Number>
 - name: <String>
- plan: <Object>
 - id: <Number>
 - name: <String>
- site: <Object>

- id: <Number>
- name: <String>

Instances Wiki

- id: <Number>
- value: <Number> // id, convenience
- name: <String>
- urlName: <String>
- category: <String>
- instanceId: <String>
- content: <String>
- contentFormatted: <String>
- format: <String>
- createdByUsername: <String>
- updatedByUsername: <String>

Networks

- id: <Number>
- value: <Number> // id, convenience
- code: <String>
- category: <String>
- name: <String>
- status: <String>
- cloudId: <Number>
- groupId: <Number>
- networkType:<Object>
 - id: <Number>
 - code: <String>
 - name: <String>
- externalId: <String>
- externalNetworkType: <String>
- networkDomain: <Object>
 - id: <Number>
 - name: <String>
- networkPool: <Object>
 - id: <Number>
 - name: <String>
- createdBy: <String>

Plans

- id: <Number>
- value: <Number> // id, convenience
- code: <String>
- name: <String>
- storage: <Integer, bytes>
- memory: <Integer, bytes>
- cores: <Number>

Resource Pools

- id: <Number>
- value: <Number> // id, convenience
- code: <String>
- externalId: <String>
- name: <String>
- serverGroupId: <Number>
- status: <String>
- regionCode: <String>
- parentPoolId: <Number>
- type: <String>

Security Groups

- id: <Number>
- value: <Number> // id, convenience
- code: <String>
- name: <String>
- externalType: <String>
- externalId: <String>
- cloudId: <Number>
- scopeMode: <String>
- scopeId: <Number>

Servers

- id: <Number>
- value: <Number> // id, convenience
- name: <String>
- displayName: <String>
- description: <String>
- category: <String>

- osType: <String>
- powerState: <String>
- lastStats: <String>
- zone: <Object>
 - id: <Number>
 - name: <String>
- capacityInfo: <Object>
 - maxStorage: <Integer, bytes>
 - maxMemory: <Integer, bytes>
 - maxCores: <Number>
 - usedMemory: <Integer, bytes>
 - usedStorage: <Integer, bytes>
- computeServerType: <Object>
 - id: <Number>
 - name: <String>
 - nodeType: <String>
 - vmHypervisor: <String>
 - containerHypervisor: <String>

Servers Wiki

- id: <Number>
- value: <Number> // id, convenience
- name: <String>
- urlName: <String>
- category: <String>
- serverId: <String>
- content: <String>
- contentFormatted: <String>
- format: <String>
- createdByUsername: <String>
- updatedByUsername: <String>

REQUEST SCRIPT The request script is used differently for Morpheus API Option List types. A Morpheus API option list type will use an internal API to return a list of objects instead of performing HTTP(S) requests to the Morpheus API. Due to this approach, the results object will not be used to generate query parameters or a JSON body. The results object will instead be used to contain a map of accepted key:value pairs that can be used to filter, sort and order the list of objects that get returned.

Below is a list of accepted key:value pairs for each object type:

Generic options available for all object types

- `max`: `<integer>` // Maximum number of results to return. Default: 25
- `offset`: `<integer>` // Offset for returned results. Default: 0
- `sort`: `<string>` // Field to sort on. Default: 'name'
- `order`: `<string>` // Order of returned values. Accepted values: 'asc', 'desc'. Default: 'asc'

Example: `results = {max: 5, order : 'desc'}`

Networks

- `zoneId`
- `siteId`
- `planId`
- `provisionTypeId`: `<Number>` // Id of the provision type (technology), filters to only networks associated with this provision type
- `layoutId`: `<Number>` // Id of an Instance Layout, ignored if `provisionTypeId` is supplied, otherwise used to look up the provision type
- `poolId`: `<Number>` // Id of a network pool, filters to only networks within the specified network pool

Instance Networks

Contains same options for Networks Morpheus API but pre-filtered for Networks applicable to a selected Instance Type.

- `phrase` : `<string>` // Fuzzy matches phrase on wiki name, urlName and content

Plans

- `zoneId`
- `siteId`
- `layoutId`
- `provisionTypeId`: `<Number>` // Id of the provision type (technology), filters to only plans associated with this provision type

Resource Pools

- `zoneId`
- `siteId`
- `planId`
- `layoutId`: `<Number>` // Id of an Instance Layout, used to get the associated provision type and filter to that provision type

Security Groups

- `zoneId` // required
- `poolId`

Clouds

- `zoneId` : `<integer>` // Database ID of cloud to return
- `tenantId` : `<integer>` // Database ID of tenant where clouds are added. Filters to only clouds added within the specified tenant. Only available in Master Tenant

- `zoneTypeId` : `<integer>` // Database ID of cloud type. Filters to only clouds with the specified cloud type
- `siteId` : `<integer>` // Database ID of group. Filters to only clouds within the specified group
- `tagName` : `<string>` // Filters to clouds with servers with tags containing the `tagName`
- `tagValue` : `<mixed>` // Requires `tagName`. Filters to clouds with servers that have tags containing the `tagName` and specified `tagValue`
- `phrase` : `<string>` // Fuzzy matches phrase on cloud name and description

Example: `results = {tenantId: 1, siteId: 1, tagName: "morpheus"}`

Instance Types Clouds

Contains same options for Clouds Morpheus API type but pre-filtered for Clouds applicable to a selected Instance Type.

- `phrase` : `<string>` // Fuzzy matches phrase on wiki name, `urlName` and content

Instances

- `appId` : `<integer>` // Database ID of app to filter by. Returns instances linked to the app
- `tenantId` : `<integer>` // Database ID of tenant where instances are located. Filters to only instances within the specified tenant. Only available in Master Tenant
- `serverId` : `<integer>` // Database ID of server. Filters to the instance that contains the specified server
- `tagName` : `<string>` // Filters to instances with tags containing the `tagName`
- `tagValue` : `<mixed>` // Requires `tagName`. Filters to instances with tags containing the `tagName` and specified `tagValue`
- `phrase` : `<string>` // Fuzzy matches phrase on instance name and description

Example: `results = {tenantId:1, phrase: "ha"}`

Groups

- `tenantId` : `<integer>` // Database ID of tenant where groups are located. Filters to only groups added within the specified tenant. Only available in Master Tenant
- `zoneTypeId` : `<integer>` // Database ID of cloud type. Filters to only groups that contain clouds with the specified cloud type
- `zoneId` : `<integer>` // Database ID of cloud. Filters to only groups that contain the cloud with the specified ID
- `siteId` : `<integer>` // Database ID of group to return
- `phrase` : `<string>` // Fuzzy matches phrase on group name and location.

Servers

- `tenantId` : `<integer>` // Database ID of tenant where servers are located. Filters to only servers within the specified tenant. Only available in Master Tenant
- `serverId` : `<integer>` // Database ID of server. Filters to the server specified by the ID
- `siteZoneId` : `<integer>` // Database ID of cloud. Filters to servers contained within the specified cloud
- `serverType` : `<string>` // Type of server. Accepted values: 'host', 'baremetal', 'vm'
- `siteId` : `<integer>` // Database ID of group. Filters to only servers contained within clouds that are added in the specified group

- `tagName` : `<string>` // Filters to servers with tags containing the `tagName`
- `tagValue` : `<mixed>` // Requires `tagName`. Filters to servers with tags containing the `tagName` and specified `tagValue`
- `phrase` : `<string>` // Fuzzy matches phrase on server name and description.

Example: `results = {max: 50, siteZoneId : 3}`

Instances Wiki

Contains same options for Instances Morpheus API type.

- `phrase` : `<string>` // Fuzzy matches phrase on wiki name, `urlName` and content

Servers Wiki

Contains same options for Servers Morpheus API type.

- `phrase` : `<string>` // Fuzzy matches phrase on wiki name, `urlName` and content

LDAP Option List Fields

LDAP URL The URL pointing to the LDAP server

USERNAME The fully qualified username (with @ suffix syntax) for the binding account

PASSWORD The password for the above account

LDAP Query The LDAP query to pull the appropriate objects. See the next section for an example use case

TRANSLATION SCRIPT Create a JS script to translate the result data object into an array containing objects with properties 'name' and 'value'. The input data is provided as 'data' and the result should be put on the global variable 'results'.

Note: Option Lists are set on one or multiple `Select List` or `Typeahead Inputs`. The Input is then set on an Instance Type, Layout, Cluster Layout, and/or Operational Workflow for input during provisioning or execution.

LDAP Query Variables

The current user and dependant parameters are loaded into the query using the `<%=phrase%>` syntax.

Listing 1: LDAP Query Variables (Example `<%=user.email%>`)

```

user {
  accountId,
  attributes,
  displayName,
  email,
  firstName,
  id,
  lastName,
  linuxUsername,
  username,
  windowsUsername
}
customOptions {

```

(continues on next page)

```
    fieldName
  }
```

Creating an Option List Based on an LDAP Query

In Morpheus version 4.2.1 and higher, Option Lists can be populated from LDAP queries. This gives users the ability to search Active Directory, capture objects, and present them as custom options where needed.

It's recommended that you connect LDAP-type Option Lists to Typeahead-type Inputs as the list of returned selections can be very large. This also allows you to select multiple options from the list, presuming you've allowed for that when creating the Input.

Populating LDAP-type Option Lists requires knowledge of LDAP query syntax. This guide provides one example and there are many publicly-available resources for help writing additional queries.

1. Create a new Option List (Library > Options > Option Lists > ADD)
2. Enter a name for the new LDAP Option List
3. Change the Type value to LDAP and the relevant fields will appear as shown in the screenshot:
4. Enter the LDAP URL in the following format (an example is also shown as a placeholder in the UI form field):

```
ldap[s]://<hostname>:<port>/<base_dn>
```

5. Enter the fully qualified username with @ suffix syntax, such as: *user@ad.mycompany.com*
6. Enter the account password
7. Enter your LDAP query. You can even inject variables into your query structure to query based on the value the user has entered into the typeahead field as shown in the example below:

```
(&(objectClass=user)(cn=<%=phrase%>*))
```

8. Finally, enter a translation script which will convert the returned LDAP object into a list of name:value pairs you can work with in Morpheus. The example script below shows the user DisplayName and sets the value to the SAMAccountName:

```
for(var x=0;x < data.length ; x++) {

    var row = data[x];
    var a = {};

    if(row.displayName != null) {
        a['name'] = row.displayName;
    } else {

        a['name'] = row.sAMAccountName;
    }

    a['value'] = row.sAMAccountName;
    results.push(a);
}
```

9. Click *SAVE CHANGES*

NEW OPTION LIST
✕

NAME

DESCRIPTION

TYPE

VISIBILITY

If the account assigned is not the master tenant, visibility will automatically change to private.

▼ Request Options

LDAP URL

USERNAME

PASSWORD

LDAP QUERY

LDAP Queries are standard LDAP formatted queries where different objects can be searched. Dependent parameters can be loaded into the query using the `<%=phrase%>` syntax.

TRANSLATION SCRIPT Create a js script to translate the result data object into an Array containing objects with properties `name`, and `value`. The input data is provided as `data` and the result should be put on the global variable `results`.

```
for(var x=0;x < data.length ; x++) {
  var row = data[x];
```

SAVE CHANGES

5.4.6 Templates

Templates can be created directly in Morpheus and/or sourced from version control, depending on the type. They can be used to help users consume IaC technologies, generate configuration files, utilize scripts or store security scan packages. Once stored, they can be used with provisioned Instances, configured as part of Tasks or Workflows, or run regularly as part of security scan Jobs. Each section below discusses template types in greater detail.

Spec Templates

Spec Templates allow Morpheus users to leverage several major Infrastructure-as-Code solutions. These are typically JSON or YAML-based configuration files which make creating and managing multiple resource types easier. Morpheus allows users to create and/or manage a collection of these templates for different solutions and from different sources.

Morpheus currently supports Spec Templates of the following types:

- Kubernetes Spec
- Helm Chart
- Terraform
- ARM Template
- CloudFormation Template

- OneView Server Profile Template
- UCS Service Profile Template

Morpheus also allows users to leverage templates pulled from URL sources, online repositories (such as GitHub), or you can write a template locally inside the “NEW SPEC TEMPLATE” modal.

Tip: To see Morpheus Spec Templates in action, take a look at our guide on [creating custom Instance Types using Terraform](#) or see our [KnowledgeBase](#) for another example where a CloudFormation Spec Template is used to create a provisionable custom Instance Type.

Creating a Spec Template

1. Navigate to Library > Templates > Spec Templates
2. Click + *ADD*
3. Complete the following fields, then click *SAVE CHANGES*:
 - **NAME**
 - **TYPE:** See the previous section for a complete list of Spec Template types
 - **SOURCE:** Local, Repository, or URL
 - **CONTENT:** If this is a local Spec Template, supply the template in this field. If the template is supplied through a URL or online repository, the CONTENT field will change to allow the user to point Morpheus to that resource
 - **VERSION:** (Only displayed on Terraform Spec Templates) Enter a Terraform version number to force a specific version when provisioning your Terraform Instance Type or App, assuming your Terraform Runtime setting (Administration > Settings > Provisioning Tab) is “auto”. If Terraform Runtime is set to “manual”, Morpheus will use the version of Terraform installed on the appliance box

File Templates

File Templates are for generating config files, such as my.cnf, elasticsearch.yml, morpheus.rb, or any text file. With full config map variable support, Template Files are dynamically generated during a Workflow phase or ad hoc via Instance actions.

File Templates can also be exposed on Instances in the Settings Tab. Ensure the Instance Type supports settings, and Category is defined in Advance Options on the Library Template config.

Note: Morpheus variables are supported in Library Templates using `<%= variable.var %>` format

Examples:

HA Proxy Config (haproxy.cfg)

- FILE NAME: haproxy.cfg
- FILE PATH: /config/haproxy.cfg
- PHASE: Pre Provision
- TEMPLATE:
- SETTING NAME: haproxyConfig

- SETTING CATEGORY: haproxy

```
#!/bin/bash

global
maxconn 256
log /dev/log local0 warning
log-tag <%=logTag%>

defaults
mode http
timeout connect 5000ms
timeout client 50000ms
timeout server 50000ms
log global

frontend http-in
bind *:<%=container.externalPort%>
default_backend servers

backend servers
# server server1 127.0.0.1:80 maxconn 32
```

mysql config (mysqld.cnf)

- FILE NAME: mysqld.cnf
- FILE PATH: /config/mysqld.cnf
- PHASE: Pre Provision

```
#!/bin/bash

[mysqld]
pid-file= /var/run/mysqld/mysqld.pid
socket= /var/run/mysqld/mysqld.sock
datadir= /var/lib/mysql
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0
explicit_defaults_for_timestamp = 1
```

Script Templates

Scripts are bash and Powershell scripts that can be attached to Node Types to always execute at the selected phase when that Node Type is provisioned, added to Workflows as Library Script Tasks, and/or executed ad-hoc on Instances.

Creating Scripts

1. Navigate to Library > Templates > Script Templates
2. Select + *ADD*
3. Enter the Following:

NAME Name of the Script in Morpheus

SCRIPT TYPE

- Bash
- Powershell

PHASE Select which phase the Script will execute when attached to a Node Type. When a script is attached to a Node Type, it will execute according to the selected phase:

Start Service Any time the Instance action `Start Service` is executed

Stop Service Any time the Instance action `Stop Service` is executed

Pre-Provision

- Containers: Script will execute against the container host before the container is provisioned
- Virtual Machines: Script will execute before any provision phase Scripts or Tasks

Provision

Script will execute once per new Instance node during the provision Phase. Provisioning will not be considered complete until all scripts and tasks in the provisioning phase are completed

Note: Any Script or Task set to the provision phase will be included in the total provision time and impact success/warn/failure provisioning status messages. As an example, your VM could be up and running but if your Script is in the provision phase and fails, provisioning will be marked as a failure.

Post-Provision Script will execute once per new Instance node after the provision phase is completed. Scripts and Tasks in the Post-Provision phase will show execution status and history, but are not considered part of the provision and do not impact provisioning status.

Pre-Deploy Script will execute on target Instance any time a deployment is run against the Instance. The Script will run prior to the deployment file(s) being written

Deploy Script will execute on target Instance any time a deployment is run against the Instance. The script will run after the deployment file(s) are written

Reconfigure Script will execute on target Instance any time a reconfigure is executed against the Instance.

Teardown Script will execute on target Instance upon Instance deletion. Script will execute against target Instance prior to the deletion/removal of resources.

SCRIPT Enter Bash or Powershell script.

Note: Morpheus variables are supported in Library Scripts using `<%= variable.var %>` format

RUN AS USER By default Scripts are execute as `morpheus-node`. To execute as another User, populate `RUN AS USER` and ensure proper user permissions & group access

SUDO Flag `SUDO` if `sudo` is required to execute the Script

To attach Scripts and templates that have been added to the Library to a Node Type, start typing the name and then select the script(s) and/or template(s).

- Multiple scripts and templates can be added to a Node Type
- Scripts and Templates can be added/shared among multiple Node Types
- The execution phase can be set for Scripts in the Scripts section
- Search will populate Scripts or Templates containing the characters entered anywhere in their name, not just the first letter(s) of the name

Security Packages

The Security Packages Section is for uploading SCAP packages which can then be consumed in Security Scan Jobs (Provisioning > Jobs).

Add a new Security Package

1. Navigate to Library > Templates > Security Packages
2. Click **+ADD > SCAP Package**
3. Provide a name in addition to a URL to source the package
4. Click **SAVE CHANGES**

Note: Currently URL is the only source option for security packages

ADD SECURITY PACKAGE
×

NAME

DESCRIPTION

ENABLED

Security Package Files

SOURCE

URL

SAVE CHANGES

Cluster Packages

Cluster Packages are created and can then be associated with Cluster Layouts. Cluster Packages themselves are created from Spec Templates. The Spec Templates and Cluster Layouts sections of Morpheus docs include more information on creating those constructs and show the steps to deploy sample builds.

Creating a Cluster Package

Navigate to the Cluster Packages List page (Library > Templates > Cluster Packages). System Cluster Packages are listed here. These are included as part of Morpheus pre-built Cluster Layouts and cannot be edited or viewed. User-created Cluster Layouts are also listed here and these may be edited. To begin creating a new Cluster Package, click + *ADD* and configure the following:

- **NAME:** A friendly name for the Cluster Package in Morpheus
- **CODE:** An identifying code for the Cluster Package for use with Morpheus API and CLI
- **DESCRIPTION:** An optional description for the Cluster Package
- **PACKAGE VERSION:** A version designation for the Cluster Package
- **ICON:** An identifying icon for the Cluster Package, this will appear in a list with other packages associated with a Cluster Layout
- **DARK MODE IMAGE:** An alternate icon for the Cluster Package used when the dark theme is enabled on the Morpheus appliance
- **TYPE:** A type selection for the package, see dropdown for options
- **PACKAGE TYPE:** A one-word descriptor for the package such as “calico”, “prometheus”, etc.
- **ENABLED:** When marked, this Cluster Package is available to be set on Cluster Layouts
- **REPEAT INSTALL:** When marked, if the package install initially fails, the installation will be attempted again
- **SPEC TEMPLATES:** Use the Typeahead field to select relevant Spec Templates to build the Cluster Package

The screenshot shows the Morpheus Library interface. At the top, there is a navigation bar with the Morpheus logo and a search bar. Below the navigation bar, there are several tabs: Operations, Provisioning, Library (selected), Infrastructure, Backups, Monitoring, Tools, and Administration. Under the Library tab, there are sub-tabs: Automation, Blueprints, Virtual Images, Options, Templates (selected), and Integrations. Below the sub-tabs, there are more sub-tabs: Spec Templates, File Templates, Script Templates, Security Packages, and Cluster Packages (selected). A search bar contains the text 'calico'. To the right of the search bar are buttons for '+ ADD' and a settings icon. Below the search bar is a table with the following columns: NAME, DESCRIPTION, TYPE, PACKAGE TYPE, and ENABLED. The table lists several Calico packages, all of which are enabled.

NAME	DESCRIPTION	TYPE	PACKAGE TYPE	ENABLED
calico 3.17.2 package	install calico networking	network	calico	✓
calico 3.17.2 package IP-in-IP Encapsulation	install calico networking	network	calico	✓
calico 3.20.0 package	install calico networking	network	calico	✓
calico 3.21.0 package	install calico 3.21 networking	network	calico	✓
calico 3.22.0 package	install calico 3.22 networking	network	calico	✓
calico 3.22.1 package	install calico 3.22.1 networking	network	calico	✓
calico 3.24.2 package	install calico 3.24.2 networking	network	calico	✓
calico 3.26.0 package	install calico 3.26.0 networking	network	calico	✓
kubernetes calico 3.25.0 package	install calico 3.25.0 networking	network	calico	✓

5.4.7 Library Integrations

The integrations section within Morpheus Library lists existing integrations and allows for the creation of new integrations which are related to automation technologies. A complete list of integrated third party technologies is available in the Administration section (Administration > Integrations). More detailed information about each Morpheus integration with third party technologies is included in our Automation Integrations section.

5.4.8 Variables

A vast number of variables are available for use in Tasks, Scripts, Templates, Resource Names, Cloud-Init User Data and Option List configs.

Important: Variables are case sensitive

Pre-Provision Vars

A subset of variables are available for Instance, Host Name and Hostnames. These can be passed inside `${ }` blocks during provisioning or in relevant policy configs. Groovy syntax can be resolved to allow for dynamic name generation as shown in some of the examples below.

Instance Naming Policy example: `${userInitials}-${cloudCode}-${platform == 'windows' ? 'W' : 'L'}-${sequence}`

Available variables for Naming Policy naming patterns include:

```

${account}
${accountId}
${accountName}
${accountNumber}
${accountType}
${cloudCode}
${cloudName}
${customerNumber}
${customOptions.fieldName}
${groupCode}
${groupName}
${instance.instanceContext} # Environment Code
${platform == 'windows' ? 'w':'l'} # results in `w` for Windows platforms and `l` for
↳Linux Platforms
${platform}
${provisionType}
${sequence} # results in 1
${sequence+100} # results in 101
${sequence.toString().padLeft(5,'0')} # results in 00001
${tenantId}
${tenant} # Tenant Name
${tenantSubdomain}
${type}
${userId}
${userInitials}
${username}
```

An example Instance Name Policy using a naming pattern with User Initials, Cloud Code, Instance Type, and a sequential number starting at 3000 is `${userInitials}-${cloudCode}-${type}-${sequence+3000}`, resulting in an Instance Name of **md-vmwd3-centos-3001** for the first instance, followed by **md-vmwd3-centos-3002** and so on.

Note: It's not recommended that users include ">", "<", "%", "\$", or "=" in naming policies.

Syntax Examples

PowerShell Example: \$app_id = "<%= instance.metadata.app_id %>"

Bash Example: HOSTNAME="<%= container.server.hostname %>"

Python Example: hostname = morpheus['server']['hostname']

HTTP Body Example: {"name": "<%= instance.createdByUsername %>"}

Note: customOptions values are defined from custom Inputs.

Common Examples

```

container.configGroup: <%=container.configGroup%>
container.configId: <%=container.configId%>
container.configPath: <%=container.configPath%>
container.configRole: <%=container.configRole%>
container.containerTypeCode: <%=container.containerTypeCode%>
container.containerTypeName: <%=container.containerTypeName%>
container.containerTypeShortName: <%=container.containerTypeShortName%>
container.cores: <%=container.cores%>
container.dataPath: <%=container.dataPath%>
container.dateCreated: <%=container.dateCreated%>
container.domainName: <%=container.domainName%>
container.environmentPrefix: <%=container.environmentPrefix%>
container.externalIp: <%=container.externalIp%>
container.hostMountPoint: <%=container.hostMountPoint%>
container.hostname: <%=container.hostname%>
container.image: <%=container.image%>
container.internalHostname: <%=container.internalHostname%>
container.internalIp: <%=container.internalIp%>
container.logsPath: <%=container.logsPath%>
container.memory: <%=container.memory%>
container.planCode: <%=container.planCode%>
container.provisionType: <%=container.provisionType%>
container.server: <%=container.server.serverTypeName%>
container.serverId: <%=container.serverId%>
container.sshHost: <%=container.sshHost%>
container.status: <%=container.status%>
container.storage: <%=container.storage%>
container.version: <%=container.version%>
customOptions: <%=customOptions.fieldName%>
evar: <%=evars.name%>
evars: <%=evars%>
group.code: <%=group.code%>
group.datacenterId: <%=group.datacenterId%>
group.location: <%=group.location%>
group.name: <%=group.name%>
instance.autoScale: <%=instance.autoScale%>
instance.configGroup: <%=instance.configGroup%>

```

(continues on next page)

```

instance.configId: <%=instance.configId%>
instance.configRole: <%=instance.configRole%>
instance.containers[0]: <%=instance.containers[0].containerTypeName%>
instance.cores: <%=instance.cores%>
instance.createdByEmail: <%=instance.createdByEmail%>
instance.createdByFirstName: <%=instance.createdByFirstName%>
instance.createdById: <%=instance.createdById%>
instance.createdByLastName: <%=instance.createdByLastName%>
instance.createdByUsername: <%=instance.createdByUsername%>
instance.deployGroup: <%=instance.deployGroup%>
instance.description: <%=instance.description%>
instance.displayName: <%=instance.displayName%>
instance.domainName: <%=instance.domainName%>
instance.environmentPrefix: <%=instance.environmentPrefix%>
instance.expireDate: <%=instance.expireDate%>
instance.firewallEnabled: <%=instance.firewallEnabled%>
instance.hostname: <%=instance.hostname%>
instance.instanceContext: <%=instance.instanceContext%> (tip: ↵
↵instanceContext = Environment)
instance.instanceLevel: <%=instance.instanceLevel%>
instance.instanceTypeCode: <%=instance.instanceTypeCode%>
instance.instanceTypeName: <%=instance.instanceTypeName%>
instance.instanceVersion: <%=instance.instanceVersion%>
instance.memory: <%=instance.memory%>
instance.metadata: <%=instance.metadata%>
instance.name: <%=instance.name%>
instance.networkLevel: <%=instance.networkLevel%>
instance.plan: <%=instance.plan%>
instance.provisionType: <%=instance.provisionType%>
instance.status: <%=instance.status%>
instance.statusMessage: <%=instance.statusMessage%>
instance.storage: <%=instance.storage%>
instance.tags: <%=instance.tags%>
instance.userStatus: <%=instance.userStatus%>
server.agentInstalled: <%=server.agentInstalled%>
server.agentVersion: <%=server.agentVersion%>
server.apiKey: <%=server.apiKey%>
server.category: <%=server.category%>
server.commType: <%=server.commType%>
server.configGroup: <%=server.configGroup%>
server.configId: <%=server.configId%>
server.configRole: <%=server.configRole%>
server.consoleHost: <%=server.consoleHost%>
server.consolePort: <%=server.consolePort%>
server.consoleType: <%=server.consoleType%>
server.consoleUsername: <%=server.consoleUsername%>
server.dataDevice: <%=server.dataDevice%>
server.dateCreated: <%=server.dateCreated%>
server.description: <%=server.description%>
server.displayName: <%=server.displayName%>
server.domainName: <%=server.domainName%>
server.externalId: <%=server.externalId%>

```

(continues on next page)

(continued from previous page)

```
server.externalIp: <%=server.externalIp%>
server.fqdn: <%=server.fqdn%>
server.hostname: <%=server.hostname%>
server.internalId: <%=server.internalId%>
server.internalIp: <%=server.internalIp%>
server.internalName: <%=server.internalName%>
server.internalSshUsername: <%=server.internalSshUsername%>
server.lastAgentUpdate: <%=server.lastAgentUpdate%>
server.lvmEnabled: <%=server.lvmEnabled%>
server.macAddress: <%=server.macAddress%>
server.managed: <%=server.managed%>
server.maxCores: <%=server.maxCores%>
server.maxMemory: <%=server.maxMemory%>
server.maxStorage: <%=server.maxStorage%>
server.name: <%=server.name%>
server.nodePackageVersion: <%=server.nodePackageVersion%>
server.osDevice: <%=server.osDevice%>
server.osType: <%=server.osType%>
server.osTypeCode: <%=server.osTypeCode%>
server.parentServerId: <%=server.parentServerId%>
server.plan: <%=server.plan%>
server.platform: <%=server.platform%>
server.platformVersion: <%=server.platformVersion%>
server.powerState: <%=server.powerState%>
server.serialNumber: <%=server.serialNumber%>
server.serverModel: <%=server.serverModel%>
server.serverType: <%=server.serverType%>
server.serverTypeCode: <%=server.serverTypeCode%>
server.serverTypeName: <%=server.serverTypeName%>
server.serverVendor: <%=server.serverVendor%>
server.softwareRaid: <%=server.softwareRaid%>
server.sourceImageId: <%=server.sourceImageId%>
server.sshHost: <%=server.sshHost%>
server.sshPort: <%=server.sshPort%>
server.sshUsername: <%=server.sshUsername%>
server.status: <%=server.status%>
server.statusMessage: <%=server.statusMessage%>
server.tags: <%=server.tags%>
server.toolsInstalled: <%=server.toolsInstalled%>
server.visibility: <%=server.visibility%>
task.results (using task code): <%=results.taskCode%>
task.results (using task name): <%=results["Task Name"]%>
task.results.value: <%=results.taskCode.key%>
zone.agentMode: <%=zone.agentMode%>
zone.cloudTypeCode: <%=zone.cloudTypeCode%>
zone.cloudTypeName: <%=zone.cloudTypeName%>
zone.code: <%=zone.code%>
zone.domainName: <%=zone.domainName%>
zone.firewallEnabled: <%=zone.firewallEnabled%>
zone.location: <%=zone.location%>
zone.name: <%=zone.name%>
zone.regionCode: <%=zone.regionCode%>
```

(continues on next page)

```
zone.scalePriority: <%=zone.scalePriority%>
cypher: <%=cypher.read('secret/hello')%>
cypher: <%=cypher.read('secret/' + zone.code)%> # Make variables more dynamic based off
↳ other variables
```

Instance

```
instance {
  adminPassword,
  adminUsername,
  apps. [],
  assignedDomainName,
  autoScale,
  backup. {},
  configGroup,
  configId,
  configRole,
  container. {},
  containers. [],
  cores,
  createBackup, true/false
  createdByEmail,
  createdByFirstName,
  createdById,
  createdByLastName,
  createdByUser. {
    username,
    displayName,
    firstName,
    lastName,
    email,
    linuxUsername,
    windowsUsername
  },
  createdByUsername,
  createUser, # true/false
  customOptions,
  deployGroup,
  description,
  displayName,
  domainName,
  environmentPrefix,
  evars: {},
  expireDate, # YYYY-MM-DD-T00:00:00Z
  expireDays,
  expose. [],
  firewallEnabled: true/false,
  hostId,
  hostname,
  id,
```

(continues on next page)

(continued from previous page)

```

instanceContext,
instanceLevel,
instanceTypeCode,
instanceTypeName,
instanceVersion,
isEC2:true/false,
isVpcSelectable, # true/false
layoutCode,
layoutId,
layoutName,
layoutSize,
lbInstances.[],
memory(bytes),
memoryDisplay, #MB/GB
metadata.{},
name,
nestedVirtualization,
networkLevel,
noAgent,
plan,
poolProviderType,
ports,
provisionType,
resourcePoolId,
scheduleStatus,
servicePassword,
serviceUsername,
smbiosAssetTag,
sslCertId,
sslEnabled, # true/false
status,
statusMessage,
storage, # bytes
tags,
userStatus,
vmwareFolderId,
}

```

Container

```

container {
    configGroup,
    configId,
    configPath,
    configRole,
    containerTypeCode,
    containerTypeShortName,
    cores,
    dataPath,
    dateCreated,

```

(continues on next page)

(continued from previous page)

```
    domainName,  
    environmentPrefix,  
    externalIp,  
    hostMountPoint,  
    hostname,  
    image,  
    internalHostname,  
    internalIp,  
    logsPath,  
    memory,  
    planCode,  
    provisionType,  
    server:{},  
    serverId,  
    sshHost,  
    status,  
    storage,  
    version,  
    containerTypeName  
}
```

Server

```
server {  
    agentInstalled,  
    agentVersion,  
    apiKey,  
    category,  
    commType,  
    configGroup,  
    configId,  
    configRole,  
    consoleHost,  
    consolePort,  
    consoleType,  
    consoleUsername,  
    dataDevice,  
    dateCreated,  
    description,  
    displayName,  
    domainName,  
    externalId,  
    externalIp,  
    fqdn,  
    hostname,  
    internalId,  
    internalIp,  
    internalName,  
    internalSshUsername,  
    lastAgentUpdate,  
}
```

(continues on next page)

(continued from previous page)

```
lvmEnabled,  
macAddress,  
managed,  
maxCores,  
maxMemory,  
maxStorage,  
name,  
nodePackageVersion,  
osDevice,  
osType,  
osTypeCode,  
parentServerId,  
plan,  
platform,  
platformVersion,  
powerState,  
serialNumber,  
serverModel,  
serverType,  
serverTypeCode,  
serverTypeName,  
serverVendor,  
softwareRaid,  
sourceImageId,  
sshHost,  
sshPort,  
sshUsername,  
status,  
statusMessage,  
tags,  
toolsInstalled,  
visibility,  
volumes {  
    name  
    id  
    deviceName  
    maxStorage  
    unitNumber  
    displayOrder  
    rootVolume  
}  
}
```

Zone (Cloud)

```
zone {
    agentMode,
    cloudTypeCode,
    cloudTypeName,
    code,
    datacenterId,
    domainName,
    firewallEnabled,
    location,
    name,
    regionCode,
    scalePriority
}
```

networkConfig

```
'networkConfig': {
  'primaryInterface': {
    'doDhcp': false,
    'dnsServers': [],
    'dnsDomain': "",
    'netmask': "",
    'gateway': "",
    'ipAddress': "",
    'doDhcp6': false,
    'gateway6': "",
    'ipv6Address': "",
    'cidr6Suffix': "",
    'dnsServers6': ""
  },
  'extraInterfaces': [
    {
      'doDhcp': false,
      'dnsServers': [],
      'dnsDomain': "",
      'netmask': "",
      'gateway': "",
      'ipAddress': "",
      'doDhcp6': false,
      'gateway6': "",
      'ipv6Address': "",
      'cidr6Suffix': "",
      'dnsServers6': ""
    },
    {
      'doDhcp': false,
      'dnsServers': [],
      'dnsDomain': "",
      'netmask': "",
```

(continues on next page)

(continued from previous page)

```

'gateway': "",
'ipAddress': "",
'doDhcp6': false,
'gateway6': "",
'ipv6Address': "",
'cidr6Suffix': "",
'dnsServers6': ""
}
]
}

```

Group (Site)

```

group {
  code,
  location,
  datacenterId,
  name
}

```

Custom Options (Inputs)

```

customOptions {
  customOptions.fieldName
}

```

Global

ex: <%= morpheus.user.id %>

```

"morpheus":{
  "user":{
    "id":value,
    "account":{
      "id":value
    },
    "username":"value",
    "displayName":"value",
    "email":"value",
    "firstName":"value",
    "lastName":"value",
    "dateCreated":0000-00-00T00:00:00Z,
    "lastUpdated":0000-00-00T00:00:00Z,
    "enabled":true/false,
    "accountExpired":true/false,
    "accountLocked":false,
    "passwordExpired":false,
    "defaultGroupId":value,

```

(continues on next page)

(continued from previous page)

```
"defaultZoneId":value,
"hasLinuxUser":true/false,
"hasWindowsUser":true/false,
"role":{
  "id":value
},
"instanceLimits":value
},
}
```

User

```
'user': {'accountId': int,
         'attributes': {samlAttributes},
         'displayName': 'string',
         'email': 'string',
         'firstName': 'string',
         'id': int,
         'lastName': 'string',
         'linuxUsername': 'string',
         'username': 'string',
         'windowsUsername': 'string',
```

Script Variables Example

Below is an example of the variables available to a script running against an Instance context.

Note: Variable maps are determined by context, configurations and permissions, actual maps may contain additional or fewer options.

```
'account': 'string',
'accountId': int,
'accountType': 'string',
'allowExisting': boolean,
'apps': [{'appContext': 'string',
          'description': 'string',
          'id': int,
          'name': 'string',
'cloud': 'string',
'cloudCode': 'string',
'cloudName': 'string',
'container': {'allowExisting': boolean,
              'certificatePath': string,
              'certificateStyle': string,
              'changeManagementExtId': int,
              'changeManagementServiceId': int,
              'cloud': 'string',
              'cloudConfig': {'agentInstall': agentInstallScript,
```

(continues on next page)

(continued from previous page)

```

        'finalizeServer': finalizeServerScript,
        'meta': metaData,
        'user': userData},
'configGroup': int,
'configId': int,
'configPath': 'string',
'configRole': int,
'containerTypeCategory': 'string',
'containerTypeCode': 'string',
'containerTypeName': 'string',
'containerTypeShortName': 'string',
'cores': int,
'coresPerSocket': int,
'createUser': boolean,
'customOptions': {'morph_ver': 'string',
'dataPath': 'string',
'dateCreated': 'string',
'domainName': 'string',
'environmentPrefix': 'string',
'evars': {},
'expireDays': 'string',
'expose': ['string'],
'exposedPorts': [{'loadBalanceProtocol': 'string',
                    'name': 'string',
                    'port': int}],
'externalIp': 'string',
'externalPort': int,
'hostMountPoint': 'string',
'hostName': 'string',
'hostname': 'string',
'hosts': {'containerName': 'string',
           'containerName': 'string',
           'containerName': 'string',
'id': int,
'image': 'string',
'instanceContext': 'string',
'instanceType': {'code': 'string',
'internalHostname': 'string',
'internalIp': 'string',
'internalPort': int,
'layout': {'code': 'string',
            'id': int},
'logsPath': 'string',
'maxCores': int,
'maxCpu': int,
'maxMemory': int,
'maxStorage': int,
'memory': int,
'memoryDisplay': 'string',
'mounts': [],
'name': 'string',
'networkId': int,
```

(continues on next page)

(continued from previous page)

```
'networkInterfaces': [{ 'id': 'string',
                        'ipAddress': 'string',
                        'ipMode': 'string',
                        'network': { 'dhcpServer': int,
                                    'group': int,
                                    'id': int,
                                    'name': 'string',
                                    'pool': int },
                        'networkInterfaceTypeId': int }],
'noAgent': boolean,
'planCode': 'string',
'portMap': {},
'ports': [{ 'displayName': 'string',
             'export': boolean,
             'exportName': 'string',
             'external': int,
             'index': int,
             'internal': int,
             'link': boolean,
             'loadBalance': boolean,
             'loadBalanceProtocol': 'string',
             'name': 'string',
             'primaryPort': boolean,
             'protocol': 'string',
             'visible': boolean },
           { 'displayName': 'string',
             'export': boolean,
             'exportName': 'string',
             'external': int,
             'index': int,
             'internal': int,
             'link': boolean,
             'loadBalance': boolean,
             'loadBalanceProtocol': 'string',
             'name': 'string',
             'primaryPort': boolean,
             'protocol': 'string',
             'visible': boolean }],
'provisionType': 'string',
'publicKeyId': int,
'server': {}
'serverId': int,
'shutdownDays': 'string',
'site': { 'accountId': int,
         'active': boolean,
         'id': int,
         'integrations': [],
         'location': 'string',
         'name': 'string',
         'visibility': 'string',
         'zones': [{}]},
'sshHost': 'string',
```

(continues on next page)

(continued from previous page)

```

    'status': 'string',
    'storage': int,
    'storageController': int,
    'type': 'string',
    'userGroup': {'id': ''},
    'version': 'string',
    'vm': boolean,
    'volumes': [{'datastoreId': int,
                  'id': int,
                  'maxIOPS': int,
                  'maxStorage': int,
                  'name': 'string',
                  'rootVolume': boolean,
                  'size': int,
                  'storageType': int,
                  'vId': int}]],
'containerName': 'string',
'coresPerSocket': int,
'createUser': boolean,
'customOptions': {'morph_ver': 'string',
'deployOptions': {},
'evars': {},
'expireDays': 'string',
'expose': ['string'],
'exposedPorts': [{'loadBalanceProtocol': 'string',
                  'name': 'string',
                  'port': int}],
'externalIp': 'string',
'group': {'code': 'string',
          'configCmdbId': 'string',
          'configManagementId': 'string',
          'datacenterId': int,
          'dnsIntegrationId': 'string',
          'location': 'string',
          'name': 'string',
          'serviceRegistryId': 'string',
'groupCode': 'string',
'groupName': 'string',
'host': ,
'hostMountPoint': 'string',
'hostName': 'string',
'hosts': {},
'input': {'backup': ,
          'cloud': {},
          'computedHostName': 'string',
          'computedName': 'string',
          'copies': int,
          'domainOptions': {}},
          'environmentVariables': {},
          'executionId': int,
          'expireDays': int,
          'group': {}},

```

(continues on next page)

```
'hostName': 'string',
'instanceContext': 'string',
'layout': {},
'metadata': {},
'name': 'string',
'plan': {},
'powerScheduleType': int,
'securityGroups': {},
'shutdownDays': int,
'type': 'string',
'version': 'string'},
'instance': {'adminPassword': 'maskedString',
  'adminUsername': 'string',
  'allowExisting': boolean,
  'apps': [{}],
  'assignedDomainName': 'string',
  'autoScale': boolean,
  'backup': {'backupRepository': int,
    'createBackup': boolean,
    'enabled': boolean,
    'jobAction': 'string',
    'jobRetentionCount': 'string',
    'providerBackupType': int,
    'showScheduledBackupWarning': boolean},
  'cloud': 'string',
  'cloudConfig': {'agentInstall': agentInstallScript,
    'finalizeServer': finalizeServerScript,
    'meta': metaData,
    'user': userData
  },
  'configGroup': int,
  'configId': int,
  'configRole': int,
  'container': {},
  'containers': [{}],
  'cores': int,
  'createBackup': boolean,
  'createUser': boolean,
  'createdByEmail': 'string',
  'createdByFirstName': 'string',
  'createdById': int,
  'createdByLastName': 'string',
  'createdByUser': {'accountId': int,
    'displayName': 'string',
    'email': 'string',
    'firstName': 'string',
    'id': int,
    'lastName': 'string',
    'linuxUsername': 'string',
    'username': 'string',
    'windowsUsername': 'string',
  },
  'createdByUsername': 'string',
```

(continues on next page)

(continued from previous page)

```

'customOptions': {'morph_ver': 'string',
'deployGroup': ,
'description': 'string',
'displayName': 'string',
'domainName': 'string',
'environmentPrefix': 'string',
'evars': {
'expireDate': date,
'expireDays': 'string',
'expose': ['string'],
'firewallEnabled': boolean,
'hostName': 'string',
'hostname': 'string',
'id': int,
'instanceContext': 'string',
'instanceLevel': 'string',
'instanceType': {'code': 'string',
'instanceTypeCode': 'string',
'instanceTypeName': 'string',
'instanceVersion': 'string',
'layout': {'code': 'string',
'id': int},
'layoutCode': 'string',
'layoutId': int,
'layoutName': 'string',
'lbInstances': [{'balanceMode': 'string',
'enabled': boolean,
'externalAddress': 'string',
'id': int,
'instanceId': int,
'loadBalancer': {'id': int},
'loadBalancerId': int,
'name': 'string',
'port': int,
'protocol': 'string',
'sslCert': 'string',
'sslRedirectMode': 'string',
'stickyMode': 'string',
'vipAddress': 'string',
'vipDirectAddress': 'string',
'vipHostname': 'string',
'vipName': 'string',
'vipPort': int,
'vipProtocol': 'string',
'vipScheme': 'string',
'vipShared': 'string',
'loadBalancerId': int,
'memory': int,
'memoryDisplay': 'string',
'metadata': {'ver': 'string',
'name': 'string',
'networkLevel': 'string',

```

(continues on next page)

```
'plan': 'string',
'ports': {},
'powerScheduleType': ,
'provisionType': 'string',
'scheduleStatus': 'string',
'servicePassword': 'maskedString',
'serviceUsername': 'string',
'shutdownDays': 'string',
'site': {'accountId': int,
        'active': boolean,
        'id': int,
        'integrations': [],
        'location': 'string',
        'name': 'string',
        'visibility': 'string',
        'zones': [{}]}
'sslCertId': int,
'sslEnabled': boolean,
'status': 'string',
'statusMessage': 'string',
'storage': int,
'tags': 'string',
'type': ,
'userGroup': {'id': 'string'},
'userStatus': 'string',
'instanceContext': 'string',
'instanceType': {'code': 'string'},
'internalIp': 'string',
'isDocker': boolean,
'layout': {'code': 'string'},
'localScriptGitId': int,
'localScriptGitRef': 'string',
'logTag': 'string',
'maxCores': int,
'maxCpu': int,
'maxMemory': int,
'maxStorage': int,
'memoryDisplay': 'string',
'morpheus': {'apiAccessToken': 'string',
             'applianceHost': 'string',
             'appliancePort': 'string',
             'applianceScheme': 'string',
             'applianceSsl': boolean,
             'applianceUrl': 'string'},
'morpheusUser': 'string',
'mounts': [],
'name': 'string',
'networkId': int,
'networkInterfaces': [{'id': 'string',
                       'ipAddress': 'string',
                       'ipMode': 'string',
                       'network': {'dhcpServer': ,
```

(continues on next page)

(continued from previous page)

```

        'group': int,
        'Id': int,
        'name': 'string',
        'pool': int},
        'networkInterfaceTypeId': int]],
'noAgent': boolean,
'platform': 'string',
'port': int,
'ports': [{'code': 'string',
            'displayName': 'string',
            'export': boolean,
            'exportName': 'string',
            'external': int,
            'index': int,
            'internal': int,
            'link': boolean,
            'loadBalance': boolean,
            'primaryPort': boolean,
            'protocol': 'string',
            'visible': boolean}],
'provisionType': 'string',
'publicKeyId': int,
'pythonAdditionalPackages': ,
'pythonArgs': ,
'pythonBinary': 'string',
'pythonScript': ,
'results': {},
'sequence': int,
'server': {'agentInstalled': boolean,
            'agentVersion': 'string',
            'apiKey': 'string',
            'category': ,
            'cloudConfig': {'agentInstall': agentInstallScript,
                             'finalizeServer': finalizeServerScript,
                             'meta': metaData,
                             'user': userData
                             },
            'commType': 'string',
            'computeTypeCode': 'string',
            'computeTypeName': 'string',
            'configGroup': int,
            'configId': int,
            'configRole': 'string',
            'consoleHost': 'string',
            'consolePort': int,
            'consoleType': 'string',
            'consoleUsername': 'string',
            'createdByUser': {'accountId': int,
                              'displayName': 'string',
                              'email': 'string',
                              'firstName': 'string',
                              'id': int,

```

(continues on next page)

```
        'lastName': 'string',
        'linuxUsername': 'string',
        'username': 'string',
        'windowsUsername': 'string',
'dataDevice': 'string',
'dateCreated': 'string',
'description': 'string',
'displayName': 'string',
'domainName': 'string',
'externalId': 'string',
'externalIp': 'string',
'fqdn': 'string',
'hostname': 'string',
'id': int,
'interfaces': [{ 'dhcp': boolean,
                  'domain': { 'fqdn': 'string',
                              'name': 'string',
                              'ouPath': 'string' },
                  'interfaceId': int,
                  'ipAddress': 'string',
                  'ipMode': 'string',
                  'ipSubnet': 'string',
                  'ipv6Address': 'string',
                  'ipv6Subnet': 'string',
                  'macAddress': 'string',
                  'network': { 'cidr': 'string',
                              'cidrMask': 'string',
                              'gateway': 'string',
                              'name': 'string',
                              'netmask': 'string',
                              'vlanId': int },
                  'networkPosition': 'string',
                  'vlanId': int }],
'internalId': int,
'internalIp': 'string',
'internalName': 'string',
'internalSshUsername': 'string',
'lastAgentUpdate': 'string',
'lvmEnabled': boolean,
'macAddress': 'string',
'managed': boolean,
'maxCores': int,
'maxMemory': int,
'maxStorage': int,
'name': 'string',
'nodePackageVersion': 'string',
'osDevice': 'string',
'osPassword': 'maskedString',
'osType': 'string',
'osTypeCode': 'string',
'osUsername': 'string',
'parentServerId': int,
```

(continues on next page)

(continued from previous page)

```

    'plan': 'string',
    'platform': 'string',
    'platformVersion': 'string',
    'powerScheduleType': ,
    'powerState': 'string',
    'publicKeyId': int,
    'serialNumber': 'string',
    'serverModel': 'string',
    'serverType': 'string',
    'serverTypeCode': 'string',
    'serverTypeName': 'string',
    'serverVendor': 'string',
    'softwareRaid': boolean,
    'sourceImageId': int,
    'sshHost': 'string',
    'sshPort': int,
    'sshUsername': 'string',
    'status': 'string',
    'statusMessage': 'string',
    'tags': {},
    'toolsInstalled': boolean,
    'uniqueId': int,
    'uuid': 'string',
    'visibility': 'string',
    'volumes': [{ 'deviceName': 'string',
                  'displayOrder': int,
                  'id': int,
                  'maxStorage': int,
                  'name': 'string',
                  'rootVolume': boolean,
                  'unitNumber': 'string',
                }],
'serverId': 'string',
'serverName': 'string',
'shutdownDays': 'string',
'site': { 'accountId': int,
          'active': boolean,
          'id': int,
          'integrations': [],
          'location': 'string',
          'name': 'string',
          'visibility': 'string',
          'zones': [{}],
        },
'sshKey': 'string',
'state': {},
'storageController': int,
'tenant': 'string',
'tenantId': int,
'tenantSubdomain': 'string',
'type': 'string',
'user': { 'accountId': int,
          'attributes': {samlAttributes},
          'displayName': 'string',
        },

```

(continues on next page)

```
'email': 'string',
'firstName': 'string',
'id': int,
'lastName': 'string',
'linuxUsername': 'string',
'username': 'string',
'windowsUsername': 'string',
'userGroup': {'id': 'string'},
'userId': int,
'userInitials': 'string',
'username': 'string',
'vm': boolean,
'volumes': [{ 'datastoreId': int,
               'id': int,
               'maxIOPS': int,
               'maxStorage': int,
               'name': 'string',
               'rootVolume': boolean,
               'size': int,
               'storageType': int,
               'vId': int}],
'zone': { 'agentMode': 'string',
          'cloudTypeCode': 'string',
          'cloudTypeName': 'string',
          'code': 'string',
          'datacenterId': int,
          'domainName': 'string',
          'firewallEnabled': boolean,
          'location': 'string',
          'name': 'string',
          'regionCode': 'string',
          'scalePriority': int}}
```

Note: Variable maps are determined by context, configurations and permissions, actual maps may contain additional or fewer options.

Spec Template Variables

5.5 Infrastructure

The heart of Morpheus is the ability to manage provisioning across any infrastructure, from bare metal to virtualized clouds and all the way to public infrastructure.

5.5.1 Groups

Overview

Groups in Morpheus define what resources a user has access to. Group access is defined by User Roles. Clouds are added to groups, and a User can only access the Clouds that are in the Groups their Role(s) gives them access to. Resources such as Networks, Datastores, Resources Pools, and Folders have additional Group access settings.

Policies applied to a Group will be enforced on all Instances provisioned or moved into that Group.

Note: Groups are not multi-tenant. A group only exists in the tenant it is created in.

The Groups view displays all current groups, includes search feature, and also enables the addition of new groups.

To View Groups:

1. Select the Infrastructure link in the navigation bar
2. Click the Groups link

UI

1. Select the Infrastructure link in the navigation bar
2. Click the Groups link

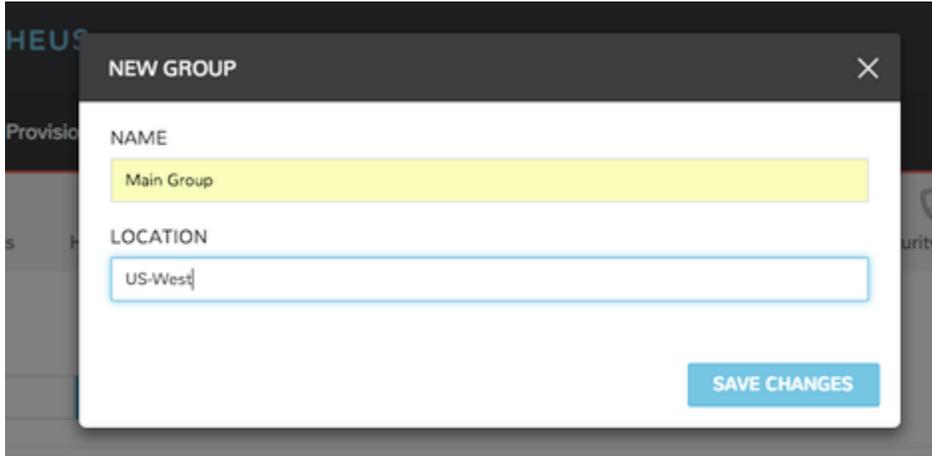
CLI

View all groups: `groups list` To use the group: `groups use <id>` or `groups use "group name"`
Json output of a specific group: `groups get <id> -j` or `groups get "group name" -j`

API

View all groups: `curl https://api.gomorpheus.com/api/groups -H "Authorization: BEARER access_token"`
View a specific group: `curl https://api.gomorpheus.com/api/groups/:id -H "Authorization: BEARER access_token"`

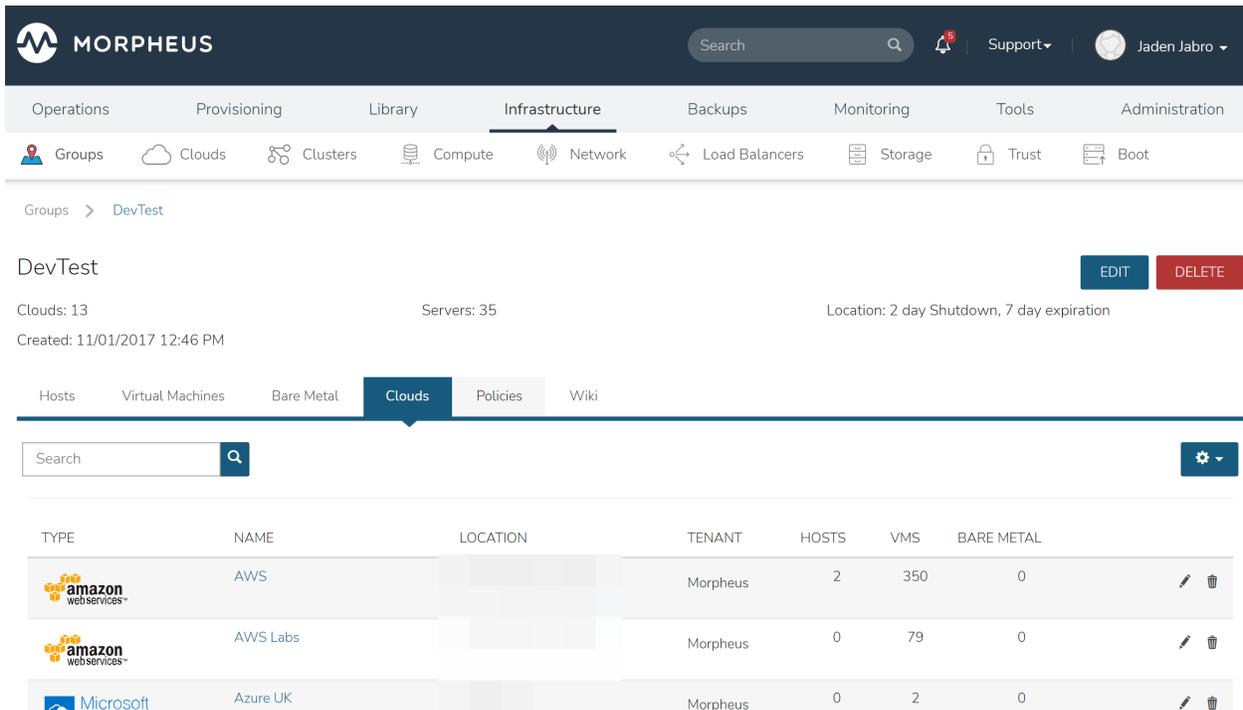
Adding Groups



To add a group:

1. Select the Infrastructure link in the navigation bar
2. Click the Groups link
3. Click the Create Group button
4. Input out the Name and Location (optional) fields
5. Click the Save Changes button to save

Managing Groups



TYPE	NAME	LOCATION	TENANT	HOSTS	VMS	BARE METAL	
	AWS		Morpheus	2	350	0	 
	AWS Labs		Morpheus	0	79	0	 
	Azure UK		Morpheus	0	2	0	 

To view a Group:

1. Select the Infrastructure link in the navigation bar
2. Click the Groups link
3. Click the Group name to view/modify

Available tabs in group view

Hosts Lists available hosts in the group and displays power, os, name, type, cloud, ip address, nodes, disc space, memory, and status. You can add a host from this tab panel by clicking Add Host.

Virtual Machines List all Virtual Machines in the Group.

Bare Metal List all Bare Metal Hosts added to the Group

Clouds Lists Clouds added to the Group. Existing Clouds or new Clouds can be added from the Group by clicking Add Cloud.

Policies Lists and allows creation or management of Policies applied to the Group.

Edit Group

To edit a group:

1. Select the Infrastructure link in the navigation bar.
2. Click the Groups link.
3. Click the name of the group you wish to edit.
4. Click the Edit button.
5. From the Edit Group Wizard modify information as needed.
6. Click the Save Changes button to save.

Delete Group

To delete a group:

1. Select the Infrastructure link in the navigation bar.
2. Click the Groups link.
3. Click the name of the group you wish to delete.
4. Click the Delete button.
5. Confirm

User Access

Important: User access to Groups is determined by their user Role(s). Group access for Roles can be configured in the Group Access section of a Roles Settings.

5.5.2 Clouds

Overview

Clouds are integrations or connections to public, private, hybrid clouds, or bare metal servers. Clouds can belong to many groups and contain many hosts. The clouds view includes clouds status, statistics, tenant assignment, and provides the option to add, edit, delete new clouds. Morpheus supports most Public Clouds and Private Clouds.

Supported Cloud Types

- Alibaba Cloud
- Amazon
- Azure (Public)
- Azure Stack (Private)
- Canonical MaaS
- Cloud Foundry
- Dell (Cloud type for PXE and manually added Dell EMC Hosts)
- DigitalOcean
- Google Cloud
- HPE (Cloud type for PXE and manually added HPE Hosts)
- Huawei
- Hyper-V
- IBM Cloud
- IBM Cloud Platform
- Kubernetes
- MacStadium
- Morpheus (Generic Cloud type for PXE/Bare Metal and manually added Hosts)
- Nutanix
- Open Telekom Cloud
- OpenStack
- Oracle Public Cloud
- Oracle VM
- Platform 9
- SCVMM
- Supermicro (Cloud type for PXE and manually added Supermicro Hosts)
- UCS
- UpCloud
- vCloud Air (OVH)
- VMWare ESXi

- VMware Fusion
- VMWare on AWS
- VMware vCenter
- VMware vCloud Director
- XenServer

Information on each cloud type can be found in the [Guides](#) section.

Creating Clouds

Clouds can be added from *Infrastructure > Clouds* or in *Infrastructure > Groups > (select Group) > Clouds*. Individual Guides for adding specific Cloud Types can be found in the [Guides](#) section.

Cloud Detail View

The Cloud Detail view shows metrics on health, sync status, current month costs, average monthly costs, resource utilization statistics, and resource counts for Container Hosts, Hypervisors, Bare Metal, Virtual Machines, and Un-managed resources.

The screenshot shows the Morpheus Cloud Detail View for a cloud named 'QA VMware'. The interface includes a navigation menu with 'Infrastructure' selected, and a sub-menu with 'Clouds' selected. The cloud name 'QA VMware' is displayed with 'EDIT', 'REFRESH', and 'DELETE' buttons. Below this, the location is 'H5 - Denver, CO', the last sync is '02/13/2024 10:21 AM', and the sync duration is '37 seconds'.

Key metrics are shown in a row of five cards:

- COST THIS MONTH:** \$15,587
- COST HISTORY:** A line graph showing costs from Sep to Feb, with a peak of 37472.96 in Jan.
- MAX CPU:** 23%
- MEMORY:** 12%
- STORAGE:** 0%

Resource counts are shown in a row of five cards:

- CONTAINER HOSTS:** 1
- HYPERVERSORS:** 3
- BARE METAL:** 0
- VIRTUAL MACHINES:** 59
- DISCOVERED:** 46

A navigation bar at the bottom includes 'Summary' (selected), Clusters, Hosts, VMs, Containers, Network, Data Stores, Resources, Policies, Profiles, Costing, and Wiki. Below this, the VMware logo is shown, and configuration details are listed in a grid:

Appliance URL: default	Time Zone: default	Domain:
Datacenter ID:	Agent Install Mode: cloudInit	Storage Mode: standard
Guidance: manual	Network Mode: standard	Local Firewall: off
Security Server: off	Costing Status: ok	

To view the Cloud List View, select the name of a Cloud to display the clouds Detail View.

EDIT Edit the setup configuration of the Cloud.

REFRESH Force a sync with the Cloud. Depending on the Cloud, choose to force a standard Cloud sync (occurs every five minutes by default) or a nightly sync. When syncing Costing data, Morpheus will force a pull of costing data for your specified period. If opting to “rebuild” the costing data, Morpheus will delete all costing data from the Cloud for that period and attempt to rebuild the data by calling the Cloud API.

DELETE Delete the Cloud from Morpheus

Important: All Instances and managed Hosts and VM’s associated with the Cloud must be removed prior to deleting a cloud.

Cloud Detail Tabs

Note: Not all tabs are available for all Cloud Types.

Clusters The Clusters tab displays clusters provisioned into the Cloud being viewed, including their status, type, name, layout, workers, and compute, memory, and storage stats. You can add a cluster by clicking *ADD CLUSTER*.

Hosts

The Hosts tab displays available hosts in the Cloud and displays power, OS, name, type, cloud, IP address, nodes, disk space, memory, and status. You can add a resource by clicking *ADD RESOURCE*, add a hypervisor host by clicking *ADD HYPERVISOR*, or perform action an action by selecting one or more Hosts and clicking *ACTIONS*.

VMs Displays an inventory of existing Instances in your Cloud configuration and provides details such as power, OS, name, type, cloud, IP address, nodes, disk space, memory, and status.

Bare Metal Setup PXE Boot in the Boot section to add bare metal servers. Once set up you can view information such as power, OS, name, type, cloud, IP address, nodes, disk space, memory, and status.

Security Groups The Security Groups tab displays a list of existing security groups in the cloud. You can add a security group to this cloud by clicking *EDIT SECURITY GROUPS*.

Load Balancers The load balancers tab panel displays available load balancers in the cloud including the name, description, type, cloud and host. You can add a load balancer from this tab by clicking *ADD LOAD BALANCER*.

Networks Displays Networks synced or added to the Cloud, including their name, type, CIDR, pool, DHCP status, visibility and targeted Tenant.

Data Stores Displays Datastores synced or added to the Cloud, including their name, type, capacity, online status, visibility, and targeted Tenant.

Resources Displays Resource Pools synced from the Cloud, including their name, description, and targeted Tenant.

Policies Manages Policies enforced on the Cloud. Setting a policy on this tab is equal to creating a policy in Administration > Policies and scoping it to the selected Cloud.

Profiles Manages Terraform, Key/Value Profiles that create custom object associated secrets and metadata that will automatically be mapped per Cloud selection during provisioning and automation.

Deleting Clouds

To delete a cloud:

1. Select the Infrastructure link in the navigation bar.
2. Select the Clouds link in the sub navigation bar.
3. Click the Delete icon of the cloud to delete.

Important: All Instances, managed Hosts and VMs must be removed prior to deleting a Cloud. To remove Instances, hosts and VMs from Morpheus without deleting the Cloud resources they represent, select Delete on the host or VM, unselect “Remove Infrastructure”, and select “Remove Associated Instances” if Instance are associated with the selected Hosts or VMs.

Cloud Profiles

Role Permissions

Access to **Profiles** tab is determined by the following role permissions:

Role: Feature Access: Admin: Profiles

- None: Cannot access Profiles tab or create/view/edit/delete profiles
- Read: Can access Profiles tab, can view profiles, cannot create/edit/delete profiles
- Full: Can access Profiles tab, can create/view/edit/delete profiles

Terraform Profiles

- Terraform Profiles allow creation of Cloud-associated tfvars secrets, allowing tf apps and specs to be provisioned across multiple clouds that required different tfvars.
- Target Cloud Terraform Profiles are automatically mapped to tf apps/specs during provisioning, no manual scoping is required.
- Terraform Profiles are encrypted in Cypher and creating a profile creates a Cypher entry with key `tfvars/profile/cloud/$cloudCode/variables``
- Terraform Profiles can be edited after creation
- Terraform Profiles are limited to one per Cloud, once one is created for the Cloud the option to create a Terraform Profile is no longer present. Edit the existing Terraform Profile to make changes at that point

Important: Since Morpheus mounts Terraform Profiles in Cypher using a mount point which contains the Cloud code value, any Clouds which have the same code will share a Terraform Profile. Create or edit Clouds to have a unique code value if they should have a unique Terraform Profile. It’s also important to understand that Morpheus does not require Clouds have a code at creation time. When Clouds are created without a code, Morpheus applies a generic non-unique code based on the Cloud type (“amazon” for AWS Clouds, as an example). This sets up a potential situation where all

Clouds of the same type have the same generic Cloud code and thus share a Terraform Profile. To avoid this situation, enter a Cloud code value at creation time or edit existing Clouds to have a unique code.

Create a Terraform Profile

1. Navigate to Infrastructure > Clouds and select a Cloud
2. Select the “Profiles” tab
3. Select + *ADD PROFILE*
4. Select Terraform Profile Type
5. Enter tfvars in the *Terraform Profile Variables* field
 - example Terraform Profile Variables

```
access_key="****accessKey****"
secret_key="*****secretKey*****"
region="us-west-1"
```

6. Select *SAVE CHANGES*

Now, when provisioning a Terraform Instance or App to the Cloud the profile was created in, the tfvars in the profile become available to Terraform. It is not necessary to manually tie this tfvars files to your App Blueprint, these tfvars will automatically be available to Terraform whenever you provision an App to this cloud.

Key/Value Store Profiles

- Key/Value Profiles (Key/Value Store) expand provisioning, automation, billing and reporting capabilities by allowing dynamic custom object specific metadata in provisioning and automation mappings
- Key/Value Profile entries are available using `<%=cloud.profile.key%>`
- Terraform Profiles are limited to one Profile per Cloud, however multiple key/value pairs can be added to a single profile

Create a Key/Value Profile

1. Navigate to `/infrastructure/clouds/` and select a Cloud
2. Select the Profiles tab
3. Select + *ADD PROFILE*
4. Select Key/Value Profile Type
5. Enter key/value entries, selecting + to add additional entries
6. Select *SAVE CHANGES*

5.5.3 Clusters

Overview

Infrastructure > Clusters is for creating and managing Kubernetes Clusters, Morpheus manager Docker Clusters, KVM Clusters, or Cloud specific Kubernetes services such as EKS, AKS and GKE.

Cluster Types

Name	Description	Provider Type
Kubernetes Cluster	Provisions by default a Kubernetes cluster consisting of 1 Kubernetes Master and 3 Kubernetes Worker nodes. Additional system layouts available including Master clusters. Custom layouts can be created.	Kubernetes
Docker Cluster	Provisions by default a Morpheus controlled Docker Cluster with 1 host. Additional hosts can be added. Custom layouts can be created. Existing Morpheus Docker Hosts are automatically converted to Clusters upon 4.0.0 upgrade.	Docker
EKS Cluster	Amazon EKS (Elastic Kubernetes Service) Clusters	Kubernetes
AKS Cluster	Azure AKS (Azure Kubernetes Service) Clusters	Kubernetes
KVM Cluster	Provisions by default a Morpheus controlled KVM Cluster with 1 host. Additional hosts can be added. Custom layouts can be created. Existing Morpheus KVM Hosts are automatically converted to Clusters upon 4.0.0 upgrade.	KVM
KVM/Docker Cluster	Provisions by default a Morpheus controlled Docker, VM and Functions* Cluster with 1 host. Additional hosts can be added.	Docker & KVM
Ext Kubernetes	Brings an existing (brownfield) Kubernetes cluster into Morpheus	Kubernetes
GKE Cluster	Google Cloud GKE (Google Kubernetes Engine) Clusters	Kubernetes

Note: Refer to clusterLayouts for supported Clouds per Cluster Type.

Requirements

- Morpheus Role permission Infrastructure: Clusters > Full required for Viewing, Creating, Editing and Deleting Clusters.
- Morpheus Role permission Infrastructure: Clusters > Read required for Viewing Cluster list and detail pages.

Cluster Permissions

- **Cluster Permissions** Each Cluster has Group, Tenant and Service Plan access permissions settings (“MORE” > Permissions on the Clusters list page).
- **Namespace Permissions** Individual Namespaces also have Group, Tenant and Service Plan access permissions settings

Kubernetes Clusters

Requirements

- Agent installation is required for Master and Worker Nodes. Refer to *Morpheus Agent* section for additional information.
- **Access to Cloud Front, Image copy access and permissions for System and Uploaded Images used in Cluster Layouts**
Image(s) used in Cluster Layouts must either exist in destination cloud/resource or be able to be copied to destination by Morpheus, typically applicable for non-public clouds. For the initial provision, Morpheus System Images are streamed from Cloud Front through Morpheus to target destination. Subsequent provisions clone the local Image.
- System Kubernetes Layouts require Master and Worker nodes to access to the following over 443 during K8s install and configuration:
 - Morpheus Appliance url (443)
 - <https://packages.cloud.google.com>
 - <https://storage.googleapis.com>
 - <https://docs.projectcalico.org>
 - <https://openebs.github.io>
 - <https://cloud.weave.works>
- Morpheus Role permission **Infrastructure**: **Clusters** > Full required for Viewing, Creating, Editing and Deleting Clusters.
- Morpheus Role permission **Infrastructure**: **Clusters** > Read required for Viewing Cluster list and detail pages.

Creating Kubernetes Clusters

Provisions a new Kubernetes Cluster in selected target Cloud using selected Layout.

Note: When deploying a highly-available Kubernetes cluster, it’s important to note that Morpheus does not currently auto-deploy a load balancer. Additionally, when Morpheus runs the `kubeadm init` command in the background during cluster provisioning, it also sets the `--control-plane-endpoint` flag to the first control plane node. This is a hard-coded behavior. To accomplish a highly-available cluster, users may wish to update the configured control plane endpoint, such as to a DNS name pointing to a load balancer. We are currently investigating updates to the product that would allow the user to specify such a DNS name prior to kicking off cluster provisioning. Additionally, users can circumvent this issue by configuring and deploying their own custom Cluster Layouts.

Morpheus maintains a number of default Kubernetes Cluster Layouts which are updated frequently to offer support for current versions. AKS & GKE Kubernetes versions will dynamically update to the providers supported versions.

Morpheus also supports creation of custom Kubernetes Cluster Layouts, a process which is described in detail in a later section.

To create a new Kubernetes Cluster:

1. Navigate to **Infrastructure > Clusters**
2. Select **+ ADD CLUSTER**
3. Select **Kubernetes Cluster**
4. Select a Group for the Cluster
5. Select **NEXT**
6. Populate the following:
 - CLOUD** Select target Cloud
 - CLUSTER NAME** Name for the Kubernetes Cluster
 - RESOURCE NAME** Name for Kubernetes Cluster resources
 - DESCRIPTION** Description of the Cluster
 - VISIBILITY**
 - Public** Available to all Tenants
 - Private** Available to Master Tenant
 - LABELS** Internal label(s)
7. Select **NEXT**
8. Populate the following:

Note: VMware sample fields provided. Actual options depend on Target Cloud

LAYOUT Select from available layouts. System provided layouts include Single Master and Cluster Layouts.

PLAN Select plan for Kubernetes Master

VOLUMES Configure volumes for Kubernetes Master

NETWORKS Select the network for Kubernetes Master & Worker VM's

CUSTOM CONFIG Add custom Kubernetes annotations and config hash

CLUSTER HOSTNAME Cluster address Hostname (cluster layouts only)

POD CIDR POD network range in CIDR format ie 192.168.0.0/24 (cluster layouts only)

WORKER PLAN Plan for Worker Nodes (cluster layouts only)

NUMBER OF WORKERS Specify the number of workers to provision

LOAD BALANCER Select an available Load Balancer (cluster layouts only) }

User Config

CREATE YOUR USER Select to create your user on provisioned hosts (requires Linux user config in Morpheus User Profile)

USER GROUP Select User group to create users for all User Group members on provisioned hosts (requires Linux user config in Morpheus User Profile for all members of User Group)

Advanced Options

DOMAIN Specify Domain override for DNS records

HOSTNAME Set hostname override (defaults to Instance name unless an Active Hostname Policy applies)

9. Select *NEXT*
10. Select optional Workflow to execute
11. Select *NEXT*
12. Review and select *COMPLETE*
 - The Master Node(s) will provision first.
 - **Upon successful completion of VM provision, Kubernetes scripts will be executed to install and configure Kubernetes**

Note: Access to the sites listed in the *Requirements* section is required from Master and Worker nodes over 443

- After Master or Masters are successfully provisioned and Kubernetes is successfully installed and configured, the Worker Nodes will provision in parallel.
 - **Provision status can be viewed:**
 - From the Status next to the Cluster in **Infrastructure > Clusters**
 - Status bar with eta and current step available on Cluster detail page, accessible by selecting the Cluster name from **Infrastructure > Clusters**
 - All process status and history is available - From the Cluster detail page History tab, accessible by selecting the Cluster name from **Infrastructure > Clusters** and the History tab - From *Operations - Activity - History* - Individual process output available by clicking *i* on target process
13. Once all Master and Worker Nodes are successfully provisioned and Kubernetes is installed and configured, the Cluster status will turn green.

Important: Cluster provisioning requires successful creation of VMs, Agent Installation, and execution of Kubernetes workflows. Consult process output from ```Infrastructure > Clusters - Details` and morpheus-ui current logs at **Administration - Health - Morpheus Logs** for information on failed Clusters.

Intra-Kubernetes Cluster Port Requirements

The table below includes port requirements for the machines within the cluster (not for the Morpheus appliance itself). Check that the following ports are open on Control-plane and Worker nodes:

Table 16: **Control-plane node(s)**

Protocol	Direction	Port Range	Purpose	Used By
TCP	Inbound	6443	Kubernetes API Server	All
TCP	Inbound	6783	Weaveworks	
TCP	Inbound	2379-2380	etcd server client API	kube-apiserver, etcd
TCP	Inbound	10250	kubelet API	Self, Control plane
TCP	Inbound	10251	kube-scheduler	Self
TCP	Inbound	10252	kube-controller-manager	Self

Table 17: **Worker node(s)**

Protocol	Direction	Port Range	Purpose	Used By
TCP	Inbound	10250	kubelet API	Self, Control plane
TCP	Inbound	30000-32767	NodePort Services	All

Adding Worker Nodes

1. Navigate to **Infrastructure - Clusters**
2. Select **v MORE** for the target cluster
3. Select **ADD (type) Kubernetes Worker**

NAME Name of the Worker Node. Auto=populated with `${cluster.resourceName}-worker-${seq}`

DESCRIPTION Description of the Worker Node, displayed in Worker tab on Cluster Detail pages, and on Worker Host Detail page

CLOUD Target Cloud for the Worker Node.

4. Select **NEXT**
5. Populate the following:

Note: VMware sample fields provided. Actual options depend on Target Cloud

SERVICE PLAN Service Plan for the new Worker Node

NETWORK Configure network options for the Worker node.

HOST If Host selection is enabled, optionally specify target host for new Worker node

FOLDER

Optionally specify target folder for new Worker node

Advanced Options

DOMAIN Specify Domain override for DNS records

HOSTNAME Set hostname override (defaults to Instance name unless an Active Hostname Policy applies)

6. Select *NEXT*
7. Select optional Workflow to execute
8. Select *NEXT*
9. Review and select *COMPLETE*

Note: Ensure there is a default StorageClass available when using a Morpheus Kubernetes cluster with OpenEBS so that Kubernetes specs or HELM templates that use a default StorageClass for Persistent Volume Claims can be utilised.

Kubernetes Cluster Detail Pages

The Kubernetes Cluster Detail page provides a high degree of monitoring and control over Kubernetes Clusters. This includes monitoring of all nodes in the Cluster, `kubectl` command line, account and role control, workload management, and more. The upper section of the page (which is persistent regardless of the currently-selected tab) provides high level costing and monitoring information, including a current aggregate metric for the CPU, memory and storage use.

The screenshot shows the Morpheus UI for a Kubernetes cluster. The top navigation bar includes the Morpheus logo, a search bar, a notification bell, a 'Support' dropdown, and a user profile 'Alex Harker'. The main navigation tabs are Operations, Provisioning, Library, Infrastructure (active), Backups, Monitoring, Tools, and Administration. A secondary navigation bar contains icons for Groups, Clouds, Clusters, Compute, Network, Load Balancers, Storage, Trust, and Boot. The breadcrumb trail is 'Clusters > QA Kubernetes Cluster'. The cluster name 'QA Kubernetes Cluster' is displayed with a green checkmark and three buttons: EDIT, DELETE, and ACTIONS. Below the cluster name, it shows 'Cloud: QA Kubernetes', 'Last Sync: 05/31/2023 02:23 PM', and 'Sync Duration: 1 second'. A 'don't delete' link is also present. The main metrics section shows four circular progress indicators: COST THIS MONTH (\$396), MAX CPU (16%), MEMORY (62%), and STORAGE CAPACITY (12%).

The upper section also includes the **ACTIONS** menu which includes the following functions:

- **REFRESH:** Forces a routine sync of the cluster status
- **PERMISSIONS:** View and edit the Group, Service Plan, and Tenant access permissions for the cluster
- **VIEW API TOKEN:** Displays the API token for the cluster
- **VIEW KUBE CONFIG:** Displays the cluster configuration
- **RUN WORKLOAD:** Run deployments, stateful sets, daemon sets, or jobs and target them to a specific namespace
- **UPGRADE CLUSTER:** Upgrade the cluster to a higher version of Kubernetes
- **ADD KUBERNETES WORKER:** Launches a wizard which allows users to configure a new worker for the cluster

Additional monitoring and control panes are located within tabs, some of which contain subtabs.

Summary

- Control
- Access
- Nodes
- Workloads
- Network
- Storage
- Containers
- Monitoring
- History
- Wiki

The summary tab contains high-level details on health and makeup of the cluster.

The Summary tab displays the following information:

- Cluster Details:**
 - Type: Kubernetes Cluster
 - Created By: Tommy Marks
 - Date Created: 02/16/2022 07:08 PM
 - Worker CPU: 13.0%
 - Worker Memory: 14.0GiB/22.8GiB
 - Worker Storage: 37.2GiB/300.0GiB
 - Enabled: Yes
 - Managed: Yes
 - Layout: MKS Kubernetes 1.23 Cluster on Ubuntu 20.04
 - API Url: <https://10.32.20.236:6443>
- Cluster Metrics:**
 - 1 CONTROLLERS
 - 3 WORKERS
 - 108 CONTAINERS
 - 26 SERVICES
 - 0 JOBS
 - 2 VOLUMES
- CPU Usage Graph:** A line graph showing CPU usage for various system components over time. The y-axis ranges from 0 to 0.8. Components include calico-apiserver, calico-system, default, kube-system, logging, monitoring, permanent, rook-ceph, and tigera-operator.
- MEMORY Usage Graph:** A bar chart showing memory usage for the same components. The y-axis ranges from 3.5 GiB to 14 GiB. rook-ceph is the primary memory consumer.

Contains the `kubectl` command line with ability to target commands to specific namespaces. The Control tab also contains the Packages subtab which displays the list of packages and their versions.

The Control tab interface includes:

- Navigation tabs: Summary, **Control**, Access, Nodes, Workloads, Network, Storage, Containers, Monitoring, History, Wiki.
- Subtabs: **kubectl**, Packages.
- Command Line: `kubectl` followed by a dropdown menu set to "all namespaces" and a text input field.
- Execution Buttons: Play, Copy, Refresh, and Close icons.

The Access Tab contains view and edit tools for Namespaces, accounts, roles, and role bindings.

The nodes tab includes a list of master and worker nodes in the cluster, their statuses, and the current compute, memory, and storage pressure on each node.

View and edit existing Pods, Deployments, Replica Sets, Daemon Sets, Stateful Sets, and Jobs. Add new Deployments, Stateful Sets, Daemon Sets, and Jobs through the ACTIONS menu near the top of the Cluster Detail Page.

View, add, and edit Services, Endpoints, Ingress and Network Policies

Summary Control Access Nodes Workloads **Network** Storage Containers Monitoring History Wiki

Services Endpoints Ingress **Policies**

Search All Namespaces

NAME	NAMESPACE	
allow-apiservert	calico-apiservert	📄 🗑️ ⚙️

View, add, and edit Storage classes, Volume claims, Volumes, Config maps, and Secrets

Summary Control Access Nodes Workloads Network **Storage** Containers Monitoring History Wiki

Storage Classes Volume Claims Volumes Config Maps **Secrets**

Search All Namespaces

NAME	NAMESPACE	
alertmanager-main	monitoring	📄 🗑️ ⚙️
alertmanager-main-generated	monitoring	📄 🗑️ ⚙️
alertmanager-main-tls-assets	monitoring	📄 🗑️ ⚙️
alertmanager-main-token-fb96l	monitoring	📄 🗑️ ⚙️
attachdetach-controller-token-swp24	kube-system	📄 🗑️ ⚙️

View a list of containers running on the cluster and restart or delete them if needed. This list can be filtered by Namespace or a specific Worker if desired.

Summary Control Access Nodes Workloads Network Storage **Containers** Monitoring History Wiki

Search All

<input type="checkbox"/>	STATUS	NAME	INSTANCE	TYPE	LOCATION	MANAGED	CPU	MEMORY	STORAGE	
<input type="checkbox"/>	▶️	calico-node-dpd4b-calico-node		docker.io/calico/node:v3.22.0	10.32.20.237	no	0	1	0	📄
<input type="checkbox"/>	▶️	calico-node-w68l9-calico-node		docker.io/calico/node:v3.22.0	10.32.20.239	no	0	1	0	📄
<input type="checkbox"/>	▶️	calico-typha-6d7b5857d8-d2zl-calico-typha		docker.io/calico/typha:v3.22.0	10.32.20.239	no	0	0	0	📄
<input type="checkbox"/>	▶️	calico-typha-6d7b5857d8-kqdf8-calico-typha		docker.io/calico/typha:v3.22.0	10.32.20.238	no	0	0	0	📄
<input type="checkbox"/>	▶️	fluent-bit-2z2jm-fluent-bit		fluent/fluent-bit:1.8.9	10.32.20.238	no	0	0	0	📄
<input type="checkbox"/>	▶️	fluent-bit-2ek4k-fluent-bit		fluent/fluent-bit:1.8.9	10.32.20.236	no	0	0	0	📄

View logs and events with filtering tools and search functionality available.

Summary Control Access Nodes Workloads Network Storage Containers **Monitoring** History Wiki

Logs Events

Search All Levels Today

NO DATA

TIMESTAMP MESSAGE

View the Cluster lifecycle history. This includes lists of automation packages (Tasks and Workflows) run against the cluster or its nodes, the success of these scripts and the output.

Summary Control Access Nodes Workloads Network Storage Containers Monitoring **History** Wiki

4/15/2022 7/28/2022 11/9/2022 2/22/2023 6/6/2023 9/18/2023

Search All

QA Kubernetes Cluster - Upgrading QA Kubernetes Cluster to v1.23.15 from v1.23.8
Start Date: 01/17/2023 11:53 AM Duration: 17s User: Tommy Marks

QA Kubernetes Cluster - Upgrading QA Kubernetes Cluster to v1.23.8 from v1.23.5
Start Date: 11/08/2022 02:54 PM Duration: 4m 54s User: Tommy Marks

QA Kubernetes Cluster - Ungrading QA Kubernetes Cluster to v1.23.5 from v1.22.3

View the Morpheus Wiki entry for this Cluster. This Wiki page may also be viewed in the Wiki section (Operations > Wiki). Edit the Wiki as desired, most standard Markdown syntax will be honored allowing the use of headings, links, embedded images, and more.

Summary Control Access Nodes Workloads Network Storage Containers Monitoring History **Wiki**

QA Cluster

Last Updated By: Alex Harker on 05/31/2023 04:05 PM

Adding External Kubernetes Clusters

Morpheus supports the management and consumption of Kubernetes clusters provisioned outside of Morpheus. These are referred to as External Kubernetes Clusters in Morpheus UI. This could be used, for example, to onboard and manage OpenShift clusters. In order to fully integrate the Kubernetes cluster with the Morpheus feature set, you may need to create a service account for Morpheus. Without first taking that step, some features may not work fully, such as listing all namespaces. The process for creating a service account and integrating the Cluster with Morpheus is described here.

First, create the Service Account within the Kubernetes cluster:

```
kubect1 create serviceaccount morpheus
```

Next, create the Role Binding:

```
kubectl create clusterrolebinding morpheus-admin \  
--clusterrole=cluster-admin --serviceaccount=default:morpheus \  
--namespace=default
```

With those items created, we can gather the API URL and the API token which will be used to add the existing cluster to Morpheus in the next step:

```
kubectl config view --minify | grep server | cut -f 2- -d ":" | tr -d " "
```

```
SECRET_NAME=$(kubectl get secrets | grep ^morpheus | cut -f1 -d ' ')  
kubectl describe secret $SECRET_NAME | grep -E '^token' | cut -f2 -d':' | tr -d " "
```

After finishing those steps, we can now create the external cluster in Morpheus. Navigate to Infrastructure > Clusters. Click + *ADD CLUSTER* and then select “External Kubernetes Cluster”. Set the following fields, you will have to advance through the pages of the wizard to see all fields indicated:

- **GROUP:** A previously created Morpheus Group
- **CLOUD:** A previously-integrated Cloud
- **CLUSTER NAME:** A friendly name for the onboarded cluster in Morpheus UI
- **RESOURCE NAME:** The resource name will be pre-pended to Kubernetes hosts associated with this cluster when shown in Morpheus UI
- **LAYOUT:** Set an associated Layout
- **API URL:** Enter the API URL gathered in the previous step
- **API TOKEN:** Enter the API Token gathered in the previous step
- **KUBE CONFIG:** Enter Kubeconfig YAML to authenticate the cluster

The above are the required fields, others may be optionally configured depending on the situation. Complete the wizard and Morpheus will begin the process of onboarding the existing cluster into management within Morpheus UI. Once things are finalized and statuses are green, the cluster can be monitored and consumed as any other cluster provisioned from Morpheus.

Operations Provisioning Library **Infrastructure** Backups Monitoring Tools Administration

Groups Clouds Clusters Compute Network Load Balancers Storage Trust Boot

Clusters > QA Kubernetes Cluster

✓ QA Kubernetes Cluster EDIT DELETE ACTIONS

Cloud: QA Kubernetes Last Sync: 11/29/2023 04:25 PM Sync Duration: 1 second

\$395 COST THIS MONTH

18% MAX CPU

40% MEMORY

13% STORAGE CAPACITY

Summary Control Access Nodes Workloads Network Storage Containers Monitoring History Wiki

	Type: Kubernetes Cluster	Created By: [redacted]	Date Created: 10/29/2023 11:17 PM
	Worker CPU: 17.0%	Worker Memory: 9.1GiB/22.7GiB	Worker Storage: 39.2GiB/300.0GiB
	Enabled: Yes	Managed: Yes	Layout: MKS Kubernetes 1.28 Cluster on Ubuntu 20.04
	API Url: [redacted]		

1 CONTROLLERS 3 WORKERS 116 CONTAINERS 23 SERVICES 3 JOBS 1 VOLUMES

Docker Clusters

Provisions a new Docker Cluster managed by Morpheus.

To create a new Docker Cluster:

1. Navigate to **Infrastructure** > **Clusters**
2. Select **+ ADD CLUSTER**
3. Select **Docker Cluster**
4. Populate the following:
 - CLOUD** Select target Cloud
 - CLUSTER NAME** Name for the Docker Cluster
 - RESOURCE NAME** Name for Docker Cluster resources
 - DESCRIPTION** Description of the Cluster
 - VISIBILITY**
 - Public** Available to all Tenants
 - Private** Available to Master Tenant

LABELS Internal label(s)

5. Select *NEXT*
6. Populate the following (options depend on Cloud Selection and will vary):

LAYOUT Select from available layouts.

PLAN Select plan for Docker Host

VOLUMES Configure volumes for Docker Host

NETWORKS Select the network for Docker Master & Worker VM's

NUMBER OF HOSTS Specify the number of hosts to be created

User Config

CREATE YOUR USER Select to create your user on provisioned hosts (requires Linux user config in Morpheus User Profile)

USER GROUP Select User group to create users for all User Group members on provisioned hosts (requires Linux user config in Morpheus User Profile for all members of User Group)

Advanced Options

DOMAIN Specify Domain for DNS records

HOSTNAME Set hostname (defaults to Instance name)

7. Select *NEXT*
8. Select optional Workflow to execute
9. Select *NEXT*
10. Review and select *COMPLETE*

EKS Clusters

Provisions a new Elastic Kubernetes Service (EKS) Cluster in target AWS Cloud.

Note: EKS Cluster provisioning is different than creating a Kubernetes Cluster type in AWS EC2, which creates EC2 instances and configures Kubernetes, outside of EKS.

Morpheus currently supports EKS in the following regions: us-east-2, us-east-1, us-west-1, us-west-2, af-south-1, ap-east-1, ap-south-1, ap-northeast-3, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-northeast-1, ca-central-1, eu-central-1, eu-west-1, eu-west-2, eu-south-1, eu-west-3, eu-north-1, me-south-1, sa-east-1, us-gov-east-1, us-gov-west-1

Create an EKS Cluster

1. Navigate to Infrastructure - Clusters
2. Select + *ADD CLUSTER*
3. Select EKS Cluster
4. Populate the following:
 - LAYOUT** Select server layout for EKS Cluster

PUBLIC IP

Subnet Default Use AWS configured Subnet setting for Public IP assignment

Assigned EIP Assigned Elastic IP to Controller and Worker Nodes. Requires available EIP's

CONTROLLER ROLE Select Role for EKS Controller from synced role list

CONTROLLER SUBNET Select subnet placement for EKS Controller

CONTROLLER SECURITY GROUP Select Security Group assignment for EKS Controller

WORKER SUBNET Select Subnet placement for Worker Nodes

WORKER SECURITY GROUP Select Security Group assignment for Worker Nodes

WORKER PLAN Select Service Plan (EC2 Instance Type) for Worker Nodes

User Config

CREATE YOUR USER Select to create your user on provisioned hosts (requires Linux user config in Morpheus User Profile)

USER GROUP Select User group to create users for all User Group members on provisioned hosts (requires Linux user config in Morpheus User Profile for all members of User Group)

Advanced Options

DOMAIN Specify Domain for DNS records

HOSTNAME Set hostname (defaults to Instance name)

5. Select *NEXT*
6. Select optional Workflow to execute
7. Select *NEXT*
8. Review and select *COMPLETE*

GKE Clusters

Provisions a new Google Kubernetes Engine (GKE) Cluster in target Google Cloud.

Note: Ensure proper permissions exist for the Google Clouds service account to create, inventory and manage GKE clusters.

Create an GKE Cluster

1. Navigate to Infrastructure - Clusters
2. Select + *ADD CLUSTER*
3. Select GKE Cluster
4. Populate the following:
 - CLOUD** Select target Cloud
 - CLUSTER NAME** Name for the GKE Cluster
 - RESOURCE NAME** Name for GKE Cluster resources/hosts

DESCRIPTION Description of the Cluster

VISIBILITY

Public Available to all Tenants

Private Available to Master Tenant

LABELS Internal label(s)

LAYOUT Select cluster layout for GKE Cluster

RESOURCE POOL Specify an available Resource Pool from the selected Cloud

GOOGLE ZONE Specify Region for the cluster

VOLUMES Cluster hosts volume size and type

NETWORKS Select GCP subnet(s) and config

WORKER PLAN Service Plan for GKE worker nodes

RELEASE CHANNEL Regular, Rapid, Stable or Static

CONTROL PLANE VERSION Select from available synced GKE k8's versions

NUMBER OF WORKERS Number of worker nodes to be provisioned

5. Select *NEXT*
6. Select optional Workflow to execute
7. Select *NEXT*
8. Review and select *COMPLETE*

5.5.4 Compute



Note: The Infrastructure Hosts page from previous versions has been renamed to Compute and updated with Container and Resource sections.

The *Infrastructure > Compute* section provides a universal stage for viewing and managing Hosts, Virtual Machines, Containers, Resources, and Bare Metal across Clouds.

In this section you can:

- View & Manage and Hosts, Virtual Machines, Containers, Resources, Bare Metal and Hypervisors
- Add manual Virtual Machines and Bare Metal Hosts
- Convert Hosts, Virtual Machines and Bare Metal to Managed

Hosts



Section Features & Filters *Record Search Custom Views CSV Export JSON Export Paging Config*

- Standard Filters: *OS Type*
- Advanced Filters: *Status Managed Server Type*

Hosts in Morpheus are Hypervisors and Container hosts that VMs and Containers are hosted on, such as ESXi Hosts and Kubernetes Master and Workers. These hosts are populated from integrated clouds, hosts and clusters provisioned from Morpheus, or manually added hosts.

Provisioning new hosts takes place in the Infrastructure > Clusters section of Morpheus. For example, provisioning a new Docker cluster in that section will begin the process of creating a Morpheus-managed Docker cluster with one host (by default). Additional hosts and custom layouts can also be created. See the [Clusters](#) section of Morpheus docs for more information.

Virtual Machines



Section Features & Filters *Record Search Custom Views CSV Export JSON Export Paging Config*

- Standard Filters: *OS Type*
- Advanced Filters: *Status Managed Server Type*

The Virtual Machines tab lists all managed and unmanaged VMs across Morpheus. Managed VMs are either provisioned by Morpheus, or are inventoried/discovered VMs that have been converted to managed. Unmanaged VMs are typically inventoried/discovered VMs from Cloud integrations.

- Virtual Machine Change Cloud

Change Cloud functionality allows a server to be reassigned to a new Cloud, which may be necessary at times for easier record keeping in Morpheus. In order to use this feature, the user must have “Infrastructure: Move Servers” permission set to “Full.”

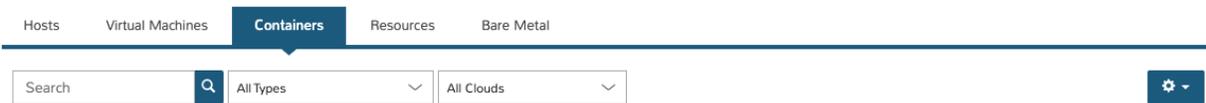
Changing Clouds might be necessary, for example, when moving a VM from one vCenter datacenter to another. We can use this tool to update the Cloud association in Morpheus as well. Other scenarios may include migrating workloads from private Cloud to public Cloud or even creating a brand new VM in a new Cloud which represents an identical workload to something pre-existing but which will be retired.

Note: For VMware Clouds, Cloud reassignment occurs automatically when a server is migrated from one cluster to another or from one resource pool to another within the same cluster. Following the next Cloud sync,

Morpheus will reflect the migration made in vCenter. In this scenario, you should not try to use the manual change Cloud tools. In fact, you will receive a warning in Morpheus logs if you attempt to do so.

To change Clouds, navigate to the VM detail page (Infrastructure > Compute > Virtual Machines > selected VM), click on the ACTIONS menu, and click Change Cloud. You can select the new target Cloud here and optionally select a new VM which the current one should be merged into. The important thing to keep in mind is that this tool is for Morpheus record keeping only. **It is not a tool which does migration work for you.** See the embedded video below for a demonstration of this feature.

Containers



Section Features & Filters *Record Search Custom Views CSV Export JSON Export Paging Config*

- Standard Filters: *Container Type Cloud*

The containers tab lists all Containers associated with Morpheus Instances accessible to the user. Note additional system level containers from Kubernetes and Docker Clusters are not listed here but are accessible in Cluster detail sections.

Resources



Section Features & Filters *Record Search Custom Views CSV Export JSON Export Paging Config*

- Standard Filters: *Resource Type Cloud*

Resources represent objects that do not map to VM or Container types in Morpheus, such as IAC resources from Terraform, Cloudformation or ARM Templates like VPC's, Gateways, Users, Policies, Brokers, API's, Endpoints, Directories, ACL's, Routes... well anything really. All resources created from IAC Templates map to iac provider resource types and Morpheus maintains a resource object record from the mapped resource.

Expand the **Resource Types** table below to see all Resource types that will be mapped to Resource objects in Morpheus:

- Resource Types **Click to Expand/Hide**

The following Resource Types are mapped when provisioning IAC templates including Terraform and Cloud-Formation templates.

ICON	RESOURCE NAME	PROVIDER
 AWS Resource	Accessanalyzer Analyzer	terraform
 AWS Resource	Acm Certificate	terraform
 AWS Resource	Acm Certificate Validation	terraform
 AWS Resource	Acmpca Certificate Authority	terraform
 Security Center	Advanced Threat Protection	terraform
 API Gateway	Amazon Api	cloudFormation
 API Gateway	Amazon Api Authorizer	cloudFormation
 API Gateway	Amazon Api Deployment	cloudFormation
 API Gateway	Amazon Api Domain Name	cloudFormation
 API Gateway	Amazon Api Integration	cloudFormation
 API Gateway	Amazon Api Integration Response	cloudFormation
 API Gateway	Amazon Api Mapping	cloudFormation
 API Gateway	Amazon Api Model	cloudFormation
 API Gateway	Amazon Api Route	cloudFormation
 API Gateway	Amazon Api Route Responese	cloudFormation
 API Gateway	Amazon Api Stage	cloudFormation
 EC2	Ami	terraform
 EC2	Ami Copy	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 EC2	Ami From Instance	terraform
 EC2	Ami Launch Permission	terraform
 Analysis Services	Analysis Server	terraform
 Analysis Services	Analysis Server	terraform
 Azure Resource	API	terraform
 Azure Resource	API Authorization Server	terraform
 Azure Resource	API Backend	terraform
 Azure Resource	API Certificate	terraform
 Azure Resource	API Diagnostic	terraform
 API Gateway	Api Gateway Account	terraform
 API Gateway	Api Gateway Api Key	terraform
 API Gateway	Api Gateway Authorizer	terraform
 API Gateway	Api Gateway Base Path Mapping	terraform
 API Gateway	Api Gateway Client Certificate	terraform
 API Gateway	Api Gateway Deployment	terraform
 API Gateway	Api Gateway Documentation Part	terraform
 API Gateway	Api Gateway Documentation Version	terraform
 API Gateway	Api Gateway Domain Name	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 API Gateway	Api Gateway Gateway Response	terraform
 API Gateway	Api Gateway Integration	terraform
 API Gateway	Api Gateway Integration Response	terraform
 API Gateway	Api Gateway Method	terraform
 API Gateway	Api Gateway Method Response	terraform
 API Gateway	Api Gateway Method Settings	terraform
 API Gateway	Api Gateway Model	terraform
 API Gateway	Api Gateway Request Validator	terraform
 API Gateway	Api Gateway Resource	terraform
 API Gateway	Api Gateway Rest Api	terraform
 API Gateway	Api Gateway Stage	terraform
 API Gateway	Api Gateway Usage Plan	terraform
 API Gateway	Api Gateway Usage Plan Key	terraform
 API Gateway	Api Gateway Vpc Link	terraform
 Azure Resource	API Group	terraform
 Azure Resource	API Group User	terraform
 Azure Resource	API Identity Provider AD	terraform
 Azure Resource	API Identity Provider Facebook	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Azure Resource	API Identity Provider Google	terraform
 Azure Resource	API Identity Provider Microsoft	terraform
 Azure Resource	API Identity Provider Twitter	terraform
 Azure Resource	API Logger	terraform
 Azure Resource	API Management	terraform
 Azure Resource	API Named Value	terraform
 Azure Resource	API OpenID Provider	terraform
 Azure Resource	API Operation	terraform
 Azure Resource	API Operation Policy	terraform
 Azure Resource	API Policy	terraform
 Azure Resource	API Product	terraform
 Azure Resource	API Product API	terraform
 Azure Resource	API Product Group	terraform
 Azure Resource	API Product Policy	terraform
 Azure Resource	API Property	terraform
 Azure Resource	API Schema	terraform
 Azure Resource	API Subscription	terraform
 Azure Resource	API User	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Azure Resource	API Version Set	terraform
 API Gateway	Apigatewayv2 Api	terraform
 API Gateway	Apigatewayv2 Api Mapping	terraform
 API Gateway	Apigatewayv2 Authorizer	terraform
 API Gateway	Apigatewayv2 Deployment	terraform
 API Gateway	Apigatewayv2 Domain Name	terraform
 API Gateway	Apigatewayv2 Integration	terraform
 API Gateway	Apigatewayv2 Integration Response	terraform
 API Gateway	Apigatewayv2 Model	terraform
 API Gateway	Apigatewayv2 Route	terraform
 API Gateway	Apigatewayv2 Route Response	terraform
 API Gateway	Apigatewayv2 Stage	terraform
 API Gateway	Apigatewayv2 Vpc Link	terraform
 App Configurator	App Configuration	terraform
 Elastic Load Balancing	App Cookie Stickiness Policy	terraform
 Application Insights	App Insights Analytics Item	terraform
 Application Insights	App Insights API Key	terraform
 Application Insights	App Insights Web Test	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Azure Resource	App Service	terraform
 Azure Resource	App Service Active Slot	terraform
 Azure Resource	App Service Certificate	terraform
 Azure Resource	App Service Certificate Order	terraform
 Azure Resource	App Service Environment	terraform
 Azure Resource	App Service Host Binding	terraform
 Azure Resource	App Service Network Swift Connection	terraform
 Azure Resource	App Service Plan	terraform
 Azure Resource	App Service Slot	terraform
 Azure Resource	App Service Source Control Token	terraform
 Auto Scaling	Appautoscaling Policy	terraform
 Auto Scaling	Appautoscaling Scheduled Action	terraform
 Auto Scaling	Appautoscaling Target	terraform
 Compute	Application Definition	terraform
 Network	Application Gateway	terraform
 Network	Application Gateway	terraform
 Application Insights	Application Insights	terraform
 Network	Application Security Group	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Network	Application Security Group	terraform
 AWS Resource	Appmesh Mesh	terraform
 AWS Resource	Appmesh Route	terraform
 AWS Resource	Appmesh Virtual Node	terraform
 AWS Resource	Appmesh Virtual Router	terraform
 AWS Resource	Appmesh Virtual Service	terraform
 AppSync	Appsync Api Key	terraform
 AppSync	Appsync Datasource	terraform
 AppSync	Appsync Function	terraform
 AppSync	Appsync Graphql Api	terraform
 AppSync	Appsync Resolver	terraform
 Athena	Athena Named Query	cloudFormation
 Athena	Athena Database	terraform
 Athena	Athena Named Query	terraform
 Athena	Athena Workgroup	terraform
 Automation	Automation Account	terraform
 Automation	Automation Certificate	terraform
 Automation	Automation Credential	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Automation	Automation DSC Config	terraform
 Automation	Automation DSC Node Config	terraform
 Automation	Automation Job Schedule	terraform
 Automation	Automation Module	terraform
 Automation	Automation Runbook	terraform
 Automation	Automation Schedule	terraform
 Automation	Automation Variable Boolean	terraform
 Automation	Automation Variable Datetime	terraform
 Automation	Automation Variable Int	terraform
 Automation	Automation Variable String	terraform
 Auto Scaling	Autoscaling Attachment	terraform
 Auto Scaling	Autoscaling Group	terraform
 Auto Scaling	Autoscaling Lifecycle Hook	terraform
 Auto Scaling	Autoscaling Notification	terraform
 Auto Scaling	Autoscaling Policy	terraform
 Auto Scaling	Autoscaling Schedule	terraform
 Compute	Availability Set	terraform
 Elastic Load Balancing	AWS ALB Listener	cloudFormation

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Elastic Load Balancing	AWS ALB Listener Certificate	cloudFormation
 Elastic Load Balancing	AWS ALB Listener Rule	cloudFormation
 Elastic Load Balancing	AWS ALB Target Group	cloudFormation
 AWS App Mesh	AWS App Mesh	cloudFormation
 AWS App Mesh	AWS App Mesh Route	cloudFormation
 AWS App Mesh	AWS App Mesh Virtual Node	cloudFormation
 AWS App Mesh	AWS App Mesh Virtual Router	cloudFormation
 AWS App Mesh	AWS App Mesh Virtual Service	cloudFormation
 Elastic Load Balancing	AWS Application Load Balancer	cloudFormation
 Auto Scaling	AWS Auto Scaling Group	cloudFormation
 Auto Scaling	AWS Auto Scaling Launch Configuration	cloudFormation
 Auto Scaling	AWS Auto Scaling Lifecycle Hook	cloudFormation
 Auto Scaling	AWS Auto Scaling Policy	cloudFormation
 Auto Scaling	AWS Auto Scaling Scheduled Action	cloudFormation
 Elastic Load Balancing	AWS Elastic Load Balancer	cloudFormation
 Elasticsearch Service	AWS Elasticsearch Domain	cloudFormation

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 AWS SDB	AWS SDB Domain	cloudFormation
 Secrets Manager	AWS Secrets Manager Resource Policy	cloudFormation
 Secrets Manager	AWS Secrets Manager Rotation Schedule	cloudFormation
 Secrets Manager	AWS Secrets Manager Secret	cloudFormation
 Secrets Manager	AWS Secrets Manager Secret Target Attachment	cloudFormation
 AWS Resource	AWS SES Configuration Set	cloudFormation
 AWS Resource	AWS SES Configuration Set Event Destination	cloudFormation
 AWS Resource	AWS SES Receipt Filter	cloudFormation
 AWS Resource	AWS SES Receipt Rule	cloudFormation
 AWS Resource	AWS SES Receipt Rule Set	cloudFormation
 AWS Resource	AWS SES Template	cloudFormation
 AWS Resource	AWS SNS Subscription	cloudFormation
 AWS Resource	AWS SNS Topic	cloudFormation
 AWS Resource	AWS SNS Topic Policy	cloudFormation
 AWS SQS	AWS SQS Queue	cloudFormation
 AWS SQS	AWS SQS Queue Policy	cloudFormation
 WAF	AWS WAF Byte Match Set	cloudFormation

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 WAF	AWS WAF IP Set	cloudFormation
 WAF	AWS WAF Regional Byte Match Set	cloudFormation
 WAF	AWS WAF Regional Geo Match Set	cloudFormation
 WAF	AWS WAF Regional IP Set	cloudFormation
 WAF	AWS WAF Regional Rate Based Rule	cloudFormation
 WAF	AWS WAF Regional Regex Pattern Set	cloudFormation
 WAF	AWS WAF Regional Rule	cloudFormation
 WAF	AWS WAF Regional Size Constraint Set	cloudFormation
 WAF	AWS WAF Regional Sql Injection Match Set	cloudFormation
 WAF	AWS WAF Regional Web ACL	cloudFormation
 WAF	AWS WAF Regional Web ACL Association	cloudFormation
 WAF	AWS WAF Regional Xss Match Set	cloudFormation
 WAF	AWS WAF Rule	cloudFormation
 WAF	AWS WAF Size Constraint Set	cloudFormation
 WAF	AWS WAF Sql Injection Match Set	cloudFormation
 WAF	AWS WAF Web ACL	cloudFormation
 WAF	AWS WAF Xss Match Set	cloudFormation
 Workspaces	AWS Workspace	cloudFormation

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Azure Resource	Azure Resource	terraform
 Azure Resource	Azure Resource	terraform
 Backup	Backup Plan	terraform
 Backup	Backup Policy File Share	terraform
 Backup	Backup Policy VM	terraform
 Backup	Backup Protected File Share	terraform
 Backup	Backup Protected VM	terraform
 Backup	Backup Selection	terraform
 Backup	Backup Storage Account	terraform
 Backup	Backup Vault	terraform
 Network	Bastion Host	terraform
 Network	Bastion Host	terraform
 Batch	Batch Account	terraform
 Batch	Batch Account	terraform
 Batch	Batch Application	terraform
 Batch	Batch Application	terraform
 Batch	Batch Certificate	terraform
 Batch	Batch Certificate	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 AWS Resource	Batch Compute Environment	terraform
 AWS Resource	Batch Job Definition	terraform
 AWS Resource	Batch Job Queue	terraform
 Batch	Batch Pool	terraform
 Batch	Batch Pool	terraform
 Bot	Bot Channel Email	terraform
 Bot	Bot Channel Email	terraform
 Bot	Bot Channel MS Teams	terraform
 Bot	Bot Channel MS Teams	terraform
 Bot	Bot Channel Registration	terraform
 Bot	Bot Channel Slack	terraform
 Bot	Bot Connection	terraform
 Bot	Bot Connection	terraform
 Bot	Bot Direct Line	terraform
 Bot	Bot Web App	terraform
 Bot	Bot Web App	terraform
 AWS Resource	Budgets Budget	terraform
 CDN	CDN Endpoint	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 CDN	CDN Endpoint	terraform
 CDN	CDN Profile	terraform
 CDN	CDN Profile	terraform
 AWS Resource	Cloud9 Environment Ec2	terraform
 Cloud Formation	Cloudformation Stack	terraform
 Cloud Formation	Cloudformation Stack Set	terraform
 Cloud Formation	Cloudformation Stack Set Instance	terraform
 CloudFront	Cloudfront Distribution	terraform
 CloudFront	Cloudfront Origin Access Identity	terraform
 CloudFront	Cloudfront Public Key	terraform
 AWS Resource	Cloudhsm V2 Cluster	terraform
 AWS Resource	Cloudhsm V2 Hsm	terraform
 Cloud Trail	Cloudtrail	terraform
 Cloud Watch	Cloudwatch Dashboard	terraform
 Cloud Watch	Cloudwatch Event Permission	terraform
 Cloud Watch	Cloudwatch Event Rule	terraform
 Cloud Watch	Cloudwatch Event Target	terraform
 Cloud Watch	Cloudwatch Log Destination	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Cloud Watch	Cloudwatch Log Destination Policy	terraform
 Cloud Watch	Cloudwatch Log Group	terraform
 Cloud Watch	Cloudwatch Log Metric Filter	terraform
 Cloud Watch	Cloudwatch Log Resource Policy	terraform
 Cloud Watch	Cloudwatch Log Stream	terraform
 Cloud Watch	Cloudwatch Log Subscription Filter	terraform
 Cloud Watch	Cloudwatch Metric Alarm	terraform
 CodeBuild	Codebuild Project	terraform
 CodeBuild	Codebuild Source Credential	terraform
 CodeBuild	Codebuild Webhook	terraform
 CodeCommit	Codecommit Repository	terraform
 CodeCommit	Codecommit Trigger	terraform
 CodeDeploy	Codedeploy App	terraform
 CodeDeploy	Codedeploy Deployment Config	terraform
 CodeDeploy	Codedeploy Deployment Group	terraform
 CodePipeline	Codepipeline	terraform
 CodePipeline	Codepipeline Webhook	terraform
 CodeStar	Codestarnotifications Notification Rule	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Cognitive Services	Cognitive Account	terraform
 Cognitive Services	Cognitive Account	terraform
 AWS Resource	Cognito Identity Pool	terraform
 AWS Resource	Cognito Identity Pool Roles Attachment	terraform
 AWS Resource	Cognito Identity Provider	terraform
 AWS Resource	Cognito Resource Server	terraform
 AWS Resource	Cognito User Group	terraform
 AWS Resource	Cognito User Pool	terraform
 AWS Resource	Cognito User Pool Client	terraform
 AWS Resource	Cognito User Pool Domain	terraform
 AWS Resource	Config Aggregate Authorization	terraform
 AWS Resource	Config Config Rule	terraform
 AWS Resource	Config Configuration Aggregator	terraform
 AWS Resource	Config Configuration Recorder	terraform
 AWS Resource	Config Configuration Recorder Status	terraform
 AWS Resource	Config Delivery Channel	terraform
 AWS Resource	Config Organization Custom Rule	terraform
 AWS Resource	Config Organization Managed Rule	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Container	Container Group	terraform
 Container	Container Group	terraform
 Container	Container Registry	terraform
 Container	Container Registry	terraform
 Container	Container Registry Webhook	terraform
 Container	Container Registry Webhook	terraform
 CosmosDB	CosmosDB Account	terraform
 CosmosDB	CosmosDB Cassandra Keyspace	terraform
 CosmosDB	CosmosDB Gremlin Database	terraform
 CosmosDB	CosmosDB Gremlin Graph	terraform
 CosmosDB	CosmosDB Mongo Collection	terraform
 CosmosDB	CosmosDB Mongo Database	terraform
 CosmosDB	CosmosDB SQL Container	terraform
 CosmosDB	CosmosDB SQL Database	terraform
 CosmosDB	CosmosDB Table	terraform
 CosmosDB	Cost Management Export Resource Group	terraform
 CosmosDB	Cost Management Export Resource Group	terraform
 AWS Resource	Cur Report Definition	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Custom	Custom Provider	terraform
 Custom	Custom Provider	terraform
 VPC	Customer Gateway	terraform
 Portal	Dashboard	terraform
 Portal	Dashboard	terraform
 Data Factory	Data Factory	terraform
 Data Factory	Data Factory Data Lake Storage	terraform
 Data Factory	Data Factory Integration Runtime	terraform
 Data Factory	Data Factory MySQL Dataset	terraform
 Data Factory	Data Factory MySQL Link	terraform
 Data Factory	Data Factory Pipeline	terraform
 Data Factory	Data Factory PostgreSQL Dataset	terraform
 Data Factory	Data Factory PostgreSQL Link	terraform
 Data Factory	Data Factory SQL Server Link	terraform
 Data Factory	Data Factory SQL Server Table	terraform
 Data Factory	Data Factory Trigger Schedule	terraform
 Data Lake	Data Lake Analytics Account	terraform
 Data Lake	Data Lake Analytics Firewall Rule	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Storage	Data Lake Filesystem	terraform
 Data Lake	Data Lake Store	terraform
 Data Lake	Data Lake Store File	terraform
 Data Lake	Data Lake Store Firewall Rule	terraform
 Data Share	Data Share Account	terraform
 Database Migration	Database Migration Project	terraform
 Database Migration	Database Migration Service	terraform
 Databricks	Databricks Workspace	terraform
 AWS Resource	Datapipeline Pipeline	terraform
 AWS Resource	Datasync Agent	terraform
 AWS Resource	Datasync Location Efs	terraform
 AWS Resource	Datasync Location Nfs	terraform
 AWS Resource	Datasync Location S3	terraform
 AWS Resource	Datasync Location Smb	terraform
 AWS Resource	Datasync Task	terraform
 DynamoDB	Dax Cluster	terraform
 DynamoDB	Dax Parameter Group	terraform
 DynamoDB	Dax Subnet Group	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 RDS	Db Cluster Snapshot	terraform
 RDS	Db Event Subscription	terraform
 RDS	Db Instance	terraform
 RDS	Db Instance Role Association	terraform
 RDS	Db Option Group	terraform
 RDS	Db Parameter Group	terraform
 RDS	Db Security Group	terraform
 RDS	Db Snapshot	terraform
 RDS	Db Subnet Group	terraform
 Compute	Dedicated Host	terraform
 Compute	Dedicated Host Group	terraform
 VPC	Default Network Acl	terraform
 VPC	Default Route Table	terraform
 VPC	Default Security Group	terraform
 VPC	Default Subnet	terraform
 VPC	Default Vpc	terraform
 VPC	Default Vpc Dhcp Options	terraform
 AWS Resource	Devicefarm Project	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 DevSpace	DevSpace Controller	terraform
 DevSpace	DevSpace Controller	terraform
 DevTest	DevTest Lab	terraform
 DevTest	DevTest Lab	terraform
 DevTest	DevTest Linux Vm	terraform
 DevTest	DevTest Linux Vm	terraform
 DevTest	DevTest Policy	terraform
 DevTest	DevTest Policy	terraform
 DevTest	DevTest Schedule	terraform
 DevTest	DevTest Schedule	terraform
 DevTest	DevTest Virtual Network	terraform
 DevTest	DevTest Virtual Network	terraform
 DevTest	DevTest Windows Vm	terraform
 DevTest	DevTest Windows Vm	terraform
 AWS Resource	Directory Service Conditional Forwarder	terraform
 AWS Resource	Directory Service Directory	terraform
 AWS Resource	Directory Service Log Subscription	terraform
 Compute	Disk Encryption Set	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 AWS Resource	Dlm Lifecycle Policy	terraform
 AWS Resource	Dms Certificate	terraform
 AWS Resource	Dms Endpoint	terraform
 AWS Resource	Dms Event Subscription	terraform
 AWS Resource	Dms Replication Instance	terraform
 AWS Resource	Dms Replication Subnet Group	terraform
 AWS Resource	Dms Replication Task	terraform
 DNS	DNS A Record	terraform
 DNS	DNS A Record	terraform
 DNS	DNS AAAA Record	terraform
 DNS	DNS CAA Record	terraform
 DNS	DNS CName Record	terraform
 DNS	DNS CName Record	terraform
 DNS	DNS MX Record	terraform
 DNS	DNS MX Record	terraform
 DNS	DNS NS Record	terraform
 DNS	DNS NS Record	terraform
 DNS	DNS PTR Record	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 DNS	DNS SRV Record	terraform
 DNS	DNS SRV Record	terraform
 DNS	DNS Txt Record	terraform
 DNS	DNS Txt Record	terraform
 DNS	DNS Zone	terraform
 DNS	DNS Zone	terraform
 AWS Resource	Docdb Cluster	terraform
 AWS Resource	Docdb Cluster Instance	terraform
 AWS Resource	Docdb Cluster Parameter Group	terraform
 AWS Resource	Docdb Cluster Snapshot	terraform
 AWS Resource	Docdb Subnet Group	terraform
 Direct Connect	Dx Bgp Peer	terraform
 Direct Connect	Dx Connection	terraform
 Direct Connect	Dx Connection Association	terraform
 Direct Connect	Dx Gateway	terraform
 Direct Connect	Dx Gateway Association	terraform
 Direct Connect	Dx Gateway Association Proposal	terraform
 Direct Connect	Dx Hosted Private Virtual Interface	terraform

Table 18 – continued from previous p

ICON	RESOURCE NAME	PROVIDER
 Direct Connect	Dx Hosted Private Virtual Interface Accepter	terraform
 Direct Connect	Dx Hosted Public Virtual Interface	terraform
 Direct Connect	Dx Hosted Public Virtual Interface Accepter	terraform
 Direct Connect	Dx Hosted Transit Virtual Interface	terraform
 Direct Connect	Dx Hosted Transit Virtual Interface Accepter	terraform
 Direct Connect	Dx Lag	terraform
 Direct Connect	Dx Private Virtual Interface	terraform
 Direct Connect	Dx Public Virtual Interface	terraform
 Direct Connect	Dx Transit Virtual Interface	terraform
 DynamoDB	Dynamodb Global Table	terraform
 DynamoDB	Dynamodb Table	terraform
 DynamoDB	DynamoDB Table	cloudFormation
 DynamoDB	Dynamodb Table Item	terraform
 EC2	Ebs Default Kms Key	terraform
 EC2	Ebs Encryption By Default	terraform
 EC2	Ebs Snapshot	terraform
 EC2	Ebs Snapshot Copy	terraform
 EC2	Ebs Volume	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 EC2	Ec2 Availability Zone Group	terraform
 EC2	EC2 Capacity Reservation	cloudFormation
 EC2	Ec2 Capacity Reservation	terraform
 EC2	EC2 Client Vpn Authorization Rule	cloudFormation
 EC2	EC2 Client Vpn Endpoint	cloudFormation
 EC2	Ec2 Client Vpn Endpoint	terraform
 EC2	Ec2 Client Vpn Network Association	terraform
 EC2	EC2 Client Vpn Route	cloudFormation
 EC2	EC2 Client Vpn Target Network Association	cloudFormation
 EC2	EC2 Customer Gateway	cloudFormation
 EC2	EC2 DHCP Options	cloudFormation
 EC2	EC2 Egress Only Internet Gateway	cloudFormation
 EC2	EC2 EIP	cloudFormation
 EC2	EC2 EIP Association	cloudFormation
 EC2	EC2 Fleet	cloudFormation
 EC2	Ec2 Fleet	terraform
 EC2	EC2 Flow Log	cloudFormation
 EC2	EC2 Host	cloudFormation

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 EC2	EC2 Instance	cloudFormation
 EC2	EC2 Internet Gateway	cloudFormation
 EC2	EC2 Launch Template	cloudFormation
 EC2	EC2 Nat Gateway	cloudFormation
 EC2	EC2 Network Acl	cloudFormation
 EC2	EC2 Network Acl Entry	cloudFormation
 EC2	EC2 Network Interface	cloudFormation
 EC2	EC2 Network Interface Attachment	cloudFormation
 EC2	EC2 Network Interface Permission	cloudFormation
 EC2	EC2 Placement Group	cloudFormation
 EC2	EC2 Route	cloudFormation
 EC2	EC2 Route Table	cloudFormation
 EC2	EC2 Security Group	cloudFormation
 EC2	EC2 Security Group Egress	cloudFormation
 EC2	EC2 Security Group Ingress	cloudFormation
 EC2	EC2 Spot Fleet	cloudFormation
 EC2	EC2 Subnet	cloudFormation
 EC2	EC2 Subnet Cidr Block	cloudFormation

Table 18 – continued from previous p

ICON	RESOURCE NAME	PROVIDER
 EC2	EC2 Subnet Network Acl Association	cloudFormation
 EC2	EC2 Subnet Route Table Association	cloudFormation
 EC2	Ec2 Traffic Mirror Filter	terraform
 EC2	Ec2 Traffic Mirror Filter Rule	terraform
 EC2	Ec2 Traffic Mirror Session	terraform
 EC2	Ec2 Traffic Mirror Target	terraform
 EC2	Ec2 Transit Gateway	terraform
 EC2	EC2 Transit Gateway	cloudFormation
 EC2	EC2 Transit Gateway Attachment	cloudFormation
 EC2	Ec2 Transit Gateway Peering Attachment	terraform
 EC2	Ec2 Transit Gateway Peering Attachment Acceptor	terraform
 EC2	Ec2 Transit Gateway Route	terraform
 EC2	EC2 Transit Gateway Route	cloudFormation
 EC2	Ec2 Transit Gateway Route Table	terraform
 EC2	EC2 Transit Gateway Route Table	cloudFormation
 EC2	Ec2 Transit Gateway Route Table Association	terraform
 EC2	EC2 Transit Gateway Route Table Association	cloudFormation
 EC2	Ec2 Transit Gateway Route Table Propagation	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 EC2	EC2 Transit Gateway Route Table Propagation	cloudFormation
 EC2	Ec2 Transit Gateway Vpc Attachment	terraform
 EC2	EC2 Transit Gateway VPC Attachment	cloudFormation
 EC2	Ec2 Transit Gateway Vpc Attachment Acceptor	terraform
 EC2	EC2 Volume	cloudFormation
 EC2	EC2 Volume Attachment	cloudFormation
 EC2	EC2 VPC	cloudFormation
 EC2	EC2 VPC Cidr Block	cloudFormation
 EC2	EC2 VPC DHCP Options Association	cloudFormation
 EC2	EC2 VPC Endpoint	cloudFormation
 EC2	EC2 VPC Endpoint Connection Notification	cloudFormation
 EC2	EC2 VPC Endpoint Service	cloudFormation
 EC2	EC2 VPC Endpoint Service Permissions	cloudFormation
 EC2	EC2 VPC Gateway Attachment	cloudFormation
 EC2	EC2 VPC Peering Connection	cloudFormation
 EC2	EC2 VPN Connection	cloudFormation
 EC2	EC2 VPN Connection Route	cloudFormation
 EC2	EC2 VPN Gateway	cloudFormation

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 EC2	EC2 VPN Gateway Route Propagation	cloudFormation
 AWS Resource	Ecr Lifecycle Policy	terraform
 AWS Resource	Ecr Repository	terraform
 AWS Resource	Ecr Repository Policy	terraform
 Elastic Container Service	Ecs Capacity Provider	terraform
 Elastic Container Service	ECS Cluster	cloudFormation
 Elastic Container Service	Ecs Cluster	terraform
 Elastic Container Service	Ecs Service	terraform
 Elastic Container Service	ECS Service	cloudFormation
 Elastic Container Service	Ecs Task Definition	terraform
 Elastic Container Service	ECS Task Definition	cloudFormation
 Elastic File System EFS	Efs File System	terraform
 Elastic File System EFS	EFS FileSystem	cloudFormation
 Elastic File System EFS	Efs Mount Target	terraform
 Elastic File System EFS	EFS Mount Target	cloudFormation
 VPC	Egress Only Internet Gateway	terraform
 EC2	Eip	terraform
 EC2	Eip Association	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Elastic Kubernetes Service	EKS Cluster	cloudFormation
 Elastic Kubernetes Service	Eks Cluster	terraform
 Elastic Kubernetes Service	Eks Fargate Profile	terraform
 Elastic Kubernetes Service	Eks Node Group	terraform
 Elastic Beanstalk	Elastic Beanstalk Application	cloudFormation
 Elastic Beanstalk	Elastic Beanstalk Application	terraform
 Elastic Beanstalk	Elastic Beanstalk Application Version	cloudFormation
 Elastic Beanstalk	Elastic Beanstalk Application Version	terraform
 Elastic Beanstalk	Elastic Beanstalk Configuration Template	cloudFormation
 Elastic Beanstalk	Elastic Beanstalk Configuration Template	terraform
 Elastic Beanstalk	Elastic Beanstalk Environment	terraform
 Elastic Beanstalk	Elastic Beanstalk Environment	cloudFormation
 ElastiCache	ElastiCache Cluster	cloudFormation
 ElastiCache	Elasticache Cluster	terraform
 ElastiCache	Elasticache Parameter Group	terraform
 ElastiCache	ElastiCache Parameter Group	cloudFormation
 ElastiCache	Elasticache Replication Group	terraform
 ElastiCache	ElastiCache Replication Group	cloudFormation

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 ElastiCache	ElastiCache Security Group	terraform
 ElastiCache	ElastiCache Security Group	cloudFormation
 ElastiCache	ElastiCache Security Group Ingress	cloudFormation
 ElastiCache	ElastiCache Subnet Group	terraform
 ElastiCache	ElastiCache Subnet Group	cloudFormation
 Elasticsearch Service	Elasticsearch Domain	terraform
 Elasticsearch Service	Elasticsearch Domain Policy	terraform
 AWS Resource	Elastictranscoder Pipeline	terraform
 AWS Resource	Elastictranscoder Preset	terraform
 Elastic Load Balancing	Elb	terraform
 Elastic Load Balancing	Elb Attachment	terraform
 EMR	EMR Cluster	cloudFormation
 EMR	Emr Cluster	terraform
 EMR	EMR Instance Fleet Config	cloudFormation
 EMR	Emr Instance Group	terraform
 EMR	EMR Instance Group Config	cloudFormation
 EMR	Emr Security Configuration	terraform
 EMR	EMR Security Configuration	cloudFormation

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 EMR	EMR Step	cloudFormation
 Messaging	Event Grid Domain	terraform
 Messaging	Event Grid Event Subscription	terraform
 Messaging	Event Grid Topic	terraform
 Messaging	Event Hub	terraform
 Messaging	Event Hub Authorization Rule	terraform
 Messaging	Event Hub Consumer Group	terraform
 Messaging	Event Hub Disaster Recovery Config	terraform
 Messaging	Event Hub Namespace	terraform
 Messaging	Event Hub Namespace Authorization Rule	terraform
 Network	Express Route Circuit	terraform
 Network	Express Route Circuit	terraform
 Network	Express Route Circuit Authorization	terraform
 Network	Express Route Circuit Peering	terraform
 Network	Express Route Circuit Peering	terraform
 Network	Express Route Gateway	terraform
 Network	Express Route Gateway	terraform
 Network	Firewall	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Network	Firewall Application Rule Collection	terraform
 Network	Firewall Nat Rule Collection	terraform
 Network	Firewall Nat Rule Collection	terraform
 Network	Firewall Network Rule Collection	terraform
 VPC	Flow Log	terraform
 AWS Resource	Fms Admin Account	terraform
 Front Door	Front Door	terraform
 Front Door	Front Door	terraform
 Front Door	Front Door Firewall Policy	terraform
 Front Door	Front Door Firewall Policy	terraform
 FSx	Fsx Lustre File System	terraform
 FSx	Fsx Windows File System	terraform
 Azure Resource	Function App	terraform
 Azure Resource	Function App Slot	terraform
 AWS Resource	Gamelift Alias	terraform
 AWS Resource	Gamelift Build	terraform
 AWS Resource	Gamelift Fleet	terraform
 AWS Resource	Gamelift Game Session Queue	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 S3 Glacier	Glacier Vault	terraform
 S3 Glacier	Glacier Vault Lock	terraform
 AWS Resource	Globalaccelerator Accelerator	terraform
 AWS Resource	Globalaccelerator Endpoint Group	terraform
 AWS Resource	Globalaccelerator Listener	terraform
 Glue	Glue Catalog Database	terraform
 Glue	Glue Catalog Table	terraform
 Glue	Glue Classifier	terraform
 Glue	Glue Connection	terraform
 Glue	Glue Crawler	terraform
 Glue	Glue Job	terraform
 Glue	Glue Security Configuration	terraform
 Glue	Glue Trigger	terraform
 Glue	Glue Workflow	terraform
 AWS Resource	Guardduty Detector	terraform
 AWS Resource	Guardduty Invite Acceptor	terraform
 AWS Resource	Guardduty Ipset	terraform
 AWS Resource	Guardduty Member	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 AWS Resource	Guardduty Organization Admin Account	terraform
 AWS Resource	Guardduty Organization Configuration	terraform
 AWS Resource	Guardduty Threatintelset	terraform
 HDInsight	HD Insight Hadoop Cluster	terraform
 HDInsight	HD Insight HBase Cluster	terraform
 HDInsight	HD Insight Kafka Cluster	terraform
 HDInsight	HD Insight ML Services Cluster	terraform
 HDInsight	HD Insight Query Cluster	terraform
 HDInsight	HD Insight R Cluster	terraform
 HDInsight	HD Insight Spark Cluster	terraform
 HDInsight	HD Insight Storm Cluster	terraform
 Healthcare API	Healthcare Service	terraform
 Healthcare API	Healthcare Service	terraform
 Storage	HPC Cache	terraform
 Storage	HPC Cache Blob Target	terraform
 Storage	HPC Cache Blob Target	terraform
 Storage	HPC Cache NFS Target	terraform
 Storage	HPC Cache NFS Target	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 AWS Resource	IAM Access Key	cloudFormation
 AWS Resource	Iam Access Key	terraform
 AWS Resource	Iam Account Alias	terraform
 AWS Resource	Iam Account Password Policy	terraform
 AWS Resource	IAM Group	cloudFormation
 AWS Resource	Iam Group	terraform
 AWS Resource	Iam Group Membership	terraform
 AWS Resource	Iam Group Policy	terraform
 AWS Resource	Iam Group Policy Attachment	terraform
 AWS Resource	Iam Instance Profile	terraform
 AWS Resource	IAM Instance Profile	cloudFormation
 AWS Resource	IAM Managed Policy	cloudFormation
 AWS Resource	Iam Openid Connect Provider	terraform
 AWS Resource	Iam Policy	terraform
 AWS Resource	IAM Policy	cloudFormation
 AWS Resource	Iam Policy Attachment	terraform
 AWS Resource	Iam Role	terraform
 AWS Resource	IAM Role	cloudFormation

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 AWS Resource	Iam Role Policy	terraform
 AWS Resource	Iam Role Policy Attachment	terraform
 AWS Resource	Iam Saml Provider	terraform
 AWS Resource	Iam Server Certificate	terraform
 AWS Resource	Iam Service Linked Role	terraform
 AWS Resource	IAM Service Linked Role	cloudFormation
 AWS Resource	Iam User	terraform
 AWS Resource	IAM User	cloudFormation
 AWS Resource	IAM User Group Addition	cloudFormation
 AWS Resource	Iam User Group Membership	terraform
 AWS Resource	Iam User Login Profile	terraform
 AWS Resource	Iam User Policy	terraform
 AWS Resource	Iam User Policy Attachment	terraform
 AWS Resource	Iam User Ssh Key	terraform
 Compute	Image	terraform
 AWS Resource	Inspector Assessment Target	terraform
 AWS Resource	Inspector Assessment Template	terraform
 AWS Resource	Inspector Resource Group	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 EC2	Instance	terraform
 VPC	Internet Gateway	terraform
 IoT Central Application	IOT Central Application	terraform
 AWS Resource	Iot Certificate	terraform
 IoT Hub	IOT Hub	terraform
 IoT Hub	IOT Hub Access Policy	terraform
 IoT Hub	IOT Hub Consumer Group	terraform
 IoT Hub	IOT Hub DPS	terraform
 IoT Hub	IOT Hub DPS Certificate	terraform
 IoT Hub	IOT Hub Endpoint Event Hub	terraform
 IoT Hub	IOT Hub Endpoint Event Hub	terraform
 IoT Hub	IOT Hub Endpoint Queue	terraform
 IoT Hub	IOT Hub Endpoint Storage	terraform
 IoT Hub	IOT Hub Endpoint Topic	terraform
 IoT Hub	IOT Hub Fallback Route	terraform
 IoT Hub	IOT Hub Route	terraform
 IoT Hub	IOT Hub Shared Access Policy	terraform
 AWS Resource	Iot Policy	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 AWS Resource	Iot Policy Attachment	terraform
 AWS Resource	Iot Role Alias	terraform
 AWS Resource	Iot Thing	terraform
 AWS Resource	Iot Thing Principal Attachment	terraform
 AWS Resource	Iot Thing Type	terraform
 AWS Resource	Iot Topic Rule	terraform
 EC2	Key Pair	terraform
 Key Vault	Key Vault	terraform
 Key Vault	Key Vault Access Policy	terraform
 Key Vault	Key Vault Access Policy	terraform
 Key Vault	Key Vault Certificate	terraform
 Key Vault	Key Vault Certificate	terraform
 Key Vault	Key Vault Key	terraform
 Key Vault	Key Vault Key	terraform
 Key Vault	Key Vault Secret	terraform
 Key Vault	Key Vault Secret	terraform
 Kinesis	Kinesis Analytics Application	terraform
 Kinesis	Kinesis Firehose Delivery Stream	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Kinesis	Kinesis Stream	terraform
 Kinesis	Kinesis Stream	cloudFormation
 Kinesis	Kinesis Stream Consumer	cloudFormation
 Kinesis	Kinesis Video Stream	terraform
 AWS Resource	Kms Alias	terraform
 AWS Resource	Kms Ciphertext	terraform
 AWS Resource	Kms External Key	terraform
 AWS Resource	Kms Grant	terraform
 AWS Resource	Kms Key	terraform
 Container	Kubernetes Cluster	terraform
 Container	Kubernetes Cluster	terraform
 Container	Kubernetes Node Pool	terraform
 Container	Kubernetes Node Pool	terraform
 Data Explorer	Kusto Cluster	terraform
 Data Explorer	Kusto Database	terraform
 Data Explorer	Kusto Database Principal	terraform
 Data Explorer	Kusto Event Data Connection	terraform
 Lambda	Lambda Alias	cloudFormation

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Lambda	Lambda Alias	terraform
 Lambda	Lambda Event Source Mapping	cloudFormation
 Lambda	Lambda Event Source Mapping	terraform
 Lambda	Lambda Function	cloudFormation
 Lambda	Lambda Function	terraform
 Lambda	Lambda Function Event Invoke Config	terraform
 Lambda	Lambda Layer Version	terraform
 Lambda	Lambda Layer Version	cloudFormation
 Lambda	Lambda Layer Version Permission	cloudFormation
 Lambda	Lambda Permission	terraform
 Lambda	Lambda Permission	cloudFormation
 Lambda	Lambda Provisioned Concurrency Config	terraform
 Lambda	Lambda Version	cloudFormation
 EC2	Launch Configuration	terraform
 EC2	Launch Template	terraform
 Elastic Load Balancing	Lb	terraform
 Elastic Load Balancing	Lb Cookie Stickiness Policy	terraform
 Elastic Load Balancing	Lb Listener	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Elastic Load Balancing	Lb Listener Certificate	terraform
 Elastic Load Balancing	Lb Listener Rule	terraform
 Elastic Load Balancing	Lb Ssl Negotiation Policy	terraform
 Elastic Load Balancing	Lb Target Group	terraform
 Elastic Load Balancing	Lb Target Group Attachment	terraform
 AWS Resource	Licensemanager Association	terraform
 AWS Resource	Licensemanager License Configuration	terraform
 Amazon Lightsail	Lightsail Domain	terraform
 Amazon Lightsail	Lightsail Instance	terraform
 Amazon Lightsail	Lightsail Key Pair	terraform
 Amazon Lightsail	Lightsail Static Ip	terraform
 Amazon Lightsail	Lightsail Static Ip Attachment	terraform
 Compute	Linux VM Scale Set	terraform
 Load Balancer	Load Balancer	terraform
 Load Balancer	Load Balancer	terraform
 Load Balancer	Load Balancer Address Pool	terraform
 Load Balancer	Load Balancer Address Pool	terraform
 Elastic Load Balancing	Load Balancer Backend Server Policy	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Elastic Load Balancing	Load Balancer Listener Policy	terraform
 Load Balancer	Load Balancer Nat Pool	terraform
 Load Balancer	Load Balancer Nat Pool	terraform
 Load Balancer	Load Balancer Nat Rule	terraform
 Load Balancer	Load Balancer Nat Rule	terraform
 Load Balancer	Load Balancer Outbound Rule	terraform
 Load Balancer	Load Balancer Outbound Rule	terraform
 Elastic Load Balancing	Load Balancer Policy	terraform
 Load Balancer	Load Balancer Probe	terraform
 Load Balancer	Load Balancer Probe	terraform
 Load Balancer	Load Balancer Rule	terraform
 Load Balancer	Load Balancer Rule	terraform
 Network	Local Network Gateway	terraform
 Log Analytics	Log Analytics Linked Service	terraform
 Log Analytics	Log Analytics Linked Service	terraform
 Log Analytics	Log Analytics Solution	terraform
 Log Analytics	Log Analytics Solution	terraform
 Log Analytics	Log Analytics Windows Event	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Log Analytics	Log Analytics Windows Event	terraform
 Log Analytics	Log Analytics Windows Performance Counter	terraform
 Log Analytics	Log Analytics Windows Performance Counter	terraform
 Log Analytics	Log Analytics Workspace	terraform
 Log Analytics	Log Analytics Workspace	terraform
 Logic App	Logic App Custom Action	terraform
 Logic App	Logic App Custom Action	terraform
 Logic App	Logic App Custom Trigger	terraform
 Logic App	Logic App Custom Trigger	terraform
 Logic App	Logic App Http Action	terraform
 Logic App	Logic App Http Action	terraform
 Logic App	Logic App Http Request	terraform
 Logic App	Logic App Http Request	terraform
 Logic App	Logic App Trigger Recurrence	terraform
 Logic App	Logic App Trigger Recurrence	terraform
 Logic App	Logic App Workflow	terraform
 Logic App	Logic App Workflow	terraform
 Machine Learning	Machine Learning Workspace	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 AWS Resource	Macie Member Account Association	terraform
 AWS Resource	Macie S3 Bucket Association	terraform
 VPC	Main Route Table Association	terraform
 Maintenance	Maintenance Configuration	terraform
 Maintenance	Maintenance Configuration	terraform
 Compute	Managed Application	terraform
 Compute	Managed Disk	terraform
 Management	Management Group	terraform
 Management	Management Group	terraform
 Management	Management Lock	terraform
 Management	Management Lock	terraform
 Maps	Maps Account	terraform
 Maps	Maps Account	terraform
 Database	MariaDb Configuration	terraform
 Database	MariaDb Database	terraform
 Database	MariaDb Firewall Rule	terraform
 Database	MariaDb Server	terraform
 Database	MariaDb Virtual Network Rule	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Compute	Marketplace Agreement	terraform
 AWS Resource	Media Convert Queue	terraform
 AWS Resource	Media Package Channel	terraform
 Media	Media Services Account	terraform
 Media	Media Services Account	terraform
 AWS Resource	Media Store Container	terraform
 AWS Resource	Media Store Container Policy	terraform
 Monitor	Monitor Action Group	terraform
 Monitor	Monitor Action Group	terraform
 Monitor	Monitor Activity Log Alert	terraform
 Monitor	Monitor Activity Log Alert	terraform
 Monitor	Monitor Autoscale Setting	terraform
 Monitor	Monitor Autoscale Setting	terraform
 Monitor	Monitor Diagnostic Setting	terraform
 Monitor	Monitor Diagnostic Setting	terraform
 Monitor	Monitor Log Profile	terraform
 Monitor	Monitor Log Profile	terraform
 Monitor	Monitor Metric Alert	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Monitor	Monitor Metric Alert	terraform
 Monitor	Monitor Scheduled Query Rules Alert	terraform
 Monitor	Monitor Scheduled Query Rules Alert	terraform
 Monitor	Monitor Scheduled Query Rules Log	terraform
 Monitor	Monitor Scheduled Query Rules Log	terraform
 MQ	MQ Broker	cloudFormation
 MQ	Mq Broker	terraform
 MQ	MQ Configuration	cloudFormation
 MQ	Mq Configuration	terraform
 MQ	MQ Configuration Association	cloudFormation
 Database	MS SQL Database	terraform
 Database	MS SQL Elastic Pool	terraform
 Database	MS SQL Security Alert Policy	terraform
 Database	MS SQL Server	terraform
 Database	MS SQL Vm	terraform
 Database	MS SQL Vulnerability Assessment	terraform
 Database	MS SQL Vulnerability Assessment Rule Baseline	terraform
 AWS Resource	Msk Cluster	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 AWS Resource	Msk Configuration	terraform
 Database	MySQL Configuration	terraform
 Database	MySQL Database	terraform
 Database	MySQL Firewall Rule	terraform
 Database	MySQL Server	terraform
 Database	MySQL Virtual Network Rule	terraform
 Messaging	Namespace Authorization Rule	terraform
 Network	Nat Gateway	terraform
 VPC	Nat Gateway	terraform
 Neptune	Neptune Cluster	terraform
 Neptune	Neptune Cluster Instance	terraform
 Neptune	Neptune Cluster Parameter Group	terraform
 Neptune	Neptune Cluster Snapshot	terraform
 Neptune	Neptune DB Cluster	cloudFormation
 Neptune	Neptune DB Cluster Parameter Group	cloudFormation
 Neptune	Neptune DB Instance	cloudFormation
 Neptune	Neptune DB Parameter Group	cloudFormation
 Neptune	Neptune DB Subnet Group	cloudFormation

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Neptune	Neptune Event Subscription	terraform
 Neptune	Neptune Parameter Group	terraform
 Neptune	Neptune Subnet Group	terraform
 NetApp	NetApp Account	terraform
 NetApp	NetApp Account	terraform
 NetApp	NetApp Pool	terraform
 NetApp	NetApp Snapshot	terraform
 NetApp	NetApp Snapshot	terraform
 NetApp	NetApp Volume	terraform
 NetApp	NetApp Volume	terraform
 VPC	Network Acl	terraform
 VPC	Network Acl Rule	terraform
 Network	Network DDos Protection Plan	terraform
 Network	Network Gateway	terraform
 Network	Network Interface	terraform
 VPC	Network Interface	terraform
 Network	Network Interface Application Gateway Backend Address Pool Association	terraform
 Network	Network Interface Application Security Group Association	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 VPC	Network Interface Attachment	terraform
 Network	Network Interface Backend Address Pool Association	terraform
 Network	Network Interface Nat Rule Association	terraform
 Network	Network Interface Security Group Association	terraform
 VPC	Network Interface Sg Attachment	terraform
 Network	Network Packet Capture	terraform
 Network	Network Profile	terraform
 Network	Network Security Group	terraform
 Network	Network Security Rule	terraform
 Network	Network Watcher	terraform
 Network	Network Watcher Flow Log	terraform
 Messaging	Notification Hub	terraform
 Messaging	Notification Hub Authorization Rule	terraform
 Messaging	Notification Hub Namespace	terraform
 AWS Resource	Opsworks Application	terraform
 AWS Resource	Opsworks Custom Layer	terraform
 AWS Resource	Opsworks Ganglia Layer	terraform
 AWS Resource	Opsworks Haproxy Layer	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 AWS Resource	Opsworks Instance	terraform
 AWS Resource	Opsworks Java App Layer	terraform
 AWS Resource	Opsworks Memcached Layer	terraform
 AWS Resource	Opsworks Mysql Layer	terraform
 AWS Resource	Opsworks Nodejs App Layer	terraform
 AWS Resource	Opsworks Permission	terraform
 AWS Resource	Opsworks Php App Layer	terraform
 AWS Resource	Opsworks Rails App Layer	terraform
 AWS Resource	Opsworks Rds Db Instance	terraform
 AWS Resource	Opsworks Stack	terraform
 AWS Resource	Opsworks Static Web Layer	terraform
 AWS Resource	Opsworks User Profile	terraform
 AWS Resource	Organizations Account	terraform
 AWS Resource	Organizations Organization	terraform
 AWS Resource	Organizations Organizational Unit	terraform
 AWS Resource	Organizations Policy	terraform
 AWS Resource	Organizations Policy Attachment	terraform
 Network	Packet Capture	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 AWS Resource	Pinpoint Adm Channel	terraform
 AWS Resource	Pinpoint Apns Channel	terraform
 AWS Resource	Pinpoint Apns Sandbox Channel	terraform
 AWS Resource	Pinpoint Apns Voip Channel	terraform
 AWS Resource	Pinpoint Apns Voip Sandbox Channel	terraform
 AWS Resource	Pinpoint App	terraform
 AWS Resource	Pinpoint Baidu Channel	terraform
 AWS Resource	Pinpoint Email Channel	terraform
 AWS Resource	Pinpoint Event Stream	terraform
 AWS Resource	Pinpoint Gcm Channel	terraform
 AWS Resource	Pinpoint Sms Channel	terraform
 Compute	Placement Group	terraform
 EC2	Placement Group	terraform
 Network	Point To Site VPN Gateway	terraform
 Policy	Policy Assignment	terraform
 Policy	Policy Assignment	terraform
 Policy	Policy Definition	terraform
 Policy	Policy Definition	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Policy	Policy Remediation	terraform
 Policy	Policy Remediation	terraform
 Policy	Policy Set Definition	terraform
 Policy	Policy Set Definition	terraform
 Database	PostgreSQL Configuration	terraform
 Database	PostgreSQL Database	terraform
 Database	PostgreSQL Firewall Rule	terraform
 Database	PostgreSQL Server	terraform
 Database	PostgreSQL Virtual Network Rule	terraform
 PowerBI	PowerBI	terraform
 Private DNS	Private DNS A Record	terraform
 Private DNS	Private DNS A Record	terraform
 Private DNS	Private DNS AAAA Record	terraform
 Private DNS	Private DNS AAAA Record	terraform
 Private DNS	Private DNS CName Record	terraform
 Private DNS	Private DNS CName Record	terraform
 Private DNS	Private DNS MX Record	terraform
 Private DNS	Private DNS MX Record	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Private DNS	Private DNS Ptr Record	terraform
 Private DNS	Private DNS Ptr Record	terraform
 Private DNS	Private DNS Srv Record	terraform
 Private DNS	Private DNS Srv Record	terraform
 Private DNS	Private DNS Text Record	terraform
 Private DNS	Private DNS Text Record	terraform
 Private DNS	Private DNS Zone	terraform
 Private DNS	Private DNS Zone	terraform
 Private DNS	Private DNS Zone Virtual Network Link	terraform
 Private DNS	Private DNS Zone Virtual Network Link	terraform
 Network	Private Endpoint	terraform
 Network	Private Link Service	terraform
 Network	Private Link Service	terraform
 Elastic Load Balancing	Proxy Protocol Policy	terraform
 Network	Public IP	terraform
 Network	Public IP Prefix	terraform
 AWS Resource	Qldb Ledger	terraform
 Messaging	Queue Authorization Rule	terraform

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 Messaging	Queue Authorization Rule	terraform
 Quicksight	Quicksight Group	terraform
 Quicksight	Quicksight User	terraform
 AWS Resource	Ram Principal Association	terraform
 AWS Resource	Ram Resource Association	terraform
 AWS Resource	Ram Resource Share	terraform
 AWS Resource	Ram Resource Share Acceptor	terraform
 RDS	Rds Cluster	terraform
 RDS	Rds Cluster Endpoint	terraform
 RDS	Rds Cluster Instance	terraform
 RDS	Rds Cluster Parameter Group	terraform
 RDS	RDS DB Cluster	cloudFormation
 RDS	RDS DB Cluster ParameterGroup	cloudFormation
 RDS	RDS DB Instance	cloudFormation
 RDS	RDS DB Parameter Group	cloudFormation
 RDS	RDS DB Security Group	cloudFormation
 RDS	RDS DB Security Group Ingress	cloudFormation
 RDS	RDS DB Subnet Group	cloudFormation

Table 18 – continued from previous page

ICON	RESOURCE NAME	PROVIDER
 RDS	RDS Event Subscription	cloudFormation
 RDS	Rds Global Cluster	terraform
 RDS	RDS Option Group	cloudFormation
 Recovery	Recovery Services Vault	terraform
 Recovery	Recovery Services Vault	terraform
 Recovery	Redis Cache	terraform
 Recovery	Redis Firewall Rule	terraform
 Redshift	Redshift Cluster	cloudFormation
 Redshift	Redshift Cluster	terraform
 Redshift	Redshift Cluster Parameter Group	cloudFormation
 Redshift	Redshift Cluster Security Group	cloudFormation
 Redshift	Redshift Cluster Security Group Ingress	cloudFormation
 Redshift	Redshift Cluster Subnet Group	cloudFormation

Bare Metal

Hosts Virtual Machines Containers Resources **Bare Metal**

Search  All OS Types

ACTIONS + BARE METAL 

Section Features & Filters *Record Search Custom Views CSV Export JSON Export Paging Config*

- Standard Filters: *OS Type*
- Advanced Filters: *Status Managed Server Type*

Bare Metal hosts are from discovered, PXE Boot or manually added Bare Metal hosts. Bare Metal hosts that are also Hypervisors will be listed in the Hosts section.

5.5.5 Network

Networks

Infrastructure > Network > Networks

Overview

The Networks section is for configuring networks across all clouds in Morpheus. Existing networks from Clouds added in Morpheus will auto-populate in the Networks section.

Networks can be configured for DHCP or Static IP assignment, assigned IP pools, and configured for visibility and account assignment for multi-tenancy usage. Inactive Networks are unavailable for provisioning use. In addition, Morpheus allows administrators to restrict management of Morpheus-created Networks through Role permissions.

Configuring Networks

DHCP

To configure a network for DHCP:

1. Navigate to *Infrastructure > Network > Networks*
2. Search for the target network
3. Edit the Network by either:
 - Select *Actions > Edit*
 - Select the Network, then select *Edit*
4. In the Network Config modal, set the DHCP flag as Active (default)
5. Save Changes

Important: The DHCP flag tells Morpheus this network has a DHCP server assigning IP Addresses to hosts. Morpheus does not act as the DHCP server, and provisioning to a network that has the DHCP server flag active in Morpheus, but no DHCP server actually on the network will in most cases cause the instance to not receive an IP address.

Note: When selecting a network with DHCP enabled during provisioning, “DHCP” will populate to the right of the selected network:

Static and IP Pools

To configure a network for Static IP Assignment:

1. Navigate to *Infrastructure > Network > Networks*
2. Search for the target network
3. Edit the Network by either:
 - Select *Actions > Edit*
 - Select the Network, then select *Edit*
4. In the Network Config modal, add the following:
 - Gateway
 - DNS Primary
 - DNS Secondary
 - CIDR ex 10.10.10.0/22
 - VLAN ID (if necessary)
 - Network Pool * Leave as “choose a pool” for entering a static IP while provisioning * Select a Pool to use a pre-configured Morpheus or IPAM Integration IP Pool
 - The Permissions settings are used for Multi-Tenant resource configuration
 - Leave settings as default if used in a single-tenant environment (only one Tenant in your Morpheus appliance)
 - To share this network across all accounts in a multi-tenant environment, select the Master Tenant and set the Visibility to Public
 - To assign this network to be used by only one account in a multi-tenant environment, select the account and set visibility to Private
 - Active
 - Leave as enabled to use this network
 - Disable the active flag to remove this network from available network options
5. Save Changes

Note: When selecting a network with DHCP disabled and no IP Pool assigned during provisioning, an IP entry field will populate to the right of the selected network(s):

Note: When selecting a network with an IP Pool assigned during provisioning, the name of the IP pool will populate to the right of the selected network(s). IP Pools override DHCP.

Advanced Options (Search Domains)

Search domains are appended to DNS searches when a **non** fully qualified domain name (short name) is queried. Search domains can be entered as comma separated values, which will be added to DNS configurations, such as */etc/resolv.conf*. These domains are injected via cloud-init or other method chosen for the virtual image.

Guest Console SSH Tunnel

In some scenarios, instances that are segregated from the Morpheus appliance by port restrictions, or other mechanisms, can cause difficulties to access the guest console via the Morpheus web UI. Guest Console SSH Tunnel settings allow the administrator to configure a jump host's settings that is dual-homed, accessible by Morpheus but also resides on the segregated network. When the guest console is configured with the SSH protocol, the traffic will be routed to the jump host, which will then relay to the target instance.

GUEST CONSOLE JUMP HOST DNS hostname or IP of the jump host to relay the traffic

GUEST CONSOLE JUMP PORT Port override, if different than 22 for SSH

GUEST CONSOLE JUMP USERNAME Username used to authenticate to the jump host

GUEST CONSOLE JUMP PASSWORD Password that is used with the username to authenticate to the jump host

GUEST CONSOLE KEYPAIR Keypair saved in Morpheus to be used in lieu of, or in addition to, the password to the jump host, which is associated with the configured username. Keypairs can be imported at: Infrastructure > Trust > Key Pairs

Subnets

Subnet details can be viewed from the *SUBNETS* tab on the detail page of a specific network. From the *SUBNETS* tab, Morpheus allows the user to search and edit existing subnets.

In an Azure VNet, you can also create new subnets with the *+ADD* button.

CREATE SUBNET ✕

TYPE

SUBNET NAME

SUBNET CIDR

The subnet's address range in CIDR notation (e.g. 192.168.1.0/24). It must be contained by the address space of the virtual network.

ACTIVE

CIDR

DHCP SERVER

ALLOW IP OVERRIDE

NETWORK POOL

▶ Group Access

▶ Tenant Permissions

SAVE CHANGES

Network Groups

Network Groups are useful for grouping networks during provisioning and scaling or grouping availability subnets together such that during provisioning vm's within an instance can be round robin provisioned across availability zones.

Adding Network Groups

1. Navigate to *Infrastructure > Network > Networks Groups*
2. Select *ADD*
3. Enter the following:

Group info:

- *Name*: Name of the Network Group in Morpheus
- *Description*: Details of the Network Group in Morpheus

Networks

- Search for and select target Networks for the Network Group
- Search for and select target Subnets for the Network Group

Group Access

- Set Group Access and Defaults for the Network Group

Tenant Permissions

- Set Tenant Visibility for Network Group

2. Select *SAVE CHANGES*

Routers

Overview

Routers can be viewed, created, and managed from the Routers tab of the Infrastructure > Networks page. Morpheus supports the creation of the following router types depending on networks that are currently configured:

- Amazon Internet Gateway
- Huawei Router
- Neutron Router
- NSX Edge Gateway
- NSX Edge Logical Router
- NSX-T Cloud Tier0 Gateway
- NSX-T Cloud Tier1 Gateway
- NSX-T Tier0 Gateway
- NSX-T Tier1 Gateway
- Open Telekom Router

Create New Router

1. Navigate to Infrastructure > Networks > Routers tab
2. Click + *ADD*
3. Select the router type and complete the fields on the resulting modal
4. Once complete, click *ADD NETWORK ROUTER*

Editing Existing Routers

1. Navigate to Infrastructure > Networks > Routers tab
2. Click on the pencil icon for the appropriate router
3. After editing router fields, click *SAVE*

Deleting Existing Routers

1. Navigate to Infrastructure > Networks > Routers tab
2. Click on the trash can icon for the appropriate router
3. Acknowledge the pop-up banner ensuring you wish to delete the router

IP Pools

Infrastructure > Network > IP Pools

Overview

The IP Pools tab in the Networks section allows you to create Morpheus-type IP Pools (which is an IP address range Morpheus can use to assign available static IP addresses to Instances) and NSX-T IP Pools. The IP Pool section also displays pools synced from IPAM integrations like Infoblox, Bluecat and others.

To add a Morpheus Network Pool

1. Click + *ADD* in Infrastructure > Network > IP Pools
2. **Enter the following:**
 - Name** A friendly name for the IP Pool in Morpheus.
 - Pool Type** Currently Morpheus-type IP Pools and NSX-T IP Pools (with a configured integration) can be created directly from Morpheus
 - Starting Address** The starting IP address of the IP Pool address range. ex: 192.168.0.2
 - Ending Address:** The ending IP address of the IP Pool address range. ex: 192.168.0.255
3. Save Changes

Note: Multiple Address Ranges can be added to a pool by selecting the + icon next to the address range.

After saving the IP pool will be available for assignment to networks in the NETWORK POOL dropdown when adding or editing a network.

Floating IPs

Overview

Morpheus supports sync and management of floating IP addresses for OpenStack, Huawei, and OTC Clouds. When these Clouds are integrated and floating IPs are present, Morpheus will automatically sync them in. Once synced, floating IPs are viewable from their own list page (Infrastructure > Network > Floating IPs) and related options are presented during provisioning, teardown, and from Instance detail pages.

Note: The Floating IPs tab is present only when supported Clouds are integrated and floating IPs are available. Additional Cloud support is planned for the future.

Floating IPs List Page

All Floating IPs known to Morpheus can be viewed on the Floating IPs List Page (Infrastructure > Network > Floating IPs). From the Floating IPs list page we can see the following:

- **IP ADDRESS:** The address for the floating IP synced from a supported Cloud
- **CLOUD:** The Cloud integration the floating IP was synced from
- **STATUS:** “Free” when the floating IP is available and “Assigned” when the floating IP is currently assigned to a workload
- **VM:** For assigned floating IPs, the VM which currently has the floating IP attached

Free floating IPs will have a gear icon () at the end of the row. Click the gear icon and select “Release Floating IP” to release from within the source cloud and remove the entry from the Floating IPs list.

IP ADDRESS	CLOUD	STATUS	VM
10.3	QA OpenStack	Free	
10.3	QA OpenStack (all)	Free	
10.3	QA OpenStack	Free	
10.3	QA OpenStack (all)	Free	
10.3	QA OpenStack (all)	Assigned	sa-opv3-ubuntu-2
10.3	QA OpenStack	Assigned	sa-opv3-ubuntu-2
10.3	QA Openstack Yoga	Assigned	
190	QA Huawei	Free	
80.1	QA OpenTelekom	Assigned	dd-otc2-ubuntu-107
80.1	QA OpenTelekom	Assigned	dd-otc2-ubuntu-106

Working with Floating IPs

When provisioning to supported Clouds, Morpheus will give the option to attach a floating IP at provision time. From the CONFIGURE tab of the provisioning wizard for supported Clouds, select the desired floating IP.

CREATE INSTANCE ✕

TYPE > GROUP > **CONFIGURE** > AUTOMATION > REVIEW

Configuration Options

VERSION: 22.04

LAYOUT: OpenStack VM

PLAN: m1.large
Cores: 4 Memory: 8 GB Price: \$168.00 / Month

RESOURCE POOL: morpheus

VOLUMES: 80 GB Volume +

NETWORKS: tm-open-net DHCP +

AVAILABILITY ZONE: nova

SECURITY GROUP: default

SERVER GROUP (AFFINITY): Select

FLOATING IP: Select

- external
- Available Ips
- 10.3 [redacted]
- 10.3 [redacted]

▶ User Config

▶ Advanced Options

PREVIOUS NEXT

During Instance teardown, Morpheus gives the option to release the floating IP.

DELETE HOST?



Warning! Deleting this host `sa-opv3-ubuntu-2` with "Remove Infrastructure" enabled will permanently delete it from the Cloud. To only delete the record but leave in the Cloud, uncheck "Remove Infrastructure". If "Inventory Existing Instances" is enabled on the Cloud, the host will be re-synced as discovered.

- Remove Infrastructure
- Remove Associated Instances
- Release Floating IP
- Force Delete

Type **DELETE** to confirm

Domains

Infrastructure > Network > Domains

Overview

The Domains section is for creating and managing domains for use in Morpheus. Domains are used for setting FQDNs, joining Domains, and creating DNS records. The Domains section is also a multi-tenant endpoint for managing domain settings across multiple tenants.

- Domains are synced in from Cloud, DNS and Network Integrations. Domains can also be user created.
- Active Domains are available for selection in the Domain dropdown when provisioning an Instance
- Default Domains can be set for Clouds and Networks in their Advanced Options sections.
- Images can be flagged to Auto-Join Domains in the Infrastructure > Network > Domains section

Important: For an Instance to auto-join a Domain, a Domain must set in the Advanced Options section of the Cloud or Network used when provisioning

Adding Domains

1. Navigate to **Infrastructure > Network > Domains**
2. Select + *ADD*
3. Enter the following:

Domain Name Ex. demo.example.com

Description Descriptive metadata for use in Morpheus

Display Name Overrides the displayed name in domain selection components, which is useful when using many OU paths

Public Zone Check for Public Zones, leave unchecked for Private Zones

Workflow Select an existing Workflow which will be applied to Instances at provision time when they are associated with the domain. This is useful for any domain-related scripting you may currently use. For example, you may want to ensure a machine is removed from the domain when it's torn down which could be accomplished by creating a Provisioning Workflow (with teardown phase Tasks) and associating the Workflow with the domain

Active Active Domains are available for selection in Domain selection fields across Morpheus. In-active Domains are removed from Domain selection fields.

Join Domain Controller Enable to have Windows instances join a Domain

Username Admin user for Domain. domain\username format required when specifying OU Path

Password Password for DC user account

DC Server (optional) Specify the URL or Path of the DC Server

OU Path (optional) Enter the OU Path for the connection string.

Guest Username (optional) If set, this will change the provisioned hosts RPC Service User after domain join. Useful when a domain policy disables the Administrator account that typically would be set as the RPC user on a host record. Morpheus will update the RPC username and password on the host(s) record after domain join with the specified Guest Username and Guest Password. The RPC username and password are used for auth during Remote Procedure Call (RPC) executions over winrm, ssh and guest tools.

Guest Password (optional) The password for the Guest User account indicated in the prior field

4. Click *Save Changes*

The Domain has been added and will be selectable in the Domain dropdown during provisioning, and in Cloud and Network settings.

Editing Domain Permissions

To edit visibility permissions for a domain, navigate to **Infrastructure > Network > Domains**. In the row for the selected domain, click **MORE > Permissions**. Within the Permissions modal, set Group and Tenant access permissions.

Note: Only resources assigned to the Master Tenant can be set as Publicly visible. If the Tenant assigned is not the Master Tenant, visibility will automatically change to private. Additionally, only Master Tenant users can set visibility for domains. Domains originating in a subtenant will always be private to that subtenant.

Group Access

Configure the domain to be visible to all Infrastructure Groups or only to selected Groups. If the domain is scoped to specific Groups, Users whose Roles do not give them Group access will not have access to the domain. Additionally, users will not be able to set the domain as the default on a Cloud which is not a part of the selected Groups.

Tenant Permissions

When set to public, all Tenants will have visibility into the domain and can join their Instances to the domain. When set to private, users can select specific Tenants which should have access to the domain.

Editing and Removing Domains

- Domains can be edited by selecting the *Actions* dropdown for the Domain and selecting *Edit*, or by selecting the icon in list views.
- Added Domains can be removed from Morpheus by selecting the *Actions* dropdown for the Domain and selecting *Remove*, or the icon in list views.

Setting the default domain on a Cloud

1. Navigate to *Infrastructure > Clouds*.
2. Edit the target Cloud.
3. Expand *Advanced Options* section.
4. In the *Domain* dropdown, select the Domain.
5. Save Changes

Setting the default domain on a Network

1. Navigate to *Infrastructure > Network*.
2. Edit the target Network.
3. Expand *Advanced Options* section.
4. In the *Domain* dropdown, select the Domain.
5. Save Changes

Selecting a Domain while provisioning an instance

1. While creating an instance, in the *Configure* section, expand the *DNS Options*.
2. Select Domain from the *Domain* dropdown.

Proxies

Overview

In many situations, companies deploy virtual machines in proxy restricted environments for things such as PCI Compliance, or just general security. As a result of this Morpheus provides out of the box support for proxy connectivity. Proxy authentication support is also provided with both Basic Authentication capabilities as well as NTLM for Windows Proxy environments. Morpheus is even able to configure virtual machines it provisions to utilize these proxies by setting up the operating systems proxy settings directly (restricted to cloud-init based Linux platforms for now, but can also be done on windows based platforms in a different manner).

To get started with Proxies, it may first be important to configure the Morpheus appliance itself to have access to proxy communication for downloading service catalog images. To configure this, visit the Administration > Settings page where a section labeled “Proxy Settings” is located. Fill in the relevant connection info needed to utilize the proxy. It may also be advised to ensure that the Linux environment’s *http_proxy*, *https_proxy*, and *no_proxy* are set appropriately.

Defining Proxies

Proxies can be used in a few different contexts and optionally scoped to specific networks with which one may be provisioning into or on a cloud integration as a whole. To configure a Proxy for use by the provisioning engines within Morpheus we must go to *Infrastructure > Networks > Proxies*. Here we can create records representing connection information for various proxies. This includes the host ip address, proxy port, and any credentials (if necessary) needed to utilize the proxy. Now that these proxies are defined we can use them in various contexts.

Cloud Communication

When morpheus needs to connect to various cloud APIs to issue provisioning commands or to sync in existing environments, we need to ensure that those api endpoints are accessible by the appliance. In some cases the appliance may be behind a proxy when it comes to public cloud access like Azure and AWS. To configure the cloud integration to utilize a proxy, when adding or editing a cloud there is a setting called “API Proxy” under “Advanced Options”. This is where the proxy of choice can be selected to instruct the Provisioning engine how to communicate with the public cloud. Simply adjust this setting and the cloud should start being able to receive/issue instructions.

Provisioning with Proxies

Proxy configurations can vary from operating system to operating system and in some cases it is necessary for these to be configured in the blueprint as a prerequisite. In other cases it can also be configured automatically. Mostly with the use of cloud-init (which all of our out of the box service catalog utilizes on all clouds). When editing/creating a cloud there is a setting for “Provisioning Proxy” in “Provisioning Options”. If this proxy is set, Morpheus will automatically apply these proxy settings to the guest operating system.

Overriding proxy settings can also be done on the Network record. Networks (or subnets) can be configured in *Infrastructure > Networks* or on the Networks tab of the relevant Cloud detail page. Here, a proxy can also be assigned as well as additional options like the *No Proxy* rules for proxy exceptions.

Docker

When provisioning Docker based hosts within a Proxy environment it is up to the user to configure the docker host proxy configuration manually. There are workflows that can be configured via the Automation engine to make this automatic when creating docker based hosts. Please see documentation on Docker and proxies for specific information.

Proxy setups can vary widely from company to company, and it may be advised to contact support for help configuring morpheus to work in the proxy environment.

Security Groups

Infrastructure > Network - Security Groups

Overview

A Security Group acts as a virtual firewall that controls the traffic for one or more Instances. When you launch an instance, you associate one or more Security Groups with the instance. You add rules to each Security Group that allow traffic to or from its associated Instances. You can modify the rules for a Security Group at any time; the new rules are automatically applied to all Instances that are associated with the Security Group.

Important: The local firewall setting must be enabled for Security Groups to be applied in the guest firewall (iptables). The local firewall setting can be enabled in Infrastructure > Clouds > Click the Cloud > Edit > Advanced Options > Local Firewall (On/Off)

Important: When the local firewall setting is enabled, Morpheus will automatically set an iptable rule to allow incoming connections on tcp port 22 from the Morpheus Appliance.

Important: If the local firewall setting is configured on a cloud that supports Security Groups natively (AWS for example), the local firewall setting is ignored and the guest firewall will not be modified. Security Groups will be attached to the instance as normal

Add Security Group

1. Navigate to *Infrastructure > Network - Security Groups*
2. Click the + *Add Security Group* button.
3. From the Security Group Wizard input a name, and description.
4. Save Changes

Add Security Group Rule

1. Navigate to *Infrastructure > Network - Security Groups*
2. Click the name of the Security Group you wish to add a rule to.
3. From the Security Group page click the + *Add Rule* button.
4. From the Rule Wizard select the rule type and input source and depending on the type selected protocol and input a port range.
5. Save Changes

Edit Security Group rule

1. Navigate to *Infrastructure > Network - Security Groups*
2. Click the name of the Security Group you wish to edit a rule in.
3. Click the edit icon on the row of the Security Group rule you wish to edit.
4. Modify information as needed.
5. Save Changes

Delete Security Group rule

1. Navigate to *Infrastructure > Network - Security Groups*
2. Click the name of the Security Group you wish to delete a rule from.
3. Click the delete icon on the row of the Security Group rule you wish to delete.

Add Cloud Security Group

To add Cloud Security Group

1. Navigate to *Infrastructure > Clouds*
2. Click the name of the desired cloud to add a Security Group
3. Click the Networks tab
4. Within the “Security Groups” section, click on a Security Group to edit its rules
5. Alternatively, click + *ADD SECURITY GROUP* to create a new one

Remove Cloud Security Group

1. Navigate to *Infrastructure > Clouds*
2. Click the name of the cloud to remove the Security Group from.
3. Click the Security Groups tab.
4. Click the *Edit Security Groups* button.
5. Click the - (Minus) button of the Security Group from the Added Security Groups list to remove.
6. Save Changes

Integrations

Overview

The Network Integrations section allows you to add and manage IPAM, DNS, and Service Registry integrations. These services can also be added in the Administration > Integrations section.

The following integrations are currently supported:

Networking

- Cisco ACI
- VMWare NSX

IPAM

- Infoblox
- Bluecat
- phpIPAM

Security

- Cisco ACI

DNS

- Microsoft DNS
- PowerDNS
- Route 53

Scoping Services

NETWORKING Networking integrations are available in the *NETWORK MODE* dropdown located under the *Advanced Options* section in Cloud configurations.

IPAM IPAM integrations will populate pools in the IP Pool section, which are available for assignment to networks in the *NETWORK POOL* dropdown when configuring a network.

SECURITY Security integrations are available in the *SECURITY SERVER* dropdown located under the *Advanced Options* section in Cloud configurations.

DNS DNS integrations will populate domains in the *Infrastructure > Network > Domains* section, and are available in the *DOMAIN* dropdown located under the *Advanced Options* section in Cloud, Group, and Network configurations, as well as in the *Configure* section of the Create Instance wizard. DNS integrations are also available in the *DNS SERVICE* dropdown located under the *Advanced Options* section in Cloud and Group configurations.

Service Registry Service Registry integrations are available in the *SERVICE REGISTRY* dropdown located under the *Advanced Options* section in Cloud and Group configurations.

Note: Infoblox will also appear as a DNS INTEGRATION option in Clouds and Groups after adding Infoblox IPAM Integration.

5.5.6 Load Balancers

Infrastructure > Load Balancers

Overview

Morpheus can provision VM or Container HaProxy Load Balancers, Amazon Elastic and Application Load Balancers, Azure Load Balancers, and integrates with several external Load Balancers, including F5, A10, Citrix, and AVI.

Once created or integrated, Load Balancers are available as an option to be added during provision time or post-provisioning.

Once a Load Balancer is added to an instance, you can manually scale or configure auto-scaling based on thresholds or schedules, and burst across clouds with cloud priority.

In the Load Balancers page there are two sections:

Load Balancers View or edit existing Load Balancers, add new Load Balancers.

Virtual Servers View and link to Instances that are attached to load balancers.

Load Balancers

The Load Balancers tab list currently available Load Balancers, which you can select, edit or delete, and is where you can create new or integrate with external Load Balancers.

Add a new Load Balancer

Select + LOAD BALANCER, chose an option, and fill in the required information:

A10 (aXAPI v3)

- API Host
- API Port
- Username
- Password
- Internal IP
- Public IP
- VIP Address
- VIP Port

Amazon ALB

- Scheme
- Internal
- Internet-Facing
- Amazon Subnets (Select + to add additional) * Specify the subnets to enable for your load balancer. You can specify only one subnet per Availability Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer.
- Amazon Security Groups (Select + to add additional)

AVI

- API Host
- API Port
- Username
- Password
- Internal IP
- Public IP
- VIP Address
- VIP Port

Azure Load Balancer

- Cloud
- Resource Group * Populated from cloud selection

Citrix NetScaler

- API Host
- API Port
- Username
- Password

F5 BigIP (v11.4+)

- API Host
- API Port
- Username
- Password
- Management URL

FortiADC

- API HOST
- API PORT
- USERNAME
- PASSWORD
- INTERFACE (synced on auth)

HaProxy Container (Internal, will create a HaProxy container, must have available docker host to provision to)

- Group
- Cloud
- Name
- Description
- Plan * Select the size of HaProxy container to be provisioned

NSX-T Load Balancer

- NSX-T
- Name
- Description
- Enabled
- Admin State
- Size
- Tier-1 Gateways
- Log Level

Upon saving your new Load Balancer will be added to the Load Balancers list and available in the Load Balancer dropdown in the Provisioning Wizard Automation Section for Instance Types that have scaling enabled.

Load Balancer Detail Pages

In the main Load Balancer page, select an existing Load Balancer to go to that Load Balancers Details Page, which lists Stats, Settings, Actions and Virtual Servers for that load balancer.

Orchestrating Load Balancers

A large part of application orchestration and automation involves tying various web services and backend services into different load balancer configurations. If the automation tool is unable to communicate or integrate with this aspect of your infrastructure, a lot of gaps will be created in the full orchestrated flow of application deployment. This is why Morpheus provides deep integration with load balancers and explicit definitions with catalog items as to how they are connected to provisioned instances. Some of the functionality includes:

- Public Cloud Load Balancer Support
- Private Cloud Load Balancer Support
- Port Type definitions (Profiles like HTTP/HTTPS or UDP)
- SSL Certificate Management and SSL Certificate Upload
- SSL Passthrough or Forced Redirect

Not only does Morpheus have an ability to provision HAProxy based load balancer containers for easy consumption in development environments, but also has direct tie ins with several Load Balancer Types:

- F5 BigIP
- A10
- Netscaler
- NSX Advanced Load Balancer
- Amazon ELB
- Amazon ALB
- Azure Load Balancer
- Fortinet
- Openstack Octavia

- HA Proxy
- NSX-T

Morpheus exposes configuration options during provisioning of an Instance relevant and common to each supported LB Integration. In some cases, Morpheus also provides direct management and sync support for VIP configurations on the various Load Balancers (such as F5, and NSX Advanced Load Balancer), However in a day to day orchestrated workflow this would not be the ideal means by which a user should consume load balancer services.

By tying the Load Balancer associations into the provisioning of instances and the definition of the instance catalog item, the lifecycle of the VIP can more easily be maintained throughout the lifecycle of whatever application may be deployed.

Setting up an Instance for Load Balancer Consumption

Several of the provided Morpheus instance types are ready to go with load balancer orchestration out of the box (Apache, Nginx, Tomcat, Node.js, etc). It is also fairly easy to extend existing generic instance types during provisioning to be tied to load balancers or to set up said catalog items in advanced for such functionality.

When creating a custom Instance Type (in Library), one can define a list of exposed ports that the node type within the instance exposes. When defining these exposed ports it prompts for a Name, Port Number, and LB Type. The LB Type is what enables load-balancer functionality. This can either be HTTP,HTTPS, or TCP. This specification helps build the correct profile for the VIP as well as setup the appropriate types of Health Monitors within the target load balancer integration.

Now, when a user consumes this custom instance type (either through single instance provisioning or full application blueprint provisioning), a section appears in the Automation phase of provisioning. Each port that is defined that exposes a load-balancer gets a dropdown to choose which load balancer integration attach to the exposed port and various prompts become available.

These prompts control features ranging from target VIP Address to selecting an SSL Certificate to be applied to the VIP. These SSL Certificates will even go so far as to create SSL Profiles in integrations for things like an F5 automatically for the application. There are also external integrations for SSL Certificate management with Venafi which allows for the consumption of certificates managed by that external system.

Once the instance is provisioned, as part of the final phase, the load balancer configuration will be applied and maintained on the instance. This association can be manipulated after the fact via the “Scale” tab found on the Instance Detail page.

Another benefit to associating load-balancers this way is that the pool members are automatically maintained during scaling events, either via auto-scaling thresholds or manual node additions / removals.

F5 Load Balancers

Add F5 Load Balancer

To add a F5 Load Balancer Integration:

1. Navigate to *Infrastructure > Load Balancers*
2. Select + *ADD*
3. Select *F5 BigIP*
4. Fill in the following:

GROUP Select the Group the Load Balancer will be available for

CLOUD Select the Cloud the Load Balancer will be available for

NAME Name of the Load Balancer in Morpheus

DESCRIPTION Identifying information displayed on the Load Balancer list page.

VISIBILITY Define Multi-Tenant permissions

API HOST IP or resolvable hostname url.

API PORT Typically 8443

USERNAME API user

PASSWORD API user password

MANAGEMENT URL Example: `https://10.30.20.31:8443/xui/`

Advanced Options (optional)

- VIRTUAL NAME
- POOL NAME
- SERVER NAME

5. Save Changes

Important: The F5 API handles SSL certificate installation by downloading the certificate from a URL the user provides. Morpheus provides the “Appliance URL” configured in global settings (Administration > Settings > Appliance) to satisfy that requirement. Make sure you have configured a valid URL in this field and that F5 can reach it.

Virtual Servers

Instances attached to an F5 will be listed in the Virtual servers tab. Virtual servers can also be manually added in this section.

Add Virtual Server

1. Navigate to *Infrastructure > Load Balancers*
2. Select F5 Integration name to drill into the detail page
3. Select + *ADD* in the VIRTUAL SERVERS tab
4. Fill in the following:
 - **NAME** Name of the Virtual Server in Morpheus
 - **DESCRIPTION** Description of the Virtual Server in Morpheus
 - **Enabled** Uncheck to keep the configuration but disable F5 availability in Morpheus
 - **VIP TYPE**
 - Standard
 - Forwarding (Layer 2)
 - Forwarding (IP)
 - Performance (HTTP)
 - Performance (Layer 4)

- Stateless
- Reject
- DHCP
- Internal
- Message Routing
- **VIP HOSTNAME** Enter Hostname of the VIP (optional)
- **VIP ADDRESS** Enter IP address for the VIP
- **VIP PORT** Enter port used for the VIP
- **SOURCE ADDRESS** Enter Virtual Server source address
- **PROTOCOL** tcp, udp, or sctp
- **PROFILES** Search for and select from available PROFILES
- **POLICIES** Search for and select from available POLICIES
- **IRULES** Search for and select from available RULE SCRIPTS
- **PERSISTENCE**
 - cookie
 - dest-addr
 - global-settings
 - hash
 - msrdp
 - sip
 - source-addr
 - ssl
 - universal
- **DEFAULT POOL** Select from available POOLS

5. Select *SAVE CHANGES*

Policies

Policies will be synced and listed in the Policies tab. These policies will be available options when creating Virtual Servers.

Pools

Create Pool

NAME Name of the POOL in Morpheus

DESCRIPTION Description of the POOL in Morpheus

BALANCE MODE

- Round Robin
- Least Connections

SERVICE PORT Specify SERVICE PORT for the POOL

MEMBERS Search for and select from available NODES

MONITORS Search for and select from available Monitors

Profiles

SSL Profiles are synced and will be created when an SSL Certificate is assigned in the Load balancer section when provisioning or editing a Load balancer on an Instance.

Monitors

Create Monitor

NAME Name of the MONITOR in Morpheus

DESCRIPTION Description of the MONITOR in Morpheus

PARENT MONITOR Select from available MONITORS

DESTINATION Specify Destination, such a *:443. Default is *:*

INTERVAL Specify Monitor Interval. Default is 5

TIMEOUT Specify Monitor Timeout. Default is 15

MONITOR CONFIG Enter monitor config.

Nodes

Create Node

NAME Name of the NODE in Morpheus

DESCRIPTION Description of the NODE in Morpheus

ADDRESS Enter node address

MONITOR Select from available MONITORS

SERVICE PORT Specify SERVICE PORT for the NODE

Rule Scripts

Rule Scripts will be synced and listed in the RULE SCRIPTS tab. These rules will be available options when creating Virtual Servers.

Citrix NetScaler



Add NetScaler Integration

To add a NetScaler Load Balancer Integration:

1. Navigate to *Infrastructure > Load Balancers*
2. Select + *ADD*
3. Select *Citrix NetScaler*
4. Fill in the following:

GROUP * Select the Group the Load Balancer will be available for.

CLOUD * Select the Cloud the Load Balancer will be available for.

NAME * Name of the Load Balancer in Morpheus.

DESCRIPTION Identifying information displayed on the Load Balancer list page.

VISIBILITY

Define Tenant Visibility

- Public: Available to all Tenants.
- Private: Only available to specified Tenant.

Tenant If Visibility is set to private, define the Tenant the Load Balancer will be available in.

API URL *

URL of the NetScaler API

- Example: <http://10.30.21.55>

API PORT *

NetScaler API port

- Example: 80

USERNAME * NetScaler service account username

PASSWORD * NetScaler service account password

VIRTUAL NAME

Naming Pattern for new NetScaler Virtual Servers

- If blank, defaults to `morph_lb_${loadBalancer.id}`

SERVICE NAME

Naming Pattern for new NetScaler Services

- If blank, defaults to `morph_service_{{container.id}}`

SERVER NAME

Naming Pattern for new NetScaler Servers

- If blank, defaults to `morph_server_{{server.id}}`

Add Load Balancer to Instance

Load Balancers can be added to Instances during Provisioning or to existing Instances. For Load Balancer settings to appear during provisioning, or for the scale tab to be available on an Instance, the instances Node Type must have a LB port defined.

Note: For Load Balancer settings to appear during provisioning, or for the scale tab to be available on an Instance, the instances Node Type must have a LB port defined.

Add Load Balancer during Provisioning

In the Instance Provisioning wizard, Load Balancers can be configured in the Automation > Load Balancer section.

1. Navigate to Provisioning > Instances.
2. Select + *ADD*.
3. Select an Instance Type that supports scaling. (ENABLE SCALING (HORIZONTAL) flagged on Instance Type configuration)
4. Proceed with Instance configuration to the Automation section.
5. Fill in the following:

VIP ADDRESS

Define IP Address for the Virtual Server

- Example: 10.30.23.191

VIP PORT

Define port for the Virtual Server

- Example: 80

VIP HOSTNAME

Define hostname that will resolve to the VIP IP.

- Example: `jwDemoHaApp59.den.example.com`

VIRTUAL SERVICE NAME Define name for the Virtual Service. Defaults to `{{instance.name}}`

BALANCE MODE

Specify balance mode for the VIP

- Least Connections
- Round Robin

STICKY MODE

Specify sticky session options for the VIP

- Source IP
- Cookie

SHARED VIP ADDRESS Select if VIP is shared, then enter DIRECT VIP ADDRESS

SSL CERT

SSL Certificate that will be applied to the VIP.

- No SSL
- Select existing Certificate from Infrastructure > Keys & Certs or from a Trust Provider Integration.

USE EXTERNAL ADDRESS FOR BACKEND NODES

- Select if traffic from LB to Backend Nodes needs to be sent to the External Addresses, or leave deselected to use Internal Addresses for Backed Nodes.

6. Optionally configure auto-scaling configuration in the Scale section

7. Select *NEXT* and provision the Instance.

After all nodes in the Instance are provisioned, the LB configuration will be added to the Instance and Virtual Servers, Services and Servers will be created and configured on the NetScaler. The Load Balancer settings and status will be visible in the Instance details page LOAD BALANCER section, with additional details, links, and configurations options available in the SCALE tab.

5.5.7 Storage

Note: In v3.5.2 STORAGE PROVIDERS has been split out into BUCKETS and FILE SHARES sections.

Overview

Infrastructure > Storage is for adding and managing Storage Buckets, File Shares, Volumes, Data Stores and Storage Servers for use with other Services in Morpheus.

Role Requirements

There are two Role permissions for the *Infrastructure > Storage* section: *Infrastructure: Storage* and *Infrastructure: Storage Browser*. *Infrastructure: Storage* give Full, Read or No access to the *Infrastructure > Storage* sections, while *Infrastructure: Storage Browser* is specific to *Buckets* and *Files Shares*. Full *Infrastructure: Storage Browser* permissions allows *Buckets* and *Files Shares* to be browsed and files and folders to be added, downloaded and deleted from the *Buckets* and *Files Shares*. Read *Infrastructure: Storage Browser* permissions allows *Buckets* and *Files Shares* to be browsed only.

Default Storage

The default Storage path for Virtual Images, Backups, Deployment Archives, Archive Service, and Archived Snapshots is `var/opt/morpheus/morpheus-ui/`. It is recommended to add Storage Buckets and File Shares for these targets in the *Infrastructure > Storage* section to avoid running out of disk space on the Morpheus Appliance.

Storage Buckets

Storage Buckets are for Backup, Archives, Deployment and Virtual Images storage targets. Buckets can be browsed and files and folders can be uploaded, downloaded or deleted from the Bucket section. Retention Policies can be set on Storage Buckets for files to be deleted or backed up to another bucket after a set amount of time.

Supported Bucket Types

- Alibaba
- Amazon S3
- Azure
- Google Cloud Storage
- Openstack Swift
- Rackspace CDN

Alibaba Buckets

To Add an Alibaba Storage Bucket:

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the BUCKETS tab, Click the + *ADD* button.
4. Select *Alibaba* from the dropdown list
5. From the NEW BUCKET Wizard input the following:

NAME Name of the Bucket in Morpheus.

ACCESS KEY Alibaba Access Key

SECRET KEY Alibaba Secret Key

REGION Enter Alibaba Region for the Bucket

BUCKET NAME Enter existing Alibaba Bucket name, or to add a new Bucket enter a new name and select *Create Bucket*.

Create Bucket Enable if the Bucket entered in BUCKET NAME does not exist and needs to be created.

Default Backup Target Sets this Bucket as the default backup target when creating Backups. If selected the option to update existing Backup configuration to use this Bucket will be presented.

Archive Snapshots Enabled to export VM snapshots to this Bucket when creating VMware Backups, after which the snapshot will be removed from the target hypervisor.

Default Deployment Archive Target Sets this bucket as the default storage target when uploading Deployment files in the *Deployments* section.

Default Virtual Image Store Sets this bucket as the default storage target when uploading Virtual Images from the *Virtual Images* section, importing Images from Instance Actions, creating Images with the *Image Builder* and when creating new images from *Migrations*.

RETENTION POLICY

None Files in the Bucket will not be automatically deleted or backed up.

Backup Old Files

This option will backup files after a set amount of time and remove them from the bucket.

DAYS OLD Files older than the set number of days will be automatically backed up to the selected Backup Bucket.

BACKUP BUCKET Search for and then select the Bucket the files will be backed up to.

DELETE OLD FILES

This option will delete files from this bucket after a set amount of days.

DAYS OLD Files older than the set number of days will be automatically deleted from the Bucket.

6. Select *SAVE CHANGES*

The Bucket will be created and displayed in the Buckets tab.

- To browse, upload, download, or delete files from this Bucket, select the name of the Bucket.
- To edit the Bucket, select the edit icon or select the name of the Bucket and select *ACTIONS - EDIT*.

Warning: Repointing a bucket that is in use may cause loss of file references. Ensure data is mirrored first.

- To delete a Bucket, select the trash icon or select the name of the Bucket and select *DELETE*.

Warning: When deleting a Bucket, all Deployment Versions and Backups associated with the Bucket will be deleted.

Amazon S3 Buckets

To Add an Amazon S3 Storage Bucket:

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the BUCKETS tab, Click the + *ADD* button.
4. Select *Amazon S3* from the dropdown list
5. From the NEW BUCKET Wizard input the following:

NAME Name of the Bucket in Morpheus.

ACCESS KEY AWS IAM Access Key

SECRET KEY AWS IAM Secret Key

BUCKET NAME Enter existing S3 Bucket name, or to add a new Bucket enter a new name and select *Create Bucket*.

CREATE BUCKET Enable if the Bucket entered in BUCKET NAME does not exist and needs to be created. If enabled, select an AWS Region to create the Bucket in.

ENDPOINT URL Optional endpoint URL if pointing to an object store other than amazon that mimics the Amazon S3 APIs.

Default Backup Target Sets this Bucket as the default backup target when creating Backups. If selected the option to update existing Backup configuration to use this Bucket will be presented.

Archive Snapshots Enabled to export VM snapshots to this Bucket when creating VMware Backups, after which the snapshot will be removed from the target hypervisor.

Default Deployment Archive Target Sets this bucket as the default storage target when uploading Deployment files in the *Deployments* section.

Default Virtual Image Store Sets this bucket as the default storage target when uploading Virtual Images from the *Virtual Images* section, importing Images from Instance Actions, creating Images with the *Image Builder* and when creating new images from *Migrations*.

RETENTION POLICY

None Files in the Bucket will not be automatically deleted or backed up.

Backup Old Files

This option will backup files after a set amount of time and remove them from the bucket.

DAYS OLD Files older than the set number of days will be automatically backed up to the selected Backup Bucket.

BACKUP BUCKET Search for and then select the Bucket the files will be backed up to.

DELETE OLD FILES

This option will delete files from this bucket after a set amount of days.

DAYS OLD Files older than the set number of days will be automatically deleted from the Bucket.

6. Select *SAVE CHANGES*

The Bucket will be created and displayed in the Buckets tab.

- To browse, upload, download, or delete files from this Bucket, select the name of the Bucket.
- To edit the Bucket, select the edit icon or select the name of the Bucket and select *ACTIONS - EDIT*.

Warning: Repointing a bucket that is in use may cause loss of file references. Ensure data is mirrored first.

- To delete a Bucket, select the trash icon or select the name of the Bucket and select *DELETE*.

Warning: When deleting a Bucket, all Deployment Versions and Backups associated with the Bucket will be deleted.

Azure Buckets

To Add an Azure Storage Bucket:

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the BUCKETS tab, Click the + *ADD* button.
4. Select *Azure* from the dropdown list
5. From the NEW BUCKET Wizard input the following:

NAME Name of the Bucket in Morpheus.

STORAGE ACCOUNT Name of the Storage Account in Azure for the Bucket

STORAGE KEY Storage Key provided from Azure

SHARE NAME Enter existing Azure Storage Share name, or to add a new Share enter a new name and select *Create Bucket* below.

CREATE BUCKET Enable if the Share entered in SHARE NAME does not exist and needs to be created.

Default Backup Target Sets this bucket as the default backup target when creating Backups. If selected the option to update existing Backup configuration to use this Bucket will be presented.

Archive Snapshots Enabled to export VM snapshots to this Bucket when creating VMware Backups, after which the snapshot will be removed from the target hypervisor.

Default Deployment Archive Target Sets this Bucket as the default storage target when uploading Deployment files in the *Deployments* section.

Default Virtual Image Store Sets this bucket as the default storage target when uploading Virtual Images from the *Virtual Images* section, importing Images from Instance Actions, creating Images with the *Image Builder* and when creating new images from *Migrations*.

RETENTION POLICY

None Files in the Bucket will not be automatically deleted or backed up.

Backup Old Files

This option will backup files after a set amount of time and remove them from the bucket.

DAYS OLD Files older than the set number of days will be automatically backed up to the selected Backup Bucket.

BACKUP BUCKET Search for and then select the Bucket the files will be backed up to.

DELETE OLD FILES

This option will delete files from this bucket after a set amount of days.

DAYS OLD Files older than the set number of days will be automatically deleted from the Bucket.

6. Select *SAVE CHANGES*

The Bucket will be created and displayed in the Buckets tab.

- To browse, upload, download, or delete files from this Bucket, select the name of the Bucket.
- To edit the Bucket, select the edit icon or select the name of the Bucket and select *ACTIONS - EDIT*.

Warning: Repointing a bucket that is in use may cause loss of file references. Ensure data is mirrored first.

- To delete a Bucket, select the trash icon or select the name of the Bucket and select *DELETE*.

Warning: When deleting a Bucket, all Deployment Versions and Backups associated with the Bucket will be deleted.

Google Cloud Storage Buckets

Note: Google Cloud Storage Buckets are associated with an existing GCP Cloud integration. Ensure the GCP Cloud integration is pre-existing before attempting to create a new Google Cloud Storage Bucket. On the initial integration and subsequent cloud syncs, Google Cloud Storage Buckets are automatically onboarded and updated.

To add a Google Cloud Storage Bucket:

1. Select the Infrastructure link in the navigation bar
2. Select the Storage link in the sub-navigation bar
3. In the BUCKETS tab, Click the + *ADD* button
4. Select *Google Cloud Storage* from the dropdown list
5. From the NEW BUCKET Wizard input the following:

NAME A friendly name for the bucket in Morpheus

STORAGE SERVICE Select the GCP cloud integration this bucket should be created in

PROJECT ID Select the Project to create the group under, Projects are a GCP-specific concept and are logical grouping for your resources. Select any existing project associated with your selected GCP cloud integration

BUCKET NAME The name for the bucket resource on the GCP side

LOCATION TYPE Select Region, Dual-Region, or Multi-Region. Buckets with a Region location type will be stored in one GCP region, such as “us-east1 (South Carolina)”. Dual-Region and Multi-Region data is stored in two (or more, in the case of multi-region) GCP regions separated by a significant physical distance. Dual-Region and Multi-Region data is geo-redundant across the multiple selected regions

LOCATION A selected GCP region (or regions, in the case of dual and multi-location data) where the data will be stored

STORAGE CLASS Select Standard, Nearline, Coldline, or Archive. The appropriate storage class will depend on how frequently the bucket data is accessed and how long the type of data in the bucket is expected to be stored. More information on storage classes can be found in [GCP Documentation](#)

ACTIVE When marked, the bucket is available for use in Morpheus

DEFAULT BACKUP TARGET Sets the bucket as the default storage option when creating backups at provision time or in the Backups section of Morpheus

DEFAULT DEPLOYMENT ARCHIVE TARGET Sets this Bucket as the default storage target when uploading deployment files in the *Deployments* section

DEFAULT VIRTUAL IMAGE STORE Sets this bucket as the default storage target when uploading virtual images from the *Virtual Images* section, importing images from Instance actions, creating images with the *Image Builder*, and when creating new images from *Migrations*

RETENTION POLICY Select None and the files in this bucket will never be deleted or backed up by Morpheus. When selecting 'Backup Old Files', Morpheus will backup files into another selected bucket once they reach a certain number of days old. When selecting 'Delete Old Files', Morpheus will delete any files that reach a certain number of days old

Dell EMC ECS Buckets

Note: A Dell EMC ECS Storage Server must be configured in *Infrastructure - Storage - Servers* prior to adding a Dell EMC ECS Bucket.

To Add a Dell EMC ECS Storage Bucket:

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the BUCKETS tab, Click the + *ADD* button.
4. Select *Dell EMC ECS Bucket* from the dropdown list
5. From the NEW BUCKET Wizard input the following:

NAME Name of the Bucket in Morpheus.

STORAGE SERVICE Select existing Dell EMC ECS Storage Server (configured in *Infrastructure - Storage - Servers*)

BUCKET NAME Enter a name for the new Dell EMC ECS bucket.

USER Dell EMC ECS User

SECRET KEY Dell EMC ECS Secret key

NAMESPACE Select Dell EMC ECS Namespace for the Bucket

STORAGE GROUP Select a Dell EMC ECS Storage Group

Default Backup Target Sets this bucket as the default backup target when creating Backups. If selected the option to update existing Backup configuration to use this Bucket will be presented.

Archive Snapshots Enabled to export VM snapshots to this Bucket when creating VMware Backups, after which the snapshot will be removed from the target hypervisor.

Default Deployment Archive Target Sets this Bucket as the default storage target when uploading Deployment files in the *Deployments* section.

Default Virtual Image Store Sets this bucket as the default storage target when uploading Virtual Images from the *Virtual Images* section, importing Images from Instance Actions, creating Images with the *Image Builder* and when creating new images from *Migrations*.

RETENTION POLICY

None Files in the Bucket will not be automatically deleted or backed up.

Backup Old Files

This option will backup files after a set amount of time and remove them from the bucket.

DAYS OLD Files older than the set number of days will be automatically backed up to the selected Backup Bucket.

BACKUP BUCKET Search for and then select the Bucket the files will be backed up to.

DELETE OLD FILES

This option will delete files from this bucket after a set amount of days.

DAYS OLD Files older than the set number of days will be automatically deleted from the Bucket.

6. Select *SAVE CHANGES*

The Bucket will be created and displayed in the Buckets tab.

- To browse, upload, download, or delete files from this Bucket, select the name of the Bucket.
- To edit the Bucket, select the edit icon or select the name of the Bucket and select *ACTIONS - EDIT*.

Warning: Repointing a bucket that is in use may cause loss of file references. Ensure data is mirrored first.

- To delete a Bucket, select the trash icon or select the name of the Bucket and select *DELETE*.

Warning: When deleting a Bucket, all Deployment Versions and Backups associated with the Bucket will be deleted.

Openstack Swift Buckets

To Add an Azure Storage Bucket:

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the BUCKETS tab, Click the + *ADD* button.
4. Select *Openstack Swift* from the dropdown list
5. From the NEW BUCKET Wizard input the following:

NAME Name of the Bucket in Morpheus.

USERNAME Openstack Swift Username

API KEY Openstack Swift API Key

BUCKET NAME Enter existing Openstack Swift Bucket name, or to add a new Bucket enter a new name and select *Create Bucket* below.

IDENTITY URL Openstack Swift Identity URL

Create Bucket Enable if the name entered in BUCKET NAME does not exist and needs to be created.

Default Backup Target Sets this bucket as the default backup target when creating Backups. If selected the option to update existing Backup configuration to use this Bucket will be presented.

Archive Snapshots Enabled to export VM snapshots to this Bucket when creating VMware Backups, after which the snapshot will be removed from the target hypervisor.

Default Deployment Archive Target Sets this Bucket as the default storage target when uploading Deployment files in the *Deployments* section.

Default Virtual Image Store Sets this bucket as the default storage target when uploading Virtual Images from the *Virtual Images* section, importing Images from Instance Actions, creating Images with the *Image Builder* and when creating new images from *Migrations*.

RETENTION POLICY

None Files in the Bucket will not be automatically deleted or backed up.

Backup Old Files

This option will backup files after a set amount of time and remove them from the bucket.

DAYS OLD Files older than the set number of days will be automatically backed up to the selected Backup Bucket.

BACKUP BUCKET Search for and then select the Bucket the files will be backed up to.

DELETE OLD FILES

This option will delete files from this bucket after a set amount of days.

DAYS OLD Files older than the set number of days will be automatically deleted from the Bucket.

6. Select *SAVE CHANGES*

The Bucket will be created and displayed in the Buckets tab.

- To browse, upload, download, or delete files from this Bucket, select the name of the Bucket.
- To edit the Bucket, select the edit icon or select the name of the Bucket and select *ACTIONS - EDIT*.

Warning: Repointing a bucket that is in use may cause loss of file references. Ensure data is mirrored first.

- To delete a Bucket, select the trash icon or select the name of the Bucket and select *DELETE*.

Warning: When deleting a Bucket, all Deployment Versions and Backups associated with the Bucket will be deleted.

Rackspace CDN Buckets

To Add a Rackspace CDN Bucket:

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the BUCKETS tab, Click the + *ADD* button.
4. Select *Rackspace CDN* from the dropdown list
5. From the NEW BUCKET Wizard input the following:

NAME Name of the Bucket in Morpheus.

USERNAME Rackspace CDN Username

API KEY Rackspace CDN API Key

REGION Enter Rackspace CDN Region

BUCKET NAME Enter existing Rackspace CDN Bucket name, or to add a new Bucket enter a new name and select *Create Bucket* below.

Create Bucket Enable if the name entered in BUCKET NAME does not exist and needs to be created.

Default Backup Target Sets this bucket as the default backup target when creating Backups. If selected the option to update existing Backup configuration to use this Bucket will be presented.

Archive Snapshots Enabled to export VM snapshots to this Bucket when creating VMware Backups, after which the snapshot will be removed from the target hypervisor.

Default Deployment Archive Target Sets this Bucket as the default storage target when uploading Deployment files in the *Deployments* section.

Default Virtual Image Store Sets this bucket as the default storage target when uploading Virtual Images from the *Virtual Images* section, importing Images from Instance Actions, creating Images with the *Image Builder* and when creating new images from *Migrations*.

RETENTION POLICY

None Files in the Bucket will not be automatically deleted or backed up.

Backup Old Files

This option will backup files after a set amount of time and remove them from the bucket.

DAYS OLD Files older than the set number of days will be automatically backed up to the selected Backup Bucket.

BACKUP BUCKET Search for and then select the Bucket the files will be backed up to.

DELETE OLD FILES

This option will delete files from this bucket after a set amount of days.

DAYS OLD Files older than the set number of days will be automatically deleted from the Bucket.

6. Select *SAVE CHANGES*

The Bucket will be created and displayed in the Buckets tab.

- To browse, upload, download, or delete files from this Bucket, select the name of the Bucket.
- To edit the Bucket, select the edit icon or select the name of the Bucket and select *ACTIONS - EDIT*.

Warning: Repointing a bucket that is in use may cause loss of file references. Ensure data is mirrored first.

- To delete a Bucket, select the trash icon or select the name of the Bucket and select *DELETE*.

Warning: When deleting a Bucket, all Deployment Versions and Backups associated with the Bucket will be deleted.

File Shares

File Shares are for Backup, Archives, Deployment and Virtual Images storage targets. File Shares can be browsed and files and folders can be uploaded, downloaded or deleted from the File Shares section. Retention Policies can be set on Storage File Shares for files to be deleted or backed up to another File Share after a set amount of time.

Supported File Share Types

- CIFS (Samba Windows File Sharing)
- Dell EMC ECS Share
- Dell EMC Isilon Share
- Local Storage
- NFSv3

CIFS File Shares

To Add a CIFS File Share:

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the FILE SHARES tab, Click the + *ADD* button.
4. Select *CIFS (Samba Windows File Sharing)* from the dropdown list
5. From the NEW FILE SHARE Wizard input the following:

NAME Name of the File Share in Morpheus.

HOST

Enter host IP or resolvable hostname Example: 192.168.200.210

USERNAME CIFS Share Username

PASSWORD CIFS Share User Password

SHARE PATH

Enter CIFS Share Path Example: cifs

Default Backup Target Sets this File Share as the default backup target when creating Backups. If selected the option to update existing Backup configuration to use this File Share will be presented.

Archive Snapshots Enabled to export VM snapshots to this File Share when creating VMware Backups, after which the snapshot will be removed from the source Cloud.

Default Deployment Archive Target Sets this File Share as the default storage target when uploading Deployment files in the *Deployments* section.

Default Virtual Image Store Sets this File Share as the default storage target when uploading Virtual Images from the *Virtual Images* section, importing Images from Instance Actions, creating Images with the *Image Builder* and when creating new images from *Migrations*.

RETENTION POLICY

None Files in the File Share will not be automatically deleted or backed up.

Backup Old Files

This option will backup files after a set amount of time and remove them from the File Share.

DAYS OLD Files older than the set number of days will be automatically backed up to the selected Backup File Share.

BACKUP File Share Search for and select the File Share the files will be backed up to.

DELETE OLD FILES

This option will delete files from this File Share after a set amount of days.

DAYS OLD Files older than the set number of days will be automatically deleted from the File Share.

6. Select *SAVE CHANGES*

The File Share will be created and displayed in the File Shares tab.

- To browse, upload, download, or delete files from this File Share, select the name of the File Share.
- To edit the File Share, select the edit icon or select the name of the File Share and select *ACTIONS - EDIT*.

Warning: Repointing a File Share that is in use may cause loss of file references. Ensure data is mirrored first.

- To delete a File Share, select the trash icon or select the name of the File Share and select *DELETE*.

Warning: When deleting a File Share, all Deployment Versions and Backups associated with the File Share will be deleted.

Dell EMC ECS File Shares

To Add a Dell EMC ECS File Share:

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the FILE SHARES tab, Click the + *ADD* button.
4. Select *Dell EMC ECS Share* from the dropdown list
5. From the NEW FILE SHARE Wizard input the following:

NAME Name of the File Share in Morpheus.

STORAGE SERVICE Select existing Dell EMC ECS Storage Server (configured in *Infrastructure - Storage - Servers*)

SHARE PATH

Enter Dell EMC ECS Share Path Example: `ecs-file-share-1`

USER Dell EMC ECS User

SECRET KEY Dell EMC ECS Secret key

Volume Size Specify volume size for the File Share (in MB)

Allowed IP's

Specify IP Addresses to limit accessibility to the File Share

Leave blank for open access Click the + symbol to the right of the first ALLOWED IPS field to add multiple IP's

NAMESPACE Select Dell EMC ECS Namespace (synced)

STORAGE GROUP Select Dell EMC ECS Storage Group (synced)

Default Backup Target Sets this File Share as the default backup target when creating Backups. If selected the option to update existing Backup configuration to use this File Share will be presented.

Archive Snapshots Enabled to export VM snapshots to this File Share when creating VMware Backups, after which the snapshot will be removed from the source Cloud.

Default Deployment Archive Target Sets this File Share as the default storage target when uploading Deployment files in the *Deployments* section.

Default Virtual Image Store Sets this File Share as the default storage target when uploading Virtual Images from the *Virtual Images* section, importing Images from Instance Actions, creating Images with the *Image Builder* and when creating new images from *Migrations*.

RETENTION POLICY

None Files in the File Share will not be automatically deleted or backed up.

Backup Old Files

This option will backup files after a set amount of time and remove them from the File Share.

DAYS OLD Files older than the set number of days will be automatically backed up to the selected Backup File Share.

BACKUP File Share Search for and select the File Share the files will be backed up to.

DELETE OLD FILES

This option will delete files from this File Share after a set amount of days.

DAYS OLD Files older than the set number of days will be automatically deleted from the File Share.

6. Select *SAVE CHANGES*

The File Share will be created and displayed in the File Shares tab.

- To browse, upload, download, or delete files from this File Share, select the name of the File Share.
- To edit the File Share, select the edit icon or select the name of the File Share and select *ACTIONS - EDIT*.

Warning: Repointing a File Share that is in use may cause loss of file references. Ensure data is mirrored first.

- To delete a File Share, select the trash icon or select the name of the File Share and select *DELETE*.

Warning: When deleting a File Share, all Deployment Versions and Backups associated with the File Share will be deleted.

Dell EMC Isilon File Shares

To Add a Dell EMC Isilon File Share:

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the FILE SHARES tab, Click the + *ADD* button.
4. Select *Dell EMC Isilon Share* from the dropdown list
5. From the NEW FILE SHARE Wizard input the following:

NAME Name of the File Share in Morpheus.

STORAGE SERVICE Select existing Dell EMC Isilon Storage Server (configured in *Infrastructure - Storage - Servers*)

SHARE PATH

Enter Dell EMC Isilon Share Path Example: `ecs-file-share-1`

Volume Size Specify volume size for the File Share (in MB)

Allowed IP's

Specify IP Addresses to limit accessibility to the File Share

Leave blank for open access Click the + symbol to the right of the first ALLOWED IPS field to add multiple IP's

NAMESPACE Select Dell EMC Isilon Namespace (synced)

STORAGE GROUP Select Dell EMC Isilon Storage Group (synced)

Default Backup Target Sets this File Share as the default backup target when creating Backups. If selected the option to update existing Backup configuration to use this File Share will be presented.

Archive Snapshots Enabled to export VM snapshots to this File Share when creating VMware Backups, after which the snapshot will be removed from the source Cloud.

Default Deployment Archive Target Sets this File Share as the default storage target when uploading Deployment files in the *Deployments* section.

Default Virtual Image Store Sets this File Share as the default storage target when uploading Virtual Images from the *Virtual Images* section, importing Images from Instance Actions, creating Images with the *Image Builder* and when creating new images from *Migrations*.

RETENTION POLICY

None Files in the File Share will not be automatically deleted or backed up.

Backup Old Files

This option will backup files after a set amount of time and remove them from the File Share.

DAYS OLD Files older than the set number of days will be automatically backed up to the selected Backup File Share.

BACKUP File Share Search for and select the File Share the files will be backed up to.

DELETE OLD FILES

This option will delete files from this File Share after a set amount of days.

DAYS OLD Files older than the set number of days will be automatically deleted from the File Share.

6. Select *SAVE CHANGES*

The File Share will be created and displayed in the File Shares tab.

- To browse, upload, download, or delete files from this File Share, select the name of the File Share.
- To edit the File Share, select the edit icon or select the name of the File Share and select *ACTIONS - EDIT*.

Warning: Repointing a File Share that is in use may cause loss of file references. Ensure data is mirrored first.

- To delete a File Share, select the trash icon or select the name of the File Share and select *DELETE*.

Warning: When deleting a File Share, all Deployment Versions and Backups associated with the File Share will be deleted.

Local Storage File Shares

Important: Local Storage refers to local to the Morpheus Appliance and the path must be owned by *morpheus-app*. Please be conscious of storage space. High Availability configurations require Local Storage File Shares paths to be shared storage paths between the front end Morpheus Appliances.

Note: To change the owner of a file path to be used as a Local Storage File Share, run `chown morpheus-app /path` on the Morpheus Appliance.

Note: Morpheus will validate path and ownership of the File Share Path.

To Add a Local Storage File Share:

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the FILE SHARES tab, Click the + *ADD* button.
4. Select *Local Storage Share* from the dropdown list
5. From the NEW FILE SHARE Wizard input the following:

NAME Name of the File Share in Morpheus.

STORAGE PATH

Enter the File Share path on the local Morpheus Appliance. Example: `/var/opt/morpheus/morpheus-ui/vms/virtual-images`

Important: High Availability configurations require Local Storage File Shares paths to be shared storage paths between the front end Morpheus Appliances.

Default Backup Target Sets this File Share as the default backup target when creating Backups. If selected the option to update existing Backup configuration to use this File Share will be presented.

Archive Snapshots Enabled to export VM snapshots to this File Share when creating VMware Backups, after which the snapshot will be removed from the source Cloud.

Default Deployment Archive Target Sets this File Share as the default storage target when uploading Deployment files in the *Deployments* section.

Default Virtual Image Store Sets this File Share as the default storage target when uploading Virtual Images from the *Virtual Images* section, importing Images from Instance Actions, creating Images with the *Image Builder* and when creating new images from *Migrations*.

RETENTION POLICY

None Files in the File Share will not be automatically deleted or backed up.

Backup Old Files

This option will backup files after a set amount of time and remove them from the File Share.

DAYS OLD Files older than the set number of days will be automatically backed up to the selected Backup File Share.

BACKUP File Share Search for and select the File Share the files will be backed up to.

DELETE OLD FILES

This option will delete files from this File Share after a set amount of days.

DAYS OLD Files older than the set number of days will be automatically deleted from the File Share.

6. Select *SAVE CHANGES*

The File Share will be created and displayed in the File Shares tab.

- To browse, upload, download, or delete files from this File Share, select the name of the File Share.
- To edit the File Share, select the edit icon or select the name of the File Share and select *ACTIONS - EDIT*.

Warning: Repointing a File Share that is in use may cause loss of file references. Ensure data is mirrored first.

- To delete a File Share, select the trash icon or select the name of the File Share and select *DELETE*.

Warning: When deleting a File Share, all Deployment Versions and Backups associated with the File Share will be deleted.

NFSv3 File Shares

Note: Configure access to the NFS folder on the NFS Provider prior to adding the NFSv3 File Share.

Note: Upon save Morpheus will create a persistent mount owned by `morpheus-app.morpheus-app` on the Morpheus Appliance for the NFSv3 File Share. The Morpheus appliance will require access to the following ports in order to mount the share: 111, 54302, 20048, 2049, 46666, 42955, 875. With some storage solutions, you may need to enable insecure, unprivileged ports, or allow non-root on the export before Morpheus is able to successfully mount the share.

To Add a NFSv3 File Share:

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the FILE SHARES tab, Click the + *ADD* button.
4. Select *NFSv3* from the dropdown list
5. From the NEW FILE SHARE Wizard input the following:

NAME Name of the File Share in Morpheus.

HOST Enter the File Share path on the local Morpheus Appliance.

EXPORT FOLDER Enter the NFSv3 Folder

Default Backup Target Sets this File Share as the default backup target when creating Backups. If selected the option to update existing Backup configuration to use this File Share will be presented.

Archive Snapshots Enabled to export VM snapshots to this File Share when creating VMware Backups, after which the snapshot will be removed from the source Cloud.

Default Deployment Archive Target Sets this File Share as the default storage target when uploading Deployment files in the *Deployments* section.

Default Virtual Image Store Sets this File Share as the default storage target when uploading Virtual Images from the *Virtual Images* section, importing Images from Instance Actions, creating Images with the *Image Builder* and when creating new images from *Migrations*.

RETENTION POLICY

None Files in the File Share will not be automatically deleted or backed up.

Backup Old Files

This option will backup files after a set amount of time and remove them from the File Share.

DAYS OLD Files older than the set number of days will be automatically backed up to the selected Backup File Share.

BACKUP File Share Search for and select the File Share the files will be backed up to.

DELETE OLD FILES

This option will delete files from this File Share after a set amount of days.

DAYS OLD Files older than the set number of days will be automatically deleted from the File Share.

6. Select *SAVE CHANGES*

The File Share will be created and displayed in the File Shares tab.

- To browse, upload, download, or delete files from this File Share, select the name of the File Share.
- To edit the File Share, select the edit icon or select the name of the File Share and select *ACTIONS - EDIT*.

Warning: Repointing a File Share that is in use may cause loss of file references. Ensure data is mirrored first.

- To delete a File Share, select the trash icon or select the name of the File Share and select *DELETE*.

Warning: When deleting a File Share, all Deployment Versions and Backups associated with the File Share will be deleted.

Volumes

Volumes sync or created in Morpheus can be viewed in *Infrastructure- Storage - Volumes*. Volumes can be added for Storage Servers integrated with Morpheus in the *Infrastructure- Storage - Servers* section.

Volumes Types

The available Volume Types list and filterable by are:

- 3Par Volume
- Alibaba Cloud SSD
- Alibaba Efficiency Disk
- Alibaba Cloud Disk
- AWS gp2
- AWS io1
- AWS sc1
- AWS st1
- Azure Volume
- Azure Disk
- Bluemix Disk
- Bluemix SAN
- Bluemix SAN
- CD ROM
- DO Disk
- ECS Block Storage
- ECS Object Storage
- ECS Shared File System
- Floppy Disk

- Google Standard
- HP Enclosure Disk
- Oracle iSCSI
- Isilon NFS Volume
- Nutanix IDE
- Nutanix SATA
- Nutanix SCSI
- Open Telekom Volume
- Openstack Disk
- Openstack Volume
- Oracle Block Volume
- Oracle Disk
- Oracle Virtual Volume
- SCVMM Datastore
- Softlayer Disk
- Softlayer SAN
- Softlayer SAN
- Disk
- UpCloud Disk
- UpCloud MaxIOPS
- NULL
- NULL
- VMWare Datastore
- VMWare Disk

CREATE VOLUME

At least one Storage Server Integration from *Infrastructure- Storage - Servers* is required to create volumes from *Infrastructure- Storage - Volumes*.

3par

To Add a 3Par Volume:

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the VolumeS tab, Click the + *ADD* button.
4. Select *3Par* from the dropdown list
5. From the CREATE VOLUME Wizard input the following:

SELECT TYPE

STORAGE SERVER Name of the 3par Storage Server added in *Infrastructure- Storage - Servers*

GROUP Select Storage Group

VOLUME TYPE 3Par Volume

Click **NEXT** Select *NEXT*

CONFIGURE

NAME Name of the Volume

VOLUME SIZE Specify size of the Volume (in MB)

PROVISION TYPE

- FULL
- TPVV
- SNP
- PEER
- UNKNOWN
- TDVV

Click **COMPLETE** Select *COMPLETE*

Dell EMC ECS

To Add a Dell EMC ECS Volume:

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the VolumeS tab, Click the + *ADD* button.
4. Select *Dell EMC ECS* from the dropdown list
5. From the CREATE VOLUME Wizard input the following:

SELECT TYPE

STORAGE SERVER Name of the DELL EMC ECS Storage Server added in *Infrastructure- Storage - Servers*

GROUP Select Storage Group

VOLUME TYPE ECS Block Storage ECS Object Storage ECS Shared File System

Click **NEXT** Select *NEXT*

CONFIGURE

NAME Name of the Volume

Click **COMPLETE** Select *COMPLETE*

Dell EMC Isilon

To Add a Dell EMC ECS Volume:

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the VolumeS tab, Click the + *ADD* button.
4. Select *Dell EMC Isilon* from the dropdown list
5. From the CREATE VOLUME Wizard input the following:

SELECT TYPE

STORAGE SERVER Name of the Dell EMC Isilon Storage Server added in *Infrastructure- Storage - Servers*

GROUP Select Storage Group

VOLUME TYPE Isilon NFS Volume

Click **NEXT** Select *NEXT*

CONFIGURE

NAME Name of the Volume

ALLOWED IP's

Specify IP Addresses to limit accessibility to the File Share

Leave blank for open access Click the + symbol to the right of the first ALLOWED IPS field to add multiple IP's

VOLUME SIZE Specify size of the Volume (in MB)

Click **COMPLETE** Select *COMPLETE*

Data Stores

Data Stores are logical divisions of underlying storage disk. Organizations may use them to divide and track cloud resources by team or department. When integrating certain Cloud types, Morpheus will onboard all existing data stores and administrators can then make them available to Groups or Tenants as needed. At provision time, when applicable based on Cloud and Layout, users can select the data store they wish to provision to.

Here within the Data Store view in the storage section, users can see a list of data stores for each Cloud. In the row for each Cloud, the storage type, associated Cloud, and permissions information are shown.

Create Data Stores

To a limited extent, data stores can be created from this view. Currently, data store creation is restricted to VMware data store creation on 3Par volumes. In order to create such a data store you would need to first have an integrated 3Par server. See the section on storage servers for more information on setting up this integration.

Note: For all other data store types, create the needed data store within the target Cloud and Morpheus will automatically sync in the data store on the next Cloud sync. You can force a Cloud sync from the Cloud Detail Page (Infrastructure > Clouds > Selected Cloud > Refresh Menu > Short).

- Navigate to Infrastructure > Storage > Data Stores
- Click *+ADD*
- Enter a Name, select a VMware Cloud, select a 3Par Volume, and select a Host Group
- Manage permissions in the Group Access and Tenant Permissions sections, if needed
- Click *SAVE CHANGES*

Manage Permissions

From this view, users can manage permissions for any data store synced from integrated Clouds. This includes setting which Groups have access to the data store, and which Tenants have access. To edit data store permissions:

- Navigate to Infrastructure > Storage > Data Stores
- Click ACTIONS > Edit
- **Groups:** Select “all” Groups or select specific Groups which should have access to the data store
- **Tenants:** Primary Tenant users can opt to make the data store available to all Tenants (public visibility) or to selected Tenants (private visibility with specific Tenants selected). Subtenant users will only be able to make data stores visible to their own Tenant
- **Active:** When marked, the data store is active and available for provisioning
- Click *SAVE CHANGES*

Servers

Add Storage Server

Adding 3Par Storage Server

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the SERVERS tab, Click the *+ ADD* button.
4. From the ADD STORAGE SERVER wizard input the following:
 - NAME** Name of the Storage Server in Morpheus
 - TYPE** Select *3Par*
 - URL** URL Of 3Par Server Example : *https://192.168.190.201:8008*
 - USERNAME** Add your administrative user account.
 - PASSWORD** Add your administrative password.
5. Select *SAVE CHANGES*

The 3Par Storage Server will be added and displayed in the Buckets tab. Buckets, Files Shares and Storage Groups will be synced in.

Adding Dell EMC ECS Storage Server

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the SERVERS tab, Click the + *ADD* button.
4. From the ADD STORAGE SERVER wizard input the following:

NAME Name of the Storage Server in Morpheus

TYPE Select *Dell EMC ECS*

URL

URL Of DELL EMC ECS Server Example : *https://192.168.190.200:4443*

Tip: The port 4443 is the api port for ECS api. This may be different depending on your configuration

USERNAME Add your administrative user account.

PASSWORD Add your administrative password.

S3 SERVICE URL (Optional) Add your S3 service url Example: *http://192.168.190.220:9020*

Note: S3 SERVICE URL is not required if you are not planning on using ECS S3.

5. Select *SAVE CHANGES*

The Dell EMC ECS Storage Server will be added and displayed in the Buckets tab. Buckets, Files Shares and Storage Groups will be synced in.

Adding Dell EMC Isilon Storage Server

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the SERVERS tab, Click the + *ADD* button.
4. From the ADD STORAGE SERVER wizard input the following:

NAME Name of the Storage Server in Morpheus

TYPE Select *Dell EMC Isilon*

URL URL Of Dell EMC Isilon Server Example : *https://192.168.190.202:8080*

USERNAME Add your administrative user account.

PASSWORD Add your administrative password.

PROVISION USER Select Provision User

PROVISION GROUP Select Provision Group

ROOT PATH

Enter Root Path Example : \

5. Select *SAVE CHANGES*

The Dell EMC Isilon Storage Server will be added and displayed in the Buckets tab. Buckets, Files Shares and Storage Groups will be synced in.

5.5.8 Trust

The Trust section is where credentials, SSH keypairs, and SSL certificates are stored. In addition, related integrations to outside technologies can be made in this section as well. Integration types include Morpheus Cypher and Hashicorp Vault for externalized credential storage. Continue onto the next section for more on standing up an external Cypher credential store.

Credentials

The credentials section allows for various credential types to be securely stored and called back when necessary, such as when creating new integrations with Cloud accounts or other outside technologies. Credentials can also be used to populate REST-based Option Lists sourced from data behind an authentication wall, as well as to run automation Tasks on remote targets that require authentication. Credentials can be securely stored internally on the appliance or stored in an external Cypher integration, more information about setting up and integrating with an external Cypher store are in the next section. The following credential pair types are currently supported:

- Access Key and Secret Key
- Client ID and Secret
- Email and Private Key
- OAuth 2.0
- Tenant, Username, and Keypair
- Username and API Key
- Username and Keypair
- Username and Password
- Username, Password, and Keypair

To create a new credential set, click + *ADD* and then select the type of credential set you'd like to store. Complete the following:

- **CREDENTIAL STORE:** Select "Internal", an integrated external Cypher store (if any), or an integrated Hashicorp Vault server (if any). See the section below for instructions on integrating with Vault or standing up and integrating with an external Cypher store.
- **NAME:** A name for the credential set in Morpheus
- **DESCRIPTION:** An optional description for the credential set
- **ENABLED:** If checked, the credential set will be available for use
- **CREDENTIAL VALUES:** Depending on the credential pair type selected (listed above), the remaining fields will be specific to the chosen type. See the next section for a more complete walkthrough on storing and using OAuth 2.0 credentials

Finally, click *ADD CREDENTIALS*. Once saved, the credential set will be available for selection where appropriate in Morpheus UI. In the screenshot below, I'm integrating a new VMware Cloud. In the credentials section, I have the following options: Creating (and using) a new Username and Password credential set (which includes the option to save internally or to an external Cypher store), choosing a previously-stored credential set, or simply entering my credentials locally and not saving them for reuse.

OAuth 2.0 Credentials

Morpheus supports storage of credential sets for retrieving temporary access tokens, through OAuth 2.0, and using the tokens to access some resource. These credential sets can be used with REST-type Option Lists to retrieve information behind this type of authentication wall. Once stored, the credential can be used with as many Option Lists as needed and potentially in other areas of the product in the future.

To create a new credential set, click + *ADD* and then select “OAuth 2.0”. Complete the following, not all fields are present or required in every context:

- **CREDENTIAL STORE:** Select “Internal” or an integrated external Cypher store (if any). See the next section for instructions on standing up and integrating with an external Cypher store
- **NAME:** A name for the credential set in Morpheus
- **DESCRIPTION:** An optional description for the credential set
- **ENABLED:** If checked, the credential set will be available for use
- **GRANT TYPE:** Client Credentials or Password Credentials
- **ACCESS TOKEN URL:** The authorization server’s token endpoint
- **CLIENT ID:** The client ID for an app registered with the target service

- **CLIENT SECRET:** The client secret, often needed when requesting access outside the context of a specific user
- **USERNAME:** (Only present with “Password Credentials” Grant Type) The username for a user with target data access
- **PASSWORD:** (Only present with “Password Credentials” Grant Type) The password for the user indicated above
- **SCOPE:** The scope of access requested to the target resource
- **CLIENT AUTHENTICATION:** “Send as basic auth header” or “Send client credentials in body” - Indicates how Morpheus should issue the token received in requests to the target resource

Once done, click *ADD CREDENTIALS*.

With the OAuth 2.0 credential set stored, they can be set on REST-type Option Lists to source data from behind a compatible authentication wall. With a REST-type Option List open (Library > Options > Option Lists), click the CREDENTIALS dropdown and select the credential set you’ve created. Alternatively, you can add a credential set directly in the add/edit Option List modal if needed. Option Lists can be associated with Select List or Typeahead-type Inputs and applied to Layouts, Instance Types, Workflows, and more to allow for customization at provision or Workflow execution time. Additional details on creating Option Lists can be found in the Library section of Morpheus docs.

Integrating Hashicorp Vault

The Hashicorp Vault integration is not included with Morpheus by default. Download the plugin from [Morpheus Exchange](#) and add the plugin to Morpheus through the [Plugins](#) section. This allows users to store credential sets completely outside of Morpheus and in Hashicorp Vault, which may be required by your organization’s IT policies.

Note: The plugin space is universal and not specific to Tenants. If Subtenant users have access to Administration > Integrations > Plugins, any integrated plugins will be available in all Tenants across the appliance. In most cases, it makes sense to restrict access to this section from Subtenant users through the associated Tenant Role. Instead integrate plugins from the Primary Tenant and expose them to various Subtenants as needed.

Once downloaded, plugins are added to Morpheus in Administration > Integrations > Plugins. Simply browse your local filesystem for the JAR file downloaded from Morpheus Exchange and its capabilities will immediately be added to the appliance. After adding the plugin, configure access for the plugin to your Vault server. Do this by clicking the Edit (pencil) button in the row for the Vault plugin. Supply a URL for your Vault server and an access token, then save your changes.

Note: When creating a Vault integration, it’s recommended that you use a long-lived token. If the token suddenly becomes invalid, Morpheus will be unable to write new credential sets to Vault and will be unable to edit or delete any existing ones. Additionally, you won’t be able to use Vault-stored credential sets elsewhere in Morpheus, such as when creating new Cloud integrations or populating REST-based Option Lists which require authentication. Should this happen, simply obtain a new token, edit the Vault integration, update the token, and save your changes.

With the plugin added, a new integration type will appear in Infrastructure > Trust > Integrations. Click + *ADD*, then “Hashicorp Vault Credentials” to get started. The fields in the list below are available for configuration but it’s possible that no configuration will be necessary. If you do not enter a new API URL and TOKEN value, these are taken from the plugin configuration set a moment ago. Similarly, The Vault Secret Engine and Secret Path can be left at default values (or empty) if those values are acceptable. If you need to override the defaults or the URL/token set on the plugin, you may do so here.

- **NAME:** A friendly name for the Vault integration in Morpheus
- **ENABLED:** When marked, this Vault integration will be available to have credentials written to it
- **API URL:** The URL for the Vault server (ex. <http://xx.xx.xx.xx:8200>)
- **TOKEN:** A valid API token for the server (see note below)
- **HASHICORP VAULT SECRET ENGINE:** Select KV Engine version 1 or 2, additional engines may be available in the future
- **ENGINE MOUNT:** If desired, enter a custom engine mount. By default, if left empty, credentials are written to the “secret/” engine mount
- **SECRET PATH:** If desired, enter a custom path and Morpheus will write new credential sets to that path. By default, if left empty, new credentials are written to “morpheus-credentials/”

When done, click *SAVE CHANGES*.

With the above process finished, this Vault integration will be available as a storage target when creating new credential sets. In Infrastructure > Trust > Credentials, after clicking + *ADD* and selecting the type of credential set to add, select the new Vault integration in the CREDENTIAL STORE field (default selection is “Internal”).

Installing and Integrating an External Cypher Appliance

The external Cypher appliance runs on a small separate VM and supports a variety of base OS distributions. Credentials are securely passed to the external appliance and can be retrieved and consumed in specific places within Morpheus UI. The download URL for the installer can be retrieved from Morpheus [Hub](#), replace the placeholder URL in the instructions below with the correct URL for the latest version of the Cypher appliance.

Begin by provisioning and updating the VM for the Cypher appliance. Then, download the installer. The following steps go through the installation process on Ubuntu but, as mentioned in the previous paragraph, many popular distributions are supported.

```
# An example URL is shown below, find the URL for the latest version and for the correct
↪distro at |morpheus| Hub
wget https://downloads.morpheusdata.com/path/to/morpheus-cypher_${version}_amd64.deb
```

Next, install and reconfigure the package.

```
sudo dpkg -i morpheus-cypher_${version}_amd64.deb
sudo morpheus-cypher-ctl reconfigure
```

After the installation and reconfigure is complete, we need to record the generated API key so we can integrate the external Cypher store with Morpheus in a later step. We can get this from the logs with the following command:

```
sudo morpheus-cypher-ctl tail

==> /var/log/morpheus-cypher/cypher/current <==
2022-02-02_15:22:27.84848 |  \  ( )  _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ | |
2022-02-02_15:22:27.84848 | | \ | | | / _ | ' _ / _ \ | ' _ \ / _ ` | | | | _ |
2022-02-02_15:22:27.84848 | | | | | ( _ | | | ( ) | | | | ( _ | | | | | _
2022-02-02_15:22:27.84848 | _ | | _ | \ _ _ | _ | \ _ _ / | _ | | \ _ _ , - | \ _ _ |
2022-02-02_15:22:27.84849   Micronaut (v3.2.2)
```

(continues on next page)

(continued from previous page)

```
2022-02-02_15:22:27.84849
2022-02-02_15:22:28.09130 15:22:28.087 [main] INFO i.m.context.env.DefaultEnvironment -
↳Established active environments: [ec2, cloud]
2022-02-02_15:22:30.15129 15:22:30.151 [main] INFO c.m.cypher.service.CypherService -
↳Root Data: null
2022-02-02_15:22:30.83499 15:22:30.834 [main] INFO c.m.cypher.service.CypherService -
↳Initialized Root Token: c90xxxx00000xxxxxx000000xxxxx000 ... Write this down as it
↳will only display once
2022-02-02_15:22:32.01282 15:22:32.012 [main] INFO io.micronaut.runtime.Micronaut -
↳Startup completed in 4749ms. Server Running: http://localhost:8080
```

Important: The API key is only shown once when the appliance is first installed. Securely store this API key for later reference or you will be unable to integrate this Cypher appliance with any other Morpheus appliances.

This completes the installation process, move to Morpheus UI to integrate the remote Cypher store with Morpheus. Cypher integrations are added in Infrastructure > Trust > Integrations. Click + *ADD* and then click Cypher. Configure the following:

- **NAME:** A name for the Cypher integration in Morpheus
- **ENABLED:** When checked, this Cypher integration is available for storing and retrieving credentials
- **API HOST:** The URL where your Cypher appliance can be reached (ex. <https://x.x.x.x/>)
- **API KEY:** The API Key we retrieved and saved in the previous step

EDIT INTEGRATION ×

NAME

ENABLED

API HOST
Warning! Using HTTP URLs are insecure and not recommended.

API KEY

Click *SAVE CHANGES* to save the new integration. Refer to the “Credentials” section above for details on storing new credential sets using the external appliance and how they can be called back in various places throughout the UI.

Key Pairs

The key pairs section enables the following actions: Add and Delete key pairs. Key pairs are commonly used by Morpheus for accessing instances via SSH. Morpheus stores key pairs to simplify administration and access across both private and public clouds.

Morpheus only accepts key pairs in PEM format (for example, a private key beginning with -----BEGIN RSA PRIVATE KEY-----). If you have a key in another format, such as OpenSSH, convert the key:

```
#No passphrase
ssh-keygen -m pem -f /path/to/key

#With passphrase
ssh-keygen -p -P "old passphrase" -N "new passphrase" -m pem -f path/to/key
```

Add Existing Key Pair

To generate a existing Key Pair:

1. Navigate to Infrastructure > Trust > Key Pairs
2. On the Key Pairs tab, click + *ADD* and select “Existing Key Pair”
3. From the Add Key Pair modal input the following as needed:
 - Name
 - Public Key
 - Private Key
 - Passphrase

Note: Certain features do not require storage of the private key.

Generate Key Pair

To generate a Key Pair:

1. Navigate to Infrastructure > Trust > Key Pairs
2. On the Key Pairs tab, click + *ADD* and select “Existing Key Pair”
3. After naming the new key pair, Morpheus will reveal both the public and private key

Note: After the private key is initially revealed it will not be shown again. If needed, you may view the public key from the Keypairs list page at any time going forward. This key pair can be associated with your Linux user details in Morpheus user settings. The public key will be added to the `authorized_keys` file on provisioned workloads where your Linux user is added at provision time.

Delete Key Pair

To Delete Key Pair:

1. Navigate to Infrastructure > Keys & Certs
2. On the Key Pairs tab, select the trash can icon at the end of any row
3. Acknowledge that you wish to delete the selected key pair

SSL Certificates

SSL certificates authenticate the identity of web servers and encrypt the data being transmitted. Morpheus stores SSL certificates to simplify administration and application of SSL certificates to Morpheus-managed resources.

Add SSL Certificate

1. Navigate to Infrastructure > Keys & Certs
2. On the SSL Certificates tab, click + *ADD*
3. From the Add SSL Certificate wizard input the following as needed:
 - Name
 - Domain Name
 - Key File
 - Cert File
 - Root Cert

Delete SSL Certificate

To Delete SSL Certificate:

1. Navigate to Infrastructure > Keys & Certs
2. On the SSL Certificates tab, select the trash can icon at the end of any row
3. Acknowledge that you wish to delete the selected SSL Certificate

Trust Integrations

Some organizations may use outside technologies to manage their key and certificates. Morpheus allows users to integrate with Venafi for trust management. Trust management integrations can be managed from the Integrations tab on the Infrastructure > Keys & Certs page. Additionally, they can be managed in Administration > Integrations.

Currently, Morpheus supports trust integration Venafi. For more detailed information on integrating Venafi with Morpheus, take a look at our [integration guide](#).

5.5.9 Boot

Overview

Morpheus provides a simple-to-use Bare Metal boot capability based on PXE. When a server boots and is redirected to the Morpheus server for the installation files, they can be configured to be simply passed an OS or Hypervisor (in which case Morpheus will see them as Bare Metal servers with no further detail) or they can be brought on as Virtual Machines or Docker Hosts. Installation of the Morpheus Agent can also be done during the initial configuration stage.

Prerequisites

In order to work with Morpheus bare metal PXE boot capabilities, some initial setup configuration is required. First, on the Morpheus server, if you have a firewall enabled, make sure port 69 is open for TFTP. Morpheus actually uses port 6969 and during installation a redirect should have been set. To check this, SSH onto the Morpheus server and run the following:

```
iptables -t nat -L -n -v
```

If the redirect is still properly, the response should include the following:

```
Chain PREROUTING (policy ACCEPT 69 packets, 9767 bytes)
pkts bytes target  prot opt in out source      destination
0      0 REDIRECT udp  --  *  *   0.0.0.0/0  0.0.0.0/0      udp dpt:69 redir ports 6969
0      0 DNAT    tcp  --  *  *   0.0.0.0/0  169.254.169.254 tcp dpt:80 to:192.168.1.156
```

Next, in Morpheus, set a default PXE root password. This password is set in Administration > Settings > Provisioning. With the default root password set, set up the redirect on the DHCP server. In addition to the DNS and Gateway settings, add Boot Server Host Name which will be the name of the Morpheus Server and Bootfile Name which should be set to `pxelinux.0`. If you are using a Linux-based DHCP server, for example on CentOS, the `dhcpd.conf` configuration will look something like the following:

```
allow booting;
allow bootp;
option option-128 code 128 = string;
option option-129 code 129 = text;
next-server xx.xx.xx.xx;
filename "pxelinux.0";
```

Note: Replace the dummy IP address in the example `dhcpd.conf` file above with your Morpheus appliance IP address.

Once you have done this, when you boot a PXE-enabled machine on the network, it will be told to access the Morpheus server and request the `pxelinux.0` file. It will do this on port 69, the default for TFTP and will be redirected to 6969 once it hits the Morpheus server. If successful you will see the “Morpheus PXE Server” menu when you boot a server. This is the default menu defined in Morpheus and supports the shipped PXE images supplied with the product. By selecting any of the choices from the “Morpheus PXE Server menu”, the install files should be downloaded and the server configured as per the supplied kickstart files. At this point, back on the Morpheus appliance, you should see the MAC address for the new server appear in the “Discovered MAC Addresses” tab of the Infrastructure > Boot page.

Troubleshooting

If you do not get the Morpheus boot menu, there are a few things to check:

- First, make sure the filename is correct. It must be `pxelinux.0`
- Next, check the TFTP server is responding by using a TFTP client to get the `pxelinux.0` file from the Morpheus server using the same host name as you have configured in the DHCP server configuration. Do this test on a machine on the same network as the machines you are trying to boot using PXE
- Leave the port number as 69 (the default) as this will also check the redirect is working
- If a GET call on the default port does not work, and the client allows (most do) try using port 6969. If this works, then the redirect is wrong
- If it will not work on either, check you can access the Morpheus server from the network you are on and also check there are no firewalls between the test network and the Morpheus Server

Mapping

Add Mapping

1. Select the Mapping tab then click the Add Mapping button.

2. From the New Mapping Wizard input the following information:

Match Pattern Mac address separated by ':' or an ip address filter

Description(optional) Description of the new mapping.

Active Flag to denote the mapping as active or disabled.

Operating System List of operating systems for the mapping.

Boot Image Lists available PXE boot images.

Answer File Lists available answer files.

Cloud Lists the available clouds.

Server Mode List of server modes:: unmanaged, Managed, Bare metal host, Container host, VM host, and Container & VM host.

3. Save

Once the mapping is added, and the target host is powered on, the {morpheus} PXE menu will load and PXE boot will start.

Edit Mapping

1. Click the edit icon on the row of the mapping you wish to edit.

2. Modify information as needed.

3. Click the Save Changes button to save.

Delete Mapping

1. Click the delete icon on the row of the mapping you wish to delete.

Boot Menus

System-seeded Boot Menus are displayed and user-created Boot Menus can be edited and deleted. User-created Boot Menus are edited or deleted by clicking on the pencil or trash can icon in the appropriate row.

Adding a Boot Menu

To begin, click + *ADD*. Available fields include:

- **NAME:** Name of the Boot Menu
- **DESCRIPTION:** Description of the Boot Menu
- **TYPE:** Select between **bios**, **uefi**, **ipxe** and **grub**
- **ENABLED:** Determines if the Boot Menu is active
- **DEFAULT MENU**
- **ROOT MENU**
- **MENU NAME**
- **BOOT IMAGE**
- **ANSWER FILE**
- **MENU CONTENT**
- **SUB MENUS**

Click *SAVE CHANGES*

Answer Files

Answer files are like lists of answers for questions that you know the setup program is going to ask but the user is not prepared to answer. They contain one or more sections, and each section contains one or more properties in the form name=value. Morpheus provides Answer Files for ESXi, CentOS, Ubuntu and XenServer, and user can add their own.

Add Answer Files

1. Click the Infrastructure link in the navigation bar.
2. Click the Boot link in the sub navigation bar
3. Select the Answer Files tab then click the Add Answer File button.
4. From the New Answer File Wizard input the following information

Name Name of the answer file.

Description(optional) Description of the new answer file.

Active Flag to denote the mapping as active or disabled.

Script Name Name of the new answer file.

Script Version Version of the new answer file.

Script The script for the new answer file.

5. Save

Edit Answer File

1. Click the Infrastructure link in the navigation bar.
2. Click the Boot link in the sub navigation bar
3. Select the Answer Files tab
4. Click the edit icon on the row of the answer file you wish to edit.
5. Modify information as needed.
6. Save Changes

Delete Answer File

1. Click the Infrastructure link in the navigation bar.
2. Click the Boot link in the sub navigation bar
3. Select the Answer Files tab.
4. Click the delete icon on the row of the answer file you wish to delete.

Images

Morpheus provides Images for ESXi, CentOS, Ubuntu and XenServer, and user can add their own Images.

Add Images

1. Click the Infrastructure link in the navigation bar.
2. Click the Boot link in the sub navigation bar
3. Select the Images tab then click the Add Image button.
4. From the Upload Virtual Image Wizard input the following information
 - Name** Name of the Image.
 - Operating System** List of available operating systems.
 - Storage Provider** List of available storage providers.
 - Image Path** Path of the image.
 - Visibility** Private or Public
 - Account** List of accounts to allow permission to this image.
5. Save Changes

Edit Image

1. Click the Infrastructure link in the navigation bar.
2. Click the Boot link in the sub navigation bar
3. Select the Images tab
4. Click the actions drop down and select edit.
5. Modify information as needed.
6. Click the Save Changes button to save.

Convert Image

1. Click the Infrastructure link in the navigation bar.
2. Click the Boot link in the sub navigation bar.
3. Select the Images tab
4. Click the *Actions* drop and select *Convert*.

Download Image

1. Click the Infrastructure link in the navigation bar.
2. Click the Boot link in the sub navigation bar.
3. Select the Images tab
4. Click the *Actions* drop and select *Download*.

Remove Image

1. Click the Infrastructure link in the navigation bar.
2. Click the Boot link in the sub navigation bar.
3. Select the Image tab.
4. Click the *Actions* drop and select *Remove*.

5.6 Backups

Morpheus built-in Backup solution provides VM, Container, Host, Database, File, Directory, Volume and Storage Provider Backup, Snapshot and Replication capabilities. Backups can be automatically configured during provisioning or manually created at any time. Backup Jobs with custom Execution Schedules and retention counts can be created and used across all environments in conjunction with configured Storage Providers. Backups can be restored over current Instances or as new Instances, and downloaded or deleted from Morpheus.

Morpheus also integrates with external services to automate availability with other providers.

5.6.1 Initial Backups Setup

Global Backup settings, Storage Providers and Execution Schedules should be configured prior to creating backups.

Global Backups Settings

Morpheus Backups can be enabled under Administration > Settings > Backups.

Scheduled Backups When enabled, configured Backups will automatically run on the set Schedule. If disabled, backups need to be manually ran.

Create Backups When enabled, Morpheus will automatically configure backup jobs for Instances.

Backup Appliance When enabled, a Backup will be created to backup the Morpheus appliance database. Select the Backup text link to edit Appliance Backup Settings and view existing Appliance Backups.

Default Backup Storage Provider Storage Providers can be configured and managed in the *Infrastructure > Storage* section.

Default Backup Schedule Schedules can be configured and managed in the Library > Automation > Power Scheduling

Backup Retention Count Default maximum number of successful backups to retain.

Backup Schedules

Backup Execution Schedules can be configured and managed in Library > Automation > Execute Scheduling. An execution schedule stores only the interval at which some execution should be run. To create a new backup job with this schedule, navigate to *Backups > Backups* and click “+ADD”. In the final step of creating the backup job we are able to select any of our created execution schedules. The Default Backup Schedule set in Administration > Settings > Backups will be selected when creating a backup job and not specifying an execution schedule.

5.6.2 Configuring Backups during Provisioning

When Backups are enabled, Backup options are presenting in the Automation tab of the Provisioning wizard.

Note: The Backup options presented in the Automation tab can be disabled using a “Create Backup” policy. See *Policies*

BACKUP TYPE Select the type for the Backup. Backup Types displayed will be filtered by available options per selected Instance Layout.

BACKUP NAME Defaults to Instance name

BACKUP TARGET Select Storage Provider target for the Backup (when applicable).

BACKUP JOB TYPE Create New, Clone, or Add to existing Job

JOB Name Defaults to Instance name

RETENTION COUNT Maximum number of successful backups to retain.

BACKUP SCHEDULE Select the schedule the Backup Job will be executed.

Backup Types displayed will be filtered by available options per selected Instance Layout. Backup Job Types include:

- File Backup
- Directory Backup

- MySQL
- MongoDB
- LVM Snapshot
- LVM Image
- LVM Migration
- Windows Migration
- Postgres
- Tar Directory Backup
- Amazon VM Snapshot
- VMWare VM Snapshot
- Fusion VM Snapshot
- Xen VM Snapshot
- Veeam VMWare VM Backup
- Veeam Hyper-V VM Backup
- Google VM Snapshot
- Commvault File/Directory Backup
- Azure VM Snapshot
- Morpheus Appliance
- Openstack VM Snapshot
- DigitalOcean VM Snapshot
- Nutanix VM Snapshot
- Softlayer VM Snapshot
- Hyper-V VM Snapshot
- VMWare VM Snapshot
- SCVMM VM Snapshot
- UpCloud VM Snapshot
- Bluemix VM Snapshot
- Alibaba VM Snapshot
- Oracle Cloud VM Snapshot
- KVM VM Snapshot
- Container Backup
- VM Backup
- Object Storage Backup

5.6.3 Summary

The Backups Summary section shows the following metrics

- Number of Configured Backups trend
- Backup Success Rate
- Number of Completed Backups
- Number of Failed Backups
- Total Size of Backups (MB) trend
- Upcoming and In Progress Backups

If a User's Role permission for Backups is set to *User*, the user will only see metrics for backups they own.

5.6.4 Backups

In the *Backups > Backups* section, currently configured Backups can be viewed and managed, and new Instance, Host and Provider backups be configured.

Note: Role permissions for Backups determine which backups will be accessible per user.

Manage an existing Backup

1. Select the Backups link in the navigation bar.
2. Select the Backups link in the sub navigation bar.
3. Select the name of the Backup to view the Backups detail page.

Create Instance Backup

To create instance backup

1. Select the Backups link in the navigation bar.
2. Select the Backups link in the sub navigation bar.
3. Click the Add Backup button.
4. From the Create Backup Wizard select the radio button Instance, then click Next.
5. Input the following:
 - Name** Name of the backup job being created.
 - Instance** Select an instance to backup from the dropdown.
6. Click Next.
7. Depending on the instance type selected in the previous step, enter additional details such as:
 - Database Name
 - Username
 - Password

- Container
 - etc..
8. Click the Next button.
 9. Schedule the backup Days, Time, Storage Provider & Retention Count.
 10. Click Complete to save.

Note: On VMware Cloud types, Morpheus will merge and consolidate the snapshots held against a VM before exporting the OVF to the storage location or share. This is so Morpheus has a full and consistent copy of the VM state.

5.6.5 Managing Backups

Overview

Backups are automatically configured and performed on each new Morpheus -provisioned Instance. Users can edit the frequency of backups. Administrators can define destination targets where backups are stored and perform all user-based tasks.

To View Backups:

Select the Backups link in the navigation bar.

Note: If backups are disabled, they are still created upon instance provisioning and can be executed manually. However, backups will not be executed on a schedule automatically. Scheduled backups must be enabled by an administrator to run automatically. To review how to enable/disable backups see [here](#).

Backup View

Review information about configuration such as: schedule, target details, total amount and successfully run backups, total and average size of backups from the Backup Page.

To Display Backup

1. Select the Backups link in the navigation bar.
2. Select the Backups link in the sub navigation bar.
3. Clicking the backup name to review its details.

Create Instance Backup

To create instance backup

1. Select the Backups link in the navigation bar.
2. Select the Backups link in the sub navigation bar.
3. Click the Add Backup button.
4. From the Create Backup Wizard select the radio button Instance, then click Next.
5. Input the following:
 - Name** Name of the backup job being created.
 - Instance** Select an instance to backup from the dropdown.
6. Click Next.
7. Depending on the instance type selected in the previous step, enter additional details such as:
 - Database Name
 - Username
 - Password
 - Container
 - etc..
8. Click the Next button.
9. Schedule the backup Days, Time, Storage Provider & Retention Count.
10. Click Complete to save.

5.7 Monitoring

5.7.1 Overview

Morpheus provides great monitoring features out of the box. Anything provisioned within Morpheus automatically gets a check created in the monitoring service. These checks are organized hierarchically in “Groups” and “Apps”. This makes it easy to gain a perspective as to what a customer or full stack facing impact is in the event of a particularly instance failure. This also takes into account redundancy layers when it comes to calculating the applications overall uptime percentage.

There are also several integrations built into the monitoring subsystem of Morpheus including App Dynamics , New Relic, and even Service Now integration.

Logs

Overview

The logging architecture backing Morpheus uses the latest and greatest technologies and standards to be able to service large amounts of log traffic as well as facilitate easy viewing. Utilizing elasticsearch behind the scenes and buffered log transmission protocols Morpheus provides a highly efficient and highly scalable solution for capturing log data from anything provisioned via the system. By utilizing common formats (syslog) it is also very easy to forward logs to external third party log services.

Configuration

Logging configuration can be setup in the Administration > Settings > Logs section. There are useful settings here, including customizing the retainment policy (7 days by default). This could be expanded to years for PCI compliance purposes or other requirements an organization might have.

Note: When increasing the retainment policy of the logging system, it may be necessary to scale out the elasticsearch cluster. Please refer to the relevant information with regards to scaling elasticsearch and advanced installation options for externalizing the elasticsearch cluster.

The Log administration section also provides options for setting custom syslog forward rules. These rules are applied on each individual host therefore keeping the Morpheus appliance itself out of the data plane. For information on different syslog formatting rules please refer to the [rsyslog](#) documentation.

Usage

Morpheus automatically sets up and configures logging for all of the standard catalog items provisioned through morpheus. This includes both Docker containers as well as virtual machines. Simply view instance-specific logs in instance detail via the “Logs” tab.

There are several filtering capabilities built into the logging UI with more being added continually. Easily toggle log level filters from the dropdown or change the date range filter using the handy date filter component. A chart is also displayed above logs representing the log counts by level over the selected time range (default last 24 hours). A handy pattern search is also available with some rather capable features based on Lucene search syntax.

Tip: It may be useful to review the [Lucene search query syntax](#) for powerful use cases.

There are several other places logs can be viewed. Not only can they be viewed across an application in app detail but also across all instances in the account. The main level Logs section provides an ability to query all logs produced by the system. It is also possible to view host-specific logs on a docker host by viewing the host detail page via [Infrastructure](#).

Note: New features are on the roadmap for the main logs section including saved searches, and handy charting dashboards for garnering insights out of log data.

Exporting Logs

Log Settings

There are three main log areas in Morpheus

- Agent Logs
- Morpheus Server Logs
- Activity / Audit Logs

Agent Logs

When Instances are deployed through Morpheus, the installed Agent captures application logs and sends them back to the Morpheus server.

In most cases, the built-in Morpheus logging features are sufficient for tracking and reviewing Agent logs. However, if needed, Morpheus supports integration with advanced logging systems. See the [log integration](#) section above for more information.

Morpheus Server Logs

The main Morpheus application log is in `/var/log/morpheus/morpheus-ui` and the latest log file is named `current`. This log is archived every 24hrs. There are a number of other log files for the individual infrastructure components as well.

If you wish to export these to an external syslog platform, do the following:

1. Once you have configured your syslog destination (edit `rsyslog.conf`), create a `morpheus-syslog.conf` file in the `/etc/rsyslog.d` directory and add the following entries

```
module(load="imfile" PollingInterval="10")
input(type="imfile" File="/var/log/morpheus/morpheus-ui/current" Tag="morpheus-ui"
↳ReadMode="2" Severity="info" StateFile="morpheus-ui")
input(type="imfile" File="/var/log/morpheus/check-server/current" Tag="check-server
↳ ReadMode="2" Severity="info" StateFile="check-server")
input(type="imfile" File="/var/log/morpheus/guacd/current" Tag="guacd" ReadMode="2"
↳Severity="info" StateFile="guacd")
input(type="imfile" File="/var/log/morpheus/elasticsearch/current" Tag=
↳"elasticsearch" ReadMode="2" Severity="info" StateFile="elasticsearch")
input(type="imfile" File="/var/log/morpheus/mysql/current" Tag="mysql" ReadMode="2"
↳Severity="info" StateFile="mysql")
input(type="imfile" File="/var/log/morpheus/nginx/current" Tag="nginx" ReadMode="2"
↳Severity="info" StateFile="nginx")
input(type="imfile" File="/var/log/morpheus/rabbitmq/current" Tag="rabbitmq"
↳ReadMode="2" Severity="info" StateFile="rabbitmq")
```

2. Restart `rsyslog`

The log files will now be forwarded to the destination you have defined.

This configuration is valid for an 'all-in-one' Morpheus server. If the infrastructure components are running on separate servers /clusters, you will need to create the relevant redirects for the logs on those boxes.

Activity Log

The final log type that may require export is the Morpheus Activity log. This tracks system changes made by users, for example create and delete instances etc.

1. To set up CEF/SIEM auditing export, you should edit the following file: `logback.xml` located at `/opt/morpheus/conf/logback.xml`.
2. Add the below appender above or below the other appenders in the `logback.xml` configuration file:

```
<appender name="AUDIT" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>/var/log/morpheus/morpheus-ui/audit.log</file>
  <rollingPolicy class="ch.qos.logback.core.rolling.
↪SizeAndTimeBasedRollingPolicy">
    <fileNamePattern>audit.%d{yyyy-MM-dd}.%i.log</fileNamePattern>
    <maxFileSize>50MB</maxFileSize>
    <maxHistory>30</maxHistory>
  </rollingPolicy>
  <encoder>
    <pattern>[%d] [%thread] %-5level %logger{15} - %maskedMsg %n</pattern>
  </encoder>
</appender>

.. note:: ``maxFileSize`` and ``maxHistory`` values can be updated as needed.
```

3. Add the below logger above or below the other loggers in the `logback.xml` configuration file (make sure it is below, not above, the appender from the previous step or an error will occur):

```
<logger name="com.morpheus.AuditLogService" level="INFO" additivity="false">
  <appender-ref ref="AUDIT" />
</logger>
```

4. Once you have done this, you need to restart the Morpheus Application server:

```
morpheus-ctl stop morpheus-ui
```

Note: Please be aware this will stop the web interface for Morpheus.

5. Once the service has stopped enter the following at the shell prompt to restart (if the service does not stop, replace `stop` with `graceful-kill` and `retry`)

```
morpheus-ctl start morpheus-ui
```

6. To know when the UI is up and running you can run the following command

```
morpheus-ctl tail morpheus-ui
```

Once you see the ASCII art show up you will be able to log back into the User Interface. A new audit file will have been created called `audit.log` and will be found in the default Morpheus log path which is `/var/log/morpheus/morpheus-ui/`

This is only an example and other configurations are possible, such as creating an appender definition for your SIEM audit database product.

morpheus-ssl nginx logs

Note: Morpheus does not put a logrotate in for Morpheus-ssl access logs

svlogd will only rotate the current file, nginx is setup to write the access logs to separate files and not stdout.

Implementation of a log rotate is left up to up to end users for files outside of the services. This is done in case end users have a log management solution.

Below is what a suggested configuration looks like for the file `/etc/logrotate.d/morpheus-nginx`:

```
/var/log/morpheus/nginx/morpheus*access.log {
    daily
    rotate 14
    compress
    delaycompress
    missingok
    notifempty
    create 644 morpheus-app morpheus-app
    postrotate
        [ ! -f /var/run/morpheus/nginx/nginx.pid ] || kill -USR1 `cat /var/run/
↪morpheus/nginx/nginx.pid`
        endscrip
}
```

Apps

App monitors are very useful for seeing an aggregation of failures or impact based on a set of checks and groups. App monitors typically correlate to Apps provisioned from Morpheus Blueprints but can also be manually created and organized. They can be great for visualizing the customer impact a failure might have or even keeping up on a screen in a NOC. To create an App monitor:

Name A friendly name for the new app monitor in Morpheus

Description An optional description value to identify the app monitor

Max Severity The maximum severity incident a failed app may create. This setting overrides check and group max severity settings

Affects Availability When checked, this failed app impacts system-wide availability calculations

App Checks Use the typeahead field to select as many checks as needed to complete the app monitor. Checks are created in Monitoring > Checks and must exist prior to creating the app monitor

Checks

The Monitoring system is composed of individual checks. A check is created for every container or vm that is provisioned through Morpheus . One interesting thing about these checks is they are type aware. There are several different built in check types that are selected based on the service or instance type that is being provisioned. These range from database type checks to web checks and message checks. They are highly configurable and also feature fallback check types for those more generic use cases.

Checks can be customized to run custom queries, check queue sizes, or even adjust severity levels and check intervals. All of these things can be controlled from the Checks sub tab within Monitoring.

Health

A check can have 4 health states. They are Healthy, Error, Warning, and Unknown. When a check test fails the system automatically reattempts the check after 30 seconds to eliminate false positives. This will convert the check into a *Error* state and raise the appropriate severity incident depending on the grouping of the check. When a check recovers it automatically goes into a *Warning* state. This will remain in the warning state until 10 successful check runs have completed.

Options

All check types have several core options and some of these default options can be configured in *Admin > Monitoring*. This includes the default check interval time. By default a check is run every 5 minutes. This can however be changed to run as frequently as once every minute.

- *Max Severity*: The maximum severity level impact for a created incident that can occur if the check fails (defaults to Critical).
- *Check Interval*: The frequency with which a check is run (default 5 minutes).
- *Affects Availability*: Whether or not this check impacts overall system availability calculations.

SSH Tunneling

In many cases when it comes to monitoring databases, and services they may not be fronted on the public ip's for external monitoring. To reach these safely, and securely Morpheus provides an SSH Tunneling mechanism for its check servers. This allows the check to be confirmed via an ssh port tunnel securely using a keypair.

Check Servers

On a base installation of Morpheus a single *check server* is installed on the appliance. This is used for running any custom user checks. This service connects to the provided rabbitmq services and can be moved off or even scaled horizontally onto sets of check servers. All other checks that are related to provisioned containers or VMs are executed by the installed agent on the guest OS or Docker host.

Check types

Web Check

A web check is useful to identify if a url is reachable and the text to match check criteria confirms if the website is loading with the expected values. The text to match character should be within the first few lines of the page source.

Use case:

Adding a check to make sure morpheus demo environment is functioning. The below check will login to the morpheus

Values to be added in Check:

- Name: "<enter name>"
- Type: Web Check
- Interval: 5 mins (Select an interval)

- Max severity: Critical
- Check the box for affects availability
- Web Url: <https://demo.morpheusdata.com/operations/dashboard> (Note: this page will load only if my login is successful. Enter the login details in Username and password fields)
- Request Method: GET
- Basic Authentication: * User: <username> * Password: <password>
- Text to Match: “Morpheus” (Login to the url and on the page of dashboard, right click and select view page source. In the forst few lines, look for a text that you want this check to verify)
- Save Changes

Push API Check

This check can be used to send an API call to morpheus from a platform to check if the push api is working. A push Check is not polled regularly by the standard monitoring system. Instead it is expected that an external API push updates as to the status of the check timed closely with the configured check interval setting. This is used to throttle the push from performing too many status updates.

Note: If a check is not heard from within the check intervals, It’s status will be updated to error and an incident will be raised as if it failed.

Use Case:

Send an API call from an app to make sure the API is not cluttered and can send checks in a 2 mins interval.

Values to be added to the check:

- Name: “<enter name>”
- Type: “Push API Check”
- Interval: 5 mins (Select an interval)
- Max severity: Critical
- Check the box for affects availability
- Copy the curl command are schedule to send this via your API. For testing we used postman to send the api call at an interval of 4 mins.
- Save Changes

MySQL Check

This check is used to run a query on a host running mysql.

Use Case:

Query localhost running mysql to query a table to check if there is any status as requested. If the status has a count of

Values to be added to the check:

- Name: “<enter name>”
- Type: “MySQL Check”
- Interval: 5 mins (Select an interval)
- Check the box for affects availability
- Host: 127.0.0.1
- Port: 3306
- DB Name: morpheus
- User: <db user name>
- Password: <password>
- Query: “select count(*) as count from request_reference where status = ‘requested’;”
- Operator: Equal
- Check results: 1
- Save Changes

Monitoring Groups

Group monitors can only contain checks and can be edited or created in *Monitoring > Groups*. Besides simply adding and removing checks to a group there are a few other useful options that can be customized in a group:

Name A friendly name for the group monitor in Morpheus

Min Checks This specifies the minimum number of checks within the group that must be happy to keep the group from becoming unhealthy

Max Severity The maximum severity incident a failed check may create. This setting overrides a check’s max severity setting

Affects Availability If checked, a failing group monitor impacts system-wide availability calculations

Checks Use the typeahead field to select pre-existing checks for the group monitor. If check(s) need to be created, this can be done in *Monitoring > Checks*

Note: Some useful information can also be seen on the detail page of a check. For example, the average response time of all checks within the group, or an aggregated check history can be viewed.

Incidents

Incident management is very important in any IT Operations environment. The ability to notify the appropriate people of an outage that requires immediate attention is critical to reducing recovery time and even preventing potential customer facing impacts. Because of this, Morpheus provides incident management features as well as external integrations out of the box.

Incidents can be found in the *Monitoring->Incidents* section. When a check fails, an incident is automatically raised. These can vary in severity based on the user configured check severities as well as the group hierarchy (representative of redundancy).

Incidents are also grouped. If an application is impacted and multiple checks fail for that application they automatically get grouped together in one Incident that can fluctuate or escalate in severity as time progresses. These incidents can be muted so as not to affect availability and they can also be resolved manually with an option to detail resolution information.

There are also integrations and API's for integrating with existing corporate workflows when it comes to incident management.

Contacts

To configure user notifications, a contact must first be created in *Monitoring > Contacts*. These contacts can be one of a few types:

- **Contact:** Used for either Email or SMS
- **Web Hook:** Used for posting a notification to a web endpoint or Alert Ops
- **Slack Hook:** Used for posting notifications to a Slack channel (ex. [\)](https://slack.com/{})
- **VictorOps:** Provides a web post format consistent with the required notification format for Victor Ops.

Most of these options provide convenient examples and information when configuring the contact. Once they are configured, contacts can freely be used to build *Alert Rules*.

Alert Rules

Alert Rules provide a powerful means to configure who gets notified in various scenarios. These scenarios include targeting specific checks, groups, or apps, and adding the appropriate recipients to be notified during a situation in which those filters are impacted.

- **Min Duration:** This setting delays notification to the recipients by the entered number of minutes required for the incident to be opened.
- **Min Severity:** Some executives might want to be notified of an outage but only if the severity impact goes above a certain level. This is very useful for scoping escalations.

To add recipients to a rule just start typing their name in the Recipients section towards the button of the edit form. An auto-complete list will start populating with contact names. Once one is selected a delivery method can be selected as well as whether or not they should be notified of any escalation changes and/or closed incidents. The delivery methods available depend on the type of contact information configured for your contact. If needed, contacts can be created or edited in *Monitoring > Contacts*.

Tip: A recipient can be in multiple alert rules and can even be configured to be notified via different methods depending on the rule. A useful example might be to alert someone via email for lower severity incidents but SMS for critical severity levels.

5.8 Tools

5.8.1 Cypher

Overview

Cypher at its core is a secure Key/Value store. But what makes cypher useful is the ability to securely store or generate credentials to connect to your instances. Not only are these credentials encrypted but by using a cypher you don't have to burn in connection credentials between instances into your apps.

Cypher keys can be revoked, either through lease timeouts or manually. So even if somebody were to gain access to your keys you could revoke access to the keys and generate new ones for your applications.

Keys can have different behaviors depending on the specified mountpoint.

Mountpoints

password Generates a secure password of specified character length in the key pattern (or 15) with symbols, numbers, upper case, and lower case letters (i.e. password/15/mypass generates a 15 character password).

tfvars This is a module to store a tfvars file for terraform app blueprints.

secret This is the standard secret module that stores a key/value in encrypted form.

uuid Returns a new UUID by key name when requested and stores the generated UUID by key name for a given lease timeout period.

key Generates a Base 64 encoded AES Key of specified bit length in the key pattern (i.e. key/128/mykey generates a 128-bit key)

vault Configures an integration between Morpheus and a Hashicorp Vault server. See below for additional configuration instructions.

- Key lease times are entered in seconds and default to 32 days (2764800 s).
 - Quick Time Reference:
 - Day: 86400
 - Week: 604800
 - Month (30 days): 2592000
 - Year: 31536000

Creating Cypher Keys

1. Navigate to Tools > Cypher and select + *ADD*
2. Configure one of the following types of Keys:

Password

A Cypher password generates a secure password of specified character length in the key pattern (or 15) with symbols, numbers, upper case, and lower case letters (i.e. password/15/mypass generates a 15 character password).

Key Pattern “password/character_length/key”

Example: password/10/mypassword

Value Leave the Value field blank for a password, as it will be generated.

Lease Enter lease time in seconds (ex. 604800 for one week)

Save changes and the password will be generated and available for use.

If your user role has Cypher: Decrypt permissions, a “DECRYPT” button will be available in the Cypher section to view the generated password.

To delete the password key, select *Actions > Remove* and confirm.

tfvars

A mountpoint to store tfvars files for Terraform App Blueprints.

Key Pattern “tfvars/key”

Example: tfvars/my-aws-account

Value The values for your tfvars file to be encrypted

Lease Enter lease time in seconds (ex. 604800 for one week)

Click *SAVE CHANGES* and the stored values will be available for use.

Note: You may also see Cloud profiles stored at the tfvars mountpoint. They will have a key pattern like: “tfvars/profile/cloud/\$cloudCode/variables”. Terraform Cloud profiles are created on the Cloud detail page (Infrastructure > Clouds > selected Cloud) under the Profiles tab. They allow Terraform apps and specs to be provisioned across multiple Clouds that require differed tfvars. See the [Cloud profiles](#) page for more.

Secret

A Cypher secret is the standard secret module that stores a key/value in encrypted form.

Key Pattern “secret/key”

- EXAMPLE: secret/mysecret

Value Add the secret value to be encrypted

Lease Enter lease time in seconds (ex. 604800 for one week)

Save changes and the secret will be encrypted and available for use.

If your Morpheus user role has Cypher: Decrypt permissions, a “DECRYPT” button will be available in the Cypher section to view the secret.

To delete the secret, select *Actions > Remove* and confirm.

UUID

A Cypher UUID Returns a new UUID by key name when requested and stores the generated UUID by key name for a given lease timeout period.

Key Pattern “uuid/key”

- Example: uuid/myuuid

Value Leave the Value filed blank for UUID, as it will be generated.

Lease Enter lease time in seconds (ex. 604800 for one week)

Save changes and the UUID will be generate and available for use.

If your user role has Cypher: Decrypt permissions, a “DECRYPT” button will be available in the Cypher section to view the generate UUID.

To delete the UUID, select *Actions > Remove* and confirm.

Key

A Cypher Key generates a Base 64 encoded AES Key of specified bit length in the key pattern (i.e. key/128/mykey generates a 128-bit key).

Key Pattern “key/bit_length/key”

- Example: key/256/mykey

Value Leave the Value filed blank for key, as it will be generated.

Lease Enter lease time in seconds (ex. 604800 for one week)

Save changes and the AES Key will be generate and available for use.

If your user role has Cypher: Decrypt permissions, a “DECRYPT” button will be available in the Cypher section to view the generate AES Key.

To delete the UUID, select *Actions > Remove* and confirm.

Vault

Use this mountpoint to store Cypher secrets in a Hashicorp Vault server backend rather than Morpheus. Additionally, you can call secrets stored in Vault from this Cypher mountpoint even if they are only saved there and not listed in the Morpheus Cypher UI. This requires installation and configuration of the Hashicorp Vault plugin. See the YouTube video embedded in this section for more information on adding the plugin, configuration, and a demonstration of its capabilities.

Note: It’s recommended that you use a long-lived token as attempts to call Vault-stored values into Tasks will stop working if the token is no longer good. In such a case you’d have to obtain a new token, delete the Cypher entry with the old token, and create a new one to restore functionality once again. Using a long-lived token will prevent the need to do this often.

Key Pattern “vault/<engineMount>/<secretPath>/data/<key>” (ex. vault/KV2/secret/data/morpheus/lab)

Value Enter your key/value pair here in valid JSON (ex. {“hello”: “world”})

Lease Enter lease time in seconds (ex. 604800 for one week)

Click *SAVE CHANGES*. The example BASH script below onboards the value stored in Vault from the `secret/data/morpheus/lab` mountpoint:

```
from_vault="<%= cypher.read('vault/KV2/secret/data/morpheus/lab') %>"  
echo $from_vault
```

Editing Cypher Keys

Cypher key types which accept user-entered values (not generated values) are editable. To edit, click the “ACTIONS” button at the end of the row for the appropriate Cypher key and then click “Edit.” Edit the values and click *SAVE CHANGES*.

Using Cypher Keys in Scripts

To use a Cypher key in a script, use the following syntax:

```
<%=cypher.read('var_name')%>
```

Example: `PASSWORD=<%=cypher.read('secret/myuserpassword')%>`

Cypher also includes an option to read a value from the `password/*` mountpoint or create one if it doesn't already exist. Use the following syntax:

```
<%=cypher.readPassword('var_name')%>
```

Example: `PASSWORD=<%=cypher.readPassword('myuserpassword')%>``

It should be noted that when Cypher keys are created using the `readPassword` function, the subsequent reads can only come from the same user. If another Morpheus user attempts to run the automation script containing the `readPassword` call, the secret value will not be read and it's very likely the script will fail. For Tasks and Workflows that need to be run by multiple users, use a pre-existing Cypher key and reference it back in the script using `read` rather than `readPassword`.

Note: You can reference the original owner of a workflow so that keys can be used in a subtenant. Example `PASSWORD=<%=cypher.read('secret/myuserpassword')%>` could be changed to `PASSWORD=<%=cypher.read('secret/myuserpassword', true)%>` within a library or a workflow and the `true` means `OWNER true`. This will keep that key in the master tenants cypher store.

5.8.2 Archives

Overview

Archives provides a way to store your files and make them available for download by your scripts and Users. Archives are organized by buckets and can be tied to any existing Bucket or File Share that may be currently integrated (for more on integrating new storage targets, see [storage documentation](#)). Thus, storage buckets in public clouds, on networked storage, or even on the appliance itself may be used to host files.

Archives List Page

To view or create Archives, navigate to Tools > Archives. At the Archives list page is a list of all currently-configured Archives. From the list view, the following details about each Archive are shown:

- **NAME:** The name for the Archive in Morpheus
- **BUCKET:** The integrated bucket or file share where files in this Archive are stored
- **# Files:** The number of files in the Archive
- **SIZE:** The total size of all files in the Archive
- **TENANTS:** When Archive visibility is set to Private, only the Tenants listed here have access to the Archive
- **VISIBILITY:** Public or Private, public Archives are available in all Tenants
- **PUBLIC URL:** Indicates whether Morpheus is automatically generating a public download URL for files in this Archive
- **ACTIONS:** Within the ACTIONS menu users may download a ZIP folder containing all files in the Archive, edit the Archive, or remove it

The screenshot shows the Morpheus web interface. At the top, there's a dark navigation bar with the Morpheus logo, a search bar, a notification bell with a '2' badge, a 'Support' link, and a user profile for 'Alex Harker'. Below this is a light-colored navigation menu with tabs for 'Operations', 'Provisioning', 'Library', 'Infrastructure', 'Backups', 'Monitoring', 'Tools' (which is active), and 'Administration'. Under the 'Tools' tab, there are sub-items: 'Cypher', 'Archives' (selected), 'Image Builder', and 'VDI Pools'. The main content area is titled 'Archives' with a help icon. On the right, there is a '+ ADD' button. Below this is a table with the following data:

NAME	BUCKET	# FILES	SIZE	TENANTS	VISIBILITY	PUBLIC URL	ACTIONS
local-wiki-images	Wiki Images File Share	1	117.7KiB		Private	No	ACTIONS ▾
wiki-image-hosting	Wiki Image Hosting	1	117.7KiB		Private	No	ACTIONS ▾

Adding an Archive

To add a new Archive, click + *ADD* from the Archives list page. Configure the following:

- **NAME:** A friendly name for the Archive in Morpheus
- **DESCRIPTION:** An optional description for the Archive
- **BUCKET:** Select an existing bucket or file share to store files in for this Archive. To integrate a new bucket or file share to use for an Archive, navigate to Infrastructure > Storage
- **VISIBILITY:** Public or Private, public Archives are available in all Tenants
- **TENANTS:** When Archive visibility is set to Private, only the Tenants selected will have access to the Archive
- **PUBLIC URL:** When marked, Morpheus will create a public download URL for all files in the Archive

Warning: Be sure that no sensitive data will be stored in the Archive if it will be configured to generate public URLs. Anyone with the public URL will be able to download the file without authentication.

Once done, click *SAVE CHANGES*

Archive Detail Page

The Archive detail contains information about the Archive configuration as well as a list of files currently stored in the Archive. The Archive detail is accessed by navigating to the Archives list page (Tools > Archives) and selecting an existing Archive. As on the Archives List Page, users can download a ZIP folder containing all files in the Archive and edit the Archive from the ACTIONS menu.

To delete the Archive, click *DELETE*. New files are added by clicking + *ADD*. When adding a new file, users may browse the file system on the local computer to select a file.

From the files list, download or delete individual files by clicking on the appropriate selection from the ACTIONS menu.

The screenshot shows the Morpheus user interface. At the top is a dark navigation bar with the Morpheus logo, a search bar, and user information for Alex Harker. Below this is a light-colored menu with categories: Operations, Provisioning, Library, Infrastructure, Backups, Monitoring, Tools (selected), and Administration. Under 'Tools', there are sub-items: Cypher, Archives (selected), Image Builder, and VDI Pools. The main content area shows the breadcrumb 'Archives > local-wiki-images'. The archive name 'local-wiki-images' is displayed with 'ACTIONS' and 'DELETE' buttons. A 'BUCKET INFO' section provides details: Bucket: Wiki Images File Share, # Files: 1, Size: 117.7KiB, Owner: Morpheus, Tenants: (empty), File Path: morpheus-archives/8e3167b673d5/, Created: 01/25/2022 09:29 AM, Visibility: Private, Last Updated: 01/26/2023 03:10 PM, and Public URL: No. Below this is a 'Files' section with a search bar and a '+ ADD' button. A table lists one file: 'mac-mini.jpg' with type 'image/jpeg' and size '117.7KiB', followed by an 'ACTIONS' button.

File Detail Page

The File Detail Page contains details about the file itself as well as private and public (if available) URLs. In the lower section are three tabs. The Links tab contains any download links which have been generated (both active and expired). The History tab contains historical information about the file including creation and deletion of download links and download events. The scripts tab contains a guide for getting started using Archive-stored files in scripts.

Archives > local-wiki-images > mac-mini.jpg

mac-mini.jpg [DOWNLOAD](#) [DELETE](#)

▼ FILE INFO

Archive Bucket: local-wiki-images	Content Type: image/jpeg	Size: 117.7KiB
Owner: Morpheus	Tenants:	Visibility: Private
File Name: mac-mini.jpg	File Path: mac-mini.jpg	Downloads: 2

▼ DOWNLOAD URLS

Private: <https://morpheus.harkerhomelab.com/tools/archives/download/local-wiki-images/mac-mini.jpg>
 Public: No

Links History Scripts [+ ADD LINK](#)

LINK URL	CREATED	EXPIRES	DOWNLOADS	STATUS	
/public-archives/link?s=979b4514a03940a8	01/26/2023 04:57 PM	01/27/2023 04:57 PM	0	Active	Download
/public-archives/link?s=1b66fb9017e943c7	01/26/2023 03:53 PM	01/26/2023 04:53 PM	0	Expired	

5.8.3 Image Builder

The Morpheus Image Builder tool creates vmdk, qcow2, vhd and raw Images from scratch. The Image Builder creates a blank VM in VMware, attaches an os iso, executes a boot script on the VM at startup via VNC which calls a preseed script which runs the unattended os installation and configuration. Morpheus then executes an ova export of the completed vmdk to target Storage provider, and converts the image to all other specified formats. The new Virtual Image records are automatically added to Morpheus and the Images are then available for use.

Requirements

- **DHCP must be enabled on the network specified for the VM in Morpheus, and network settings configured for DHCP in M**
The blank VM must get network configuration via DHCP upon boot. Static IP assignment is not possible.
- **Hypervisor Console must be enabled on the Target Cloud** Morpheus utilizes VNC to pass the boot script to the VM.
- **VM must be able to reach and resolve the Morpheus appliance url over 443** The boot script calls to the Morpheus appliance to get the preseed script.
- **Valid Linux iso set for the Virtual Image. Windows is not supported.** The iso can exist in the target cloud or uploaded to Morpheus

Note: `cloud-init` enabled must be disabled on the iso Virtual Image settings.

- **Access to target ESXi host(s) over 443 and ESXi hostname dns resolution**

– Same requirement as Hypervisor Console and Image upload/download to/from vCenter.

- Valid Boot Script
- Valid Pre-seed script
- Valid Storage Provider configure for ova export of generated image.

Sample Scripts

Sample Boot Script

```
<wait5><tab> text ks=<%=preseedUrl%><enter>
```

Note: <%=preseedUrl%> is a Morpheus variable that will populate with the Morpheus appliance url.

Sample Preseed Script

```
# CentOS 7.x kickstart file - ks.cfg
#
# For more information on kickstart syntax and commands, refer to the
# CentOS Installation Guide:
# https://access.redhat.com/documentation/en-US/Red\_Hat\_Enterprise\_Linux/7/html/Installation\_Guide/sect-kickstart-syntax.html
#
# For testing, you can fire up a local http server temporarily.
# cd to the directory where this ks.cfg file resides and run the following:
#   $ python -m SimpleHTTPServer
# You don't have to restart the server every time you make changes. Python
# will reload the file from disk every time. As long as you save your changes
# they will be reflected in the next HTTP download. Then to test with
# a PXE boot server, enter the following on the PXE boot prompt:
#   > linux text ks=http://<your_ip>:8000/ks.cfg

# Required settings
lang en_US.UTF-8
keyboard us
rootpw password
authconfig --enableshadow --enablemd5
timezone UTC

# Optional settings
install
cdrom
user --name=cloud-user --plaintext --password password
unsupported_hardware
network --bootproto=dhcp
firewall --disabled
selinux --permissive
bootloader --location=mbr --append="biosdevname=0 net.ifnames=0"
```

(continues on next page)

(continued from previous page)

```
text
skipx
zerombr
clearpart --all --initlabel
autopart --type=plain
firstboot --disabled
reboot

%packages --nobase --ignoremissing --excludedocs
openssh-clients
# Prerequisites for installing VMware Tools guest additions.
# Put in kickstart to ensure first version installed is from install disk,
# not latest from a mirror.
kernel-headers
kernel-devel
gcc
make
perl
curl
wget
bzip2
dkms
patch
net-tools
git
# Core selinux dependencies installed on 7.x, no need to specify
# Other stuff
sudo
nfs-utils
open-vm-tools
-fprintd-pam
-intltool
-biosdevname

# unnecessary firmware
-aic94xx-firmware
-atmel-firmware
-b43-openfwfw
-bfa-firmware
-ipw*-firmware
-irqbalance
-ivtv-firmware
-iwl*-firmware
-libertas-usb8388-firmware
-ql*-firmware
-rt61pci-firmware
-rt73usb-firmware
-xorg-x11-drv-ati-firmware
-zd1211-firmware
%end

%post
```

(continues on next page)

```
# configure vagrant user in sudoers
echo "%cloud-user ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers.d/cloud-user
chmod 0440 /etc/sudoers.d/cloud-user
cp /etc/sudoers /etc/sudoers.orig
sed -i "s/^\(.*requiretty\)$/#\1/" /etc/sudoers
# keep proxy settings through sudo
echo 'Defaults env_keep += "HTTP_PROXY HTTPS_PROXY FTP_PROXY RSYNC_PROXY NO_PROXY"' >> /
→etc/sudoers
%end
```

5.8.4 VDI Pools

The VDI Pools section of Morpheus Tools provides a management area for defining VDI Pools and VDI Apps that a user can consume within the [Virtual Desktop Persona](#).

Pools can be either persistent or non-persistent and have various controls pertaining to idle pools and minimum or maximum sizes. The idea here is to make sure a server is always quickly available to accommodate user demand.

Morpheus leverages its Instance Types concept for provisioning servers within the VDI Pool. All the options available during Instance provisioning is available for setting the base server configuration. This includes Workflows, domain joins, tagging, image selection and more.

A timeout setting can also be applied to release pool allocations from a user once they have disconnected their session. For non-persistent pools, a good recommendation is ten minutes whereas, for a persistent pool, a sensible recommendation would be around one hour.

Pool behavior changes depending on the pool type. In a non-persistent pool, when a timeout period expires, the VM is destroyed and a new one is allocated for use. This functionality will change based on the cloud technology in a future update allowing for potential recycling of the VMs. In a Persistent pool, when the lease timeout expires, the Instance will shutdown until the user requests it again in the future. It is important to note that lease timeouts auto-extend for as long as the user is logged into or browsing any area of the Morpheus application. Once the user closes their browser or logs out of their session, the timeouts will no longer auto-extend.

Configuring Access to VDI Pools

Access to the Virtual Desktop Persona and individual VDI pools is handled through the user Role and, where applicable, Tenant Role. When creating a new Role, access is restricted to the Virtual Desktop Persona and all VDI pools by default. To grant access:

1. Navigate to the Role (Administration > Roles > Selected Role)
2. Access the Personas tab
3. Toggle the Virtual Desktop permission to “Full” or “None”
4. Access the VDI Pool Access tab
5. Toggle access to selected VDI pools to “Full” or “None”, you can also toggle permission on all pools to “Full” or “None” with the Global Access selection
6. Role changes are saved automatically, there is no need to manually save

Additionally, users should have a Linux and/or Windows username and password configured in their user profiles in order for virtual desktop login to be as seamless as possible. User profiles are accessed by clicking on the user’s name in the upper-right corner of the application windows and clicking USER SETTINGS. The section to enter Windows and Linux account credentials is in the right column of the page.

Creating VDI Templates

Note: The following guide focuses on VMware and Windows but is applicable to most cloud environments. Morpheus also supports Linux virtual desktops.

1. Create a thin-provisioned VM in the VMware vCenter console. It's recommended you allocate at least 50 GB of storage, at least 2 vCPU and at least 8 GB RAM on the template. Smaller VMs can be deployed from this template later.
2. Install a Windows operating system, there is no need to supply a license during deployment.
3. Supply the initial username and password
4. Install any updates, applications or optimizations. See the next section for recommended optimizations for the most performant virtual desktop experience
5. Shutdown the VM and convert to a template. Optionally, you can also use the Linked Clones (VMware) process which is described in a later section

Suggested Optimizations

Reducing display and input delays is key to providing the best virtual desktop experience for the user. Consider the following optimizations for VDI desktops and servers:

- Disable desktop wallpapers
- Implement Roaming Profiles
- Enforce WDDM remote display driver
- Re-enable local Administrator
- Delete the initial created user profile
- Clean up any unneeded installer packages

Additionally, there are a number of OS optimization tools available on the Internet which are specific to VDI use cases.

Linked Clones

Linked Clones are a feature of VMware which references snapshots of a VM to deploy from. This adds the advantage of quicker clone times and the ability to more easily share small modifications to a file system. Morpheus supports Linked Clones but recommends them for VDI workloads only.

Note: Linked Clones are not templates but rather powered down VMs.

1. Locate the VM you desire to have the Linked Clone in Morpheus. If it's not currently managed by Morpheus, navigate to the appropriate Cloud (Infrastructure > Clouds), find the VM on the "VMs" tab, and click "Convert to Managed" from the ACTIONS menu
2. In the CONVERT TO INSTANCE modal, select "No Agent Install" in the AGENT field
3. If snapshots are already on the VM, these will now be synced by Morpheus. If you have not yet created a snapshot, do so in the vCenter console (and refresh the Cloud integration in Morpheus afterward) or from the ACTIONS menu in Morpheus itself. Be sure to take a snapshot of a powered-off VM and give the snapshot a name that will be identifiable for administrators

4. From the Instance detail page in Morpheus, navigate to the Backups tab to find the snapshot
5. Select “More” and create the Linked Clone
6. The Linked Clone will now appear in the Morpheus Virtual Image repository (Library > Virtual Images), ready to use with your custom Layouts

Note: You should modify the Virtual Image to “Force Guest Customization” unless you `sysprep` your VM at shutdown time

Creating or Editing a VDI Pool

VDI pools are configured from the Tools menu (VDI Pools selection). The following information is displayed in the VDI pools list view, bear in mind some fields may be hidden depending on how you’ve configured your VDI pools list view (gear icon):

- **TYPE:** An icon indicating the machine type associated with the pool. Morpheus includes many logos out of the box and also allows users to set their own custom icons
- **NAME:** The friendly name given to the VDI pool
- **PERSISTENT:** A check mark will appear when the VDI pool is configured for persistent virtual desktops
- **ENABLED:** A check mark will appear when the VDI pool is enabled and visible to users whose Role permissions allow them access
- **POOL USAGE:** A graph representing the usage of the VDI pool. The total length of the bar represents the maximum pool size based on the configuration. Green segments represent available virtual desktops, blue segments represent reserved virtual desktops, yellow segments represent virtual desktops which are being prepared, and gray segments represent additional pool capacity which could be made available depending on how many virtual desktops are currently reserved and how many idle machines you’ve configured the pool to keep available
- **DESCRIPTION:** A description of the virtual desktop type, if provided

VDI POOLS

TYPE	NAME	PERSISTENT	ENABLED	POOL USAGE	
	Suse	✓	✓		
	Windows		✓		

Create a VDI pool by selecting **+ ADD** from the VDI Pools tab or edit an existing one by clicking on the pencil icon from the appropriate row. Configure the following, fields containing a vertical blue bar along the left edge are required:

- **NAME:** A friendly name for the VDI pool in Morpheus
- **DESCRIPTION:** A description of the virtual desktop type
- **MIN IDLE:** The minimum number of virtual desktops that should remain idle and ready
- **INITIAL POOL SIZE:** The number of virtual desktops that will be prepared when the pool is created or enabled

- **MAX IDLE:** The maximum number of virtual desktops that remain idle and ready. Machines will be shut down as necessary when this number is exceeded due to users vacating their machines
 - **MAX SIZE:** The total number of virtual desktops this pool can have. Additional users will not be able to access machines once this number is reached
 - **LEASE TIMEOUT (MINUTES):** The user lease time on a virtual desktop they've reserved. The lease will continue to auto-renew itself as long as the user is logged into Morpheus. Once the user has logged out and the lease timeout period has expired, the machine will be released as appropriate based on your configuration
 - **PERSISTENT:** Pools with persistent virtual desktops will reserve a machine for each user in order to preserve settings, installed applications, work files and more. Machines in persistent pools will be shut down rather than destroyed when they are no longer in use
 - **RECYCLABLE:** When enabled, the VDI Instance will revert back to a snapshot and become available once again after the user has logged out and the VDI session has expired. This behavior will not apply to VDI pools which are also configured to be persistent because in that configuration the Instance is merely stopped and saved for the user's next session. This feature is currently only available for Cloud types which support snapshot management (VMware, Nutanix, and vCD)
 - **ALLOW COPY** Enables or disables the ability for the VDI user to copy contents from the VDI instance to the local clipboard
 - **ALLOW PRINTER** When enabled, users local system printers can be targeted from the VDI Instance
 - **ALLOW HYPERVISOR CONSOLE:** When checked, native cloud console will be enabled (if available) rather than using Morpheus-native RDP/SSH capability
 - **AUTO CREATE LOCAL USER UPON RESERVATION:** When marked, the user configured in Morpheus user settings will be created when the machine is initially accessed. If unchecked or if there is no user configured in Morpheus user settings, ensure the machine is joining a domain or there is a known user on the machine image in order to allow access
 - **ENABLED:** When marked, the initial pool size will begin to deploy once the VDI pool is saved. The icon for this desktop environment will also be presented to Virtual Desktop Persona users
 - **CONFIGURE:** Click this button to configure the deployment configuration each system will use. The wizard is identical to the Instance provisioning wizard meaning all available Instance Types, Workflows, and more are available to virtual desktop machine creation. Consult the steps above to see an example VDI image prep walkthrough
 - **LOGO:** Upload or select a logo to represent the virtual desktop type to users
 - **VDI APPS:** Optionally select one or more frequently-used applications the user can launch directly. Users will also have the option to launch into the desktop
 - **VDI GATEWAY** Select a configure VDI Gateway for VDI sessions to be redirected to. VDI sessions will be redirected to the gateway when a gateway is specified.
- Guest Console SSH Tunnel (optional)** A Jump Host can be configured for VDI session connections. Morpheus will tunnel through the Jump Host when connecting Guest Console sessions for VDI. This is not applicable for Hypervisor Console connections.
- **GUEST CONSOLE JUMP HOST** Jump Host IP address or hostname used to connect to the Jump Host for Guest Console sessions to VDI Instances
 - **GUEST CONSOLE JUMP USERNAME** Jump Host Username used to connect to the Jump Host for Guest Console sessions to VDI Instances
 - **GUEST CONSOLE JUMP PORT** Jump Host Port used to connect to the Jump Host for Guest Console sessions to VDI Instances

- **GUEST CONSOLE JUMP PASSWORD** Jump Host Password used to connect to the Jump Host for Guest Console sessions to VDI Instances (optional if key specified)
- **GUEST CONSOLE KEYPAIR** Jump Host SSH Key used to connect to the Jump Host for Guest Console sessions to VDI Instances (optional if password specified)

Note: A Guest Console Keypair included here must be a local keypair, not a synced keypair.

EDIT VDI POOL

NAME: Windows

DESCRIPTION: Windows 10 Desktop

MIN IDLE: 1
Sets the minimum number of idle instances on standby in the pool. The pool will always try to maintain this number of available instances on standby.

INITIAL POOL SIZE: 1
The initial size of the pool to be allocated on creation.

MAX IDLE: 2
Sets the maximum number of idle instances on standby in the pool. If the number of idle instances supersedes this, the pool will start removing instances.

MAX SIZE: 2
Max limit on number of allocations and instances within the pool.

LEASETIMEOUT (MINUTES): 60

Creating or Editing a VDI Apps

VDI Apps allow users to launch directly into commonly-used apps rather than the OS desktop. Currently, VDI Apps only work with RDP Windows Instances, taking advantage of native Windows Remote Application functionality. Natively-hosted remote desktop applications can only be presented from Windows 10 Enterprise and Education. Other versions of Windows 10 can present remote applications using the procedure below:

1. Open the Windows Registry Editor
2. Locate the following entry: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Terminal Server\TSAppAllowList`
3. Navigate to `fDisabledAllowList` and set its value to “1” in the right-hand pane
4. Add a new key under `TSAppAllowList` and name it “Applications”
5. Add a new key under “Applications” using any name you’d like
6. Within this new key, create two new string values, one called “Name” and one called “Path”
7. The string value for “Name” should describe the application (ex. “Notepad”)
8. The string value for “Path” should be the absolute path to the executable for that application (ex. “C:\Windows\System32\notepad.exe”)

VDI Apps are created by selecting + *ADD* from the VDI Apps tab or edit an existing one by clicking on the pencil icon from the appropriate row. Configure the following, fields containing a vertical blue bar along the left edge are required:

- **NAME:** A friendly name for the VDI App in Morpheus
- **DESCRIPTION:** A description of the virtual app type

- **LAUNCH PREFIX:** A reference to the remote app registry prepended with two pipes (||). For example, we might create a registry “Chrome” for a Chrome browser VDI App and the associated launch prefix would be “||Chrome”
- **LOGO:** Upload or select a logo to represent the virtual app type to users

VDI Gateways

The Morpheus Worker is a light weight distributed worker daemon as well as a scalable VDI Gateway. Currently, the features center around VDI Gateway but will expand to support full plugin workloads as well as agent relay capabilities.

Adding VDI Gateways to Morpheus

VDI Gateways can be linked to a Morpheus appliance and then used in VDI Pool configurations. VDI sessions will be redirected to configured gateways instead of the Morpheus appliance when a VDI Gateway is specified for a VDI Pool.

Note: A VDI Gateway is a separate VM or container Instance used to route users to VDI Instances. The Morpheus VDI Gateway section is for configuring a connection to a VDI Gateway, not creating the gateway Instance itself.

- **NAME** Specify a name for the VDI Gateway in Morpheus. Note that the VDI Gateway Name is not used when connecting to the gateway
- **DESCRIPTION** Specify a description for the VDI Gateway in Morpheus. (optional)
- **GATEWAY URL** The url of the VDI Gateway. This url is used to connect to the gateway, and should match the the worker url of the VDI Gateway.

Upon creation, the VDI Gateway record will produce an API KEY. This API KEY needs to be specified in the `morpheus-worker.rb` file on the API Gateway itself under `worker['apikey'] = '$API_KEY'`. Once the gateway object is created you will need to configure it as the default gateway in Morpheus global settings (Administration > Settings > Appliance). Scroll down to the “Default Console Gateway” setting and select the gateway object you’ve just created. Continue on to the next section to actually install the gateway and configure it with your API key.

VDI Gateway VM Install

A VDI Gateway VM is installed and configured similarly to a Morpheus appliance via `rpm` or `deb` package.

Note: VDI Gateway Package URLs are available at <https://morpheushub.com> in the downloads section.

Requirements

Table 19: **Supported VDI Gateway Operating Systems**

OS	Version(s)
Amazon Linux	2
CentOS	7.x, 8.x
Debian	9, 10, 11
RHEL	7.x, 8.x
SUSE, SLES	12
Ubuntu	16.04, 18.04, 20.04

- **Memory:** 4 GB RAM minimum recommended for default installations supporting up to 20 concurrent sessions. Add 50 MB RAM per additional concurrent session
 - **Storage:** 10 GB storage minimum recommended. Storage is required for VDI Gateway Packages and log files
 - **CPU:** 4-core minimum recommended
 - Network connectivity to and from Morpheus appliance and from users to the VDI Gateway over TCP 443 (HTTPS)
 - Superuser privileges via the `sudo` command for the user installing the Morpheus VDI Gateway package
 - Access to base `yum` or `apt` repos. Access to Optional RPM repos may be required for RPM distros
1. Download the target distro & version package for installation in a directory of your choosing. The package can be removed after successful installation.

```
wget https://downloads.morpheusdata.com/path/to/morpheus-worker-$version.distro
```

2. Validate the package checksum matches source checksums. For example:

```
sha256sum morpheus-worker-$version.distro
```

3. Next install the package using your selected distribution's package installation command and your preferred opts. Example, for RPM:

rpm:

```
sudo rpm -ihv morpheus-worker-$version.$distro

Preparing...                               ##### [100%]
Updating / installing...
 1:morpheus-worker-5.3.1-1.$distro        ##### [100%]
Thank you for installing Morpheus Worker!
Configure and start the Worker by running the following command:

sudo morpheus-worker-ctl reconfigure
```

4. Configure the gateway by editing `/etc/morpheus/morpheus-worker.rb` and updating the following:

```
worker_url 'https://gateway_worker_url' # This is the gateway URL the |morpheus|
↪appliance can resolve and reach on 443
worker['appliance_url'] = 'https://morpheus_appliance_url' # The resolvable URL or
↪IP address of |morpheus| appliance which the gateway can reach on port 443
worker['apikey'] = 'API KEY FOR THIS GATEWAY' # VDI Gateway API Key generated from
↪|morpheus| Appliance VDI Pools > VDI Gateways configuraiton
```

Note: By default the `worker_url` uses the machine's hostname, ie `https://your_machine_name`. The default `worker_url` value can be changed by editing `/etc/morpheus/morpheus-worker.rb` and changing the value of `worker_url`. Additional appliance configuration options are available below.

5. After all configuration options have been set, run `sudo morpheus-worker-ctl reconfigure` to install and configure the worker, nginx and guacd services:

```
sudo morpheus-worker-ctl reconfigure
```

The worker reconfigure process will install and configure the worker, nginx and guacd services and dependencies.

Note: Configuration options can be updated after the initial reconfigure by editing `/etc/morpheus/morpheus-worker.rb` and running `sudo morpheus-worker-ctl reconfigure` again.

- Once the installation is complete the morpheus worker service will automatically start and open a web socket with the specified Morpheus appliance. To monitor the startup process, run `morpheus-worker-ctl tail` to tail the logs of the worker, nginx and guacd services. Individual services can be tailed by specifying the service, for example `morpheus-worker-ctl tail worker`

VDI Gateway Docker Install

To Use VDI Gateway within a Docker container, a few pieces of information are needed.

Firstly, in Morpheus, go to **Tools > VDI Pools > VDI Gateways** and create a new VDI Gateway Record. Be sure to set the HTTPS URL as Morpheus will need to be able to redirect the user's browser to that page. An API Key will be generated. Make note of this as you will need it later.

Now Simply run with:

```
docker run -d -p 8443:8443 -e MORPHEUS_SELF_SIGNED=true -e MORPHEUS_KEY=[apiKey] -e MORPHEUS_URL=https://my.morpheusAppliance.url morpheusdata/morpheus-worker:latest`
```

This will setup an HTTPS self-signed exposed port on 8443 for the vdi gateway. It is highly recommended to use valid certificates on your VDI Gateways. It could be terminated at the VIP or a p12 SSL File can be used and configured for the container.

If the docker entrypoint detects a file at `/etc/certs/cert.p12`, SSL Will be enabled on port 8443 instead. be sure to set environment variables `MORPHEUS_SSL_ALIAS` and `MORPHEUS_SSL_PASSWORD` when using p12 files.

If you wish to run in HTTP mode and SSL terminate at the VIP, you can run the container like so:

```
docker run -d -p 8080:8080 -e MORPHEUS_SELF_SIGNED=true -e MORPHEUS_KEY=[apiKey] -e MORPHEUS_URL=https://my.morpheus.url morpheusdata/morpheus-worker:latest
```

VDI Gateway Helm Chart Installation

First, configure the Helm repository:

```
helm repo add morpheusdata https://gomorpheus.github.io/helm-charts-morpheus/
```

Next, install the Morpheus worker using `helm install`. You can specify each parameter using `--set key=value[, key=value]` arguments as in the following example:

```
helm install morpheus-worker --set replicaCount="1" morpheusdata/morpheus-worker
```

Alternatively, you can create a values YAML file and pass an argument as in the following example:

```
helm install -f values.yaml morpheus-worker morpheusdata/morpheus-worker
```

Upgrading the workers node(s) is as simple as refreshing the repo and using `helm upgrade`:

```
helm repo update
helm upgrade -f values.yaml morpheus-worker morpheusdata/morpheus-worker
```

To uninstall, use one of the following:

```
helm uninstall morpheus-worker
```

or

```
helm delete morpheus-worker --purge
```

Note: `helm delete` removes all the Kubernetes components associated with the chart and deletes the release.

The following table lists the configurable parameters of the Sentry chart and their default values:

Parameter	Description	Default
<code>image.repository</code>	Image repository	morpheusdata/morpheus-worker
<code>image.tag</code>	Image tag. Possible values listed here.	5.3.1-4
<code>image.pullPolicy</code>	Image pull policy	IfNotPresent
<code>env.MORPHEUS_KEY</code>	API Key for Morpheus Worker	
<code>env.MORPHEUS_URL</code>	Morpheus FQDN with protocol	
<code>env.MORPHEUS_SELF_SIGNED</code>	Is Morpheus using a Self Signed Certificate	false
<code>service.type</code>	Kubernetes service type for the GUI	ClusterIP
<code>service.port</code>	Kubernetes port where the GUI is exposed	8989
<code>livenessProbe.initialDelaySeconds</code>	Initial delay (seconds) for liveness monitoring	5
<code>livenessProbe.timeoutSeconds</code>	Timeout (seconds) before health check considered unhealthy	5
<code>livenessProbe.periodSeconds</code>	Poll interval (seconds) between health checks	10
<code>livenessProbe.failureThreshold</code>	Number of failed polls before restarting service	3
<code>replicaCount</code>	Number of Replicas if AutoScaling False	1
<code>autoscaling.enabled</code>	Enable AutoScaling	false
<code>autoscaling.minReplicas</code>	Minimum number of Replicas	1
<code>autoscaling.maxReplicas</code>	Maximum number of Replicas	100
<code>autoscaling.targetCPUUtilizationPercentage</code>	CPU Threshold for AutoScaling	80
<code>autoscaling.targetMemoryUtilizationPercentage</code>	Memory Threshold for AutoScaling	
<code>ingress.enabled</code>	Enables Ingress	false
<code>ingress.annotations</code>	Ingress annotations	{}
<code>ingress.path</code>	Ingress path	/
<code>ingress.hosts</code>	Ingress accepted hostnames	chart-example.local
<code>ingress.tls</code>	Ingress TLS configuration	[]
<code>resources</code>	CPU/Memory resource requests/limits	{}
<code>nodeSelector</code>	Node labels for pod assignment	{}
<code>tolerations</code>	Toleration labels for pod assignment	[]
<code>affinity</code>	Affinity settings for pod assignment	{}

5.9 Administration

There are several administrative integrations built into Morpheus that make it great to work with within any organization ranging from small to large. Especially, with its built in white label support and multitenancy capabilities, managed service providers have a wide range of capabilities when it comes to managing customer accounts and users.

5.9.1 Tenants

Overview

A Tenant in Morpheus is an isolated environment with unique users and workloads. The Master Tenant is the default Tenant in Morpheus, created upon installation. All other Tenants outside of the Master Tenants are Subtenants.

- The Master Tenant is the default Tenant created during the installation of Morpheus
- All Tenants created after installation are Subtenants. Only one Master Tenant can exist
- The Master Tenant creates and controls all Subtenants.
- Tenants are isolated environments with unique users, workloads, and Groups
- The Master Tenant can share or assign its resources to Subtenants
- Subtenants cannot share their resources with other Tenants
- Subtenants cannot see resources from other Subtenants
- Subtenants can only access Master Tenant resources that have been set to Public visibility or specifically assigned to the Subtenant

Roles

There are two Role types in Morpheus, Tenant Roles and User Roles. Understanding these Role types is key to effectively administering Role permissions in Morpheus. These two Role types are discussed in greater detail in this section.

Tenant Roles

Tenant Roles set the maximum permission levels for Users in the Tenant. User Role permissions will not exceed the permissions of the Tenant Role.

- Tenant Roles set the maximum permissions for a Tenant
- User Roles in a Tenant cannot exceed the permissions of the Tenant Role
- A Tenant Role can be assigned to one or multiple Tenants
- Tenant Roles determine Cloud access for the Subtenant such that all Clouds in the Master Tenant which have visibility set to Public will show as options in the Tenant Role Cloud Access tab
- Only Master Tenant Clouds given access in the Tenant Role will be accessible in the Subtenant

Important: Tenant Roles cap permissions on all Subtenant User Roles. User Roles can be created in the Subtenant with lesser permissions than the Tenant Role allows. Tenant Roles are designed for a Master Tenant Admin to set max permissions for the Subtenant, and a Subtenant Admin to configure User Roles inside the Subtenant.

User Roles

User Roles determine feature, Group, and Instance Type access for all Users. In a multi-Tenant environment, there are two types of User Roles: Single-Tenant User Roles and Multi-Tenant User Roles.

- **Single-Tenant User Roles:** These exist solely in the Tenant they are created in. All Roles created in a Subtenant will be Single-Tenant User Roles
- **Multi-Tenant User Roles:** The Master Tenant can create Multi-Tenant User Roles. These Roles are automatically seeded into Subtenants and can be assigned to Subtenant Users. Changes to Multi-Tenant User Roles made in the Master Tenant are propagated to all Subtenants. However, once a Multi-Tenant User Role is edited inside a Subtenant, it is no longer linked to the Multi-Tenant User Role and becomes its own unique Role. It will no longer receive propagated changes.

Note: Multi-Tenant User Roles are intended to make Subtenant User Role creation easier, so Master Tenant Users do not have to re-create the same base Subtenant Users Roles for every Subtenant. Multi-Tenant User Roles are not a single Role across Tenants, but more like a template that creates new Subtenant User Roles that can then be managed in the Sub Tenant.

Tenants

The Tenants page displays a list of all Tenants. This page enables users to Create, Edit, and Delete Tenants. The list of Tenants displays the Tenant Name, Role, Total Instances, Total Users, and the Created Date.

Click the Tenant Name to drill into the Tenant View where you can again Edit or Delete the Tenant, as well as Create Users, Edit Users, and Delete Users users belonging to the Tenant.

Create Tenants

To Create Tenants:

1. Select the Administration link in the navigation bar.
2. Select the Tenants link in the sub navigation bar.
3. Click the Create Tenant button.
4. From the New Tenant wizard input:
 - Name
 - Description (optional)
 - Subdomain
 - Base Role
 - Currency
5. Within the Advanced Options section, track customer data related to the Tenant if needed:
 - Account Number
 - Account Name
 - Customer Number
6. Click the *Save Changes*

Edit Tenant

To edit a Tenant:

1. Select the Administration link in the navigation bar.
2. Select the Tenants link in the sub navigation bar.
3. Click the Edit pencil icon on the row of the Tenant to edit.
4. Edit the Edit Tenant settings.

Disabling Tenant

When disabling a tenant, they are not able to login and cannot be impersonated by another tenant. However all of their information will still remain in Morpheus and they may still receive notifications and alerts.

To disable a Tenant:

1. Select the Administration link in the navigation bar.
2. Select the Tenants link in the sub navigation bar.
3. Click the Edit pencil icon on the row of the Tenant to edit.
4. Uncheck the Enabled box.

Delete Tenant

To delete a Tenant:

1. Select the Administration link in the navigation bar.
2. Select the Tenants link in the sub navigation bar.
3. Click the Delete trashcan icon on the row of the Tenant to delete.
4. Confirm

Tenant Users

The Tenant View displays a list of users belonging to the Tenant and their Name, Username, Email, and Role.

From this page: Create, Edit, and Delete users within the Tenant.

Important: In versions 3.1.1 and 2.12.5 and later, a Multi-Tenant User Role must be created prior to adding Subtenant Users or the User will not save. In previous versions a default Multi-Tenant Role was seeded. Due to customer requests, the seeded role was removed and a Multi-Tenant Role must be created by the Master Tenant for Subtenant Users.

Create Tenant User

To create a Tenant User:

1. Select the Administration link in the navigation bar.
2. Select the Tenants link in the sub navigation bar.
3. Click the Tenant Name on the row of the Tenant where the user will be added.
4. Click + *ADD USER*
5. From the New User wizard, input the fields below:
 - First Name
 - Last Name
 - Username
 - Email address
 - Role (to be inherited by the user)
 - Password
 - Any default Windows or Linux credentials

Click *SAVE CHANGES*

Important: In versions 3.1.1 and 2.12.5 and later, a Multi-Tenant User Role must be created prior to adding Subtenant Users or the User will not save. In previous versions a default Multi-Tenant Role was seeded. Due to customer requests, the seeded role was removed and a Multi-Tenant Role must be created by the Master Tenant for Subtenant Users.

Edit a Tenant User

To edit a User:

1. Select the Administration link in the navigation bar.
2. Select the Tenants link in the sub navigation bar.
3. Click the specific Tenant name from the row of available Tenants.
4. Click the Edit pencil icon for your selected Tenant.
5. Edit User information

Note: Name, Username, Passwords and e-mail addresses cannot be edited on Users created from Identity Source Integrations.

Click *SAVE CHANGES*

Delete Tenant User

To delete a Tenant User:

1. Select the Administration link in the navigation bar.
2. Select the Tenants link in the sub navigation bar.
3. Click the Tenant Name from the row for the Tenant containing the user.
4. Click the Delete trashcan icon of the row of the user to delete.
5. Confirm

Subtenant User Login

Subtenant Users can have the same Username as the User on the Master Tenant or any other Tenant. Subtenant Users will now have to login using the subdomain prefix.

Important: Subtenant users will no longer be able to login from the main login page without specifying their subdomain.

Example: I have a username subuser that belongs to a tenant with the subdomain subaccount. When logging in from the main login url, I would now need to enter in: subaccount\subuser

Configuring Tenants and Resources for Multi-Tenancy

A very common scenario for Managed Service Providers is the need to provide access to resources on a customer by customer basis. Several administrative features are available in Morpheus to ensure customer resources are properly scoped and isolated. With its built multi-tenancy capabilities and white label support, managed service providers have a wide range of capabilities when it comes to managing customer Tenants and users.

Tenants

There are essentially two types of Tenants in Morpheus

- Master Tenant
- Sub Tenants

During the initial setup of a Morpheus Appliance, the Master Tenant is created. All Tenants created in addition to this Master Tenant are sub-Tenants. There can only be one Master Tenant, and sub-Tenants cannot become the Master Tenant. The delineation between the Master Tenant and sub-Tenants is important to understand for properly scoping resources across Tenants.

Creating Tenants

The Master Tenant is created during the initial appliance setup. Additional sub-Tenants can be created in the Administration > Tenants section.

The Tenants page displays a list of all Tenants. This page enables users to: Create, Edit, and Delete Tenants. The list of Tenants displays the Tenant Name, Role, Total Instances, Total Users, Status (active or inactive) and the Created Date. Click the Tenant Name to drill into the Tenant View where you can edit or delete the Tenant, as well as create, edit and delete users belonging to the Tenant.

Note: At least one Tenant in addition to the Master Tenant is required to scope resources across Tenants.

To create a new sub-Tenant

1. Select the Administration link in the navigation bar.
2. Select the Tenants link in the sub navigation bar.
3. Click the *+Create Tenant* button.
4. From the New Tenant wizard input * Name (Required) * Description * Base Role * Currency (for pricing)

The Base Role defines a role set from which all roles created within the Tenant will inherit.

Note: In prior versions, we could set Limits when creating a Subtenant. These could restrict the amount of storage, memory, and CPUs that can be collectively provisioned by all users in the Tenant. In more recent versions, this functionality has been rolled into Policies (Administration > Policies). When creating a Policy, we are able to specify a Tenant to which the Policy should apply.

Click the *Save Changes* button.

NEW TENANT

NAME

DESCRIPTION

ENABLED

SUBDOMAIN
The SubDomain is used for creating a direct login url in Identity Sources or as a login prefix to identify the tenant i.e. 'subdomain\username'

BASE ROLE
This is the primary role of the tenant. All roles created within the tenant must inherit this role as the base role.

CURRENCY

[SAVE CHANGES](#)

Viewing Tenants

To View an individual Tenant page, select the Tenant name from the main Tenants section.

The screenshot shows the Morpheus Administration interface. At the top, there is a navigation bar with the Morpheus logo and the word 'MORPHEUS'. Below the navigation bar, there are several menu items: Operations, Provisioning, Library, Infrastructure, Backups, Monitoring, Tools, and Administration (which is highlighted). Under Administration, there are sub-menu items: Tenants, Plans & Pricing, Roles, Users, Integrations, Policies, Health, and Settings. The main content area shows the 'Tenants > A Tenant' breadcrumb trail. Below this, there is a section titled 'A Tenant' with a search bar and a '+ CREATE USER' button. To the right of the search bar, there are buttons for 'IDENTITY SOURCES', 'ACTIONS', 'EDIT', and 'DELETE'. Below the search bar, there is a table with columns for 'DISPLAY NAME', 'USERNAME', 'EMAIL', and 'ROLE'. The table contains two rows of data. The first row shows a user with the role 'Morpheus Admin, Sub Account Admin, Subtenant Multirole'. The second row shows a user with the role 'Sub Account Admin'. Each row has a 'MORE' button with a dropdown arrow.

From inside the Tenant view, we can edit or delete the Tenant, as well as click into any of the Tenant’s users.

Tenant Users

To create a new user within the Tenant:

Click the *CREATE USER* button, then from the New User wizard input the fields below:

- First Name
- Last Name
- Username
- Email
- Role
- Password
- Confirm Password

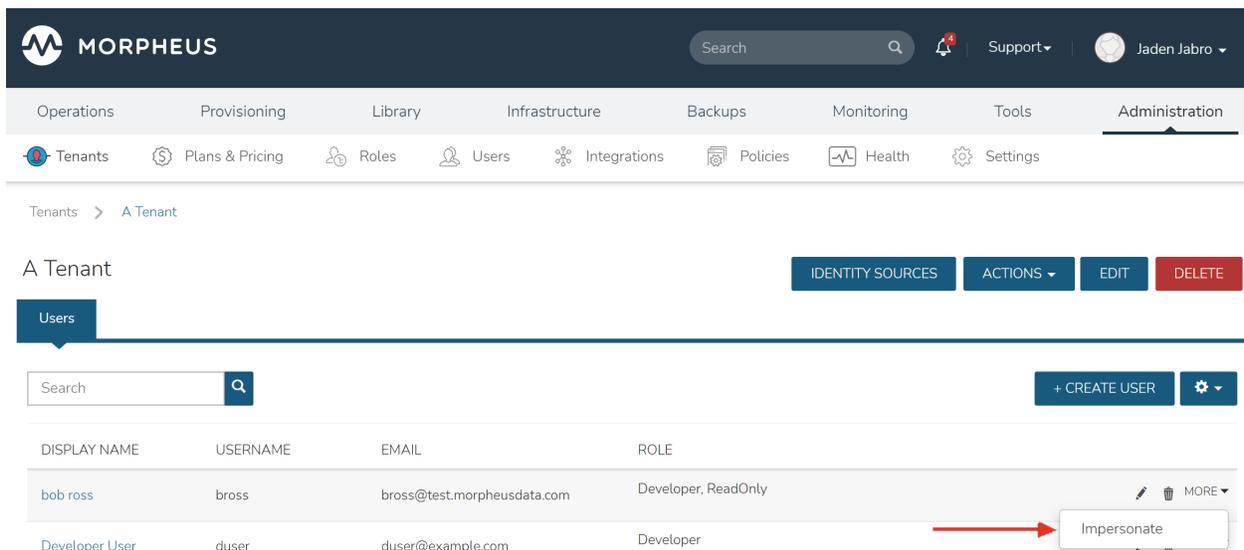
Click *Save Changes*.

Note: Users are specific to each Tenant. Users created in the Master Tenant or other sub-Tenants will only have access to the Tenant they are created in.

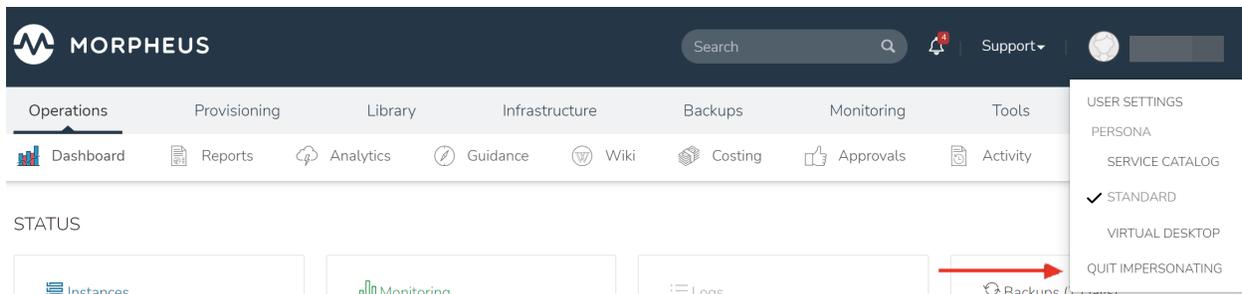
Impersonate Tenant User

Morpheus allows admin users in the Master Tenant to impersonate any user in the Subtenants to see the application as if they are that user. To impersonate a user, you must be logged in as a user with the “Impersonate User” feature enabled in the assigned role.

From inside a Tenant detail page (containing the list of that Tenant’s users), and in the specific user’s ACTIONS drop down, select “Impersonate”.



This will log you in as that user in their respective Tenant. To log out of the impersonate users Tenant, select the username in the header, and then select “Quit Impersonating”



Resources

In the Master Tenant, resources can be configured with private or public visibility:

- Private Visibility: Only available to the assigned Tenant.
- Public Visibility (option available in Master Tenant only): Available across all Tenants

Resources in the Master Tenant can also be assigned directly to Subtenants. When a resource is assigned to a Subtenant, it is only available for that Subtenant, and its visibility is automatically set to private. Public visibility is not an option for any resource assigned to or created in a Subtenant.

From the Master Tenant, the following resources can be configured for public visibility across all Tenants, or assigned to individual sub-Tenants

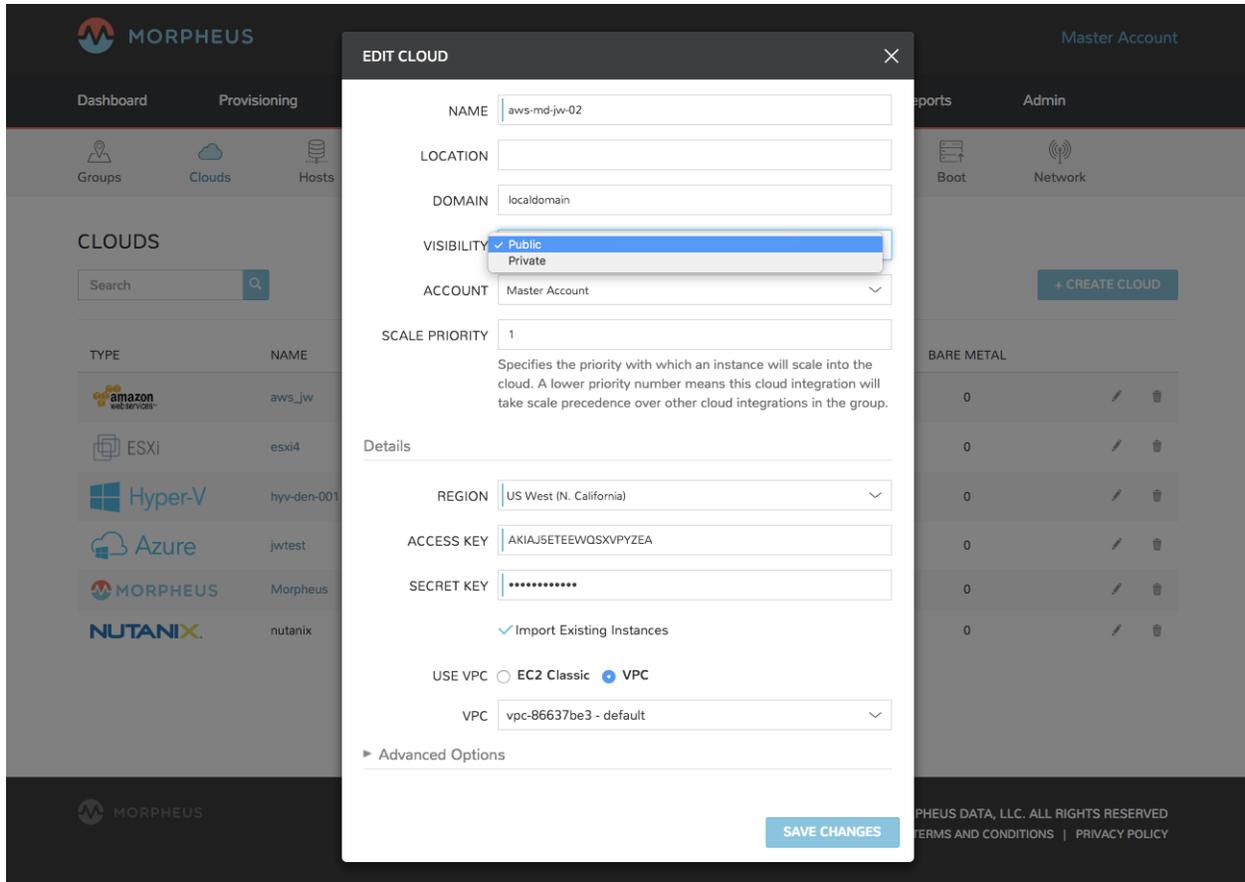
- Clouds
- Hosts
- Virtual Machines
- Networks
- Datastores
- Resource Pools
- Folders
- Virtual Images
- Library Instance Types
- Pricing
- Policies
- Workflows
- Roles

Note: Virtual Image Blueprints can be made available to multiple select Tenants when set to private.

Cloud Visibility & Assignment

To set the visibility of a Cloud to Public (shared across all Tenants) or Private (only available to the assigned Tenant):

1. Navigate to Infrastructure > Clouds
2. Select either the pencil/edit icon on the end of the cloud row, or click the name of the cloud and select “Edit” in the cloud page.
3. From the “Visibility” drop down, select either “Public” or “Private”
4. Select *Save Changes* in the footer of the Edit Cloud modal.



When a cloud is set to Public visibility, it is available to be added to Subtenants. All Subtenants created after a Master Tenant cloud is set to public will automatically have clouds with public visibility added, and a group will be created for each available cloud matching the cloud name in the new Subtenant(s).

For Tenants created prior to a Master Tenant cloud being set to public visibility, the Subtenant will have the option to add that cloud but it will not automatically be added.

While the cloud will be available for Subtenants, the resources available in that cloud to the Subtenant(s) depends on the visibility or assignment of the individual resources.

Note: A Subtenant user must have sufficient role permissions and cloud access to add publicly available clouds. Master Tenant clouds settings cannot be edited from Subtenants.

Assign a Cloud to a Tenant

Important: When assigning a Cloud to a Tenant, all resources for that Cloud will only be available to the assigned Tenant. If a cloud is created in the Master Tenant and assigned to a sub-Tenant, it will no longer be available for use by the Master Tenant or any other sub-Tenants, although it can be assigned back to the Master Tenant, or to another sub-Tenant.

It may be preferable for service providers to share or assign their cloud resources, such as specific hosts, networks, resources pools and datastores, across sub-Tenants, rather than an entire cloud.

To assign a cloud from the Master Tenant to a Sub-Tenant

1. Navigate to Infrastructure, Clouds
2. Select either the pencil/edit icon on the end of the cloud row, or click the name of the cloud and select “Edit” in the cloud page.
3. From the “Tenant” drop down, select the Tenant to assign the cloud to. The visibility will automatically be set to “Private” when a cloud is assigned to a sub-Tenant.
4. Select *Save Changes* in the footer of the Edit Cloud modal.

When a cloud is assigned to a sub-Tenant, or assigned to the Master Tenant with private visibility, that cloud and all of its resources are only available to the assigned Tenant. The Master Tenant still maintains control and visibility, and can edit the cloud settings or re-assign the cloud.

Individual Resource Visibility & Assignment

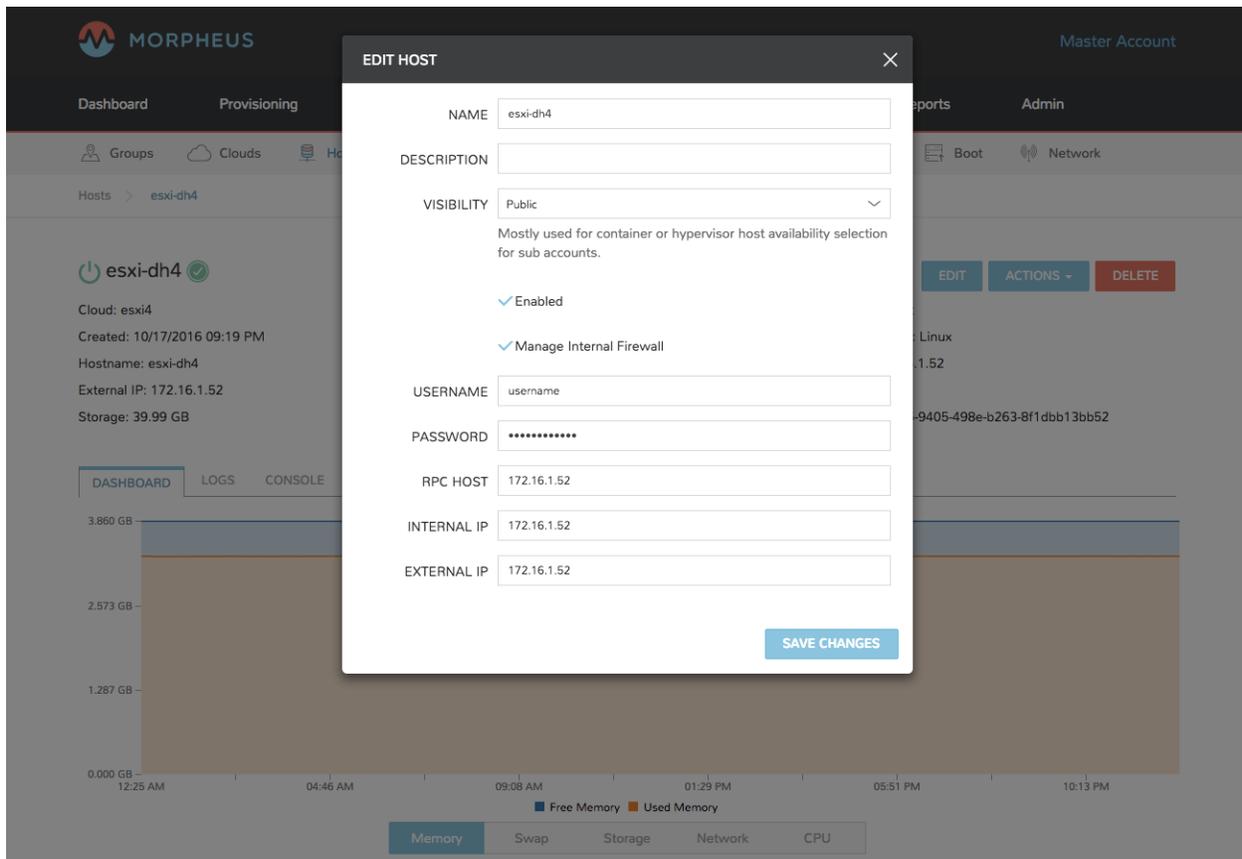
Similar to clouds, individual resources from the Master Tenant can be set to public and available to sub-Tenants, or assigned to sub-Tenants.

By default, any host, virtual machine, bare metal server, network, resource pool, datastore or blueprint added, created or inventoried by an Tenant is assigned to that Tenant. If these resources are in the Master Tenant, they can be assigned to sub Tenants. Assigning one of these resources will make it unavailable to the Master Tenant, but it will still be visible and editable by the Master Tenant. This allows Master Tenant resources to be isolated for use by sub-Tenants while still under the control of the Master Tenant.

Resources assigned to sub-Tenants from the Master Tenant will be visible and available for use by that sub-Tenant, however they cannot be edited or re-assigned by the sub-tenant.

Set the Visibility of a Host, Virtual Machine or Bare metal Server to Public or Private

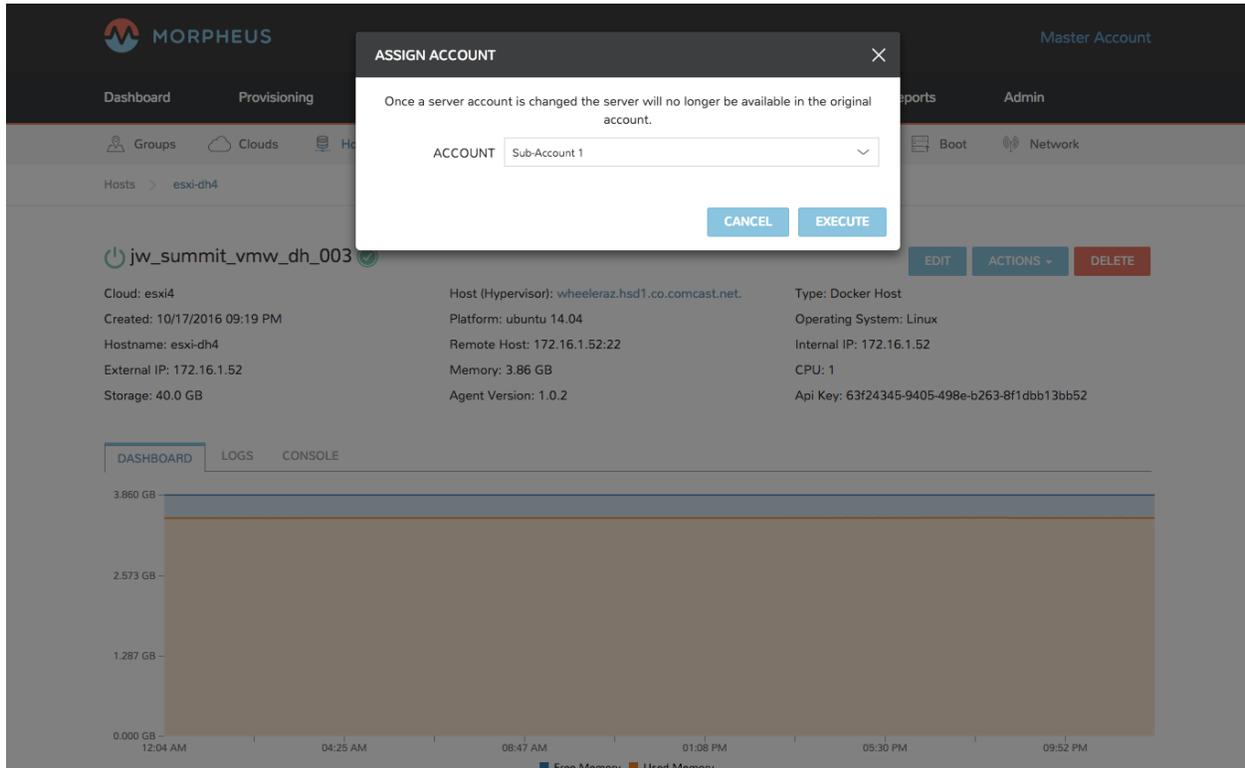
1. From the Master Tenant, navigate to Infrastructure, Hosts
2. Select either the Hosts, Virtual Machines or Bare Metal tab
3. Click the name of the resource
4. Select *Edit* in the resource page to bring up the config modal
5. From the “Visibility” drop down, select either “Public” or “Private”
6. Select *Save Changes*



Assigning a Host, Virtual Machine, or Bare Metal server to an Tenant

1. From the Master Tenant, navigate to Infrastructure, Hosts

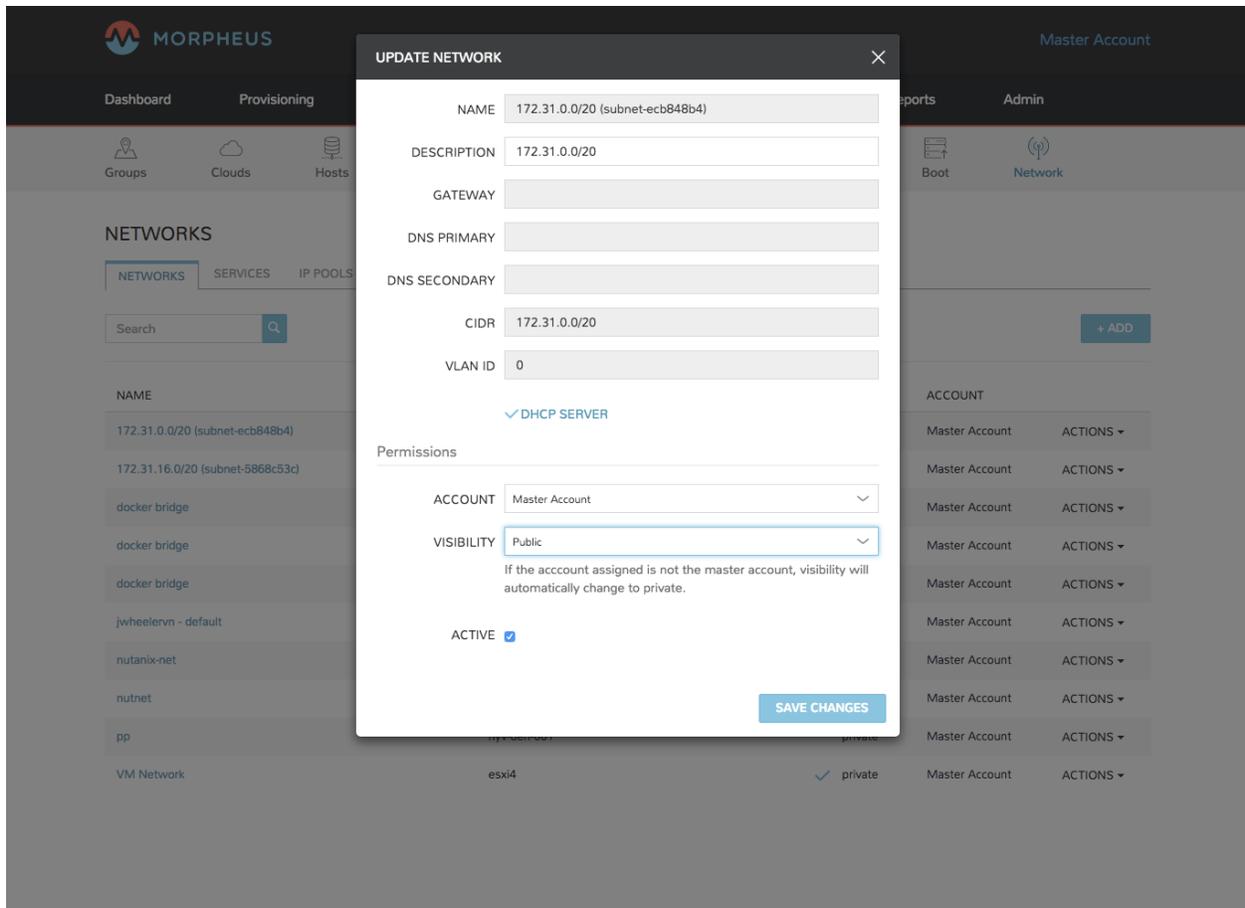
2. Select either the Hosts, Virtual Machines or Bare Metal tab
3. Click the name of the resource
4. From the “Actions” dropdown in the the resource page, select Assign Tenant
5. In the Assign Tenant modal, select the Tenant to assign the resource to.
6. Select *Execute* in the modal



The resource will now be assigned and available for use by the assigned Tenant. If assigned to a sub-Tenant, the Master Tenant will maintain visibility and control.

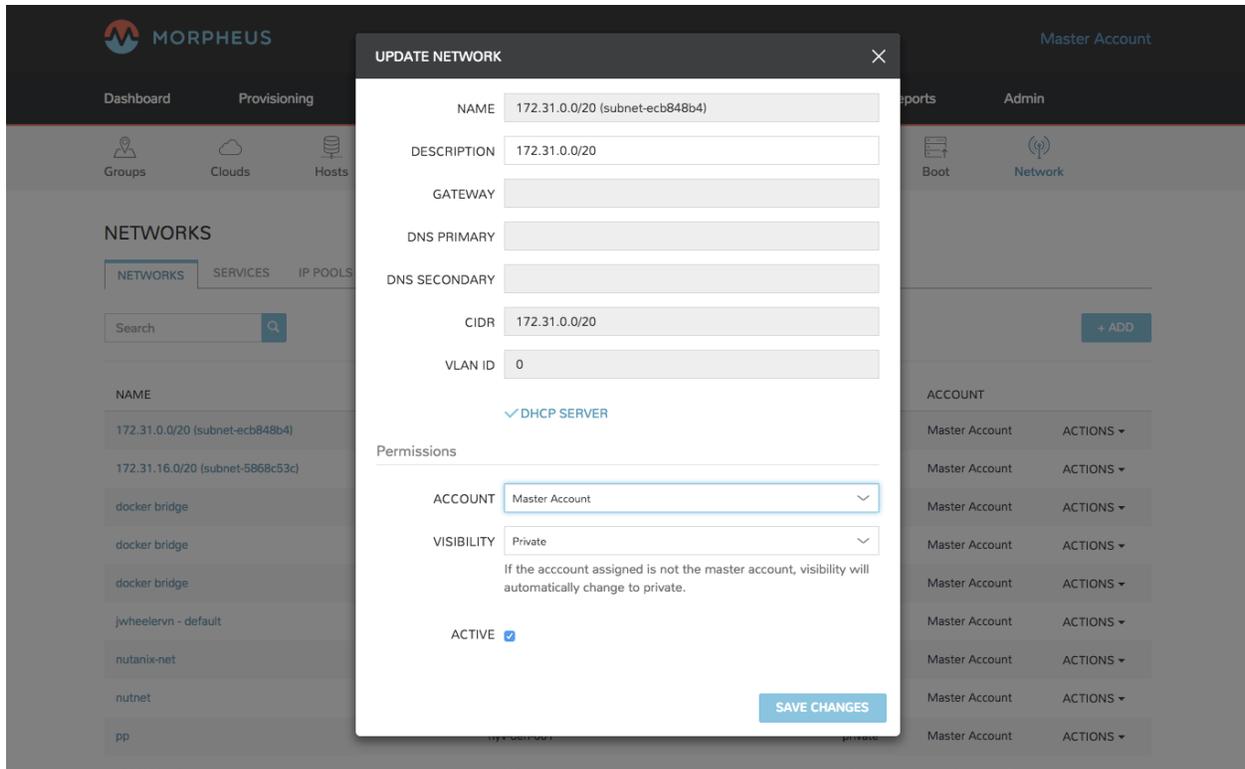
Set the Visibility of a Network to Public or Private

1. From the Master Tenant, navigate to Infrastructure, Network
2. Select either the pencil/edit icon in the network row, or click the name of the network and select “Edit” in the network page.
3. From the “Visibility” drop down, select either “Public” or “Private”
4. Select *Save Changes* in the modal



Assign a Network to a Tenant

1. From the Master Tenant, navigate to Infrastructure, Network
2. Select either the pencil/edit icon in the network row, or click the name of the network and select "Edit" in the network page.
3. From the "Tenant" drop down, select an Tenant to assign the network to.
4. Select *Save Changes* in the lower the modal



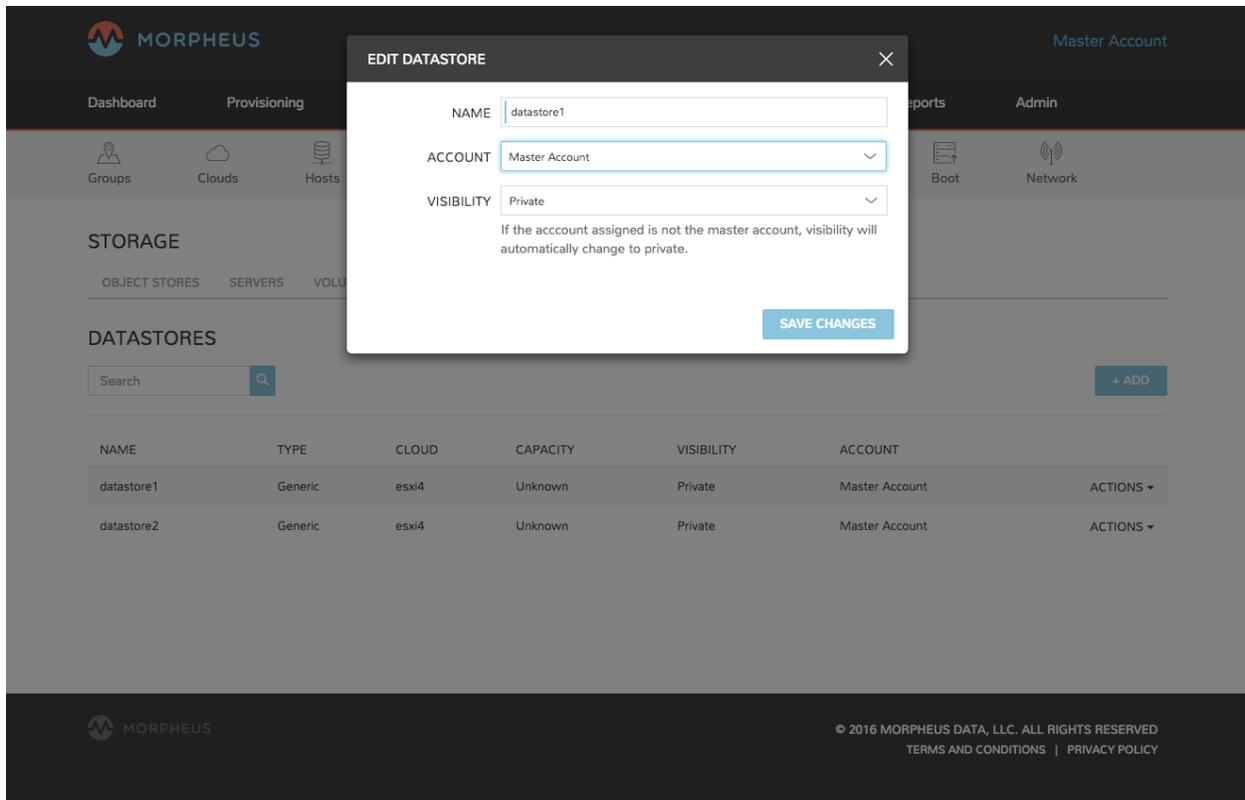
The Network will now be assigned and available for use by the assigned Tenant. If assigned to a sub-Tenant, the Master Tenant will maintain visibility and control.

Set the Visibility or assign a datastore to an Tenant

1. From the Master Tenant, navigate to Infrastructure, Storage
2. Select the “Data Stores” tab
3. Select Edit from the “Actions” dropdown in the datastores row
4. From the “Visibility” drop down, select either “Public” or “Private”
5. From the “Tenant” drop down, select the Tenant to assign the datastore to.

Note: If assigned to a sub-tenant, the visibility will be automatically set to private.

6. Select *Save Changes* in the modal

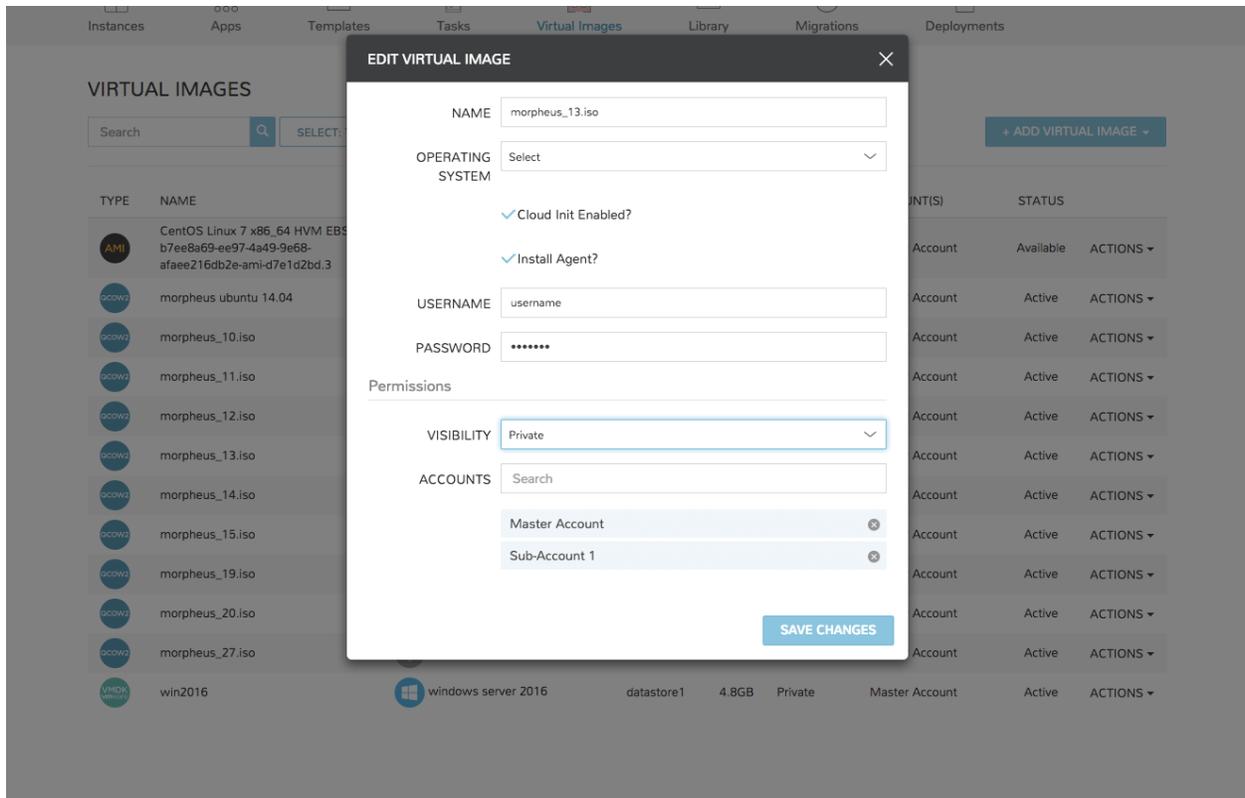


Set the Visibility or assign a Virtual Image to an Tenant

1. From the Master Tenant, navigate to Provisioning, Virtual Images
2. Select Edit from the “Actions” dropdown in the Virtual Images row
3. From the “Visibility” drop down, select either “Public” or “Private”. Public will share the
4. From the “Tenant” field, start typing the name of the Tenant to assign the Virtual Image to. Matching Tenants will populate, then select the Tenant to add.

Note: Virtual Images can be set to Private, but accessible to more than one Tenant

#. Repeat step 4 for all Tenants requiring access to the virtual image. .. To remove access for an Tenant, click the “x” next to the Tenant name #. Select *Save Changes* in the modal



The Virtual Image will now be available for use by the assigned Tenants.

5.9.2 Identity Sources

Administration > Tenants > (Selected Tenant) > Identity Sources Administration > Users > Identity Sources

Overview

Morpheus can integrate with many of the most common identity source technologies, such as Active Directory, Okta, and many others. These can be configured via the *Identity Sources* button on any Tenant detail page (Administration > Tenants > Selected Tenant) or on the Users list page (Administration > Users). These integrations map roles within these sign-on tools to equivalent roles in Morpheus so at first log in users are assigned the appropriate role.

Active Directory

Overview

Active Directory is Microsoft's primary authentication service. It is widely used in enterprise organizations and even in Microsoft cloud services. While Active Directory also supports LDAP protocol support (which Morpheus can integrate with as well), Morpheus includes a dedicated identity integration type specifically for Active Directory. By integrating Active Directory, Morpheus administrators can fully offload the work of managing the user lifecycle to Active Directory. Creating new users, applying roles to users, updating basic user data, disabling users, and more can be handled in Active Directory and automatically filtered down to Morpheus. This section includes an example integration walkthrough as well as additional details on the integration feature set.

Note: Caution should be used when integrating more than one Active Directory identity source with the same Morpheus Tenant. You must ensure the users on each identity source are unique users or that the two domains use different naming conventions for users.

How It Works

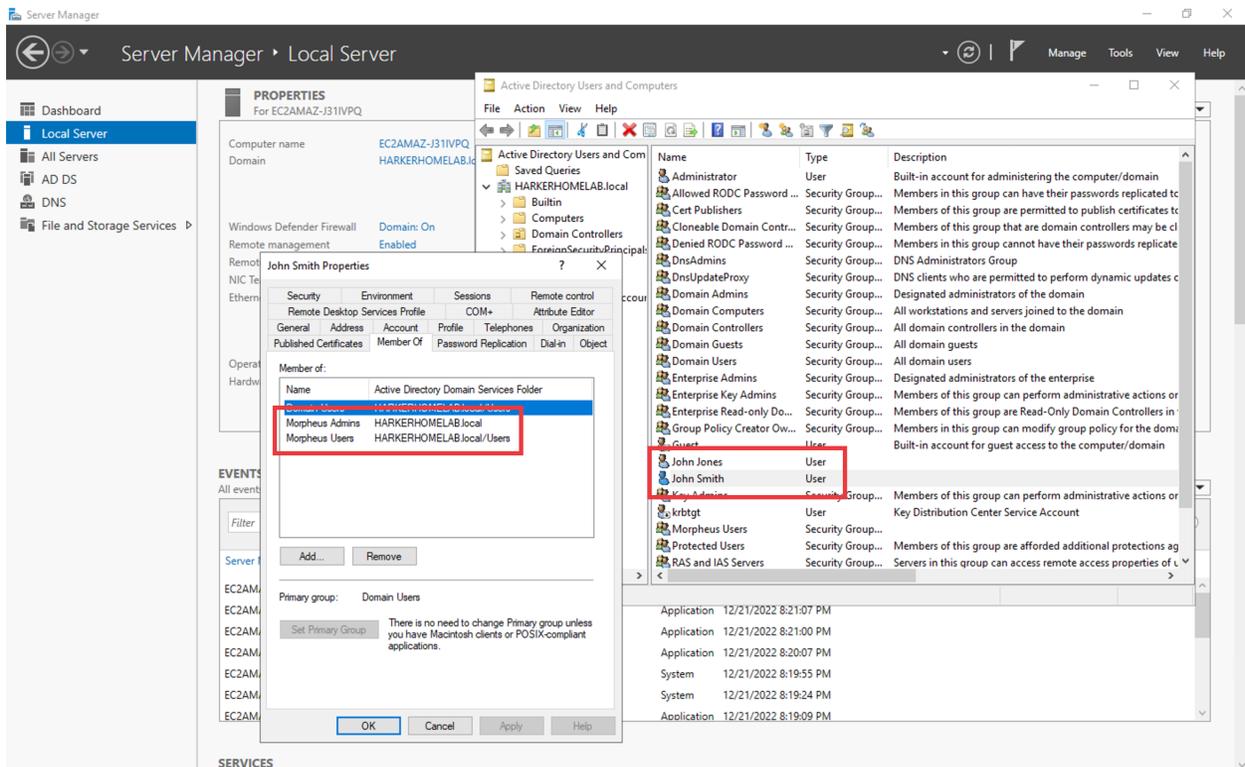
In Morpheus, identity sources (if used) are configured per Tenant. In order to see an identity integration or create a new one, navigate to a Tenant detail page (Administration > Tenants > Selected Tenant) and click *IDENTITY SOURCES*. From this page we can add a new identity source or click the pencil () icon next to an existing identity source to view or edit it. Any configured identity source will apply to just one Tenant and they cannot be shared. One scenario where this is especially useful is in an MSP appliance where each Tenant is a siloed environment for a specific customer. The customer's own existing Active Directory server and groups can be leveraged to build Morpheus user accounts with correct role mapping automatically.

When configuring an Active Directory integration in Morpheus, AD groups are mapped to Morpheus roles. When the user logs in for the first time, Morpheus adds the new user account with correct name, email address, and applies one or more roles depending on configuration. Going forward, Morpheus will sync down any changes to the user, including any role changes based on changes to the user's associated AD groups or updated passwords. Additionally, disabling a user in AD will prevent them from accessing Morpheus.

Important: Morpheus will connect over port 389 for non-secure LDAP and port 636 for secure LDAP. Ensure that Morpheus is able to talk to the AD server on the proper port.

Example Integration

In a simple example, we have an Active Directory server which has two groups relevant to Morpheus, "Morpheus Users" and "Morpheus Admins." We will configure Morpheus so that only users in the "Morpheus Users" group can access Morpheus in any capacity. Users who are also in the "Morpheus Admins" group will take on the System Admin role in Morpheus. We'll see later when the integration is configured how Morpheus roles can be mapped to AD groups. In the same AD server, I have two users. John Smith is in groups "Morpheus Users" and "Morpheus Admins". John Jones is only in the "Morpheus Users" group.



Knowing the AD scheme and the requirements for Morpheus user roles, we can begin the process of creating the integration. Identity integrations are specific to each Tenant so begin by navigating to the Tenant detail page (Administration > Tenants > Selected Tenant) and clicking *IDENTITY SOURCES*. On setting the type to “Active Directory,” the form will update with the needed fields. Note the following basic fields:

- **AD SERVER:** The IP address or hostname for the Active Directory Server
- **DOMAIN:** The AD domain in which the relevant users and groups reside
- **BINDING USERNAME:** A server user which has access to relevant objects on the AD server. In my example, I’ve used the in-built Administrator user which is the easiest option. Other users may be used depending on your organization’s IT security policies but the integration with Morpheus will not work properly if the user does not have the needed access
- **BINDING PASSWORD:** The password for the user in the prior field

With the basic configurations completed, the remaining configurations will affect Morpheus user and role generation. A REQUIRED GROUP is optional and is a group the user *must* be in to have any Morpheus access. Here we require that users be in the “Morpheus Users” group. A DEFAULT ROLE is required and will be assigned to all users regardless of any additional roles they may be assigned based on their AD group membership. Beyond that, all other Morpheus roles will be listed here and an AD group name can be associated with as many as you required. In this example, we are giving users in the “Morpheus Admins” group the Morpheus System Admin role. Though we did not use them, it’s worth pointing out that ENABLE ROLE MAPPING PERMISSION will give administrators in the Tenant the ability to update the AD role mappings (though they will not have access to the core integration fields such as AD SERVER, DOMAIN, or binding user details). MANUAL ROLE ASSIGNMENT allows users to manually update Morpheus roles outside of the automatic mappings created by the AD integration.

EDIT IDENTITY SOURCE
✕

Identity Source

TYPE

ACTIVE

NAME

DESCRIPTION

Active Directory Connection

AD SERVER

DOMAIN

USE SSL Yes No

BINDING USERNAME

BINDING PASSWORD

REQUIRED GROUP

Include Member Groups

Role Mappings

DEFAULT ROLE

AZRA CATALOG USER

MULTI-TENANT ADMIN

SYSTEM ADMIN

TEST MULTITENANT

USER ADMIN

Role Options

ENABLE ROLE MAPPING PERMISSION

MANUAL ROLE ASSIGNMENT

DEACTIVATE
DELETE
SAVE CHANGES

With the above integration steps completed, users can now log into Morpheus and a user account with correct roles will automatically be created. In our example case, John Smith has logged in and we can see he is assigned the default role as well as the System Admin role based on his AD group associations. Going forward, Morpheus will continue to sync any changes to these users. For example, Morpheus roles may be updated based on changing AD groups or user access may be completely revoked by disabling the user in AD.

The screenshot shows the Morpheus Administration interface. The top navigation bar includes the Morpheus logo, a search bar, a notification bell with 2 alerts, a 'Support' dropdown, and the user profile 'Alex Harker'. Below this is a secondary navigation bar with tabs for Operations, Provisioning, Library, Infrastructure, Backups, Monitoring, Tools, and Administration. Under the Administration tab, there are sub-tabs for Tenants, Plans & Pricing, Roles, Users, Integrations, Policies, Health, and Settings. The 'Users' sub-tab is active, showing a 'User Groups' dropdown. A search bar is present above a table of users. To the right of the search bar are buttons for 'IDENTITY SOURCES', '+ CREATE USER', and a settings gear. The table has columns for DISPLAY NAME, USERNAME, EMAIL, and ROLE. The first row shows 'Alex Harker' with role 'System Admin'. The second row shows 'John Smith' with role 'Azra Catalog User, System Admin', which is highlighted with a red box. A 'MORE' dropdown menu is visible at the end of the second row.

DISPLAY NAME	USERNAME	EMAIL	ROLE
Alex Harker	aharker	aharker@morpheusdata.com	System Admin
John Smith	jsmith	jsmith@example.com	Azra Catalog User, System Admin

Adding an Active Directory Integration

1. Navigate to Administration > Tenants
2. Select a Tenant
3. Select *IDENTITY SOURCES*
4. Select + *ADD IDENTITY SOURCE*
5. Set the TYPE to “Active Directory”
6. Populate the following:

Name A friendly name in Morpheus for the AD integration

AD Server The Hostname or IP address of AD Server

Domain The AD domain in which the relevant user and group objects reside

USE SSL Indicates whether SSL should be used for communication with the AD server. Morpheus will connect over port 389 for non-secure LDAP and port 636 for secure LDAP, ensure Morpheus can connect to the AD server over the correct port

Binding Username A username for a service account which has access to relevant objects (users, groups, etc.). For ease, the “Administrator” user may be used

Binding Password The password for the above account

Required Group The AD group users must be in to have access (optional, see example in the prior section)

Default Role The default role a user is assigned when they are in the required group or if no specific group mapping applies to the user (see example in prior section)

ENABLE ROLE MAPPING PERMISSION When selected, Tenant users with appropriate rights to view and edit Roles will have the ability to set role mapping for the Identity Source integration. This allows the Tenant user to edit only the role mappings without viewing or potentially editing the basic Identity Source configuration (AD server, domain, binding user details, etc)

MANUAL ROLE ASSIGNMENT When selected, administrators can manually edit Roles for users created through this identity source integration from the user detail page (Administration > Users > Selected user)

Note: For more on Identity Source role mapping permissions, see the [associated guide](#) in our KnowledgeBase.

7. Select *SAVE CHANGES*.

Now allowed AD users can login to Morpheus via their Active Directory credentials and a User will be automatically generated to Morpheus with matching metadata and mapped Role permissions.

Note: Sub-tenant Morpheus API authentication for Active Directory generated users is not currently supported.

Troubleshooting

If you're unable to get the Active Directory integration to work, the following troubleshooting steps may be useful to ensure your appliance can talk to the Active Directory server.

1. Open firewall ports

Source: Morpheus appliance

Destination: AD server's FQDN or IP address

Non-SSL AD integration: TCP-389

SSL AD integration: TCP-636

2. Checking open LDAP connections from the Morpheus appliance

Connect to a Morpheus appliance box and run the following:

```
$ sudo lsof i- | grep :ldap
```

3. Check LDAP connectivity from the Morpheus appliance

Connect to a Morpheus appliance box and run the following. Be sure to replace the placeholder values in the command with the correct values for your environment:

```
$ ldapsearch -x -h xx.xx.xx.xx -D "binding-user@acme.com" -W -b "cn=users,  
↪dc=acme,dc=com"
```

4. Run tcpflow from the Morpheus appliance for non-SSL enabled AD identity Integrations

Use tcpflow from the Morpheus appliance and then start the identity source configuration once again. Keep in mind this will only work for AD servers which are not SSL enabled:

```
$ sudo tcpflow -i any -c -v port 389
```

5. Check the AD and domain controllers event logs

Check the event logs for LDAP queries from the Morpheus appliance to ensure network connectivity.

Azure Active Directory SSO (SAML)

Azure Active Directory Single Sign-on can be added as a Identity Source in Morpheus using the SAML Identity Source Type. The Azure AD SSO configuration is slightly different than other SAML providers, and this guide will assist in adding a Azure AD SSO Identity Source.

Create Azure Enterprise Application

1. Login to the [Azure Portal](#)
2. Navigate to: **Azure Active Directory > Enterprise Applications**
3. Click the **+ New application** button at the top
4. Click the **+ Create your own application** button at the top
5. Ensure **Integrate any other application you don't find in the gallery (Non-gallery)** is selected and enter a name for the app. Common examples are: **MorpheusSSO**
6. Click the **Create** button at the bottom and wait for it to complete
7. Once created, you'll be in the **Overview** of the application created. Navigate to the **Single sign-on** section from the left pane
8. Choose **SAML** as the Single sign-on method
9. Copy both the **Login URL** and **Logout URL** in Step 4, we'll need these in some of the next steps
10. Before we can continue configuring the application, the configuration needs to be generated in Morpheus for more data

Create an Azure AD SAML Integration in Morpheus

Azure requires inputting the **Identifier (Entity ID)** and **Reply URL (Assertion Consumer Service URL)** in the Azure SSO configuration before it provides the Endpoints and Certificate necessary to add the Integration into Morpheus. In order to get the **Identifier (Entity ID)** and **Reply URL (Assertion Consumer Service URL)** to input into Azure SSO configuration, we need to create a **Azure AD SAML SSO** integration in Morpheus first.

To add the integration:

1. Login to Morpheus
2. Navigate to **Administration > Tenants**
3. Click a tenant hyperlink
4. Click the **IDENTITY SOURCES** button in the Tenant detail page
5. Click the **+ ADD IDENTITY SOURCE** button
6. Select **Azure AD SAML SSO** from the **TYPE** dropdown
7. Add
 - Name
 - (Optional) Description
 - Paste the **Login URL** copied from Azure into the **LOGIN REDIRECT URL** field
 - Paste the **Logout URL** copied from Azure into the **SAML LOGOUT REDIRECT URL** field

- This is the minimum information needed for now, which will let us generate the details needed from Morpheus. We'll return to this configuration page later to enter more information.
- Click the *SAVE CHANGES* button

Important: Setting SAML REQUEST to “No Signature” and SAML RESPONSE to “Do Not Validate Assertion Signature” is allowed but not recommended for security reasons.

Upon saving, the *Entity ID* (Identifier (Entity ID)) and *SP ACS URL* (Reply URL (Assertion Consumer Service URL)) will be provide in the Identity Source list view. Copy these for use in Azure SSO configuration.



Configure Azure Enterprise Application

This guide assumes an Azure AD Enterprise Application has already been created. Please refer to documentation above, if this has not already been configured.

- Navigate to: **Azure Active Directory > Enterprise Applications > Single sign-on**
- Choose SAML as the Single sign-on method
- On Step 1 (Basic SAML Configuration), click the Edit button and enter the following:
 - Identifier (Entity ID)** Enter the Entity ID URL from the Morpheus Identity Source Integration above
 - Reply URL (Assertion Consumer Service URL)** Enter the SP ACS URL from the Morpheus Identity Source Integration above
 - Logout URL** Enter the following format: `https://yourUrl/login/` If this is a sub tenant, the format may instead be the following: `https://yourUrl/login/account/1` The login URL can be found under *IDENTITY SOURCES* in the tenant
- On Step 2 (Attributes and Claims), click the Edit button
- Click the Add a group claim button at the top
- Choose All groups and ensure Group ID is selected for the Source attribute dropdown

Note: You can also choose Security groups, which ever makes more sense for the organization

- Close the pane and return to the Enterprise Application in the Single sign-on section
- On Step 3 (SAML Certificates), click the Download link next to Certificate (Base64) and Federation Metadata XML

Note: The files will download, keep them available for later configuration in Morpheus

- Navigate to Users and Groups in the left pane
- Click the Add user/group button

11. Add Azure groups to this application that will be able to login to Morpheus

Note: Note the object ID for each of these groups, as they will be used later when configuring Morpheus to map the group to roles

12. Once groups have been added, click the Assign button at the bottom

Configure the Azure AD SAML Integration in Morpheus

1. Login to Morpheus using Username and Password, as usual
2. Navigate to Administration > Tenants
3. Click a tenant hyperlink
4. Select *IDENTITY SOURCES* in the Tenant detail page
5. Click the pencil (edit) next to the integration created previously
6. Ensure the SAML REQUEST field is set to Self Signed

Note: A custom RSA signature can be used here if needed, if required by the organization

7. Ensure the SAML RESPONSE field is set to Validate Assertion Signature

Note: With this setting, if the assertion signature ever changes in the Azure Enterprise Application, this would need to be updated to match

8. Edit/view the downloaded Federation Metadata XML (.xml extension) file from the previous section

Note: It is recommended to use Microsoft Edge, or another browser, to view the contents

9. In the Federation Metadata XML file, locate the `<X509Certificate>` `</X509Certificate>` under the `<Signature>` section. Copy the entire contents between the `<X509Certificate>` and `</X509Certificate>`, it is very long
10. Paste the value copied from the Federation Metadata XML file into the Public Key (Optional) box, below the SAML RESPONSE dropdown

Configure Role Mappings

Role mappings will map Azure AD Groups to Morpheus Roles. Azure AD users will be assigned Roles in Morpheus upon signing in based on their Group Membership in Azure AD.

Important: Use an Azure Groups Object ID, not Group name, when entering Role Mappings. Example: 7626a4a2-b388-4d9b-a228-72ce9a33bd4b

DEFAULT ROLE Role a Azure AD user will be assigned by default upon signing in to Morpheus using this Identity Source.

REQUIRED AZURE AD GROUP OBJECT ID Object ID of Azure AD Group a user must be a member of to be authorized to sign in to Morpheus. Users not belonging to this Group will not be authorized to login to Morpheus. This field is optional, and if left blank, any user from the Azure AD App will be able to sign in to Morpheus and will be assigned the Default Role if no Role Mappings match AD Group membership.

GROUP ASSERTION ATTRIBUTE NAME Enter `http://schemas.microsoft.com/ws/2008/06/identity/claims/groups` for Azure AD SSO

Additional Role Mappings The existing Roles in Morpheus will be listed. To map a Morpheus Role to an Azure AD Group, enter the Object ID of the desired Azure AD Group in the *Role Attribute Value* field for the corresponding Morpheus Role.

Important: Use an Azure Groups Object ID, not Group name, when entering Role Mappings. Example: 7626a4a2-b388-4d9b-a228-72ce9a33bd4b

ENABLE ROLE MAPPING PERMISSION When selected, Tenant users with appropriate rights to view and edit Roles will have the ability to set role mapping for the Identity Source integration. This allows the Tenant user to edit only the role mappings without viewing or potentially editing the Identity Source configuration.

MANUAL ROLE ASSIGNMENT When selected, administrators can manually edit Roles for users created through this identity source integration from the user detail page (Administration > Users > Selected user).

Note: For more on Identity Source role mapping permissions, see the [associated guide](#) in our KnowledgeBase.

Once populated, select *SAVE CHANGES* and the SAML identity source integration will be added. The Identity Source can be edited anytime to deactivate or change Role Mappings or other values.

Note: If Role mappings are edited after Azure AD SSO users have signed into Morpheus, currently logged in users will need to log out of Morpheus for the new Role mappings to take effect, when applicable.

1. Under the Role Azure Group Mappings section, verify the DEFAULT ROLE dropdown has the role in Morpheus selected that all users will be assigned by default
 - It is recommended that this role contains no permissions, which ensures that anyone who authenticates gets no access
2. Under the Role Azure Group Mappings section, you will see role names listed. Next to these are text boxes with Assertion Attribute Mappings inside. Enter group object IDs from Azure into these text boxes. This will map the Azure AD groups to specific roles in Morpheus
3. Finally, click Save Changes at the bottom of the page

Here is an example of the configuration above:



Azure Group Lookups

When a user in azure ad has more that 150 group attributes, Azure does not include the group claims in the SAML response, and Morpheus is required to query Microsoft Graph to obtain the users group attribute values. When there are users that are members of more that 150 groups, populate the Azure Group Lookups section in order for those users to be able to use the Azure AD SAML SSO integration, otherwise no groups will be obtained and proper role mappings cannot occur.

AZURE TENANT ID Add Azure AD Tenant ID if user group membership will exceed 150. See *Copy Directory (tenant) and Application (client) IDs* for information on obtaining an Azure AD Tenant ID

AZURE APP ID Add Azure AD Application (Client) ID if user group membership will exceed 150. See *Copy Directory (tenant) and Application (client) IDs* for information on obtaining an Azure AD Application (Client) ID

AZURE APP SECRET Add Azure Application (Client) Secret if user group membership will exceed 150. See *Generate a Client Secret* for information on creating an Azure Application (Client) Secret

ROLE LINK ATTRIBUTE NAME default: <http://schemas.microsoft.com/claims/groups.link>. This is not normally changed.

Logging Into Morpheus with Azure AD SAML

1. Navigate to the Morpheus URL
2. A new button will appear to allow sign-in using Azure AD SAML, with the same name as the integration. Click the button .. image:: /images/integration_guides/identity_sources/azure_ad_saml/sign_in_page.png
width 30%
3. Sign-in with your Microsoft/Azure account .. image:: /images/integration_guides/identity_sources/azure_ad_saml/ms_signin.png
width 20%

Note: If no local users other than the System Admin have been created, “USERNAME AND PASSWORD” option will not be displayed, only the SAML option.

Okta

Overview

Morpheus allows users to integrate an Okta deployment for user management and authentication. In Morpheus, identity sources are added on a per-Tenant basis and Morpheus allows you to map Okta user groups to Morpheus user groups. User accounts are automatically created with matching metadata and role permissions when users are authenticated.

Adding an Okta Integration

1. Navigate to Administration > Tenants
2. Select a Tenant
3. Select *IDENTITY SOURCES*
4. Select + *IDENTITY SOURCE*
5. Choose TYPE: “Okta”
6. Populate the following, then select *SAVE CHANGES*:

Name Unique name for authentication type

Description A description for your new Okta Identity Source

Okta URL Your Okta URL

Administrator API Token Your Okta Administrator API Token

Required Group The Okta group that users must be in to have access (optional)

Default Role The default role a user is assigned if no group is listed under an Okta user that maps within the Morpheus Role Mappings section

ENABLE ROLE MAPPING PERMISSION When selected, Tenant users with appropriate rights to view and edit Roles will have the ability to set role mapping for the Identity Source integration. This allows the Tenant user to edit only the role mappings without viewing or potentially editing the Identity Source configuration.

MANUAL ROLE ASSIGNMENT When selected, administrators can manually edit Roles for users created through this identity source integration from the user detail page (Administration > Users > Selected user).

Note: For more on Identity Source role mapping permissions, see the [associated guide](#) in our KnowledgeBase.

Now, allowed Okta users can log into Morpheus via their Okta credentials and a user will be automatically generated within Morpheus with matching metadata and mapped Role permissions.

Note: If you’ve created multi-tenant roles, these will also appear here and can be mapped to Okta user groups allowing you to map users to equivalent user groups in Morpheus.

OneLogin

Adding OneLogin Identity Source Integration

1. Navigate to Administration > Tenants
2. Select the Tenant to add the Identity Source Integration
3. Select *IDENTITY SOURCES*
4. Select + *IDENTITY SOURCE*
5. Enter the following:

TYPE OneLogin

NAME Name of the Identity Source Integration in Morpheus

DESCRIPTION Optional Description of the Identity Source

ONELOGIN SUBDOMAIN**example: morpheus-dev**

Warning: Please verify the subdomain carefully. An invalid subdomain will cause authentication attempts by OneLogin users to fail.

ONELOGIN REGION Specify US or EU region**API CLIENT SECRET** OneLogin API Client Secret from the Settings - API section in OneLogin portal**API CLIENT ID** OneLogin API Client ID from the Settings - API section in OneLogin portal**REQUIRED ROLE** Enter a role of OneLogin users logging into Morpheus must have at least this OneLogin role to gain access to Morpheus.**DEFAULT ROLE** The default Morpheus Role applied to users created from OneLogin Integration if no other role mapping is specified below**ROLE MAPPINGS** Existing Morpheus Roles will be listed with fields to enter OneLogin Roles to map to. Users with OneLogin roles matching the role mappings will be assigned the appropriate Role(s) in Morpheus when signing in.**ENABLE ROLE MAPPING PERMISSION** When selected, Tenant users with appropriate rights to view and edit Roles will have the ability to set role mapping for the Identity Source integration. This allows the Tenant user to edit only the role mappings without viewing or potentially editing the Identity Source configuration.**MANUAL ROLE ASSIGNMENT** When selected, administrators can manually edit Roles for users created through this identity source integration from the user detail page (Administration > Users > Selected user).

Note: For more on Identity Source role mapping permissions, see the [associated guide](#) in our KnowledgeBase.

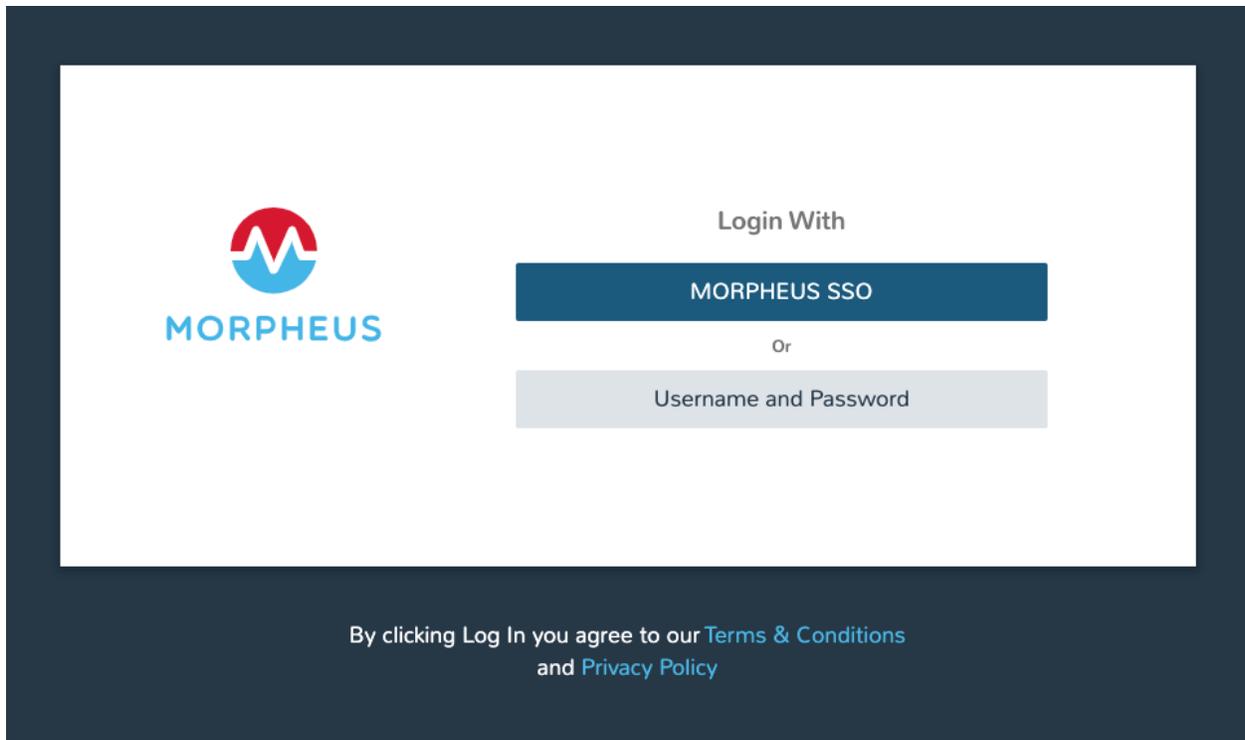
1. Select *SAVE CHANGES* and the OneLogin Integration will be added.

Users can now login to Morpheus with OneLogin credentials. The first Login will create a user in Morpheus matching the Username, email and Password from OneLogin. If a **REQUIRED ROLE** is specified in the Identity Source settings, only users with that Role in OneLogin will be able to login to Morpheus.

Important: OneLogin users will not authenticate in Morpheus if there is an existing Morpheus User with matching username or email address.

SAML Integration**Overview**

The Morpheus SAML identity source integration allows customers to add user SSO to Morpheus, authenticated by external login SAML providers.



Adding a SAML Integration

To add a SAML integration:

1. Navigate to Administration > Tenants
2. Select a tenant.
3. Select *IDENTITY SOURCES* in the Tenant detail page
4. Select + *ADD IDENTITY SOURCE*.
5. Select *SAML SSO* from the *TYPE* field
6. Add a Name and optional Description for the SAML integration

NEW IDENTITY SOURCE ✕

Identity Source

TYPE

NAME

DESCRIPTION

SAML SSO Configuration

LOGIN REDIRECT URL

SAML LOGOUT REDIRECT URL

INCLUDES SAML REQUEST PARAMETER

SAML REQUEST

SAML RESPONSE

Encryption RSA Private Key (Optional)

Copy

Advanced Validation Options

Role Mappings

DEFAULT ROLE

ROLE ATTRIBUTE NAME

REQUIRED ROLE ATTRIBUTE VALUE

LITTLE ACCESS

SUPER ADMIN - MULTI-TENANT

SYSTEM ADMIN - MULTI-TENANT

Role Options

ENABLE ROLE MAPPING PERMISSION

MANUAL ROLE ASSIGNMENT

Assertion Attribute Mappings Show/Hide

There are 4 sections with fields that need to be populated depending on the desired configuration:

- SAML Configuration
- Role Mappings
- Role Options
- Assertion Attribute Mappings

SAML Configuration

LOGIN REDIRECT URL This is the SAML endpoint Morpheus will redirect to when a user signs into Morpheus via SAML

SAML LOGOUT REDIRECT URL The URL Morpheus will POST to when a SAML user logs out of Morpheus

INCLUDES SAML REQUEST PARAMETER Yes (recommended) - the AuthN request will be sent via the ?SAMLRequest= parameter in the URL (GET)

No - the AuthN request will be submitted in the body of the request (POST)

Note: The SAML SP documentation should mention which binding to use but GET is most common

SAML REQUEST No Signature - No signature is used on the SAML request

Self Signed - A self-signed X.509 Certificate is generated after clicking *SAVE CHANGES*. This signature value can be used by the SAML SP to verify the authenticity of the request

Custom RSA Signature - Import a custom RSA Private Key and respective X.509 Certificate. This signature value can be used by the SAML SP to verify the authenticity of the request

SAML RESPONSE Do Not Validate Assertion Signature - The SAML response signature from the SAML SP will not be validated

Validate Assertion Signature - The SAML response signature from the SAML SP will be validated. Enter the SAML SP X.509 certificate in the **Public Key** field. This must be in PEM format

Important: Setting SAML REQUEST to “No Signature” and SAML RESPONSE to “Do Not Validate Assertion Signature” is allowed but not recommended for security reasons.

Role Mappings

DEFAULT ROLE Role any SAML user will be assigned by default

ROLE ATTRIBUTE NAME The name of the attribute/assertion field that will map to Morpheus roles, such a MemberOf

REQUIRED ROLE ATTRIBUTE VALUE Attribute/assertion value that a user must be assigned/a member of to be authorized, such as group or role in the SAML SP. This is obtained from the attribute/assertion defined in the ROLE ATTRIBUTE NAME field

<Morpheus ROLE NAME> Additional roles that can be mapped to a user, which will add to the DEFAULT ROLE. Attribute value that a user must be assigned/a member of to be authorized, such as group or role in the SAML SP. This is obtained from the attribute/assertion defined in the ROLE ATTRIBUTE NAME field

Note: For more on Identity Source role mapping permissions, see the [associated guide](#) in our KnowledgeBase.

Role Options

ENABLE ROLE MAPPING PERMISSION When selected, Tenant users with appropriate rights to view and edit Roles will have the ability to set role mapping for the Identity Source integration. This allows the Tenant user to edit only the role mappings without viewing or potentially editing the Identity Source configuration.

MANUAL ROLE ASSIGNMENT When selected, administrators can manually edit Roles for users created through this identity source integration from the user detail page (Administration > Users > Selected user).

Assertion Attribute Mappings

GIVEN NAME ATTRIBUTE NAME SAML SP field value to map to Morpheus user First Name

SURNAME ATTRIBUTE NAME SAML SP field value to map to Morpheus user Last Name

EMAIL ATTRIBUTE SAML SP field value to map to Morpheus user email address

SAML SSO Configuration

LOGIN REDIRECT URL

SAML LOGOUT REDIRECT URL

INCLUDES SAML REQUEST PARAMETER

SAML REQUEST

SAML RESPONSE

Copy

Advanced Validation Options

Assertion Attribute Mappings Show/Hide

GIVEN NAME ATTRIBUTE NAME

SURNAME ATTRIBUTE

Once populated, select *SAVE CHANGES* and the SAML identity source integration will be added.

In the *Identity Sources* section, important information for configuration of the SAML integration is provided. Use the SP ENTITY ID and SP ACS URL for configuration on the external login SAML provider side.

Note: In some cases, the SAML provider may need these values before providing the LOGIN REDIRECT URL and other values. When creating the integration, the NAME and LOGIN REDIRECT URL can contain any values, then selecting SAVE CHANGES will generate the above values. The NAME and LOGIN REDIRECT URL can be edited later, once the SAML configuration is created in the SAML provider.

- ENTITY ID
- SP ACS URL
- LOGIN REDIRECT URL

- SP METADATA

SAML SSO	Okta - SAML	ENTITY ID:	https://[REDACTED]/saml/eDKL60P25
		SP ACS URL:	https://[REDACTED]/externalLogin/callback/eDKL60P25
		LOGIN REDIRECT URL:	https://[REDACTED] [REDACTED]/sso/saml
		SP METADATA:	VIEW

Sample Metadata code output:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><EntityDescriptor entityID=
↳ "https://someip.com/saml/eDKL60P25" xmlns="urn:oasis:names:tc:SAML:2.0:metadata">
↳ <SPSSODescriptor AuthnRequestsSigned="false" WantAssertionsSigned="true"
↳ protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"><NameIDFormat>
↳ urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</NameIDFormat>
↳ <AssertionConsumerService index="0" isDefault="true" Binding=
↳ "urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://someip.com/
↳ externalLogin/callback/eDKL60P25"/></SPSSODescriptor></EntityDescriptor>
```

Note: Different SAML providers will have different field names and requirements. An Okta SAML Dev environment was used for the example integration in this article.

Okta SAML SSO

For Okta SAML integration, the following fields are mapped:

- LOGIN REDIRECT URL : Identity Provider Single Sign-On URL
- ENTITY ID: Audience URI (SP Entity ID)
- SP ACS URL: Single sign on URL

Onelogin SAML SSO

For Onelogin SAML integration, the following fields are mapped:

- LOGIN REDIRECT URL : SAML 2.0 Endpoint (HTTP)
- SAML LOGOUT REDIRECT URL : SLO Endpoint (HTTP)
- SIGNING PUBLIC KEY : X.509 Certificate
- ENTITY ID: ACS (Consumer) URL Validator
- SP ACS URL: ACS (Consumer) URL

5.9.3 Plans & Pricing

Overview

Service Plans determine the amount of compute resources available to each Instance. When provisioning new Instances from Morpheus, a plan is selected which determines the number of CPU cores, amount of memory and the amount of storage available to the associated machines. Additionally, when converting discovered instances in integrated clouds to Morpheus-managed Instances, the user selects a plan which best fits the instance as it is currently configured. When Instances are reconfigured, a new plan may be selected which redefines the compute resources which should be available to the Instance.

Plans can be as specific or open-ended as the user would like, restricting the user to the resources defined in the plan or allowing the user to increase those amounts at provision time. Price sets are associated with plans, which is how Morpheus can compute cost values even for Instances running in private, on-prem Clouds.

The Plans & Pricing section is where Plans, Prices Sets and Prices are managed. These concepts are associated with each other in the following way: Prices are added to Prices Sets, and Price Sets are added to Plans. Plans include Service Plans, Load Balancer Price Plans, Virtual Image Prices Plans and Snapshot Price Plans. Price sets can be agnostic and available for any Plan, or specific to a Region Code, Cloud or even a specific Resource Pool. Prices are configured per Price Type and can be added to Price Set configurations that support the Price Type. Master Tenant Prices can apply to all Tenants or be assigned to a single Tenant.

A set of Service Plans are seeded by Morpheus for private cloud provision types as well as some public clouds. Additional Plans and Prices are synced in for other public clouds when the corresponding cloud types are added to an appliance. Users can create new Service Plans or edit the system-seeded Service Plans for Private Cloud types. Service Plans configurations cannot be created or edited for Public Cloud provision types.

Plans

Plans types include Service Plans and Price Plans. Service Plans determine the memory, storage and cores configuration during provisioning and reconfigure.

Service Plans

A set of Service Plans are seeded by Morpheus for private cloud provision types as well as some public clouds. Additional Plans and Prices are synced in for other public clouds when the corresponding cloud types are added to an appliance. Users can create new Service Plans or edit the system-seeded Service Plans for Private Cloud types. Service Plans configurations cannot be created or edited for Public Cloud provision types.

Service Plan Configuration

Note: Not all fields listed below are available for every provision type. After selecting the provision type, the correct fields for that type of Service Plan will be revealed. Not all fields are required to save a valid Service Plan

- **NAME:** The name of the Service Plan in Morpheus
- **ACTIVE:** Inactive Service Plans are not available for selection during provisioning or reconfigure. New discovered records cannot be associated with deactivated Plans when converting to managed resources. Any resources attached to a Plan will continue to be associated if the Plan is later deactivated
- **CODE:** A unique identifier for use in Morpheus API and CLI

- **DISPLAY ORDER:** Configures the order in which plans are displayed relative to other plans associated with the same provision type. Note that Plans will be displayed in low-to-high order based on the Display Order property. This is reversed from Layouts which are displayed in high-to-low order
- **PROVISION TYPE:** Determines the resource Provision Type this Service Plan is available for when provisioning, reconfiguring and converting discovered resources to managed
- **REGION CODE:** (Optional) Limits availability of the Service Plan to Clouds with the specified Region Code
- **STORAGE:** The default storage size of the root volume (in MB or GB)
- **CUSTOMIZE ROOT VOLUME:** Allows the root volume size to be customized during provisioning or reconfigure. Custom Range limits, if set, will apply
- **CUSTOMIZE EXTRA VOLUMES:** Allows the extra volume sizes to be customized during provisioning or reconfigure. Custom Range limits, if set, will apply
- **ADD VOLUMES:** Allows additional volumes to be added during provisioning or reconfigure
- **MEMORY:** The amount of memory included with the plan (in MB or GB)
- **CUSTOM MEMORY:** Allows the amount of memory to be customized during provisioning or reconfigure. Custom Range limits, if set, will apply
- **CORE COUNT:** The number of virtual CPU cores included with the plan
- **CUSTOM CORES:** Allows the number of virtual CPU cores to be customized during provisioning or reconfigure. Custom Range limits, if set, will apply
- **CORES PER SOCKET:** Determines core distribution across sockets. CORES PER SOCKET cannot be larger than CORE COUNT, and CORE COUNT must be divisible by CORES PER SOCKET. For example four CORES with two CORES PER SOCKET means two sockets would have two cores each assigned. Four CORES with one CORE PER SOCKET would have four sockets with one core each assigned, and four CORES with four CORES PER SOCKET would have one socket with four cores assigned
- **TOTAL STORAGE:** When custom storage is enabled for the plan, this sets a minimum and maximum total storage allowed (all disks combined)
- **PER DISK SIZE:** When custom storage is enabled for the plan, this sets the minimum and maximum storage for each disk
- **CUSTOM MEMORY RANGE:** The minimum and maximum allowed amount of memory for the Plan when CUSTOM MEMORY is enabled for the Plan
- **CUSTOM CORES RANGE:** The minimum and maximum allowed amount of virtual CPU cores for the Plan when CUSTOM CORES is enabled for the Plan
- **SOCKETS:** The minimum and maximum allowed sockets range for the Plan when CUSTOM CORES is enabled for the Plan
- **CORES PER SOCKET:** The minimum and maximum allowed cores per socket for the Plan when CUSTOM CORES is enabled for the Plan
- **PRICE SETS:** In the Price Sets tab, associate Price Sets with the Plan. See [Add Price Set\(s\) to Plan](#)

Tip: Custom Range storage and memory values units (GB/MB) are inherited from the :STORAGE:: and :MEMORY:: GB/MB settings in the same Plan. For example, if :STORAGE: is configured for for 40 GB, a custom range for Storage would also be in GB.

Create Service Plan

Morpheus offers a few distinct types of service plans, each with different data fields to track and valid Price Set types which can be associated. For more on Service Plan configuration, see the preceding section.

Price Sets can be added to the Plan at creation time so often it makes sense to create the Prices and associate them with Price Sets before creating the Plan. Additional instructions for creating Prices and Price Sets are in the next section. With the Price Sets ready, continue with the instructions below to create Price Plans of various types.

1. Navigate to Administration > Plans & Pricing (/admin/service-plans)
2. Click the + *ADD* dropdown and select the appropriate Plan type
3. Configure details for the Plan on the General tab, the configuration options will depend on the Plan type. See the section above for a detailed description of each configuration option available for Service Plans
4. On the Price Sets tab, associate all relevant Price Sets with the Plan. The desired Price Sets must already exist. If needed, you may save the Plan at this point and come back to associate Price Sets later
5. Click *SAVE CHANGES*

Add Price Set(s) to Plan

1. Select a Price Unit to list available Price Sets which have the selected Price Unit. Available Price Sets are filtered to show only those which are relevant for the Plan type you've selected
2. Select a Price Set from the list and click + *ADD*
3. Add additional Price Sets if desired (multiple Price Sets can be associated with a Plan)
4. Added Price sets can be removed from a Plan by clicking the  icon
5. Once all Price Sets have been added, select *SAVE CHANGES*

Tip: The +*ADD* button must be clicked after selecting a Price Set or it will not be added to a Plan.

Service Plan Permissions

Group and Tenant Access permissions determine availability of a Service Plan.

- Group Access determines what Groups the Service Plan will be available in for Provisioning and Reconfigure.
- Group Access settings only apply to the Tenant they are configured in, as Groups are not multi-tenant.
- For multi-tenant environments, the Master Tenant can set Tenant Permissions to determine if the Service Plan is available to all Tenants (public visibility) or assign the Service Plan to one or multiple Tenants.

Removing Plans

Plans can be removed by navigating to the Service Plans list page (Administration > Plans & Pricing > Plans), opening the ACTIONS menu for a Plan, and then selecting “Remove”. Removing a Plan makes it no longer visible, however, it does not hard delete the Plan as the record must remain for any existing associations’ usage and price tracking. Synced Plans should be deactivated rather than removed.

Load Balancer Price Plans

Load Balancer Price Plans configure pricing for Load Balancers and Load Balancer Virtual Servers.

Load Balancer Price Plan Configuration:

- **NAME:** The name of the Load Balancer Price Plan in Morpheus
- **ACTIVE:** When Active, Prices in Price Sets added to the Price Plan will apply to associated resources as configured
- **CODE:** A unique identifier for use in Morpheus API and CLI
- **LOAD BALANCER TYPE:** Select the load balancer type the Price Plan should be associated with
- **PRICE SETS:** In the Price Sets tab, associate Price Sets with the Plan. Load Balancer Price Sets may be associated with Load Balancer-type and Load Balancer Virtual Server-type Prices. For more, see [Add Price Set\(s\) to Plan](#)

Virtual Image Price Plans

Virtual Image Price Plans configure pricing for Virtual Images. Pricing is associated per Price and Price Set configurations, but in general applies to Images with location records in the Price Plans specified Cloud type. Virtual Image Price Plans are for setting pricing for Virtual Image storage, not for adding pricing to resources that use Virtual Images associated with Virtual Image Price Plans. Virtual Image Price Plans do not apply to uploaded Virtual Images that have not been copied to a location in the specified Cloud type yet.

Virtual Image Price Plan Configuration:

- **NAME:** The name of the Virtual Image Price Plan in Morpheus
- **ACTIVE:** When Active, Prices in Price Sets added to the Price Plan will apply to associated resources as configured
- **CODE:** A unique identifier for use in Morpheus API and CLI
- **CLOUD TYPE:** Virtual Image Pricing applies to Virtual Images with location records in the specified Cloud type
- **PRICE SETS:** In the Price Sets tab, associate Price Sets with the Plan. Virtual Image Price Sets may be associated with Storage-type Prices. For more, see [Add Price Set\(s\) to Plan](#)

Snapshot Price Plans

Snapshot Price Plan Configuration:

- **NAME:** The name of the Snapshot Price Plan in Morpheus
- **ACTIVE:** When Active, Prices in Price Sets added to the Price Plan will apply to associated resources as configured
- **CODE:** A unique identifier for use in Morpheus API and CLI
- **CLOUD TYPE:** Snapshot Pricing applies to Snapshots with location records in the specified Cloud type
- **PRICE SETS:** In the Price Sets tab, associate Price Sets with the Plan. Snapshot Price Sets must have at least one Storage-type Price but may also have Datastore-type Prices. For more, see [Add Price Set\(s\) to Plan](#)

Price Sets

Price Sets combine Prices and then attach to Plans. Prices must be created prior to creating Price Sets, but it is recommended to review the Price Set Type options prior to creating Prices.

Price Unit Select the Price Unit to use for the Price Set.

- Minute
- Hour
- Day
- Month
- Year
- Two Year
- Three Year
- Four Year
- Five Year

Note: Only Prices configured with matching Price Units can be used in a Price Set. “Month” is equivalent to 30 days by default. For AWS, month is 30.5 days. For Azure, month is 30.4 days.

Type Price Set types determine which Prices are available to make up the set. This selection will filter the values returned in the Prices field at the bottom of the modal.

Note: It’s helpful to make note of the Prices options below before creating Price Sets.

- **Everything:** ‘Everything’ price sets require one or more ‘Everything’ price types and may include ‘Platform’ or ‘Software’ price types
- **Compute + Storage:** ‘Compute + Storage’ price sets require at least one of each ‘Memory’, ‘CPU’, and ‘Disk Only’ price types and may include ‘Platform’ or ‘Software’ price types
- **Component:** ‘Component’ price sets require at least one of each ‘Memory’, ‘Cores’, ‘CPU’, and ‘Storage’ price types and may include ‘Platform’ or ‘Software’ price types

- **Load Balancer:** ‘Load Balancer’ price sets require at least one ‘Load Balancer’ price type and may include ‘Load Balancer Virtual Server’ price types. Load Balancer price sets are the only type which can be associated with Load Balancer Price Plans
- **Virtual Image:** ‘Virtual Image’ price sets require at least one ‘Storage’ price type. Virtual Image price sets are the only type which can be associated with Virtual Image Price plans
- **Snapshot:** ‘Snapshot’ price sets require at least one ‘Storage’ price type and may also include ‘Datastore’ price types. Snapshot price sets are the only type which can be associated with Snapshot Price plans
- **Software/Service:** ‘Software/Service’ Price Sets require at least one ‘Software/Service’ Price type

Apply Price Changes to Usage If marked, when saving a Price Set (new Price Set or saving changes to an existing one), usage records will be restarted for servers affected by the pricing change.

Prices Search for and select Prices to be added to the Price Set. One of each Price Type required for the Price Set Type selected must be added for the Price Set to save.

Prices

Price Types

- Everything: One price for all resources Storage, CPU, Memory, and Disks
- Memory + CPU
- Memory Only (per MB)
- Cores Only (per core)
- Disk Only (per GB)
- Platform: Select from Windows, Linux (generic), or one of several specific Linux distributions
- Software/Service: Add prices for software licenses which may be included with your price sets
- Datastore (per GB)
- Load Balancer
- Load Balancer Virtual Server

Price Units

- Minute
- Hour
- Day
- Month
- Year
- Two Year
- Three Year
- Four Year
- Five Year

Currency

- AED
- ARS

- AUD
- BRL
- BWP
- CAD
- CHF
- CLF
- CLP
- DKK
- EUR
- GBP
- IDR
- ILS
- JOD
- JPY
- KRW
- MAD
- MNT
- MYR
- MXN
- NOK
- NZD
- PLN
- ROL
- RUB
- SAR
- SEK
- SGD
- THB
- TRL
- USD
- USN
- VND
- XAF
- XCD
- XOF
- XPF

- ZAR (South African Rand)

Cost The base cost of the resource(s). The Price will match the Cost unless a Price Adjustment is added.

Price Adjustment

- **None:** Default, no markup added and Price will match Cost
- **Fixed Markup:** A fixed amount added to the Cost. Price will equal Cost + Markup.
- **Percentage Markup:** Adds a percentage markup to Cost. Price equals $Cost + (Cost \times Markup \%)$
- **Custom Price:** Sets a Price independent from the Cost. If the Cost changes, a Custom Price will not.

Price A computed value of the final price including the cost plus any applicable markup.

Apply Price Changes to Usage If marked, when saving a Price Set (new Price Set or saving changes to an existing one), usage records will be restarted for servers affected by the pricing change.

5.9.4 Roles

Overview

Within Morpheus is a wide array of role based access control capabilities. These roles can be managed within the Admin > Roles section of the Morpheus UI as well as through the API or CLI. They are designed to be robust enough to fit within a wide array of enterprise and managed service provider scenarios so they can be a bit hard to grasp at first, but should make sense once a few simple concepts are explained. There are two types of roles within Morpheus called Tenant and User based roles. Both sets of roles allow restrictions to be imposed on a user at the feature access level. Entire sections within the appliance UI can be hidden based on the specified access levels for features within Morpheus. Features have different access scopes that can be selected from and can range depending on the specific feature. The most common scope set involves none, read, and full. Instance Type access is also common among both role types which allow the administrator to restrict which service catalog items they are allowed to provision within Morpheus .

There are several handy tricks for creating new roles within Morpheus and users can be assigned more than one role. When a user is assigned more than one role, permissions are granted by the role with the highest level of scope access. This allows roles to be built with small subsets of features and combined to grant different individuals relevant permission control. With resource permissions (that is, all types of permissions other than Feature permissions), a default access can be given as opposed to a specific (Full or None) permission for any resource. A specific permission will always supersede a default permission regardless of whether it's more permissive or more restrictive. In other cases (default vs default OR specific vs specific) the more permissive access will be given.

Note: Feature access control not only applies to the Morpheus UI but also applies to the public developer API. It is sometimes necessary to logout and back in for changes to a users feature access level to be respected.

Role Types

Tenant Roles

A Tenant based role (formerly called an Account based role) is used to ensure access control enforcement across an entire tenant with many sub-users. This allows the subtenant to manage their own set of internal user based roles without worrying master tenant involvement in setting them up. The master tenant is the only tenant able to create and manage these types of roles. When editing a Tenant, a singular tenant role can be assigned to the account. Users within the tenant can be assigned roles but those user based roles will never be able to supersede the level of access granted

by the tenant role. This allows a super administrator the ability to restrict access at the department or organization level without having to worry about per user access control within said tenant.

Tenant roles also have an additional section not in User based roles related to Cloud Access. Cloud Access allows the master tenant the ability to assign cloud integration resources to specific subtenants or groups of subtenants. An example would be granting access to a specific VMware cluster only to a subset of tenants using the tenant based role control.

User Roles

User roles can be created by any tenant given permission at the tenant role level. These allow tenants to manage their own sets of users and their levels of access. They also allow tenants to control which users have access to specific “Groups” for provisioning into within Morpheus. Groups are not cross tenant and therefore need to be controlled within the individual tenant in Morpheus.

Master tenant users are able to create a special type of user role called a multi-tenant user role. A multi-tenant user role is copied / duplicated down to all subtenants within Morpheus. These can be viewed as canned role templates available to new tenants when their account is first created. Any changes made to the main role are propagated down to the subtenants version of the shared role so long as the subtenant users have not previously adjusted/changed that role. The moment a subtenant makes adjustments to the shared role within their account, it is unlinked from the parent role and treated entirely independently. In order to relink the Role in the Subtenant, a Master Tenant user would have to edit the Role, uncheck MULTITENANT USER ROLE, save the Role, check MULTITENANT USER ROLE once again, then save the Role once again.

Another note about user roles is that when a user role is copied down to a subtenant, the permission scopes cannot supersede the tenant’s assigned tenant role. If they do they are automatically downgraded when propagated to the specific tenant. Any changes made to the tenant role will automatically ensure roles within the tenant are downgraded appropriately.

Note: Master Tenant administrators may edit permissions for Roles in other Tenants by viewing the Tenant detail page (Administration > Tenants > Selected Tenant) and accessing the Roles tab. From there, select the Role to edit and make changes on the resulting Role detail page.

Multi-Tenant User Role Lock

As discussed above, Multi-Tenant User Roles are made available within all Subtenants and can be thought of as ‘canned’ user role sets provided to the Subtenant. Master Tenant administrators can prevent changes to these ‘canned’ user roles by marking the box labeled ‘MULTITENANT LOCKED’ when creating or editing the role. In addition to preventing Subtenant administrators from modifying permissions of the role copied down to their Subtenant, this option also ensures Master Tenant administrators can propagate new changes to that role. Modification of the role by the subtenant (if allowed) breaks the link back to the Master Tenant and the copy of the role within the Subtenant will become its own unlinked role.

Note: Multi-tenant role lock applies only to permission sets on the Features, Report Types, Personas and VDI Pools tabs. Permissions in other tabs (such as Groups, Instance Types, Blueprints or Catalog Item Types) tabs are not locked. Similarly, changes made to the role on these tabs in the master tenant are not synced down.

Editing User Roles in other Tenants

Administrators in the Primary Tenant have the unique ability to edit feature permissions for User Roles that exist within other Tenants (Subtenants). In order to view the Roles within the Tenant, navigate to the Tenant detail page (Administration > Tenants) and select the Roles tab. Click the pencil () icon to the right of a Role in the list to edit basic information, such as the name and description of the Role. Click on the name of the Role to view its complete permission set and edit the permissions if desired. This will update the feature access rights of users in the selected Tenant which have the Role.

Roles and Identity Sources

It is very common for large Enterprises to have an existing identity source that they would like to plug in to Morpheus for authentication. This includes services like LDAP, Active Directory, OKTA, Jump Cloud, One Login, and SAML. When using these services it becomes important to configure a role mapping between the Morpheus role assignments to the equivalent identity source groups/roles the user belongs to. This is configurable within the identity source management UI. Sections are provided allowing things like LDAP groups to be directly mapped to specific roles within Morpheus. If a user matches more than one LDAP/role group then both sets of roles are applied to the user automatically. Configuring Identity Sources is done in Tenant management or user management in Administration > Tenants or Administration > Users, and has to be configured on a per-tenant basis. Additionally, administrators may opt to lock users to their mapped role in Morpheus or keep the roles unlocked to manually administer roles in one-off scenarios.

Role Permissions

Note: Permission options for sub-tenant user roles will only list options permitted by the Tenant role applied to the sub-tenant. Sub-Tenant user roles permissions cannot exceed permissions set by the overriding Tenant Role.

User Role Permission Sections

Features Controls User access level for UI sections and features in Morpheus. The complete feature permissions grid is included below.

Groups Controls User access level for Groups. Groups are not a Multi-Tenant construct, only Groups created in the current Tenant will be visible.

Instance Types Controls User access level for Instance Types. Only Instance Types created in the current Tenant or those created in the Master Tenant and shared with the current Tenant will be available.

Blueprints Controls User access level for Blueprints during App provisioning. Only Blueprints created in the current Tenant or those created in the Master Tenant and shared with the current Tenant will be available.

Report Types Controls User access for each report type in the Reports section (Operations > Reports). The user must also have Operations: Reports access granted under the Feature permissions tab.

Personas Controls User access to Morpheus Personas, at the time of this writing Users may be given access to the Standard (full Morpheus experience), Virtual Desktop (VDI), or Service Catalog Personas

Catalog Item Types Controls User access to Catalog Item types within the Service Catalog Persona. Only Catalog Items created in the current Tenant or those created in the Master Tenant and shared with the current Tenant will be available.

Cluster Types Controls user access to Cluster types. Only Cluster types allowed for the Role may be added in Infrastructure > Clusters (assuming the Role also has feature access to applicable permissions related to adding Clusters)

VDI Pools Controls User access to VDI Pools which are currently configured (Tools > VDI Pools) via the Virtual Desktops Persona view

Workflows Controls User access to configured Workflows, including the ability to view, edit, execute, or apply to Library item configurations

Tasks Controls User access to configured Tasks, including the ability to view, edit, execute, or apply to Workflows

Tenant Role Permission Sections

Features Controls Tenant access level for sections and features in Morpheus. The complete feature permissions grid is included below.

Clouds Controls Tenant access level for Clouds. This list includes Clouds integrated from the Master Tenant and shared publicly. Tenants given this Tenant Role will have either Full, Read, or None access levels to a given Cloud. See the section below for more information on Cloud Access levels.

Instance Types Controls Tenant access level for Instance Types. Only Instance Types created in the current Tenant or those created in the Master Tenant and shared with the current Tenant will be available.

Blueprints Controls Tenant access level for Blueprints during App provisioning. Only Blueprints created in the current Tenant or those created in the Master Tenant and shared with the current Tenant will be available.

Report Types Controls Tenant access for each report type in the Reports section (Operations > Reports). The Tenant must also have Operations: Reports access granted under the Feature permissions tab.

Personas Controls Tenant access to Morpheus Personas, at the time of this writing Users may be given access to the Standard (full Morpheus experience) or Service Catalog Personas

Catalog Item Types Controls Tenant access to Catalog Item types within the Service Catalog Persona. Only Catalog Items created in the current Tenant or those created in the Master Tenant and shared with the current Tenant will be available.

Cluster Types Controls Tenant access to Cluster types. Only Cluster types allowed for the Role may be added in Infrastructure > Clusters within the associated Tenant

VDI Pools Controls Tenant access to VDI Pools which are currently configured (Tools > VDI Pools) via the Virtual Desktops Persona view

Workflows Controls Tenant access to Workflows. Only selected Workflows will be shared down to Tenants having the associated Role

Tasks Controls Tenant access to Tasks. Only selected Tasks will be shared down to Tenants having the associated Role

Cloud Access Levels

When creating or editing a Tenant Role, Master Tenant administrators can choose which publicly-visible Clouds to share with Tenants having the current Role. Access can be completely restricted or administrators can choose between Read and Full access.

Read Access

Master Tenant administrators can opt to give other Tenants read-level access to any integrated Clouds through the Tenant Role. A read-only Cloud allows the Master Tenant to assign resources for viewing by Subtenant users but not allow them to perform any actions or provision new Instances.

With read-level access, Subtenant users will have access to the Cloud detail page (Infrastructure > Clouds). From the Summary subtab on the Cloud detail page, high-level information on Cloud resources are shown regardless of specific resources that have been shared with the Subtenant. Other metrics, such as costing, resource percentages (CPU, Memory, and Storage), VM counts and host counts will only be populated when servers in the Cloud have been assigned to the Tenant.

Full Access

With full access, Subtenant users also have access to the Cloud detail page (Infrastructure > Clouds > Specific Cloud) and see the same level of detail as Subtenants with read-only rights. However, with full access, Subtenant users can also perform many actions including the addition of Clusters, Hosts, and VMs, changing networks, and more. This cloud will also be selectable as a provisioning target for Subtenant users when deploying Instances or Apps.

Feature Access Permissions

Feature Access settings control permissions for sections and objects in Morpheus. Permission options include:

None Hidden or inaccessible for user

Read User can view but cannot edit or create

Full User has full access

User User can access Objects they have created or own

Group User can access Objects assigned to or shared with Groups the User has access to

Remote Console: Provisioned Remote Console tab will only appear after instance is successfully provisioned.

Remote Console: Auto Login RDP and SSH only, controls if user is auto-logged in to Remote Console or presented with login prompt.

Role Mappings Gives User Access to Role Mappings config in `/admin/roles` for configuring Identity Source Role Mappings without providing Access to other Identity Source configuration settings.

- **Admin Permission Options**

Per- mission Name	Per- mission Options	Feature Access	Description	Recommendations	Tenant Role Recommendations
Admin: Ansible	None, Full	Allows or disallows the ability to edit existing Ansible integrations	Ansible integrations are shown on the Integrations list page (Administration > Integrations). Users with access may view and edit them here.	This permission is recommended for those responsible for administering Morpheus, including creating integrations with third-party technologies, specifically Ansible	
Admin: Appliance Settings	None, Full	Allows or disallows access to the Appliance and License tabs in Administration > Settings	The Appliance tab in Administration > Settings is where Morpheus administrators would configure the appliance URL, Tenant and User management, email, proxy, and currency settings. Additionally, defining which Clouds are available for integration within Morpheus is done on this page. On the License tab information about the current Morpheus license may be viewed and a new license may be applied when needed.	This permission is recommended to only be assigned to Roles utilized within the Master Tenant. Those responsible for configuring currency, email, and proxy settings for Cloud API access will need this permission.	This permission is recommended to be set to None on the Tenant Role to restrict this access for all Subtenant Users.
Admin: Backup Settings	None, Full	Allows or disallows access to Administration > Settings > Backups. Master Tenant administrators have additional settings for appliance backups and defaults on this page.	The Backup Settings page is where users define the default Morpheus backup bucket, backup schedule, and retention count. Additionally, if given to a Master Tenant user they will have the ability to enable scheduled backups, create backups, and backup appliance.	This permission is recommended for those responsible for enabling backups and setting default backup buckets within Morpheus.	
Admin: Clients	None, Full	Allows or disallows access to the Clients tab in global settings (Administration > Settings)	The Clients settings section is where API clients are created and edited. Default clients may have their validity and refresh periods edited but cannot be deleted. User-created API clients may be edited or deleted	This permission is recommended for those responsible for administering API access.	
Admin: Distributed Workers	None, Full	Allows or disallows access to Administration > Integrations > Distributed Workers Tab			
Admin: Environments	None, Full	Allows or disallows access to the Environments tab in Administration > Settings > Provisioning. When given	The Environments tab is where named environments such as development or production are created and given a description as well as a code for use within the API. A	This permission is recommended for those responsible for defining envi-	

- **API Permission Options**

Per- mis- sion Name	Per- mis- sion Op- tions	Feature Ac- cess	Description	Recommendations	Tenant Role Recommendations
API: Billing	None, Read, Full	Allows or disallows access to invoices and projects via Morpheus API/CLI.	The invoices API/CLI is used to generate bills and gather highly-granular costing data for supported Clouds. Read access allows list and get functions and Full allows access to post (refresh).	This permission is recommended for those responsible for generating invoices or projects.	It is recommended this setting be set to None on the Tenant Role to restrict access for Subtenant users.
API: Ex- ecu- tion Re- quest	None, Full	Allows or disallows access to an API endpoint.	This endpoint allows users to execute scripts on Instances, containers, or hosts and then polls for a response.	This permission is recommended for those responsible for arbitrary API script execution.	It is recommended this setting be set to None on the Tenant Role to restrict access for Subtenant users.

- **Backups Permission Options**

Per- mis- sion Name	Per- mis- sion Op- tions	Feature Access	Description	Rec- om- men- da- tions	Ten- ant Role Rec- om- men- da- tions
Back- ups	None, View, Read, User, Full	Determines access to the Backups section of Morpheus UI, including the Summary, Jobs, Backups, and History subpages. The "User" permission allows access only to backup objects the user owns.	The Summary subpage allows the user to see the number of configured backups, the success rate, recent failures, and the size of the backups, as well as, the upcoming and in-progress backups. The Jobs subpage is where backup jobs may be created, cloned, edited or deleted. On create, a name, code (for use within the API), retention count, and schedule are selected (Note: Selectable schedules are defined Execution Schedules which are created in the Library > Automation). On the backups subpage, a list of configured backups is provided and new backups may be created or on-demand backups may be executed. On create, the place where the target exists is selected (Instance, Host, or Provider), the source is selected and a name is defined as well as the selected execution schedule. On the History subpage both the backups and restores tabs are available. Names, statuses, start times, durations and size may be viewed.	This permission is recommended for those responsible for performing the backup and restoration of workloads.	
Back- ups: In- te- gra- tions	None, Read, Full	Determines access to the Backups > Integrations page.	From this page, backup integrations may be created, edited, or deleted. The page also provides the status of existing integrations. On create the integration product is selected and all associated connection and authentication information must be provided. Additionally, visibility is set to either public or private. Integrations available include Avamar, Commvault, Rubrik, Veeam, and Zerto.	This permission is recommended for those responsible for the integration between Morpheus and backup technologies.	It is recommended this setting be set to None on the Tenant Role to restrict access for Sub-tenant users.

• Catalog Permission Options

Permission Name	Permission Options	Feature Access	Description	Recommendations	Tenant Role Recommendations
Catalog (Formerly Service Catalog: Catalog)	None, Full	Determines access to Provisioning > Catalog and Catalog in the Service Catalog Persona view	The Catalog page displays the complete list of Catalog Items that can be ordered from the Service Catalog	This permission is recommended for users who will order items from the Service Catalog	
Catalog: Dashboard (Formerly Service Catalog: Dashboard)	None, Read	Determines access to Pro-CatDasJ and Dashboard in Service Catalog Persona view	The Catalog Dashboard contains featured Catalog Items, recently-ordered Catalog items and Inventory items. The Catalog Dashboard is the default landing page for the Service Catalog Persona view	This permission is recommended for users who will use the Service Catalog	
Catalog: Inventory (Formerly Service Catalog: Inventory)	None, Full	Determines access to Pro-CatDasJ and Dashboard in Service Catalog Persona view	The Inventory is the complete list of user-owned items provisioned from the Service Catalog	This permission is recommended for users who will use the Service Catalog and need to be able to view details on the items they've provisioned from the Catalog	

- **Infrastructure Permission Options**

Per- mission Name	Per- mission Options	Feature Access	Description	Recom- menda- tions	Ten- ant Role Rec- om- men- da- tions
In- fras- truc- ture: Boot	None, Read, Full	Determines access to the Integrations > Boot page, including the Mapping, Boot Menus, Answer Files, Images, and Discovered MAC Addresses tabs.	Morpheus includes a PXE Server to provide for rapid bare metal provisioning. The Boot page is where users may add, edit, or delete answer files, as well as, manage their own images or use existing ones. Boot menus and mappings are also managed here and discovered MAC addresses are displayed.	This per- mission is recom- mend for those responsible for bare metal provi- sioning.	
In- fras- truc- ture: Cer- tifi- cates	None, Read, Full	Determines access to the SSL Certificates tab on the Infrastructure > Keys & Certs page.	The SSL Certificates page is where certificates may be uploaded and managed. These certificates may then be used within Morpheus when orchestrating load balancers.	This per- mission is recom- mended for person- nel who will be or- ches- trating and provision- ing load balancers.	
In- fras- truc- ture: Clouds	None, Read, Group, Full	Determines access to the Infrastructure > Clouds page. The “Group” permission limits the Cloud list page (Infrastructure > Clouds) to show only Clouds in their assigned Groups.	The Cloud page is where new Clouds are inte- grated with Morpheus and existing Cloud integra- tions are managed. This includes creating a code for use within the API, the location, visibility, ten- ant, whether or not it should be enabled, and if VMs should be automatically powered on. Additionally, Clouds may be integrated from the Clouds tab of a Group detail page.	This per- mission is recom- mended for those responsible for con- figuring RBAC as well as those respon- sible for Morpheus Cloud Inte- grations.	
In- fras- truc- ture: Clus- ters	None, Read, Group, Full	Determines access to the Infrastructure > Clusters page.	The Clusters page allows you to create and manage Kubernetes, Docker, and KVM Clusters, as well as Cloud-specific Kubernetes services such as EKS.	This per- mission is recom- mend for those creating and managing containers or container	

In- fras- truc- ture:	None, Read, Full	Determines access to the Infrastructure	The Hosts page provides for viewing and manag- ing hosts, virtual machines, and bare metal hosts. On the bare metal hosts page, hosts may come from PXE, or be discovered with the OpenStack	This per- mission is recom- mend	
--------------------------------	------------------------	---	---	---	--

- **Library Permission Options**

Permission Name	Permission Options	Feature Access	Description	Recommendations	Tenant Role Recommendations
Library: App Blueprints (Formerly Provisioning: Blueprints)	None, Read, Full	Determines access to the Library > Blueprints > App Blueprints page.	The Blueprints page allows for the creation of pre-configured, multi-tier application definitions which can be deployed via the Apps page. With this permission the blueprint type of Morpheus is available.	This permission is recommended for those responsible for defining Morpheus-type Blueprints.	
Library: Blueprints - ARM (Formerly Provisioning: Blueprints - ARM)	None, Provision, Full	Determines access to ARM-type Blueprints on the Library > Blueprints > App Blueprints page. The “Provision” permission allows for provisioning Apps from ARM Blueprints without the ability to create or edit them.	The Blueprints page allows for the creation of pre-configured, multi-tier application definitions which can be deployed via the Apps page. With this permission the blueprint type of ARM is available.	This permission is recommended for those responsible for defining ARM blueprints.	
Library: Blueprints - CloudFormation (Formerly Provisioning: Blueprints - CloudFormation)	None, Provision, Full	Determines access to CloudFormation-type Blueprints on the Library > Blueprints > App Blueprints page. The “Provision” permission allows for provisioning Apps from CloudFormation Blueprints without the ability to create or edit them.	The Blueprints page allows for the creation of pre-configured, multi-tier application definitions which can be deployed via the Apps page. With this permission the blueprint type of CloudFormation is available.	This permission is recommended for those responsible for defining CloudFormation blueprints.	
Library: Blueprints - Helm (Formerly Provisioning: Blueprints - Helm)	None, Provision, Full	Determines access to Helm-type Blueprints on the Library > Blueprints > App Blueprints page. The “Provision” permission allows for provisioning Apps from Helm Blueprints without the ability to create or edit them.	The Blueprints page allows for the creation of pre-configured, multi-tier application definitions which can be deployed via the Apps page. With this permission the blueprint type of Helm is available.	This permission is recommended for those responsible for defining Helm blueprints.	
Library: Blueprints - Kubernetes (Formerly Provisioning: Blueprints - Kubernetes)	None, Provision, Full	Determines access to Kubernetes-type Blueprints on the Library > Blueprints > App Blueprints page. The “Provision” permission allows for provisioning Apps from Kubernetes Blueprints without the ability to create or edit them.	The Blueprints page allows for the creation of pre-configured, multi-tier application definitions which can be deployed via the Apps page. With this permission the blueprint type of Kubernetes is available.	This permission is recommended for those responsible for defining Kubernetes blueprints.	
568					Chapter 5. Worker

- **Lifecycle Permission Options**

Per- mis- sion Name	Per- mis- sion Op- tions	Feature Access	Description	Recommendations	Ten- ant Role Rec- om- men- da- tions
En- vi- ron- ment Vari- ables	None, User, Read, Full	Allows access to the En- vironments tab of the In- stance detail page	Allows Instance environment variables to be edited. If set to “User” level only environment variables of Instances owned by the currently logged in user may be edited.	This permission is rec- ommended for those needing management control over Instances	
Life- cy- cle: Ex- tend Ex- pi- ra- tions	None, User, Full	Determines if the user can extend an expiration or shutdown Policy	Allows the user to extend au- tomated shutdown or expira- tion policies set against any workload (“full” permission) or against the user’s own work- loads (“user” permission)	This permission is recommended for administrators or those who need to be able to extend Policies set against their own workloads (“user” level permission)	
Power Con- trol	None, User, Full	Allows access to power state controls for Instances and servers, including stopping, starting, restart- ing and suspending.	Allows the user to change the current power state of In- stances and servers	This permission is rec- ommended for those needing management control over Instances	
Re- con- fig- ure	None, User, Full	Allows general access to Instance and server recon- figure (resize) feature. See additional reconfigure per- missions below for more granular control over spe- cific reconfigure function- ality.	Allows general access to reconfigure features for In- stances and servers. “User” level permission allows only Instances and servers owned by the currently logged in user to be reconfigured.	This permission is rec- ommended for those needing management control over Instances	
Re- con- fig- ure: Change Plan	None, User, Full	Allows the user to change the Instance service plan	When reconfiguring, the user may change the service plan associated with the Instance. “User” level permission al- lows only Instances owned by the currently logged in user to have their plans changed.	This permission is rec- ommended for those needing management control over Instances	
Re- con- fig- ure: Disk Add	None, User, Full	Allows the user to add disks to an Instance or server during reconfigure.	When reconfiguring, the user may add disks to the selected Instance or server. “User” level permission allows only Instances owned by the cur- rently logged in user to have their disks changed.	This permission is rec- ommended for those needing management control over Instances	
Re- con- fig- ure: Disk	None, User, Full	Allows the user to change the datastore or volume type during reconfigure.	When reconfiguring, the user may update datastore or vol- ume types. “User” level per- mission allows only Instances owned by the currently logged	This permission is rec- ommended for those needing management control over Instances	
570	Change Type		in user to have their disk types changed.	Chapter 5. Worker	
	Re- con-	None, User,	Allows the user to modify an attached disk during re-	This permission is rec- ommended for those	

- **Monitoring Permission Options**

Per- mis- sion Name	Per- mis- sion Op- tions	Feature Access	Description	Recom- menda- tions	Tenant Role Recommen- dations
Mon- i- tor- ing	None, Read, User, Full	Determines level of access to the Monitoring section of Morpheus UI, including the Status, Apps, Checks, Groups, Incidents, Contacts, and Alert Rules subpages. The “User” permission will allow access only to objects the user owns.	The Checks page is where automatically-created checks are customized or new checks are created. The Groups and Apps pages are where checks may be grouped. The Incidents page is where incidents are created upon Check failure. The Contacts page is where contacts may be added for notifications. The Alert Rules page is where notification are configured.	This permission is recommended for those responsible for monitoring applications, incidents, or configuring notifications.	
Mon- i- tor- ing: Logs (Formerly Logs)	None, Read, User, Full	Determines level of access to the Logs section of Morpheus UI. The “User” permission will allow access only to objects the user owns.	Monitoring > Logs is where Instance and Server logs may be viewed (also must be enabled in order to view Appliance logs from Administration > Health > Morpheus Logs Logs when health permission is also enabled).	This permission is recommended for those who should have access to Instance and Server logs.	Setting permission to Full on the Tenant Role will give Subtenant users full access to all logs appliance-wide, including to workloads living in other Tenants, for any Subtenant users who also have Full permission on their User Role. It’s recommended that you set this permission to User on the Tenant Role so that Subtenant users are not able to see logs for workloads other than their own.

- **Networks Permission Options**

Per- mis- sion Name	Per- mis- sion Op- tions	Feature Access	Description	Recommen- dations	Ten- ant Role Recom- men- da- tions
Net- works: Read, DHCP Re- lays	None, Full	Determines access to the DHCP Relays in applicable network integrations	Allows DHCP Relays to be viewed, created and managed	This permission is recommended for those tasked with network management	
Net- works: Read, DHCP Servers	None, Full	Determines access to the DHCP Servers in applicable network integrations	Allows DHCP Servers to be viewed, created and managed	This permission is recommended for those tasked with network management	
Net- works: Read, Do- mains	None, Full	Determines access to the Domains tab on the Infrastructure > Network page.	The Domains page is where network domains are managed. Domains are used for setting FQDNs, joining Windows Instances to domains, and creating A-Records with DNS integrations. On create the domain controller and credentials for domain join must be provided.	This permission is recommended for those responsible for Morpheus DNS and domain-join integrations.	
Net- works: Read, Fire- walls	None, Man- age Rules, Full	Determines access to the Firewall tab on applicable network integrations detail pages. When the “Manage Rules” permission is given, users have read-only access to firewall groups and the ability to create and manage firewall rules on those groups	The Firewall tab is where network firewall groups and rules are viewed, created and managed	This permission is recommended for those tasked with network security management	
Net- works: Read, Float- ing IPs	None, Full	Determines access to the Floating IPs tab on the Network list page (Infrastructure > Network)	The Floating IPs tab is where Floating IPs from supported Clouds are listed once synced into Morpheus Users may also release unattached Floating IPs from this page and link through to any workloads which have Floating IPs attached	This permission is recommended for those tasked with network management	
Net- works: Read, IP Pools	None, Full	Determines access to the IP Pools tab on the Network list page (Infrastructure > Network)	The IP Pools tab is where IP pools from various networks are displayed. Detail pages for IP pools can also be accessed here	This permission is recommended for those tasked with IP address management	

Net- works: Read, In- te-	None, Full	Determines access to the Integrations tab on the Network list page (Infrastructure > Network)	The integrations tab is where network integrations can be viewed, added and managed. Additionally, the detail pages for network inte-	Chapter 5 Worker This permission is recommended for those	
---------------------------------------	---------------	---	---	---	--

- **Operations Permission Options**

Per- mis- sion Name	Per- mis- sion Op- tions	Feature Access	Description	Recommendations	Ten- ant Role Rec- om- men- da- tions
Op- er- a- tions: Ac- tiv- ity	None, Read	Deter- mines access to the Ac- tivity and History tabs on the Oper- ations > Activity page.	The Activity page displays four types of recent activities: Provisioning, Alerts, Backups, and Permissions.	This permission is recommended for those responsible to monitor or view activities and their statuses within Morpheus.	
Op- er- a- tions: Alarms	None, Read, Full	Deter- mines access to the Alarms tab in the Activity section (Opera- tions > Health)	The Alarms tab is where alarms are listed and acknowledgement actions can be taken against them	This permission is recommended for those responsible for monitoring	
Op- er- a- tions: An- a- lyt- ics	None, Read, Full	Deter- mines access to the Oper- ations > Analytics page.	The Analytics page gives administrators the ability to break down costs and usage, then filter the results by relevant delineations including Groups, Clouds, Tenants or even tag values.	This permission is recommended for those responsible for understanding utilization and costs.	
Op- er- a- tions: Ap- provals	None, Read, Full	Deter- mines access to the Oper- ations > Approvals page.	When a Provision Approval-type Policy is enabled for a Group or Cloud, an approval request will be created on each relevant provision attempt. These approvals can be handled directly in Morpheus or dealt with in ServiceNow with a properly-configured integration.	This permission is recommended for those responsible for approving, denying, or canceling approval requests.	
Op- er- a- tions: Bud- gets	None, Read, Full	Deter- mines access to the Oper- ations > Budgets page.	The Budgets page is where budgets are created and applied to clouds, tenants, users, or groups.	This permission is recommended for those responsible for managing budgets.	
Op- er- a- tions: Dash- board	None, Read	Deter- mines access to the Oper- ations > Dash- board	The Dashboard page is a single pane of glass showing quick, easy-to-read performance and configuration information about the Morpheus environment.	“Read” permission is recommended for all users. When set to None, Chapter 5 > Worker Reports becomes the default landing page and attempts to go to the Dashboard page will fail.	

- **Projects Permission Options**

Per- mis- sion Name	Per- mis- sion Op- tions	Feature Ac- cess	Description	Recommendations	Tenant Role Recom- menda- tions
Projects	None, Read, Full	Determines ac- cess to Projects through Mor- pheus API	Projects are used to as- sociate resources together and apply common tags to their invoices	This permission is recom- mended for those responsi- ble for cost analysis and in- voice reporting	

- **Provisioning Permission Options**

Per- mis- sion Name	Per- mis- sion Op- tions	Feature Access	Description	Recommendations	Tenant Role Recom- men- da- tions
Pro- vi- sion- ing: Ad- min- is- tra- tor	None, Full	When editing an Instance (Provisioning > Instances > selected Instance > EDIT button), this permission determines access to changing the owner of an Instance.	Allows you to change the owning user of an Instance.	This permission is recommended for those responsible to ensure all instances are owned by appropriate personnel.	
Pro- vi- sion- ing: Ad- vanced Node Type Op- tions	None, Full	This allows or disallows access to the “Extra Options” field of the Node Types tab on the Library page when the Node Type Technology value is set to “VMware”.	When VMware technology type is selected for a new or existing Node Type (Library > Blueprints > Node Types), the “Extra Options” field will be available in the VMware VM Options section. These allow defining advanced vmx-file parameters during provisioning.	This permission is recommended for those responsible for managing VMware Node Types (images).	
Pro- vi- sion- ing: Apps	None, Read, User, Full	Determines access to the Provisioning > Apps page. The “User” permission will allow access to only object the user owns.	The Apps page allows Instances to be grouped and tiered logically into Apps. From this page, Apps can be deployed from existing Blueprints and Instances can be added to existing Apps. Security groups and environmental variables (Linux Only) may be added and edited. The App log, history, and monitoring tabs may be viewed.	This permission is recommended for those responsible for provisioning.	
Pro- vi- sion- ing: Code De- ploy- ments	None, Read, Full	Determines access to the Deployments tab on the Provisioning > Code page.	The Deployments page provides the ability to use git, fetch from a url, or upload a file to be utilized during the provisioning of an Instance or pushed to an existing Instance.	This permission is recommended for those responsible for providing and managing software.	
Pro- vi- sion- ing: Code Inte- gra- tions	None, Read, Full	Determines access to the Integrations tab on the Provisioning > Code page.	From this page code integrations may be created, edited, or deleted. Integrations available include Git, Github, and Jenkins.	This permission is recommended for those responsible for the integration between Morpheus and code repositories and services.	
Pro- vi- sion- ing: Code Repos- itories	None, List Files, Read, Full	Determines access to the Deployments tab on the Provisioning > Code page.	The Code Repositories contains the repositories integrated with Morpheus allowing users to browse repositories folders and files and view file contents from any branch, trigger a refresh, and create tasks,	This permission is recommended for those responsible for providing and managing software.	

- **Security Permission Options**

Per- mis- sion Name	Per- mis- sion Op- tions	Feature Access	Description	Recommendations	Tenant Role Recommendations
Security: Scanning	None, Read, Full	Determines access to the Security Packages tab on the Jobs list page (Provisioning > Jobs), Security Scanning type Jobs, and Security Subtab inside the Software tab on a server detail page where the results of security scans are viewed	Allows access to view, create, and run security scans on existing systems, as well as view the results of previously-run scans	This permission is recommended for those responsible for security compliance of existing systems	

- **Snapshots Permission Options**

Per- mis- sion Name	Per- mis- sion Op- tions	Feature Access	Description	Recommendations	Tenant Role Recommendations
Snapshots	None, Read, Full	Determines access to the “Create Snapshot” function in the Actions menu on an Instance detail page (Provisioning > Instances > selected Instance).	If utilizing a VMware Cloud, the ability to create snapshots is available on the Instance detail page (Provisioning > Instances > selected Instance).	This permission is recommended for Instance owners who should be allowed to take snapshots.	
Snapshots: Linked Clone	None, Full	For VMware Cloud Instances, this controls access to the ability to create linked clone snapshots on the Instance detail page			

- **Tools Permission Options**

Per- mis- sion Name	Per- mis- sion Op- tions	Feature Access	Description	Recom- menda- tions	Tenant Role Recom- menda- tions
Tools: Archive	None, Read, Full	Determines access to the Tools > Archives page.	Archives provides a way to store files and make them available for download by scripts and users. Archives are organized by buckets. Each bucket has a unique name that is used to identify it in URLs and Scripts.	This permission is recommended for those responsible for storage or scripts which will use the Archive.	
Tools: Cypher	None, Read, User, Full, Full De- crypt	Determines access to the Tools > Cypher page. The “User” permission will allow access only to objects the user owns. The “Full Decrypt” permission will allow for decryption of secrets.	Secure key/value store. Cypher keys can be used in scripts.	Recommended for those who need to store or use security key value pairs.	
Tools: Image Builder	None, Read, Full	Determines access to the Tools > Image Builder page, Image Builds, Boot Scripts, and Preseed Scripts tabs.	The Morpheus Image Builder tool creates vmdk, qcow2, vhd and raw images. The Image Builder creates a blank VM in VMware, attaches an OS ISO, executes a boot script on the VM at startup via VNC, which calls a preseed script that runs the unattended OS installation and configuration. Morpheus then executes an OVA export of the completed vmdk to the target storage provider and converts the image to all other specified formats.	Recommended for those who are responsible for image creation.	
Tools: Kuber- netes	None, Read, User, Full	Allows for the management of Kubernetes clusters via the API (may be deprecated in the near future).	Allows for the management of Kubernetes clusters via the API	This permission is recommended for those who need to manage Kubernetes clusters via the API.	It is recommended this permission is set to None on the Tenant Role to restrict access for Subtenant users.

• Virtual Desktop Permission Options

Per- mis- sion Name	Per- mis- sion Op- tions	Feature Ac- cess	Description	Recommen- dations	Ten- ant Role Rec- om- men- da- tions
Vir- tual Desk- top: Copy/Paste	None, Full	Allows copy and paste access from the virtual desktop terminal	Enables the user to copy and paste values from a virtual desktop session into the paste buffer of their local computer		
Vir- tual Desk- top: Local Printer	None, Full	Enables printing from a virtual desktop session to a locally installed printer			
Tools: VDI Pools	None, Read, Full	Allows for the management of virtual desktop (VDI) pools.	Enables the user to access the VDI Pools section (TooVDI) and view existing pools (with “read” permission) or create and edit pools (with “full” permission). Related API functions are also granted with this feature permission.	This permission is recommended for those needing to manage VDI pools	

Creating Roles

User Roles

User Roles can be single or multitenant. A Multitenant User Role is automatically copied into all existing subtenants as well as placed into a subtenant when created. Useful for providing a set of predefined roles a Customer can use. The Multitenant Locked option prevent subtenant from modifying FEATURE ACCESS settings in the Role. Note Group, Instance Type and Blueprint Access settings will still be editable as Groups are unique per Tenant, and Instance and Blueprints can be a mix of unique and shared items.

Important: Multitenant Roles still need to be configured/managed by each subtenant, as Groups are unique per Tenant, and Instance and Blueprints can be a mix of unique and shared items.

Note: User Roles cannot exceed Tenant Role permissions. If a Multitenant User Role has higher permissions than the Tenant Role assigned to a subtenant, the Multitenant User Role permissions in that Tenant will automatically be reduced to match the Tenant Role permissions.

Create a Single Tenant User Role

1. In the Master Account, navigate to Administration > Roles
2. Select + *CREATE ROLE*
3. Enter a name for the Role and optional Description
4. For TYPE, select “User Role”
5. Leave the “Multi-tenant Role” checkbox blank.
6. Optionally select an existing Role to copy in the COPY FROM ROLE dropdown. * This will configure the new Role with the same configuration as the selected role to copy. A new role that is not copied from another role will be generated with all permissions set to NONE.
7. Select *SAVE CHANGES*

After saving the Role will be created, and you will be redirected to the Roles Permissions settings.

Create a MultiTenant User Role

A Multitenant User Role is automatically copied into all existing subtenants as well as placed into a subtenant when created. Useful for providing a set of predefined roles a Customer can use. The Multitenant Locked option prevent subtenant from modifying FEATURE ACCESS settings in the Role. Note Group, Instance Type and Blueprint Access settings will still be editable as Groups are unique per Tenant, and Instance and Blueprints can be a mix of unique and shared items.

1. In the Master Account, navigate to Administration > Roles
2. Select + *CREATE ROLE*
3. Enter a name for the Role and optional Description
4. For TYPE, select “User Role”
5. Optionally select an existing Role to copy in the COPY FROM ROLE dropdown. * This will configure the new Role with the same configuration as the selected role to copy. A new role that is not copied from another role will be generated with all permissions set to NONE.
6. Select the MULTITENANT ROLE checkbox
7. Optionally select the MULTITENANT LOCKED checkbox * When enabled, the FEATURE ACCESS settings in the Role will not be editable by subtenants. Group, Instance Type and Blueprint Access settings will still be editable as Groups are unique per Tenant, and Instance and Blueprints can be a mix of unique and shared items.
8. Select *SAVE CHANGES*

After saving the Role will be created, and you will be redirected to the Roles Permissions settings.

Important: Multitenant Roles still need to be configured/managed by each subtenant, as Groups are unique per Tenant, and Instance and Blueprints can be a mix of unique and shared items.

Note: User Roles cannot exceed Tenant Role permissions. If a Multitenant User Role has higher permissions than the Tenant Role assigned to a subtenant, the Multitenant User Role permissions in that Tenant will automatically be reduced to match the Tenant Role permissions.

Tenant Roles

A Tenant Role sets the highest possible permissions for a Tenant. User Roles within that Tenant cannot exceed those of the Tenants assigned Tenant Role. Tenant Roles can be assigned to single or multiple Tenants, and do not apply to the Master Account.

To create a Tenant Role:

1. In the Master Account, navigate to Administration > Roles
2. Select + *CREATE ROLE*
3. Enter a name for the Role and optional Description
4. For TYPE, select “Tenant Role”
5. Optionally select an existing Role to copy in the COPY FROM ROLE dropdown. * This will configure the new Role with the same configuration as the selected role to copy. A new role that is not copied from another role will be generated with all permissions set to NONE.
6. Select *SAVE CHANGES*

After saving, the Role will be created and you will be redirected to the Roles Permissions settings.

Setting an Optional Landing URL

On login, by default, Users will land on the Dashboard page (Operations > Dashboard). However, User and Tenant Roles may have custom landing URLs set. Setting a custom landing URL on the Tenant Role will apply globally to all Users within the Tenant. Landing URLs set on the User Role will supersede the landing URL set on the Tenant Role (if any). No Tenant Role is applied to the Master Tenant so the custom landing URL can only be set on the User Role for these Users.

To set a custom landing URL, create or edit a Role. Set the custom landing URL in the “LANDING URL” field. For example, set `/provisioning/instances` and the Users having that Role will land on the Instances list page each time they log in. Other paths may be used depending on the desired landing page.

5.9.5 Users & User Groups

Users

Overview

The Users page displays a list of all Users. The following fields are surfaced for each User:

- Tenant
- Display Name
- Username
- Email
- Role

Users which are grayed out in the list are currently inactive and cannot log in. From the Actions menu in each User row, the option is given to Impersonate the User, Edit, or Remove the User.

In Morpheus 4.2.1 and higher, click on the hyperlinked Display Name of the User to see a page detailing their effective Role permissions. This is especially useful for Users in multiple Roles where it might otherwise be difficult to determine their exact rights. This page looks identical to a User Role create/edit page except none of the fields are editable. Edit the User Role permissions for the User if changes need to be made.

Note: Some User data created through an Identity Source integration (such as Active Directory) is not editable in Morpheus, as it is synced from the Identity Source.

Create User

Users can be created from Administration > Users or Administration > Tenants > (selected Tenant) > Users tab`.

Note: Authorized Identity Source Users will be automatically created upon first sign in.

To create a User:

1. Navigate to either Administration > Users or Administration > Tenants > Select a Tenant`.
2. Select + *CREATE USER*.
3. From the New User Wizard input:

Username & Email

- First Name
- Last Name
- Username
- Email address

Receive Notifications Enable to receive Provisioning and Policy email notifications.

Roles Role(s) to be inherited by the user. If multiple roles are selected, the higher permission levels of one role will override the other role(s).

Password Password must contain at least one uppercase letter, one lowercase letter, a number, and a symbol.

Enabled If unchecked, the user will no longer be able to sign into Morpheus, but their user data will remain.

Password Expired If enabled, the User will be forced to create a new password upon next login. The expired password cannot be used again.

Linux Settings Creates a User with the supplied Username, Password and/or Key-pair on Linux Instances when “Create my User” is selected during provisioning, or a User Group is added to an Instance of which this Morpheus user is a member of.

Windows Settings Creates a User with the supplied Username, Password and/or Key-pair on Windows Instances when “Create my User” is selected during provisioning, or a User Group is added to an Instance of which this Morpheus user is a member of.

Important: Please ensure password entered is allowable by Windows.

Note: Instance Resource Limits for a user are now configured through *Policies*

1. Select *SAVE CHANGES*.

Edit User

User settings can be edited from Administration > Users, Administration > Tenants > Select a Tenant > Users tab`, or from *User Settings*.

Note: Some User data from Users created via an Identity Source Integration such as Active Directory is not editable in Morpheus, as it is synced with the Identity Source.

To edit a User from the Administration > Users Section:

1. Select the Administration link in the navigation bar.
2. Select the Users link in the sub navigation bar.
3. Click **ACTIONS** on the row of the user to edit.
4. Select **EDIT** in the ACTIONS dropdown.
5. Make changes.
6. Select *SAVE CHANGES*.

To edit a User from the Administration > Tenants > Select a Tenant > Users tab`:

1. Select the Administration link in the navigation bar.
2. Select the Tenants link in the sub navigation bar.
3. Select a Tenant
4. Click **ACTIONS** on the row of the user to edit.
5. Select **EDIT** in the ACTIONS dropdown.
6. Make changes.
7. Select *SAVE CHANGES*.

User Settings

Additional settings for a User can be found in the User Settings section, including:

- User Photo
- Default Group
- Default Cloud
- API Access

To access User Settings:

1. Select your name in the header
2. Select *User Settings*

To edit the User you are currently logged in as from *User Settings*:

1. Select your name in the header
2. Select *User Settings*
3. Make changes.
4. Select *SAVE*.

API Access

API and CLI Access Tokens can be regenerated from the *User Settings* section.

To regenerate a CLI or API Access Token:

1. Select your name in the header
2. Select *User Settings*.
3. Select *API ACCESS* under the *Windows Settings* section.
4. Select *ACTIONS* for the Client ID the token will be generated for.
5. Select *Regenerate*.
6. Copy the Generated Access Token.

Important: The Access Token will be masked after User Setting are saved.

7. Select *SAVE*.

Delete User

To delete a User from the Administration > Users Section:

1. Select the Administration link in the navigation bar.
2. Select the Users link in the sub navigation bar.
3. Select **ACTIONS** on the row of the user to delete.
4. Select **REMOVE** in the ACTIONS dropdown.
5. Confirm

To delete a User from the Administration > Tenants > Select a Tenant > Users tab`:

1. Select the Administration link in the navigation bar.
2. Select the Tenants link in the sub navigation bar.
3. Select a Tenant
4. Click **ACTIONS** on the row of the user to delete.
5. Select **REMOVE** in the ACTIONS dropdown.
6. Confirm

User Groups

Overview

User Groups can be selected during provisioning to add each group members credentials to the Instance. User Groups can be configured for sudo access and in Linux will assign Group members to a groupId in linux.

Creating User Groups

1. Navigate to Administration > Users
2. Select the USER GROUPS tab.
3. Select + *CREATE USER GROUP*.
4. Enter the following:
 - NAME** Name of the User Group
 - DESCRIPTION** Optional User Group Description
 - SERVER GROUP** Name of the groupId to assign Group members to in linux.
 - SUDO ACCESS** Enable to give Group members sudo access
 - USERS** Search for and select existing Users to add to the User Group.
5. Select *SAVE CHANGES*.

Editing User Groups

1. Navigate to Administration > Users
2. Select the USER GROUPS tab.
3. Select the **ACTIONS** dropdown next to the target User Group.
4. Select **EDIT**
5. Make changes, add or remove users from the group.
6. Select *SAVE CHANGES*.

Adding a User Group when Provisioning

1. When provisioning, in the CONFIG section expand the USER section.
2. Select an existing Group from the USER GROUP dropdown.
3. Users will be created for members in the selected User Group on the provisioned Instance(s).

5.9.6 Integrations

Integrations

Note: To add and edit certain integrations such as Ansible Tower, vRO, Chef and Puppet, the user must have FULL permission for “Library: Integrations” in addition to “Admin: Integrations”.

To add an integration select + *ADD* and choose your integration. Many Morpheus-supported integrations can be configured in this section, though not all. Some integrations, such as networking integrations, must be configured within their own areas of the application. The following integrations can be configured in this section:

- Chef
- Puppet
- Ansible
- Salt Masters
- Ansible Tower
- vRealize Orchestrator
- Microsoft DNS
- PowerDNS
- Route 53
- Bind DNS
- Git
- Github
- Docker Repositories
- Jenkins
- ServiceNow
- Cherwell
- Remedy
- Venafi

Please see the *Guides* for each specific integration type for more detailed information on setup steps and features supported by the integration.

Packages

Overview

The `/administration/packages` is where Morpheus packages (`.morpkg`, `.mpkg`, or `.mopkg`) can be uploaded to appliances. Morpheus packages contain Library objects, such as Instance Types, Layouts, Node Types, Spec Templates and Cluster Layouts. Morpheus packages consist of library objects as code compiled into a simple (`$.morpkg`, `$.mpkg`, or `$.mopkg`) file, allowing for agile distribution, updated and sharing of Library configurations.

The addition of `/administration/packages` in v7.0.0 is targeted for uploading future Morpheus provided packages, however users will be able to create, distribute and/or import custom Morpheus packages.

Role Permissions

Access and capabilities for the **Packages** section is determined by the following role permissions:

Role: Feature Access: Admin: Plugins

- None: Cannot access Admin: Plugins section
- Full: Access to Admin: Plugins and ability to upload Morpheus packages (.morpkg, .mpkg, or .mopkg)

Uploading Packages

/administration/packages is targeted for uploading future Morpheus provided packages's, however users will be able to create, distribute and/or import custom Morpheus packages. Additional information on creating custom packages will be provided.

Plugins

Overview

Morpheus is extendable with custom plugins for Task types, UI tabs, reports, approvals, cypher, and more. Plugins are added from the Plugins tab of the Integration page under Administration (Administration > Integrations > Plugins). Simply browse for a local plugin file (.jar) to add it to the UI. Custom plugins can also be edited or deleted by clicking on the pencil or trash can icons in the corresponding row.

Morpheus maintains a repository of internally-developed and vetted plugins at [Morpheus Exchange](#). These plugins can be downloaded and added to Morpheus using the instructions in the prior paragraph. Alongside the download link, you can also find helpful readme information that discusses how the plugin can be used and which versions of Morpheus it's compatible with.

With at least one plugin integrated, Morpheus will show details on each plugin from the Plugins List View. The following information is displayed:

- **NAME:** The name given to the plugin
- **DESCRIPTION:** A description value (if any) coded into the plugin
- **FILE NAME:** The .jar filename
- **VERSION:** The plugin version number
- **STATUS:** The status of the plugin, such as “loaded” when the plugin is ready for use
- **STATUS MESSAGE:** A status message (if any) for the plugin
- **ENABLED:** If the plugin is enabled, a check mark appears here. Disabled plugins are also grayed out

Additional information about each plugin can be viewed by clicking on the pencil (edit) icon. Most of the information in this modal is read-only but you can enable or disable plugins from this pane.

Please visit the [Morpheus Developer Portal](#) for Plugin Architecture SDK documentation and help getting started with custom Plugin development.

Search

Operations Provisioning Library Infrastructure Backups Monitoring Tools Administration

Tenants Plans & Pricing Roles Users Integrations Policies Health Settings

Integrations Packages **Plugins** Distributed Workers

Search

NAME	DESCRIPTION	FILE NAME	VERSION	STATUS	STATUS MESSAGE	ENABLED
Digital Ocean Plugin		morpheus-cloud-plugin-1.1.0-all.jar	1.1.0	Loaded		<input checked="" type="checkbox"/>

Plugin Upload

Upload Plugin File No file chosen

Distributed Workers

Overview

The Morpheus distributed worker is installed using the same package as the VDI Gateway worker. Organizations which have already deployed VDI Gateway(s) can use the same worker for both purposes if desired, you'd simply need to update configuration in `/etc/morpheus/morpheus-worker.rb` and run a `reconfigure`. When creating a distributed worker or VDI Gateway object in Morpheus UI, an API key is generated. Adding one or both types of API keys to the worker configuration file determines if the worker is running in VDI gateway and/or distributed worker mode.

Supported Cloud Types

The following Cloud/Zone types support Distributed Workers

- vmware
- vmwareCloudAws
- nutanix
- openstack
- xenserver
- macstadium

Installation

A distributed worker VM is installed and configured similarly to a Morpheus appliance via `rpm` or `deb` package.

Note: Package URLs for the distributed worker are available at <https://morpheushub.com> in the downloads section.

Note: The distributed worker requires that the Morpheus appliance has a trusted SSL certificate. This can be accomplished by configuring a public trusted SSL certificate on the Morpheus appliance (or load balancer) or ensure the certificate and chain are added to the Java Keystore of the Distributed Worker, to trust the certificate.

Requirements

Table 20: **Supported Operating Systems**

OS	Version(s)
Amazon Linux	2
CentOS	7.x, 8.x
Debian	10, 11
RHEL	7.x, 8.x
SUSE SLES	12
Ubuntu	18.04, 20.04, 22.04

- **Memory:** 4 GB RAM minimum recommended
- **Storage:** 10 GB storage minimum recommended. Storage is required for installation packages and log files
- **CPU:** 4-core minimum recommended
- Network connectivity to the Morpheus appliance over TCP 443 (HTTPS)
- Superuser privileges via the `sudo` command for the user installing the Morpheus worker package
- Access to base `yum` or `apt` repos. Access to Optional RPM repos may be required for RPM distros

Important: In order to proxy VMware vCenter Cloud traffic through a Distributed Worker, you must have a static public DNS entry for the internal IP address of the vCenter appliance. If this is not done, everything may appear to be working properly when configuring the Cloud but problems will arise at provision time. This is not a Morpheus limitation but is a limitation of the VMware SDK client which does not natively support proxies.

Download the appropriate package from Morpheus Hub based on your target Linux distribution and version for installation in a directory of your choosing. The package can be removed after successful installation.

```
wget https://downloads.morpheusdata.com/path/to/morpheus-worker-$version.distro
```

Validate the package checksum as compared with the values indicated on Hub. For example:

```
sha256sum morpheus-worker-$version.distro
```

Next, install the package using your selected distribution's package installation command and your preferred options. Example, for RPM:

rpm:

```
$ sudo rpm -ihv morpheus-worker-$version.$distro
Preparing...                               ##### [100%]
Updating / installing...
 1:morpheus-worker-x.x.x-1.$distro         ##### [100%]
Thank you for installing Morpheus Worker!
Configure and start the Worker by running the following command:
```

(continues on next page)

With the worker configured in Morpheus, the next step is to update supported Cloud integrations which should be proxied through the worker. Select the desired Cloud from the Clouds List Page (Infrastructure > Clouds) and click *EDIT* from the chosen Cloud's Detail Page. Within the Advanced Options section, choose a configured worker from the DISTRIBUTED WORKER dropdown menu. Click *SAVE CHANGES*.

CHANGE MANAGEMENT: Please Select

AGENT INSTALL MODE: Cloud Init / Unattend (when available)

API PROXY: Select

DISTRIBUTED WORKER: Select, My Worker

► Provisioning Command

SAVE CHANGES

With the API key in hand and configuration complete in Morpheus UI, head back to the worker box. Configure the gateway by editing `/etc/morpheus/morpheus-worker.rb` and updating the following:

```
worker_url 'https://gateway_worker_url' # This is the worker URL the Morpheus.
↪ appliance can resolve and reach on 443
worker['appliance_url'] = 'https://morpheus_appliance_url' # The resolvable.
↪ URL or IP address of Morpheus appliance which the worker can reach on port.
↪ 443
worker['apikey'] = 'API KEY FOR THIS GATEWAY' # VDI Gateway API Key generated.
↪ from Morpheus Appliance VDI Pools > VDI Gateways configuraiton. For worker.
↪ only mode, a value is still required but can be any value, including the
↪ 'API KEY FOR THIS GATEWAY' default template value
worker['worker_key'] = 'DISTRIBUTED WORKER KEY' # Distributed Worker API Key.
↪ from Administration > Integrations > Distributed Workers configuration
```

Note: By default the `worker_url` uses the machine's hostname, ie `https://your_machine_name`. The default `worker_url` value can be changed by editing `/etc/morpheus/morpheus-worker.rb` and changing the value of `worker_url`. Additional appliance configuration options are available below.

After all configuration options have been set, run `sudo morpheus-worker-ctl reconfigure` to install and configure the worker, nginx and guacd services:

```
sudo morpheus-worker-ctl reconfigure
```

The worker reconfigure process will install and configure the worker, nginx and guacd services and dependencies.

Tip: If the reconfigure process fails due to a missing dependency, add the repo that the missing dependency can be found in and run

Note: Configuration options can be updated after the initial reconfigure by editing `/etc/morpheus/morpheus-worker.rb` and running `sudo morpheus-worker-ctl reconfigure` again.

Once the installation is complete the morpheus worker service will automatically start and open a web socket with the specified Morpheus appliance. To monitor the startup process, run `morpheus-worker-ctl tail` to tail the logs of the worker, nginx and guacd services. Individual services can be tailed by specifying the service, for example `morpheus-worker-ctl tail worker`

Highly-Available (HA) Deployment

If desired, multiple distributed worker nodes may be associated to the same Morpheus appliance to eliminate a single point of failure should a distributed worker node go down. Configure each distributed worker node using the same worker key (process described in the prior section) and add redundancy using as many additional workers nodes as needed. When multiple worker nodes are using the same worker key, proxy calls will always go through the primary worker node when possible. The primary node is the first worker node configured using a specific worker key. When necessary, automatic failover will take place and another active worker node will be used. While proxy calls will always try to use the primary node when available, Morpheus Agent communications can be balanced equally across worker nodes by placing a VIP in front of your distributed workers.

5.9.7 Policies

Overview

Policies add governance, ease of use, cost-savings, and auditing features to Morpheus. Morpheus enables end users to create Policies scoped to Users, Roles, Groups, Clouds, Tenants, Networks, Plans, and Global scoping to give Admins full control and governance over their environments. The available scoping will vary from one Policy type to another. See the section below for information on each Policy type and guides for more complex Policy implementation.

Policy Types

Approve Delete Sets an approval requirement for deleting Instances or Apps within the Policy scope. When setting the Policy, users have the option of using Morpheus Approvals or an Approval Integration such a ServiceNow. On delete request, Instances will be shut down and only deleted if approved.

Approve Provision Sets an approval requirement for provisioning Instances or Apps within the Policy scope. When setting the Policy, users have the option of using Morpheus Approvals or an Approval Integration such a ServiceNow.

Approve Reconfigure Sets an approval requirement for reconfiguring Instances and servers within the Policy scope. When setting the Policy, users have the option of using Morpheus Approvals or an Approval Integration such a ServiceNow.

Approve Workflow Execute If enabled, when Workflows are executed on workloads within the Policy scope, an Approval is generated. This could apply when a Workflow is executed from the Workflows list page or from the detail page for an Instance or server. Approvals can be targeted to Morpheus internal Approvals or targeted to a third-party integration (such as ServiceNow). The Workflow will not begin to execute until after the approval is granted.

Backup Creation Disable or enable the ability to create a backup when provisioning an instance.

Backup Targets A master account can determine storage provider options for backups with Backup Targets policies.

Budget Sets a maximum total combined price for all instances in the Group, Cloud, Tenant or owned by the User this policy is applied to.

Cluster Resource Name The name of Cluster hosts (master and workers) when creating Kubernetes, Docker and KVM Clusters. Pre-populates a fixed or editable Resource Name value for the cluster using `${variable}` naming

patterns and/or text, including `{sequence}` numbering. Toggle whether sequence numbers are reusable (after the resource using them is destroyed) by enabling [Reuse Naming Sequence Numbers](#) in Administration > Settings

Cypher Access Granularly set LIST, READ, WRITE, and DELETE access to arbitrary Cypher secret paths scoped globally or to specific Roles and Users. See the section below for a guide to establishing a Cypher access policy.

Delayed Delete Delayed Delete Policies allow for soft deletion of Instances and Apps. Instead of deleting immediately, Instances and Apps with a Delayed Delete policy applied will be shutdown upon deletion request and hidden by default from the UI. The Instance/App will then be in **Pending Removal** status. In order to see Instances pending deletion on the Instances list page (Provisioning > Instances), you must filter for “Pending Removal” status. These Instances will not show when filtered for “All Statuses”

Expiration Sets an expiration timeframe in days after which the Instance will be deleted. Extensions can be auto-approved or require approval immediately or after x amount of auto-extensions using Morpheus Approvals or an Approval Integration. See Morpheus [Knowledge Base](#) for more information about Expiration policies

File Share Storage Quota Sets a Storage Quota for File Share usage (in GB) to scoped User, Role, Tenant or Global.

Hostname The hostname or computer name which is set in the OS and DNS. On some platforms, hostnames are restricted by length, spaces, and/or special characters. Pre-populates a fixed or editable name for hostnames/machine names using `{variable}` naming patterns and/or text, including `{sequence}` numbering. Toggle whether sequence numbers are reusable (after the resource using them is destroyed) by enabling [Reuse Naming Sequence Numbers](#) in Administration > Settings

Instance Name Pre-populates a fixed or editable name for Instance Names using `{variable}` naming patterns and/or text, including `{sequence}` numbering. Toggle whether sequence numbers are reusable (after the resource using them is destroyed) by enabling [Reuse Naming Sequence Numbers](#) in Administration > Settings. Note that it's not recommended administrators include “>”, “<”, “%”, “\$”, or “=” in naming policies

Max Containers Sets the max number of Containers for the Group or Cloud the Policy is added to.

Max Cores Sets the max number of total of Cores combined for Instances in the Group or Cloud the Policy is added to, includes the option to include or exclude container resources in the Policy.

Max Hosts Sets the max number of total Hosts in the Group or Cloud the Policy is added to.

Max Load Balancer Pools Sets the max number of load balancer pools within the policy scope

Max Memory Sets the max number of total of RAM combined for Instances in the Group or Cloud the Policy is added to, includes the option to include or exclude container resources in the Policy.

Max Pool Members Sets the maximum number of members in a load balancer pool

Max Snapshots Set the maximum number of Snapshots that may be stored for each Instance or VM within the scope. Once the limit is met, Morpheus will warn the user when attempting to create more snapshots until the number is reduced

Max Storage Sets the max number of total of Storage combined for Instances in the Group or Cloud the Policy is added to, includes the option to include or exclude container resources in the Policy.

Max Virtual Servers Sets the maximum number of virtual servers within the policy scope

Max VMs Sets the max number of Virtual Machines for the Group or Cloud the Policy is added to.

Message of the Day (MOTD) Message of the Day Policy for displaying Alerts in Morpheus. Configurable as a pop-up or full-page notification with Info, Warning and Critical message types.

Note: Requires role permission: Admin: Message Of the Day set to “Full” to create and manage MOTD Policies.

Network Quota Limits the number of networks that can be created within the policy's scope

Object Storage Quota Sets a Storage Quota for Object Storage usage (in GB) to scoped User, Role, Tenant or Global.

Power Scheduling Adds a Power Schedule for the Instances in a Group or Cloud. Power Schedules can be created in Library > Automation > Power Scheduling

Router Quota Limits the number of routers that can be created within the policy's scope

Shutdown Sets a shutdown timeframe in days upon provision after which the Instance will be stopped. Extensions can be auto-approved or require approval immediately or after x amount of auto-extensions using Morpheus Approvals or an Approval Integration.

Storage Server Storage Quota Sets a Storage Quota for selected Storage Server (in GB), applied Globally or per specified Tenants.

Tags Requires the user to add compliant Tags at provision time, this can be enforced on a strict or passive basis

Note: Tag scanning and enforcement is currently only available for Azure, Amazon, Google, and VMware clouds. For a more comprehensive guide on implementing Tag Policies, see the associated article in our [KnowledgeBase](#).

User Creation Controls the "CREATE YOUR USER" flag in the User Config options during provisioning do be always disabled, always enabled, enabled by default, or disabled by default.

User Group Creation Forces User Creation of members in the selected User Group during Provisioning.

Workflow Forces execution of selected Workflow for Instance Provisioning.

Creating Policies

Policies can be created in three different locations.

- Administration > Policies
- Infrastructure > Groups > Group > Policies
- Infrastructure > Clouds > Cloud > Policies

Policies can be disabled and re-enabled at anytime.

Important: Precedence is applied to matching or conflicting Policies in the following order: Cloud > Group > Role > User > Global.

To create a Global Policy:

1. Navigate to Administration > Policies
2. Select + *ADD Policy* and choose from the available policy types.
3. Refer to Policy Type sections below for Configuration options.
4. Under Filter next to scope select *Global*
5. Select *SAVE CHANGES*

To create a Policy for a User:

1. Navigate to Administration > Policies
2. Select + *ADD Policy* and choose from the available policy types.
3. Refer to Policy Type sections below for Configuration options.
4. Under filter next to scope select *User* a drop down menu will appear below allowing you to select a user
5. Select *SAVE CHANGES*

To create a Policy for a Role:

1. Navigate to Administration > Policies
2. Select + *ADD Policy* and choose from the available policy types.
3. Refer to Policy Type sections below for Configuration options.
4. Under filter next to scope select *Role* a drop down menu will appear below allowing you to select a Role
5. **For APPLY INDIVIDUALLY TO EACH USER IN ROLE**
 - Select for Max Resource/Quota Policies to be calculated per user
 - Leave unselected to calculate by total usage of all users within that Role.
6. Select *SAVE CHANGES*

To create a Policy for a Cloud:

Note: Resource Limitation Policies apply to all Instances in the Cloud the Policy is added to. Approval, Naming, Power, Shutdown and Expiration Policies apply to Instances created or moved into the Group after the Policy is enabled.

1. Navigate to Infrastructure > Clouds
2. Select a Cloud by clicking on the name of the Cloud to go to the Cloud Detail page.
3. Select the POLICIES tab in the Cloud Detail page.
4. Select + *ADD* and choose from the available policy types.
5. Refer to Policy Type sections below for Configuration options.
6. Select *SAVE CHANGES*

To create a Policy for a Group:

Note: Resource Limitation Policies apply to all Instances in the Group the Policy is added to. Approval, Naming, Power, Shutdown and Expiration Policies apply to Instances created after the Policy is enabled.

1. Navigate to Infrastructure > Groups
2. Select a Group by clicking on the name of the Group to go to the Group Detail page.
3. Select the POLICIES tab in the Group Detail page.

4. Select + *ADD* and choose from the available policy types.
5. Refer to Policy Types sections below for Configuration options.
6. Select *SAVE CHANGES*

Policy Types

Cypher Policies

Morpheus allows administrators to set robust Cypher policies, which determine global, role, and/or specific user access to configured Cypher secret mount points. A number of considerations should be made when deploying Cypher Access policies, including how user role permissions will interact with the policy and how conflicts between overlapping policies are resolved.

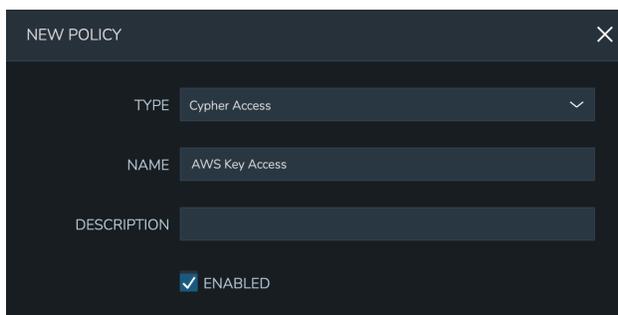
Role Permissions

User Role permissions (Administration > Roles) greatly affect Cypher access. Cypher access Role permissions are set from the Features tab of the selected Role under “Tools: Cypher”. The Role permission should be set based on the highest level of access to any one individual Cypher entry needed for the specific Role. For example, if the Role needs no access to any Cypher entries, set the feature permission to “None” and hide the Cypher UI from the Role completely. Alternatively, if the Role needs to use and decrypt even one Cypher entry, set the feature permission to “Full Decrypt”. The complete set of available permissions are below:

- **NONE:** Cypher UI hidden
- **READ:** Cypher UI present, entries can be listed
- **USER:** Cypher UI present, user sees and can use their own entries, user can create new entries
- **FULL:** Cypher UI present, user sees and can use all entries, user can create new entries, user cannot decrypt any entries
- **FULL DECRYPT:** Full access to Cypher features including the ability to decrypt secrets

Cypher Access Policies

Like other Policy types, Cypher Access Policies are created in Administration > Policies. Click + *ADD POLICY* to create new. Set the type to “Cypher Access” and the relevant configuration options will be displayed. In addition to the type, enter a name for the Policy in the top section.



NEW POLICY

TYPE: Cypher Access

NAME: AWS Key Access

DESCRIPTION:

ENABLED

In the next section, enter the key path to which the Policy will apply. In addition to static entries that point to one specific Cypher entry, this field supports pattern matching with regex. For example, enter “.*” to refer to all Cypher entries or “secret/.*” to refer to all entries under the secret mount point.

In addition to the path, set the privileges users in the Policy scope should have on the indicated path.

- **LIST:** See the entries on the indicated path listed in the Cypher UI
- **READ:** Decrypt the entries on the indicated path
- **WRITE:** Add new entries on the indicated path
- **UPDATE:** Edit Cypher entries on the indicated path. This is future functionality, the ability to update Cypher entries does not currently exist in the product
- **DELETE:** Delete entries on the indicated path

The screenshot shows a configuration window titled 'Config'. It has a 'KEY PATH' field containing 'secret/aws/*'. Below this, there are five checkboxes for permissions: 'READ' (checked), 'WRITE' (checked), 'UPDATE' (unchecked), 'DELETE' (unchecked), and 'LIST' (checked).

Finally, set the scope for the Policy. Cypher Access Policies support Global, Role, and User scope. For example, you may want to block off sets of Cypher entries for various departments within your organization. If you have existing Roles in Morpheus for each department, you can set up Role-scoped Policies to ensure they can only list, use, and add Cypher entries which are relevant to their own department.

The screenshot shows a 'Filter' configuration window. It has three dropdown menus: 'SCOPE' set to 'Role', 'ROLE' set to 'cypher_access_test_role', and 'TENANTS' set to 'Search'. There is a checkbox labeled 'APPLY INDIVIDUALLY TO EACH USER IN ROLE' which is currently unchecked. A blue 'SAVE CHANGES' button is located at the bottom right.

Important: When Cypher Access Policies conflict, the Policy with the longest path string length (typically the most specific) takes precedence. For example, a Policy giving LIST and READ access to “secret/aws/*” would be superseded by a Policy giving NO access to “secret/aws/my-secret-key”. In such a case, the user would see everything at the “secret/aws/*” path except the one indicated in the more specific Policy. When Policies targeting the same path differ only in their scope, the following scope precedence is applied: Role > User > Global. For example, if a Role-scoped Policy targeting “*” grants LIST and READ while a User-scoped Policy targeting the same path grants LIST, the user would be granted the rights in the Role-scoped Policy.

Cloud Profiles

Terraform Cloud Profiles are created on each Cloud detail page (Infrastructure > Clouds > Selected Cloud > Profiles Tab), encrypted in Cypher, and create a Cypher entry that is visible both on the Profile tab of the Cloud detail page and in Cypher. When added to a Cloud they create a Cypher entry at path `tfvars/profile/cloud/$cloudCode/variables`. If a Cloud profile Cypher entry is restricted by a Cypher Access policy, it will be (or will not be) listable/readable/deletable as dictated by the Policy but still will be viewable from the Cloud detail page if the user has sufficient permissions. Restricting or granting access to Cloud profiles via Policy does not affect access on the Cloud. Other Role permissions, such as “Admin: Profiles”, “Infrastructure: Clouds”, and Cloud/Group access must be used to restrict access via Cloud detail pages.

Example Policy

In my example organization, I have one department that needs access to AWS-related secrets and another department that needs access to Azure-related secrets. There are many other secrets stored in my appliance but I don't want either of these departments to access any of those.

secret/cypher_policy_test_1	DECRYPT	04/04/2022 02:16 PM	04/05/2022 02:32 PM	ACTIONS ▾
secret/cypher_policy_test_2	DECRYPT	04/04/2022 02:16 PM	04/04/2022 02:46 PM	ACTIONS ▾
secret/example/aws/access-key	DECRYPT	04/05/2022 07:28 PM	04/05/2022 07:28 PM	ACTIONS ▾
secret/example/aws/my-special-key	DECRYPT	04/05/2022 07:28 PM	04/05/2022 07:28 PM	ACTIONS ▾
secret/example/aws/secret-key	DECRYPT	04/05/2022 07:27 PM	04/05/2022 07:27 PM	ACTIONS ▾
secret/example/azure/secret1	DECRYPT	04/05/2022 07:29 PM	04/05/2022 07:29 PM	ACTIONS ▾
secret/example/azure/secret2	DECRYPT	04/05/2022 07:30 PM	04/05/2022 07:30 PM	ACTIONS ▾
secret/hello	DECRYPT	04/12/2021 02:40 PM	04/01/2022 10:06 PM	ACTIONS ▾
secret/QA Kubernetes Cluster-kube-cluster-admin-kubeconfig	DECRYPT	02/16/2022 07:11 PM	03/14/2022 12:54 PM	ACTIONS ▾

For the first department, I've set up a Policy that allows them to list and read (including use and decryption rights) AWS secrets. A second Policy specifically excludes them from seeing one specific entry. The Policy with the more specific path will supersede the more generic Policy that includes a wildcard.

POLICIES			
Search	TYPE: CYPHER ACCESS ▾	SCOPE: ALL ▾	+ ADD POLICY
	Cypher Exclusion	Cypher: - key path: secret/example/aws/my-special-key - none	cypher_access_test_role Accounts: All
	Cypher Policy - AWS	Cypher: - key path: secret/example/aws/* - read, list	cypher_access_test_role Accounts: All

By impersonating the user, we see they indeed have access to just the two desired Cypher entries.

Cypher ?

Search + ADD

KEY	VALUE	LEASE REMAINING	LAST UPDATED	LAST ACCESSED	
secret/example/aws/access-key	DECRYPT		04/05/2022 07:28 PM	04/05/2022 07:28 PM	ACTIONS ▾
secret/example/aws/secret-key	DECRYPT		04/05/2022 07:27 PM	04/05/2022 07:27 PM	ACTIONS ▾

For the second department, I have set up a Policy that allows them to list and read (including use and decryption rights) Azure secrets.

POLICIES

Search + ADD POLICY

TYPE: CYPHER ACCESS SCOPE: ALL

	Cypher Exclusion	Cypher - key path: secret/example/aws/my-special-key - none		cypher_access_test_role	Accounts: ALL		
	Cypher Policy - AWS	Cypher - key path: secret/example/aws/* - read, list		cypher_access_test_role	Accounts: ALL		
	Cypher Policy - Azure	Cypher - key path: secret/example/azure/* - read, list		cypher_access_test_role	Accounts: ALL		
	tm - all	Cypher - key path: secret/tm/vh.* - read, write, update, delete, list		Tommy Marks	Accounts: ALL		

By impersonating the user once again, we see they indeed have access only to Azure entries.

Cypher ?

Search + ADD

KEY	VALUE	LEASE REMAINING	LAST UPDATED	LAST ACCESSED	
secret/example/azure/secret1	DECRYPT		04/05/2022 07:29 PM	04/05/2022 07:29 PM	ACTIONS ▾
secret/example/azure/secret2	DECRYPT		04/05/2022 07:30 PM	04/05/2022 07:30 PM	ACTIONS ▾

Expiration Policies

Expiration policies set an expiration timeframe for any instance provisioned into the cloud, role, group or by the user the policy is added to. When an instance expires, it is terminated and deleted.

Configuration options for expiration policies:

Expiration Type

- User Configurable- expiration timeframe is editable during provisioning
- Fixed Expiration- user cannot change expiration timeframe

Expiration Days Configures the number of days the instance is allowed to exist before being removed.

Renewal Days If the instance is renewed, this is the number of days by which the expiration date is increased.

Notification Days This allows an email notice to be sent out X days before the instance is set to expire.

Notification Message Customizable message for notification emails. The default message is Instance `{{instance?.name}}` is set to expire on `{{instance?.expireDate}}`

Auto Approve Extensions Enable this to auto-approve extension requests, bypassing approval workflows.

Instances with expirations show the time until expiration in the instance detail pane. Instances with active expiration policies can be extended by selecting the EXTEND NOW button in the instance detail pane. The extension length is set in the policy by the RENEWAL DAYS field.

Expirations can also be added to any instance during provisioning by entering the number of days in the EXPIRATION DAYS field in the Lifecycle section of the automation section of the provisioning wizard. Expiration can be added to any instance even if no policies have been created.

Note: Expiration and Shutdown Policies will be enforced on Instances created when converting a discovered host to managed.

Instance and Host Names

Naming Policies will populate a fixed or editable name for instances, hosts and hostnames. The Name Pattern field uses `${variable}` string interpolation.

NAMING TYPE

User Configurable Naming pattern will pre-populate during provisioning but can be edited by the user.

Fixed Name Naming pattern will pre-populate during provisioning and cannot be changed.

NAME PATTERN The Name Pattern field uses Static text and/or `${variable}` string interpolation, such as `morpheus${cloudCode}${type}${sequence+3000}`

An example Instance Name Policy using a naming pattern with User Initials, Cloud Code, Instance Type, and a sequential number starting at 3000 is `${userInitials}-${cloudCode}-${type}-${sequence+3000}`, resulting in an Instance Name of **md-vmwd3-centos-3001** for the first instance, followed by **md-vmwd3-centos-3002** and so on.

Commonly used variables for naming patterns include:

```

${groupName}
${groupCode}
${cloudName}
${cloudCode}
${type}
${accountId}
${account}
${accountType}
${platform}
${platform == 'windows' ? 'w':'l'} # results in `w` for Windows platforms and `l`
↳for Linux Platforms
${userId}
${username}
${userInitials}
${provisionType}
${instance.instanceContext} # Environment Code
${sequence} # results in 1
${sequence+100} # results in 101
${sequence.toString().padLeft(5,'0')} #results in 00001
```

Cloud codes and Group codes are fields found in their respective configuration panes.

AUTO RESOLVE CONFLICTS Morpheus will automatically resolve naming conflicts by appending a sequential -number to the name when enabled.

Shutdown Policies

Shutdown policies dictate the number of days an instance is allowed to run before it is shut down. Shutdown is consistent across cloud types i.e.: in VMware, a VM is powered off. In AWS, an instance is stopped. Etc.

Configuration options for shutdown policies:

Shutdown Type

User Configurable Shutdown timeframe is editable during provisioning.

Fixed Expiration User cannot change shutdown timeframe during provisioning.

Expiration Days Configures the number of days the instance is allowed to exist before being shut down.

Renewal Days If the instance is renewed, this is the number of days by which the shutdown date is increased.

Notification Days This allows an email notice to be sent out X days before the instance is set to shut down.

Notification Message Customizable message for notification email.

Auto Approve Extensions Enable this to auto-approve extension requests, bypassing approval workflows.

Note: Expiration and Shutdown Policies will be enforced on Instances created when converting a discovered host to managed.

Provision Approval

Morpheus Provision Approvals enable an approval workflow via internal Morpheus approval or via ServiceNow workflow. If a ServiceNow integration is present, the ServiceNow option is enabled. The Approval workflow to be selected is dynamically created by querying the ServiceNow Workflow table in the integrated ServiceNow instance.

This ServiceNow approval integration enables users to use the Morpheus Self-Service provisioning portal to provision new instances and still respect the required ServiceNow business approval workflow.

Power Schedules

Power Schedules set daily times to shutdown and startup instances. Power schedule can be created and managed in Library > Automation > Power Scheduling

Note: Power Schedule Policies will apply to Instances created in a Group or Cloud after the Policy is enabled, and will not apply to pre-existing Instances.

Configuration options for Power Schedule Policies:

DESCRIPTION Add details about your Policy for reference in the Policies tab.

Enabled Policies can be edited and disabled or enabled at any time. Disabling a Power Schedule Policy will prevent the Power Schedule from running on the Groups Instances until re-enabled.

ENFORCEMENT TYPE

- User Configurable: Power Schedule choice is editable by User during provisioning.
- Fixed Schedule: User cannot change Power Schedule setting during provisioning.

POWER SCHEDULE Select Power Schedule to use in the Policy. Power schedule can be added in Library > Automation > Power Scheduling

TENANTS Leave blank for the Policy to apply to all Tenants, or search for and select Tenants to enforce the Policy on specific Tenants.

Max Resources

Max Resource policies allow setting quotas for Clouds, Groups, Roles or Users for maximum amount of Memory, Storage, Cores, Hosts, VM's, or Containers that can be created in the Cloud, Group, Role or by the User the Policy is assigned to.

Configuration options for Max Resources Policies:

Max Containers Sets the maximum combined total of Containers in Instances per Policy Scope.

Max Cores Sets the maximum combined total of Cores in Instances per Policy Scope.

Max Hosts Sets the maximum total of Hosts per Policy Scope.

Max Memory Sets the maximum combined total of RAM (capacity) for Instances per Policy Scope.

Max Storage Sets the maximum combined total of Storage (capacity) for Instances per Policy Scope.

Max VMs Sets the maximum total of managed Virtual Machines per Policy Scope.

TENANTS Leave blank for the Policy to apply to all Tenants, or search for and select Tenants to enforce the Policy on specific Tenants.

User Creation

The User Creation policy controls the “CREATE YOUR USER” flag in the User Config options during provisioning do be always disabled, always enabled, enabled by default, or disabled by default.

Configuration options for User Creation Policies:

TYPE User Creation

DESCRIPTION Description to identify the policy config

Enabled Policies enforcement can be disabled or enabled at any time.

ENFORCEMENT TYPE

- User Configurable: User Creation choice is editable by User during provisioning.
- Fixed: User cannot change User Creation setting during provisioning.

CREATE USER Check to allow or force user creation. Uncheck to disable by default or force no user creation.

TENANTS Leave blank for the Policy to apply to all Tenants, or search for and select Tenants to enforce the Policy on specific Tenants.

5.9.8 Health

Morpheus Health


MORPHEUS

Support
Jaden Jabro

Operations
Provisioning
Library
Infrastructure
Backups
Monitoring
Tools
Administration

Tenants
Plans & Pricing
Roles
Users
Integrations
Policies
Health
Settings

Morpheus Health
Morpheus Logs

HEALTH SUMMARY

✓
Overall

✓
CPU

✓
Memory

✓
Database

✓
Elastic

✓
Queue

HEALTH LEVELS


MORPHEUS CPU


SYSTEM CPU


MORPHEUS MEMORY


SYSTEM MEMORY


USED SWAP

CPU

Processor Count: 8	Process Time: 17 days 22 hours	Morpheus CPU: 10.39%
System CPU: 16.82%	System Load: 2.16	

MEMORY

Morpheus Memory: 15.0GiB	Morpheus Used Memory: 10.9GiB	Morpheus Free Memory: 4.0GiB
Morpheus Memory Usage: 73.08%	System Memory: 62.7GiB	System Used Memory: 28.1GiB
System Free Memory: 4.5GiB	System Memory Usage: 92.77%	System Swap: 1.9GiB
Free Swap: 1.8GiB		

DATABASE

Lifetime Connections: 34863	Aborted Connections: 0	Max Used Connections: 88
Max Connections: 151	Threads Running: 4	Threads Connected: 21
Slow Queries: 0	Temp Tables: 0	Handler Read First: 0
Buffer Pool Free: 5229	Open Tables: 2000	Table Scans: 0
Full Joins: 0	Key Read Requests: 60	Key Reads: 15
Engine Waits: 0	Lock Waits: 0	Handler Read Rnd: 164
Engine IO Writes: 6466226	Engine IO Reads: 36304508	Engine IO Double Writes: 224802
Engine Log Writes: 3192436	Engine Memory: 131.1MiB	Dictionary Memory: 7.5MiB
Buffer Pool Size: 8192	Free Buffers: 937	Database Pages: 7082
Old Pages: 2625	Dirty Page Percent: 0.00%	Max Dirty Pages: 0.00%
Pending Reads: 0	Insert Rate: 0.84 / s	Update Rate: 11.02 / s
Delete Rate: 0.36 / s	Read Rate: 25,005.75 / s	Buffer Hit Rate: 100.00%
Read Write Ratio: 0.573	Uptime: 2 days 13 hours	

The Morpheus Health section provides an overview of the health of your Morpheus appliance. It includes an appliance health summary in the following areas:

- **CPU:** Appliance CPU usage is checked. If usage is greater than 50%, this indicator will be in a yellow or warning state. If Morpheus is unable to complete the check, it will be in a red or error state. Depending on appliance

performance and how frequently this indicator is in a warning state, it may be necessary to upgrade to increase CPU. The **Overall** health indicator will mirror the CPU health indicator

- **Memory:** If swap usage is above 60% or Morpheus memory usage is above 95%, this indicator will be in a yellow or warning state. If Morpheus is unable to complete the check for any reason, it will be in a red or error state. Depending on appliance performance and how frequently this indicator is in a warning state, it may be necessary to increase swap, upgrade the appliance to add memory, or consider a different appliance architecture for those using single-node appliances
- **Storage:** If utilization of the filesystem mounted at “/” exceeds 80%, this indicator will be in a yellow warned status. Above 90% will put this indicator in red or error status
- **Database:** The database is checked. If the number of database connections exceeds the configured maximum number of connections or if any test queries are reported as being slow, this indicator will be in a yellow or warning state. If Morpheus is unable to communicate with the database, it will be in a red or error state. In the database section further down the page, you can check the number of maximum used connections against the number of max connections. In the case of database connections exceeding the maximum, consider increasing the maximum settings connection
- **Elastic:** Elasticsearch is polled for the health status of each index. If any indices are not reporting a “green” health status, this indicator will be in a yellow or warning state.
- **Queues:** RabbitMQ queues are checked. Any queues containing more than 1000 messages are considered to be in an error state. Appliance Queue health is given in a yellow or warning status when any queues are in such an error state. In the Queues section further down the page you can see the individual Queues listed and which have messages piling up. When the appliance is unable to complete the check for any reason, this indicator will be in a red or error state

Health Levels

Health levels provide a live representation of the current memory and CPU load on the appliance. Bear in mind that in an HA appliance, this data will be specific to the appliance node you happen to be using. By default, Morpheus does not include any endpoint or UI tool which can show you the currently used app node. However, a plugin has been developed which can surface this information if needed. See [this thread](#) in the Morpheus official forums for additional details about accessing and using the plugin.

- **Morpheus CPU:** Instantaneous amount of CPU capacity in use by Morpheus processes
- **System CPU:** Instantaneous amount of CPU capacity in use by all processes
- **Morpheus Memory:** Instantaneous amount of system memory currently in use by Morpheus processes (see the Knowledge Base article linked in the TIP box below for more information on how Morpheus claims and manages available memory)
- **System Memory:** Instantaneous amount of total system memory currently claimed (this is commonly a high percentage, see the TIP box below)
- **Used Swap:** Instantaneous amount of total available system swap in use
- **Storage:** The instantaneous percentage utilization of the filesystem mounted at “/”

Tip: It’s common to see a high percentage of system memory being used [due to the way Morpheus allocates and manages memory](#). If Morpheus is performing well, high system memory use is not necessarily an indicator that any action needs to be taken.

Additional System Health Indices

CPU

- Processor Count
- Process Time
- Morpheus CPU
- System CPU
- System Load

MEMORY

- Morpheus Memory
- Morpheus Used Memory
- Morpheus Free Memory
- Morpheus Memory Usage
- System Memory
- System Used Memory
- System Free Memory
- System Memory Usage
- System Swap
- Free Swap

DATABASE

- Lifetime Connections
- Aborted Connections
- Max Used Connections
- Max Connections
- Threads Running
- Threads Connected
- Slow Queries
- Temp Tables
- Key Reads
- Handler Reads
- Buffer Pool Free
- Open Tables
- Table Scans
- Full Joins
- Key Read Requests
- Key Reads
- Engine Waits

- Lock Waits
- Handler Reads
- Engine IO Writes
- Engine IO Reads
- Engine IO Double Writes
- Engine Log Writes
- Engine Memory
- Dictionary Memory
- Buffer Pool Size
- Free Buffers
- Database Pages
- Old Pages
- Dirty Page Percent
- Max Dirty Pages
- Pending Reads
- Insert Rate
- Update Rate
- Delete Rate
- Read Rate
- Buffer Hit Rate
- Read Write Ratio
- Uptime

ELASTIC

- Status
- Cluster
- Node Count
- Data Nodes
- Shards
- Primary Shards
- Relocating Shards
- Initializing
- Unassigned
- Pending Tasks
- Active Shards

Note: Warning status is typical for Elasticsearch

Elastic Nodes

- Node
- Master
- Location
- Heap Usage
- Memory Usage
- CPU Usage
- 1M Load
- 5M Load
- 15M Load

Elastic Indices

- Health
- Index
- Status
- Primary
- Replicas
- Doc
- Count
- Primary
- Size
- Total Size

Queues

- Queue Count
- Busy Queues
- Error Queues

Morpheus Logs

The Morpheus logs section aggregates appliance-specific logs into one list. If needed, users can export the logs by clicking *EXPORT*. This action triggers a download containing the last 10,000 log entries as a `.log` file.

Morpheus Health

Morpheus Logs

900 -
675 -
450 -
225 -
0

02:00 AM 04:00 AM 06:00 AM 08:00 AM 10:00 AM 12:00 PM

Search [] All Levels [v] Today [v]

	TIMESTAMP	MESSAGE
INFO	02/24/2022 01:56:51 PM	[RxCachedThreadScheduler-438] remove vm: com.morpheus.ComputeServer :
INFO	02/24/2022 01:56:51 PM	[RxCachedThreadScheduler-438] remove vm: com.morpheus.ComputeServer :
INFO	02/24/2022 01:56:51 PM	[RxCachedThreadScheduler-438] Removing Vms for VMWare Cloud
INFO	02/24/2022 01:56:34 PM	[RxCachedThreadScheduler-413] remove vm: com.morpheus.ComputeServer :
INFO	02/24/2022 01:56:34 PM	[RxCachedThreadScheduler-413] remove vm: com.morpheus.ComputeServer :
WARN	02/24/2022 01:56:34 PM	[RxCachedThreadScheduler-435] 1 organization is supported.. found 0

5.9.9 Settings

The Administration > Settings section sets global configuration parameters for the Morpheus appliance, whitelabeling, provisioning, monitoring, backups, logs, software licenses, and the license for Morpheus itself.

Appliance

Appliance Settings

Appliance URL The default URL used for Agent install and Agent functionality. All Instances and Hosts must be able to resolve and reach this URL over 443 for successful agent install and communication.

Note: Alternate Appliance URLs can be configured per Cloud in the *Edit Cloud > Advanced Options* section.

Internal Appliance URL (PXE) For PXE-Boot your appliance needs to be routable directly with minimal NAT masquerading. This allows one to override the default appliance url endpoint for use by the PXE Server. If this is unset, the default appliance url will be used instead.

API Allowed Origins A CORS-related field which specifies the origins that are allowed to access the Morpheus API. For example, if you were designing a web application which needed to make AJAX calls to Morpheus API. The origins should be specified here. By default, all origins are allowed. When this field is filled, an exclusive whitelist of allowed origins is established.

Cloud Sync Interval Data is refreshed through cloud integrations at the interval specified here in seconds, the default value is 300 seconds (five minutes). Appliances managing a very large number of clouds may be adversely affected by setting this value too low.

Usage Retainment Determines how many days to keep account usage (metered costing data) records. Retainment period is not set by default. Usage records will remain indefinitely if Usage Retainment is not set. Note this does not affect generated Invoice records.

Incident Retainment Enter the number of days Morpheus should keep incident records in the database. In general, this setting can be left alone but in certain cases may need to be adjusted as very large incident database tables can affect the stability of the application.

Stats Retainment Select 30, 60 or 90 days period for stats retainment. Selecting a larger period gives the ability to analyze stats, such as Instance metrics, over a longer period of time. For example, in the Monitoring tab of an Instance detail page, users can select a 60 or 90-day analysis period if the stats have been retained that long

Denied Hosts A comma-delimited list of IP addresses and/or hostnames which should not be allowed sources for HTTP Tasks or REST-populated Option Lists.

Approved Hosts A comma-delimited list of IP addresses and/or hostnames which are the only approved sources for HTTP Tasks or REST-populated Option Lists. By entering any values here, all others are automatically denied.

Enable SSL Verification of Agent (Communications) Enabling SSL Verification of Agent Communications requires a valid Certificate be installed on the Appliance.

Disable SSH Password Authentication Only allow ssh login using SSH keys. When true, SSH Password Authentication will not be enabled for VM's and Hosts provisioned after the setting is enabled.

Default Appliance Locale Sets the default language and region for all users on the Morpheus appliance. Users with individual language preferences may also override this selection on their User Settings page

Default Console Gateway Select a configured Morpheus Worker as a console gateway or VDI gateway. For more on installation and configuration of a gateway, see the [VDI Gateways section](#) of Morpheus documentation.

Tenant Management Settings

Registration Enabled If enabled, the appliance login screen will have a “NEED AN ACCOUNT? SIGN UP HERE” link added, enabling new Tenant registration.

Default Tenant Role Sets the default Tenant Role applied to Tenants created from Tenant Registration.

Default User Role Sets the default User Role applied to the User created from a Tenant Registration.

User Management Settings

Min Password Length User passwords must at least be as many characters in length as the entered value

Min Password Uppercase User passwords must include at least as many uppercase characters as the entered value

Min Password Numbers User passwords must include at least as many numerals as the entered value

Min Password Symbols User passwords must include at least as many special characters as the entered value

Session Expires (Minutes) A user session is forcibly logged out after the entered number of minutes of inactivity

Session Warning (Minutes) A pop-up warning is shown to the user when they have been inactive for the number of minutes entered. Example: If sessions are set to expire after 90 minutes, warn the user after 60 minutes if you intend to provide 30 minutes advance warning

Expire Password After (Days) User account passwords will expire after the entered number of days. Enter 0 or leave the field empty to opt out of this feature.

Disable User After Attempts (Number of Attempts) Disable a User account after a specified number of failed login attempts. Enter 0 or leave the field empty to opt out of this feature.

Disable User If Inactive For (Days) Disable a User account if inactive for the entered number of days. The User will not be able to log into the appliance again until another User with sufficient rights enables the account. Enter 0 or leave the field empty to opt out of this feature.

Send warning email before deactivating (Days) Enter the number of days prior to account deactivation that a warning email should be sent. For example, enter “5” to warn the User when they are five days short of the deactivation time entered in the prior field. Enter 0 or leave the field empty to opt out of this feature.

Email Settings

In this section, you can configure an SMTP server for email notification delivery. You will need to provide Morpheus the following information, your mail server systems administrator can assist you in filling these fields and with the preferred encryption method.

- From Address
- SMTP Server
- SMTP Port
- SSL Enabled
- TLS Encryption
- SMTP User
- SMTP Password

We recommend that you add the Morpheus server to your SMTP whitelist as well as using user authentication as an additional security measure.

Once you have added your SMTP server information into Morpheus, scroll down to the bottom of the page and press the blue SAVE button which can be found under the Enabled Clouds section.

When you have saved your SMTP server settings in the Morpheus appliance you will then need to restart the UI. To restart the morpheus-ui, connect to your Morpheus server via SSH and run the below command:

```
sudo morpheus-ctl restart morpheus-ui
```

Important: If you do not restart morpheus-ui, the notifications will not be sent. Please note it can take up to three minutes for the UI to become reachable again.

Twilio SMS Settings

Configure SMS text message delivery for Morpheus alerts. Previously, customers could use Morpheus’ own account for delivery, but for security reasons clients must now supply their own. Complete the fields indicated below and then restart all Morpheus nodes to apply the changes.

- **Account SID:** Twilio Account SID
- **SMS From #:** The “From” number to receive the text message from
- **Auth Token:** The Twilio API authentication token for your account

Proxy Settings

The Morpheus Appliance can be configured to communicate through a Proxy server for Cloud API's and Agent communication back to the Appliance.

Note: Additional Proxy configuration is available in the *Infrastructure > Network > Proxies* section. Added Proxies can be scoped to Clouds in the *Edit Cloud > Advanced Options* section of the Cloud.

Add a Global Proxy server by entering the following:

- Proxy Host
- Proxy Port
- Proxy User
- Proxy Password
- Proxy Domain
- Proxy Workstation

Currency Settings

In Morpheus, Tenants are separate environments which can be defined as using currencies that are unique from one Tenant to the next. In addition, these currencies may be different from the currency in which Price Sets have been defined. In order to present pricing to Subtenant users in their designated currency, Morpheus allows for integration with currency conversion services “open exchange rates” and “fixer.io”. This article goes through the process of setting up the integration and how it works to determine pricing conversions.

Integrating With a Currency Exchange Provider

1. Navigate to Administration > Settings > Appliance
2. Under the Currency Settings heading, make a “Currency Provider” selection
3. Enter your “Provider API Key”

The service is now integrated and can be used as described in the next section.

Consuming Currency Exchange in Morpheus

Currency exchange data is synced from the integrated provider once every 12 hours. When needed, Morpheus will use this cached data to present currency conversions rather than hitting the API directly each time. This limits the total number of API hits and reduces costs.

Exchanged currency values will be shown under conditions similar to the following scenario:

A user is working in a Subtenant configured for Currency B. The user is attempting to provision an instance with pricing sets that have only been defined in Currency A. Morpheus will convert the pricing data from currency A to Currency B for this user (and all users in this Subtenant) since price conversion has been enabled.

Enabled Clouds (Types)

Controls which types of Cloud can be created.

- When a Cloud type is disabled, it will be removed from the available options when adding new Clouds in Infrastructure > Clouds. Existing Clouds are not affected by changes to this setting.

Whitelabel

Whitelabel Settings

Overview

Morpheus Tenants can be WhiteLabeled with custom Logos, Colors, Copy, and custom CSS. Sub-Tenants can be individually white-labeled, or the Master Tenant Whitelabel can apply to all Sub-Tenants.

Enable Whitelabel Turns on the configured Whitelabel settings. Disabling will return the Appliance to the default colors and logos, but the configured options will remain saved and will apply if Whitelabel is re-enabled.

Appliance Name Replaces Morpheus in page titles.

Header Logo Top left header logo. Uploaded image is resized to 38 pixels high with a proportional width at that height.

Disable Support Menu Enable this flag to hide the support dropdown menu in the header.

Support Menu Links Customize support links. Label Code can be used for translations and is optional. Be sure to specify fully qualified url if linking to external sites.

Security Banner

The Security Banner section in `/admin/settings#!whitelabel` displays content on the login screen for Security and Co

- Applicable at Global and Tenant levels
- Security Banner input field accepts plain text and markdown
- Content is displayed below login section in scoped `/login/auth` pages.

Footer Logo Footer Logo in bottom left. Uploaded image is resized to 27 pixels high with a proportional width at that height.

Login Logo Logo shown on Login screen. Uploaded image is resized to 192 pixels wide with an unbound height proportional to that locked width.

Favicon Must be a .ico file type.

Reset When selected and Whitelabel settings are saved, associated logo is returned to blank default value.

Colors

Update Colors by entering HEX value or selecting the Color Selector pop-up next to each field and selecting a color.

- Header Background
- Header Foreground
- Nav Background
- Nav Foreground
- Nav Hover
- Primary Button Bg
- Primary Button Fg
- Primary Button Hover Bg
- Primary Button Hover Fg
- Footer Background
- Footer Foreground
- Login Background

Override CSS

Override CSS settings by entering CSS in *Override CSS* field.

Example: (this will add one continues background image to the Header)

```
header #topHeader {
    background-image: url(http://image_url.png);
}
header {
    background-image: url(http://image_url.png);
}
```

Copy

Add custom Copyright String, Terms of Use, Privacy Policy contained in the Footer text and links in the App and on the login page and emails.

Available Copy fields

- Copyright String
- Terms and Privacy String
- Terms of Use
- Privacy Policy

Note: Terms of Use and Privacy Policy Footer links will load internal pages at https://appliance_url/privacy-policy and https://appliance_url/terms-of-use displaying the entered info as plain text. The Terms and Privacy String will update the legal text displayed on the

Morpheus login page. This field takes any custom HTML markup allowing you to link to the internal legal pages or to your own outside legal pages if you prefer.

UI Loading Page

When the Morpheus UI is restarted or loading, a default “Morpheus is Loading” page is displayed. This page can be changed by adding the following to `/etc/morpheus/morpheus.rb` and adjusting the values.

Note: `morpheus-ctl reconfigure` must be ran for any chnages to `/etc/morpheus/morpheus.rb` to take effect.

```
nginx['web_root_internal'] = "/opt/morpheus/embedded/nginx/html"
nginx['loading_pages']['max_loops'] = 6 * 10 # 10 secs per loop x 6 times to get 60.
↳seconds * 10 to get to 10 minutes
nginx['loading_pages']['timeout_page'] = '/timeout.html'
nginx['loading_pages']['iteration_time'] = 10_000
nginx['loading_pages']['loading_page_title'] = 'Morpheus Loading'
nginx['loading_pages']['loading_page_h1'] = 'Morpheus is Loading...'
nginx['loading_pages']['loading_page_h2'] = 'please wait'
nginx['loading_pages']['timout_page_title'] = 'Morpheus timeout, please try again...'
nginx['loading_pages']['timout_page_h1'] = 'Timeout waiting for Morpheus to load, click.
↳below to try again.'
nginx['loading_pages']['failure_page_title'] = 'Morpheus Server Error'
nginx['loading_pages']['failure_page_h1'] = 'Morpheus Server Error'
nginx['loading_pages']['failure_page_h2'] = 'Please contact your system administrator.
↳for assistance.'
```

Provisioning

Provisioning Settings

Allow Cloud Selection Displays or hides Cloud Selection dropdown in Provisioning wizard.

Allow Host Selection Displays or hides Host Selection dropdown in Provisioning wizard.

Require Environment Selection Forces users to select and Environment during provisioning

Show Pricing Displays or hides Pricing in Provisioning wizard and Instance and Host detail pages.

Hide Datastore Stats On Selection Hides Datastore utilization and size stats in provisioning and app wizards

Cross-Tenant Naming Policies Enable for the sequence value in naming policies to apply across tenants

Reuse Naming Sequence Numbers When selected, sequence numbers can be reused when Instances are removed. Deselect this option and Morpheus will track issued sequence numbers and use the next available number each time.

Deployment Archive Store Default Storage Provider for storing Deployment Archives.

Note: Storage Providers can be configured and managed in the *Infrastructure > Storage* section.

Cloud-Init Settings

Morpheus can add global users for Linux and Windows at provision time. Cloud-init/Cloudbase-Init or VMware Tools installed on the provisioned virtual images is required.

Linux

- **Username:** Enter User to be added to Linux Instances during provisioning.
- **Password:** Enter password to be set for the above Linux user.
- **KeyPair:** Select KeyPair to be added for the above Linux user.

Note: Either a password, keypair, or both can be populated for the Linux user. Keypairs can be added in the *Infrast*tructure > Keys & Certs section.

Windows Settings

- **Administrator Password:** Enter password to be set for the Windows Administrator User during provisioning.

PXE Boot Settings

Default Root Password Enter the default password to be set for Root during PXE Boots.

App Blueprint Settings

Determines the Default Blueprint Type selected in new App Wizard

- Morpheus
- ARM Template
- CloudFormation
- Terraform
- Kubernetes Spec
- Helm Chart

Terraform Settings

- **Terraform Runtime:** Select “auto” or “manual”. When selecting “auto”, Morpheus will automatically download and use the Terraform version indicated in the VERSION field on the Spec Templates that make up a Terraform Instance type or Blueprint. When selecting “manual”, Morpheus will use the version of Terraform installed on your appliance.

Monitoring

Morpheus Monitoring Settings

Auto Create Checks When enabled a Monitoring Check will automatically be create for Instances and Apps.

Availability Time Frame The number of days availability should be calculated for. Changes will not take effect until your checks have passed their check interval.

Availability Precision The number of decimal places availability should be displayed in, can be anywhere between 0 and 5. Instance availability is shown on the Instance detail page (and used elsewhere) and refers to the percentage of time the workload is up.

Default Check Interval The default interval to use when creating new checks.

Note: Monitoring Checks can be manually configured if *Auto Create Checks* is disabled.

Service Now

ServiceNow Monitoring Integration Settings

Note: A ServiceNow Integration must be already configured in Administration > Integrations to enable the ServiceNow Monitoring Integration.

Enabled Enables the ServiceNow Monitoring Integration

Integration Select from a ServiceNow Integration added in Administration > Integrations

New Incident Action The Service Now action to take when a Morpheus incident is created.

Close Incident Action The Service Now action to take when a Morpheus incident is closed.

Incident Severity Mapping

Morpheus Severity	ServiceNow Impact
Info	Low/Medium/High
Warning	Low/Medium/High
Critical	Low/Medium/High

Logging Settings

Overview

Morpheus contains a built-in logging solution that aggregates logs from hosts and services. Logs are displayed, searchable, and filterable in the Instance, App, Host and global Logs (Monitoring > Logs) sections. Logs can also be forwarded using Syslog forward rules to any external solution that supports Syslogs.

The logs displayed in the Instance, App, Host and overall Logs (Monitoring > Logs) sections are only from managed VMs and Hosts that have the Morpheus Agent installed. Morpheus Agent will watch `/var/logs` for any `.log` file and report them back accordingly. Containerized Instances can be configured to show additional logs by configuring the

LOG FOLDER in the Library NODE TYPE. Logs from any .log file in the specified folder will be forwarded by the Morpheus Agent to the Morpheus appliance or forwarded with Syslog forward rules.

Note: The *Logs* section does not contain Morpheus appliance logs, which can be found in `/var/log/morpheus/` and in Administration > Health.

Logs are stored in ElasticSearch and retention can be set by adjusting the Availability Time Frame in the Administration > Settings > Monitoring section. Logging can also be disabled with a simple toggle switch just above the Availability Time Frame configuration.

The screenshot shows the Morpheus Administration interface. The top navigation bar includes the Morpheus logo, a search bar, a notification bell, a 'Support' dropdown, and a user profile for 'Alex Harker'. Below the navigation bar, there are tabs for 'Operations', 'Provisioning', 'Library', 'Infrastructure', 'Backups', 'Monitoring', 'Tools', and 'Administration'. Under the 'Administration' tab, there are sub-tabs for 'Tenants', 'Plans & Pricing', 'Roles', 'Users', 'Integrations', 'Policies', 'Health', and 'Settings'. The 'Logging Settings' page is displayed for the 'servicenow' service. It features a 'Logs Enabled' toggle switch that is turned on. Below this is an 'Availability Time Frame' input field set to '7' days, with a 'SAVE' button. Underneath, there is a 'Syslog Forwarding Rules' section with a table for adding rules. The table has columns for 'Name', 'Rule', and 'Action', and a 'QUICK ADD' button.

Backups

Backup Settings

The Backup settings page allows you to enable or disable scheduled backups, select a default backup bucket, and administer global settings related to backups. Changes to global settings only affect new backups going forward and do not affect existing backups.

Note: Appliance backups are subject to a two-hour time limit to complete the backup. Automated backup attempts will be abandoned and will fail once this time limit is exceeded.

Morpheus Backup Settings

Scheduled Backups Enable automatic scheduled backups for provisioned instances

Create Backups When enabled, Morpheus will automatically configure instances for manual or scheduled backups

Backup Appliance When enabled, a backup will be created for the Morpheus appliance database. Select the [Backup](#) text link to view or edit settings related to the appliance backup

Default Backup Bucket Select an existing bucket as the default for future backup runs. Click the [Infrastructure Storage](#) text link to add a new storage bucket to Morpheus if needed

Default Backup Schedule Choose a default schedule interval for automated backups. The available selections in this dropdown menu are Execution Schedules defined in [Library > Automation > Execute Scheduling](#)

Default Backup Retention Choose the default number of backups to be retained for automated Instance and appliance backup jobs

Guidance

Overview

Morpheus guidance is an important tool that makes recommendations for resource and cost optimization. It analyzes CPU, memory, and storage activities over time to make intelligent recommendations on sizing and power state. These recommendations can free up resources and save organizations significant amounts of money over time. Out of the box, Morpheus is configured for sensible thresholds used in making these recommendations but they can be edited here if needed.

Power Settings

Morpheus will recommend shutting down a resource if *all* three of the baselines in this section are exceeded:

- **Average CPU (%):** Shutdown will be recommended if the average CPU usage is below this value. Values over 100% are possible as this value factors the number of CPU cores. Default value: 75
- **Maximum CPU (%):** Shutdown will be recommended if the CPU usage never exceeds this value. Values over 100% are possible as this value factors the number of CPU cores. Default value: 500
- **Network Threshold (bytes):** Shutdown will be recommended if the average network bandwidth is below this value. Default value: 2000 bytes/second

CPU Up-size Settings

CPU up-size will be recommended when *both* of the following baselines are exceeded for a resource:

- **Average CPU (%):** CPU up-size is recommended if the average CPU percentage exceeds this value (and other criteria are also met). Default value: 50
- **Maximum CPU (%):** CPU up-size is recommended if the maximum CPU percentage exceeds this value. Default value: 99

Memory Up-size Settings

Memory up-size will be recommended when both of the following thresholds are met for a resource:

- **Minimum Free Memory (%)**: Memory up-size will be recommended if free memory dips below this value. Default value: 10

Memory Down-size Settings

Memory down-size will be recommended when both of the following thresholds are met for a resource:

- **Average Free Memory (%)**: Memory down-size is recommended if the average free memory is above this value. Default value: 60
- **Maximum Free Memory (%)**: Memory down-size is recommended if free memory has never dipped below this value. Default value: 30

Clients

Overview

Morpheus includes pre-configured OAuth clients and allows the user to create as many additional clients as they'd like. The pre-configured clients are editable but cannot be deleted. Once configured, access tokens may be generated or re-generated from the *API Access section*. Their expiration times may be viewed as well. Client settings are available only in the Primary Tenant and affect all Tenants.

Creating an OAuth Client

To create a new OAuth Client, click + *ADD* and configure the following:

- **CLIENT ID**: A reference name for the client in Morpheus
- **SECRET**: An optional OAuth client secret
- **ACCESS TOKEN VALIDITY INTERVAL (SECONDS)**: The length of time (in seconds) during which the token should be enabled
- **REFRESH TOKEN VALIDITY INTERVAL (SECONDS)**: The length of time (in seconds) during which the refresh token should be enabled

Once the client is configured, click *SAVE CHANGES*.

Editing and Deleting OAuth Clients

From the OAuth client list view (Administration > Settings > Clients), click the pencil () or trash can () icons to edit or delete the OAuth Client.

Note: Pre-configured Morpheus-default clients may be edited but not deleted. User-created clients may be edited or deleted.

Environments

Overview

The Environments section is where you create and manage your environment labels, which are available in the *Environment* dropdown during Instance or App provisioning. An Instance's environment label can be changed by editing the Instance.

Creating Environments

1. Select + *Create Environment*

2. Populate the following for the New Environment:

Name The friendly name for the environment in Morpheus

Code Shortcode used for API and CLI

Description Environment description displayed on the Environments list page

Display Order The order in which environments are presented when provisioning, a value of "0" will position the environment at the top of the list

Visibility

- *Private*: Available only in the Tenant the environment is created in
- *Public*: Available for all Tenants. Public is only applicable for environments created in the the Master Tenant.

Note: User-created environments can be edited, hidden, or removed from the Actions menu on the environments list page. Morpheus-default environments can only be hidden from users during provisioning.

Software Licenses

Overview

The License section is for automating the application of Licenses to Instances while provisioning. Licenses can be added to Morpheus and then attached to images. Morpheus will then apply the license to Instances provisioned using the images with license attached. Licenses can be configured for single or multiple Tenants.

Creating Licenses

1. Select + *Create License*

2. In the New License modal, enter the following:

- **License Type** Windows
- **Name** Name of the License in Morpheus
- **License Key** Enter the License Key
- **Org Name** The Organization Name (if applicable) related to the license key
- **Full Name** The Full Name (if applicable) related to the license key

- **Version** The License Version
- **Copies** The Number of copies available on the License
- **Description** License description displayed in the Licenses list in Morpheus, helpful for identifying the License after creation
- **Virtual Images**

Search for existing Virtual Images by name and select to attach the image to the license.

Note: Virtual Images are synced from Clouds or added in the Library > Virtual Images section.

- **Tenant Permissions** Search for and select the Tenant(s) the License will be available for. Multiple Tenants can be added.

3. Save Changes

Provisioning with Licenses

When a Virtual Image is added to a license, Morpheus will automatically apply the License to Instances configured with the Virtual Image during provisioning, including Instance Types with a Node Type that is configured with the Virtual Image, or if the image is selected when using generic Cloud Instances types (VMware, AWS, Nutanix, Openstack etc). Virtual Images can be removed from a License by editing the License.

Managing Licenses

Created Licenses details are displayed in the License page, including the number of copies applied per License, the Tenants added to the License, and the Virtual Images attached to the License.

The Name, Version, Copies, Description, Virtual Images and Tenant Permissions are editable but selecting the *Actions* dropdown on a License.

Note: License Types, Keys, Org Names and Full Names are not editable after a license has been created.

License can also be removed using the *Actions* dropdown on a License.

License

Overview

Morpheus requires a valid license for provisioning new Instances, Apps and Hosts, and converting existing Instances and Hosts to managed. Licenses can be applied and updated in this section, and the current license status can be checked.

Note: Morpheus is licensed for a certain number of concurrent workload elements (WLEs) that may be managed or inventoried at any one time. See our [Knowledge Base](#) for specific information on the types of WLEs that count against Morpheus licensing.

Current License

If a License Key has already been applied, the License status is shown in the *Current License* section:

Tenant Name Company name the License was generated for.

Start Date Date and time the current License started.

End Date Date and time the current License expires.

Space Amount of used and unused Managed RAM under the current License.

EXAMPLE: On a 1 TB License with 182 GB of RAM under management, the Space section will show *Used Space 182.9GB Unused Space 841.0GB*

Note: Once a current License expires or has reached its Space limit, users will no longer be able to provision new Instances, Apps, Hosts, or Bare Metal, or convert existing Hosts, Virtual Machines, or Bare Metal to managed. Morpheus will otherwise continue to function.

Upgrade License Key

To add a new or update an existing License:

1. Copy the License Key into the License Key field
2. Click *UPDATE*

If valid, the new License will be applied.

Request new License

Licenses can be requested at <https://morpheusdata.com>, or by contacting support@morpheusdata.com or sales@morpheusdata.com.

Utilities

System administrators have access to a utilities panel with the following options:

- **Reindex all searchable data:** Execute
- **Toggle Maintenance Mode:** Enable

Note: Maintenance mode cleanly places Morpheus into a state where maintenance can be performed on the appliance. This drains any active sessions and queues so an auto-scaling group can scale down. It also drains active sessions across services. Restarting Morpheus UI disables maintenance mode.

Note: When using Morpheus in a Highly Available (HA) environment, it is important to navigate to a node directly and enable maintenance mode, as opposed to using the load balancer virtual IP (VIP). A local host entry to the specific node may be required to ensure the correct node enters maintenance mode. In fact, it is recommended to use the [analogous API endpoint](#) to toggle a specific node into maintenance mode to avoid redirects back to the VIP address.

A Morpheus node in maintenance mode can still be accessible through the load balancer VIP/target group and can queue requests but will not process anything in queue, while in maintenance mode. A node can be removed/paused

from the load balancer VIP or have VIP health checks implemented, if the node UI/API will become inaccessible due to maintenance.

5.9.10 User Settings

User settings are accessed by clicking on your display name in the far upper-right corner of the application window. In this dropdown menu, click on the “USER SETTINGS” link.

User Photo

Upload a custom image for your user avatar that is displayed in the top header and user administration sections. **Suggested Photo Dimensions: 128 x 128**

User Settings

The fields included in this section are described below. By entering any new values in these fields and clicking *SAVE*, the existing value will be overwritten.

- **Theme:** Default or Dark mode. The Theme setting is not available when Whitelabeling is enabled on the current Tenant
- **Username:** Your Morpheus username
- **First Name:** Your first name (together with Last Name makes up your display name)
- **Last Name:** Your last name (together with First Name makes up your display name)
- **Email:** Your email address
- **Password:** Enter a value and save changes to update your password. The value in the Confirm field below must match
- **Confirm:** Confirm the new password you’ve entered
- **RECEIEVE NOTIFICATIONS** Determines if provisioning notifications are emailed to this User

User Photo



Suggested Photo Dimensions:
128 x 128

UPLOAD PHOTO

User Settings

Username

First Name

Last Name

Email

Password

Confirm

RECEIVE NOTIFICATIONS

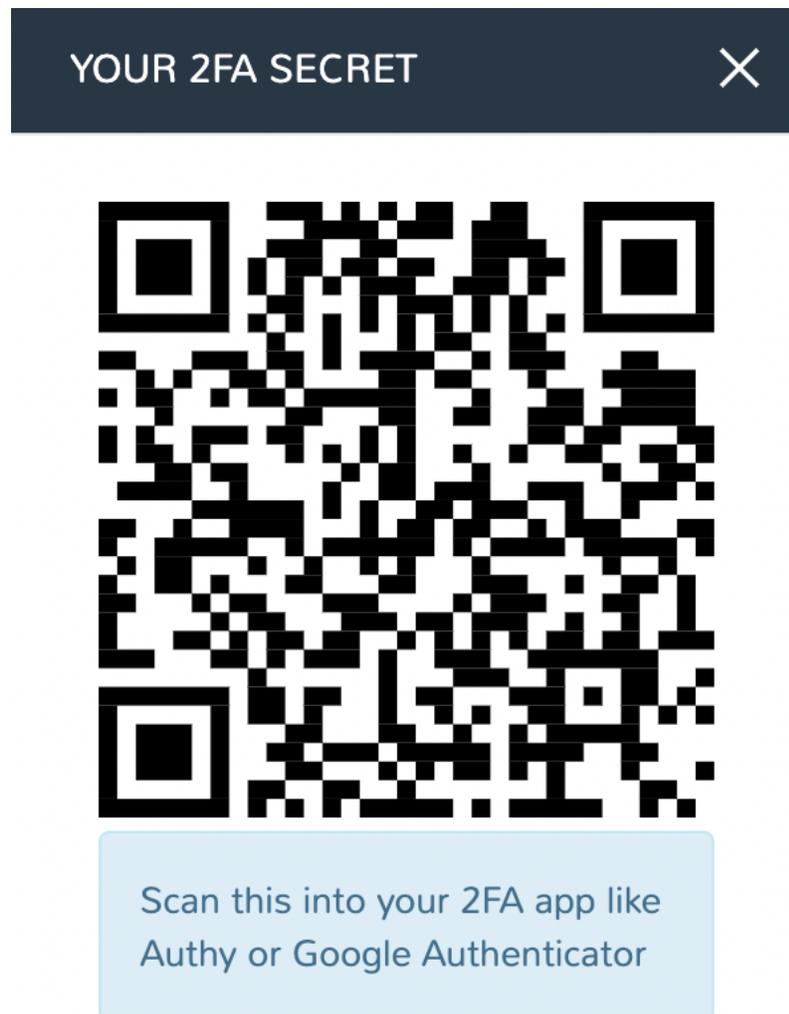
API ACCESS

2 Factor Authentication

Morpheus supports two-factor authentication (2FA) for local user accounts as well as those authenticating through Active Directory and LDAP identity sources. Authentication is handled through a 2FA app such as Authy or Google Authenticator. Other common methods for handling 2FA, such as through email or SMS text message are not currently supported. Two-factor authentication is handled on a per-user basis through the User Settings section. There is not currently a way for an administrator to enforce the use of two-factor authentication appliance-wide.

Setting Up Two-Factor Authentication

When two-factor authentication isn't yet set up, this section contains a single button: *ENABLE 2FA*. To get started, click this button and Morpheus will prompt for your password. After entering the password, you'll be shown a QR code which can be scanned into your authenticator application of choice. Once the QR code is shown, 2FA is active and the supplemental code will need to be entered each time the user logs in.



On subsequent login attempts, the user will be prompted to enter a 2FA code after successful entry of the username and password. Retrieve this code from the 2FA app you set up in the prior section and enter it to complete the login

process.

Disabling Two-Factor Authentication

When two-factor authentication is set up, this section contains two buttons: *DISABLE 2FA* and *GET 2FA CODE*. To generate a new QR code and configure an authenticator app, click *GET 2FA CODE*. Once you generate a new QR code, the old one is no longer valid. At that point you must reconfigure your authenticator app or you will not be able to access your account on the next login attempt. Generating a new QR code requires your password.

To disable 2FA, click *DISABLE 2FA*. This action does not require a password.

Handling User Lock-Out

If a user loses the device they've configured for authentication or if they cannot proceed through 2FA login for any other reason, an administrator should impersonate the user's account, reset their password, disable 2FA, then share the new temporary password with the user. At that point, the user can login, reset their password to something more secure, and re-enable 2FA (if desired).

Preferences

- **Default Group:** Sets the default Group selection when provisioning
- **Default Cloud:** Sets the default Cloud selection when provisioning
- **Default Persona:** Sets the default Persona used when logging in
- **Default Locale:** Sets the user's preferred language and region, this setting will override the global locale for the individual user

Preferences

Default Group	<input type="text" value="Labs"/>
Default Cloud	<input type="text" value="Azure Labs"/>
Default Persona	<input type="text" value="Standard"/>

Linux Settings

When provisioning a Linux-based resource and opting to have your user created during the provisioning process, the credentials entered in this section will be used to seed that user into the provisioned resource.

- **Username:** The username that will be used with your Linux user
- **Password:** The password that will be used with your Linux user (optional if specifying key)
- **Confirm:** Confirm your entered password. These must match in order for the new password value to be saved
- **SSH Key:** Select a pre-existing SSH key pair object in Morpheus. Required of not specifying password and creating your user during provisioning, or required if ssh password authentication has been disabled.

Warning: If your users Linux Settings password and/or key are not defined, and ‘Create User’ is enabled during provisioning (default), a random password will be generated but not exposed and you will not be able to login with your user.

Linux Settings

Username	<input type="text" value="morphUser"/>
Password	<input type="password" value="....."/>
Confirm	<input type="password" value="....."/>
SSH Key	<input type="text" value="morpheus-3-D4XB8L"/> ▼

Windows Settings

When provisioning a Windows-based resource and opting to have your user created during the provisioning process, the credentials entered in this section will be used to seed that user into the provisioned resource.

- **Username:** The username that will be used with your Windows accounts
- **Password:** The password that will be used with your Windows accounts
- **Confirm:** Confirm your entered password. These must match in order for the new password value to be saved

Warning: If your users Windows Settings password is not defined, and ‘Create User’ is enabled during provisioning (default), a random password will be generated but not exposed and you will not be able to login with your user.

Windows Settings

Username	<input type="text" value="morphUser"/>
Password	<input type="password"/>
Confirm	<input type="password"/>

API Access

Click the *API Access* button to expand the “API ACCESS” modal. In this modal you can generate or refresh access tokens that can be used with Morpheus API and Morpheus CLI.

If no token yet exists for a particular “CLIENT ID”, click *ACTIONS* and then Generate. If a token has expired, we can also regenerate that token by clicking *ACTIONS* and then Regenerate. After regenerating a particular token, you would need to ensure any scripts using those tokens are updated.

If needed, Primary Tenant administrators may configure the expiration periods for existing clients or create new clients from Morpheus global settings (Administration > Settings > Clients). See *client configuration documentation* for more details.

- **morph-api:** Used for Morpheus API access and should be the default token-type used
- **morph-cli:** Used for Morpheus CLI access
- **morph-automation:** Used by the internal Task engine and by jRuby-type Tasks to make API calls. It shouldn't be used externally for other types of access or in external automation. It is surfaced in the UI so users can see if a token exists and can clear it when necessary
- **morph-customer:** This token is available for legacy implementations and was previously recommended for custom API access (similar to the morph-api token). It's not recommended for use but is still available to maintain support for legacy custom automation which may still be in use on customer sites

After navigating away from the User Settings page, the complete access and refresh tokens will be masked from view. If these are lost or compromised, you can eliminate a token completely by clicking *ACTIONS* and then Clear. If you need to generate a new token for the same Client ID, click *ACTIONS* and then Regenerate.

API ACCESS ✕				
CLIENT ID	ACCESS TOKEN	REFRESH TOKEN	EXPIRES	
morph-api			09/25/2021 10:42 PM	ACTIONS ▾
morph-automation			09/25/2020 11:06 PM	ACTIONS ▾
morph-cli			06/12/2020 09:16 PM	ACTIONS ▾
morph-customer			09/25/2021 10:33 PM	ACTIONS ▾

DONE

Note: Access Tokens are only displayed/available after generation. Copy new Tokens and store appropriately before navigating from /user-settings, they will not be displayed again.

5.10 Personas

Personas are alternate views in Morpheus UI. A user's access to the various Personas is controlled by Role permissions. At present, there are three Persona types: Standard, Service Catalog, and Virtual Desktop. The Standard Persona is the typical default view. The Service Catalog Persona is a simplified view where users are presented with pre-configured Instance types, Blueprints, and Workflows to choose from based on their Role. The Virtual Desktop Persona allows administrators to grant user access to remote workstations and applications.

The rest of this section goes through the use of the Service Catalog Persona and Virtual Desktop Persona in greater detail, including how administrators can curate catalog item and virtual desktop selections for their users.

5.10.1 Service Catalog Persona

The Service Catalog Persona presents a simplified catalog where users can select and deploy Instances or Blueprints with pre-defined configuration with just a few clicks and without presenting an overwhelming list of options.

Configuring Catalog Item Access

Within the Service Catalog Persona, users are presented with Catalog Items based on their User Role. Additionally, Catalog Item access can be set on the Tenant Role to restrict certain items from all users in the Tenant. By default, User Roles have no access to any catalog items (and no access to the Service Catalog Persona). When enabling Service Catalog Persona access for User Roles, you will also need to give access to some or all Catalog Items.

Configuring Global Access:

- **Full:** Gives access to all Catalog Items
- **Custom:** Gives access to individually-selected items from the list below
- **None:** No access is given to any Catalog Items

Tip: When giving Custom access, be sure to set access on some of the individual catalog items to Full in order to reveal those items to the Role group.

Dashboard

The default page for the Service Catalog Persona is the Dashboard. The Dashboard shows a selection of featured catalog items, a listing of the last few items the user has ordered, and a selection from the user's inventory.

MORPHEUS Catalog User

Dashboard Catalog Inventory

CATALOG ITEMS

- Apache on VMware**
Apache on Ubuntu provisioned to VMware
[ORDER](#)
- Easy Peasy**
[ORDER](#)
- Resource**
My Awesome App
A Blueprint App Deployment
[ORDER](#)

RECENTLY ORDERED

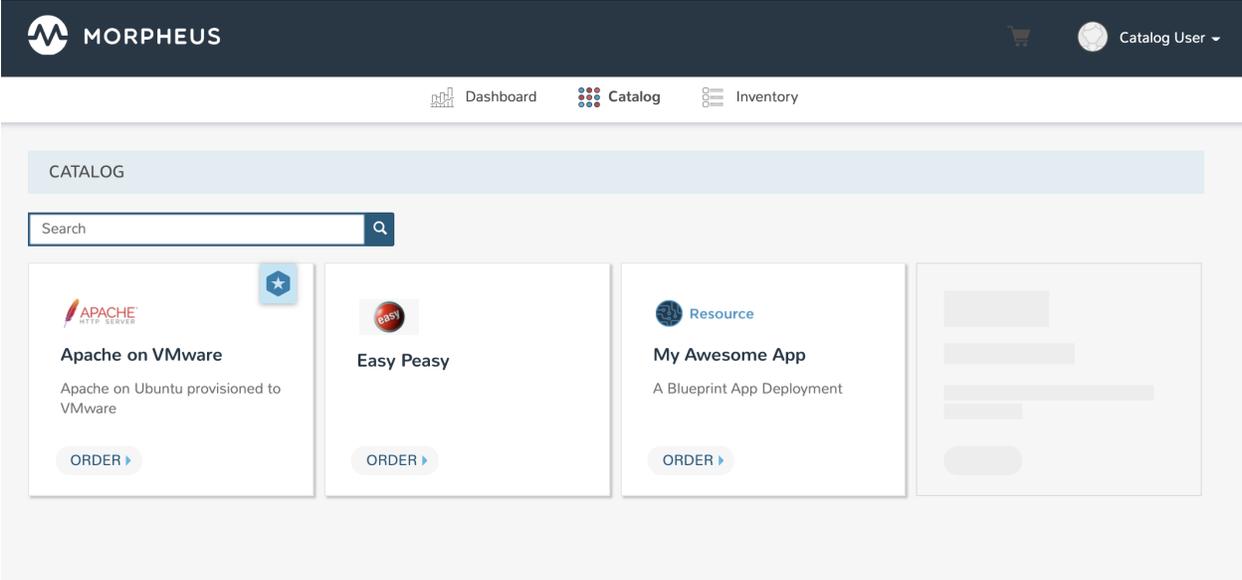
- Easy Peasy**
Ordered On: 09/17/2020
[ORDER AGAIN](#)

INVENTORY

TYPE	NAME
------	------

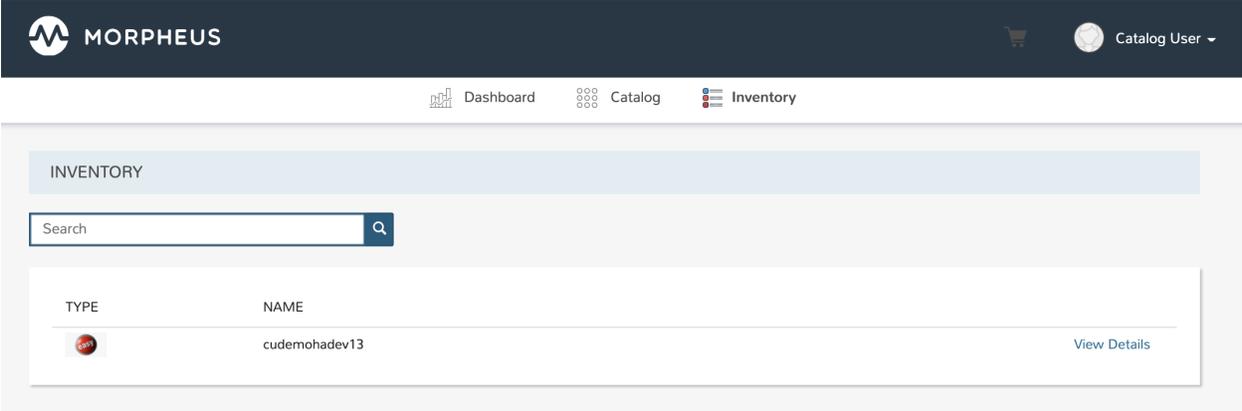
Catalog

The catalog shows the complete list of pre-defined catalog items available to the user for provisioning. Catalog items are not created in the Service Catalog Persona. For more on creating catalog items, see the [Self Service](#) section (Tools > Self Service) of Morpheus docs.



Inventory

The Inventory Page reveals the complete list of items which have been ordered by the user and provisioned. Users will only see their own items in this section.



Inventory Detail

Access the detail page for an item in your inventory by clicking the View Details link. The page displayed will look very similar to an Instance or App detail page in the Standard Persona. Highly detailed information on the health of the Instance or App are shown here. We can also take actions against Instances such as running Tasks or Workflows, reconfiguring the Instance, controlling the power state, and more.

The screenshot shows the Morpheus web interface. At the top, there's a navigation bar with 'MORPHEUS' logo, a shopping cart icon, and 'Catalog User'. Below it are navigation links for 'Dashboard', 'Catalog', and 'Inventory'. The main content area shows the item 'cudemohadev13' with a star icon and 'Env: Dev Plan: 1 CPU, 512MB Memory'. There are 'ACTIONS' and 'DELETE' buttons. A row of metrics includes: STATUS (warning icon), HEALTH (question mark icon), LAST BACKUP (minus icon), AVAILABILITY (100.00%), RESPONSE TIME (N/A), MAX CPU (0%), MEMORY (3%), and STORAGE (0%). Below this is an 'INFO' section with a 'CentOS' logo and details: Group: Labs, Owner: Catalog User, Cores: 1, Price: \$33.50 / Month, Cloud: vCenter 180, Layout: VMware VM, Memory: 512.0MiB, Source Image: Morpheus CentOS 7.5 v4, Date Created: 09/17/2020 02:27 PM, Version: 7.5, Total Storage: 10.0GiB, Provision Time: 4 minutes 28 seconds. A 'VMS' section contains a table with columns: STATUS, NAME, TYPE, CLOUD, LOCATION, COMPUTE, MEMORY, STORAGE, and ACTIONS. The table has one row for 'cudemohadev13' with a warning icon in the status column. At the bottom, there's a 'Summary' tab and a memory usage bar chart showing 'USED MEMORY', 'CACHE MEMORY', and 'MAX MEMORY'.

Placing Orders

From the Catalog page, select the tile for your chosen item to see any custom options that may need to be set prior to provisioning.

The screenshot shows the Morpheus application interface. At the top, there is a dark blue header with the Morpheus logo and the name 'MORPHEUS' on the left. On the right, there is a shopping cart icon with a '1' notification, and a user profile icon for 'Alex Harker'. Below the header, there are navigation links for 'Dashboard', 'Catalog', and 'Inventory'. The main content area shows a breadcrumb trail 'Catalog > Order'. The primary item is 'AWS APACHE', with the Apache logo and the text 'This will provision an Apache EC2 instance on AWS'. Below this, there are two dropdown menus: 'Environment' with 'Select' and 'Plan' with 'No Options Found'. An 'ADD TO CART' button is located at the bottom right of the item card.

Once the item is in the cart, make any additional selections to complete the order. Once finished, proceed to the cart by clicking on the cart icon at the top of the application window. Click “Review Order”. When reviewing your order, each selected item is listed along with its estimated cost. The total estimated cost for the entire order is also computed.

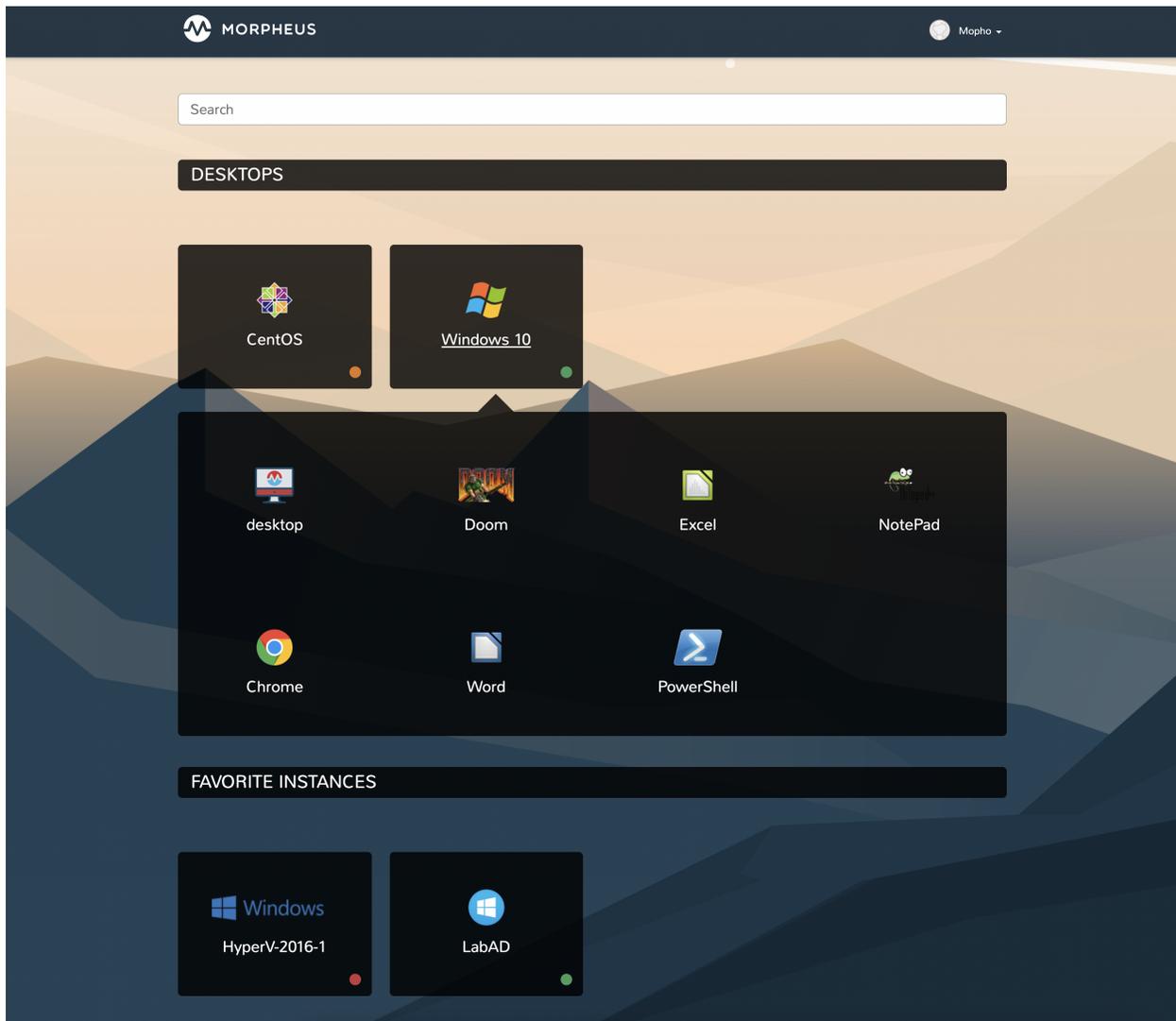
The screenshot shows the 'Review Order' page in the Morpheus application. The header and navigation are identical to the previous screenshot. The main content area is titled 'Review Order'. It displays a single item in the cart: 'Apache Docker' with the Apache logo. To the right of the item name are edit and delete icons. Below the item name, the price is listed as 'Price: \$0.725 / Month'. At the bottom right of the item card, the total is shown as 'Total: \$0.725 / Month'. A 'PLACE ORDER' button is located at the bottom right of the review area.

Once *PLACE ORDER* is clicked, the provisioning process will begin and the user is redirected back to the catalog page. Any new orders can be viewed in Inventory and additional details can be accessed through the Inventory item detail page.

5.10.2 Virtual Desktop Persona

The Morpheus VDI Persona provides a virtual desktop environment to grant users access to workstations and applications in a secure manner. Deploy pools of virtual machines on any supported Morpheus Cloud for users to reserve and use! Morpheus leverages open source client technologies, such as Apache Guacamole, to provide a performant and secure virtual desktop client for the end user while wrapping its frontend in a completely new framework. For information on configuring VDI pools for consumption in the Virtual Desktop Persona, see the [Tools section](#) of Morpheus docs.

Note: This is not an integration with existing VDI Pool Managers such as VMWare Horizon, Citrix VDI, or Nutanix XiFrame. It is a standalone Morpheus feature.



Key Features

- VDI Pool Management
- Virtual Desktop Persona
- RDP/SSH/VNC Console Support
- RDP Remote App Support
- Clipboard Copy/Paste
- HiDPI Support
- Auto Compression Scanning based on User Bandwidth
- Audio Playback (RDP)
- Local Printer (RDP)
- Auto-Resize
- Auto-Login based on Morpheus User Settings
- Customizable User Background

Configuring Access to the Virtual Desktop Persona

Access to the Virtual Desktop Persona and individual VDI pools is handled through the user Role and, where applicable, Tenant Role. When creating a new Role, access is restricted to the Virtual Desktop Persona and all VDI pools by default. To grant access:

1. Navigate to the Role (Administration > Roles > Selected Role)
2. Access the Personas tab
3. Toggle the Virtual Desktop permission to “Full” or “None”
4. Access the VDI Pool Access tab
5. Toggle access to selected VDI pools to “Full” or “None”, you can also toggle permission on all pools to “Full” or “None” with the Global Access selection
6. Role changes are saved automatically, there is no need to manually save

Launching a VDI Instance

VDI Instances are launched from the Virtual Desktops Persona. Depending on Role permissions, your account may default to this view or may even restrict you solely to this view. To access the Virtual Desktops Persona from the standard view or from another Persona, click on the user’s name in the upper-right corner of the application window. When available, this dropdown menu will list the standard Morpheus Persona view as well as any other Personas the user has permission to access. Click on Virtual Desktop to access the Virtual Desktop Persona.

The Virtual Desktop Persona view lists out each of the virtual desktop types they can access. Click on the desired virtual desktop type to launch it. If there are virtual apps available for any listed desktop types, they are presented in a flyout menu alongside a “desktop” option to access the base OS over an individual app. Items categorized as “Desktops” are VDI pools configured in the Tools menu of the Standard Persona. Items categorized as “Instances” are Instances favorited by the current user in the Standard Persona (if they have access and they have favorited any Instances). Clicking on an Instance tile offers quick access to the Instance console.

Important: Virtual Desktops are launched in a pop-up window. Be sure your web browser is not blocking pop-ups or create an appropriate exception for Morpheus virtual desktop pop-ups.

Changing the Virtual Desktop Persona Background

The Morpheus Virtual Desktop Persona includes default backgrounds for an elegant look out of the box. If desired, users may change this background to suit personal taste or organizational branding. This change is unique to each individual account. At this time, there is no option for appliance-wide whitelabeling for Virtual Desktop Persona backgrounds.

1. Click on the user's name in the far upper-right corner of the application window
2. Click USER SETTINGS
3. The section for "VDI Settings" is in the lower-right corner of the page
4. Mark the RESET box and click *SAVE* to reset the Virtual Desktop Persona to the default background
5. Click "Browse" and upload a local image to add a custom background
6. Click *SAVE*

5.10.3 Persona Access

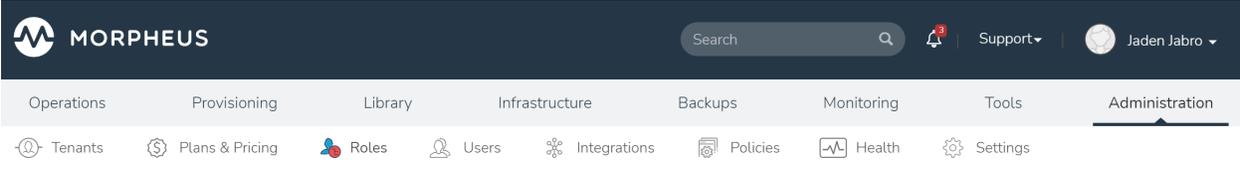
Configuring Persona Access

Access to Personas is controlled by a user's Role. Additionally, Persona access can be configured on the Tenant Role to set maximum Persona access for any user in the Tenant. By default, new Roles and Roles that existed prior to the creation of Personas will only have access to the Standard Persona. If desired, new Roles can be configured to have access to one or both Personas and existing Roles can be edited in the same way.

Tip: It's recommended to set access to all Personas to "None" if you intend not to use Personas at all. Under this configuration, Morpheus gives access only to the Standard Persona and hides the Persona selection menu from the user. New Roles and Roles that existed prior to creation of the Personas feature are pre-configured in this way.

Edit Persona access on a Role with the following steps:

1. Navigate to Administration > Roles
2. Select the desired Role to edit
3. Go to the Personas tab
4. Allow access to one or both Personas as needed, changes are saved automatically



Roles > A Tenant Role

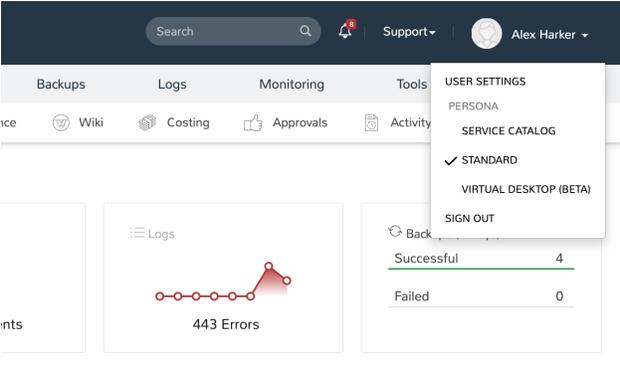
A Tenant Role

PERMISSIONS

Features	Clouds	Instance Types	Blueprints	Report Types	Personas	Catalog Item Types	VDI Pools
Default Persona <input type="text" value="Select"/>							
NAME							ACCESS
Standard							Full
Service Catalog							Full
Virtual Desktop							Full

Accessing Alternate Personas

Switch Personas by clicking on your name in the upper-right corner of the application window. If your Role gives you access to any additional Personas, they will be listed here.

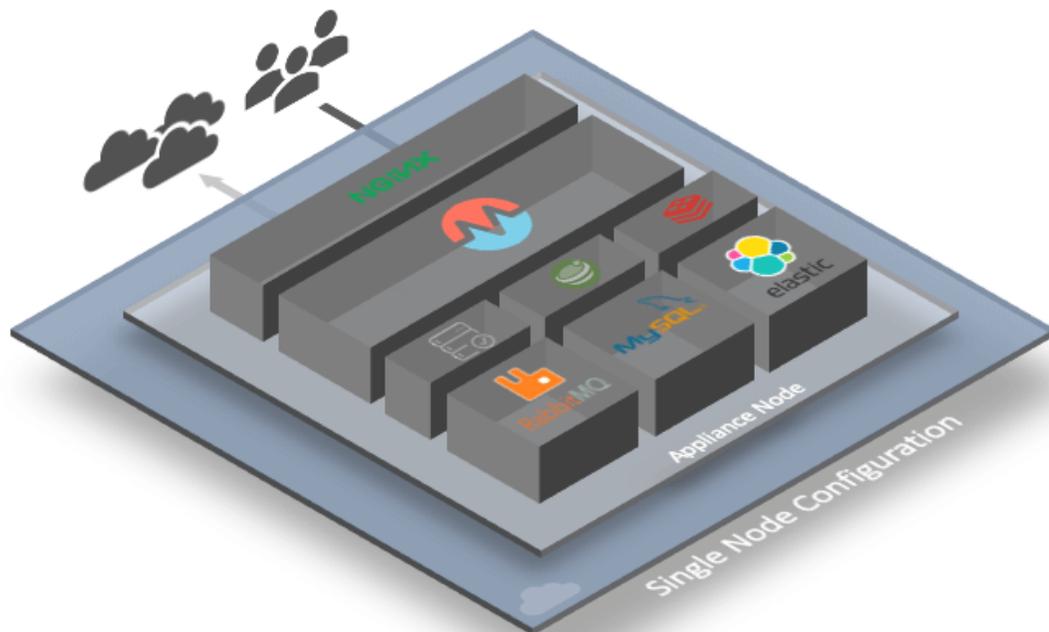


5.11 Diagrams

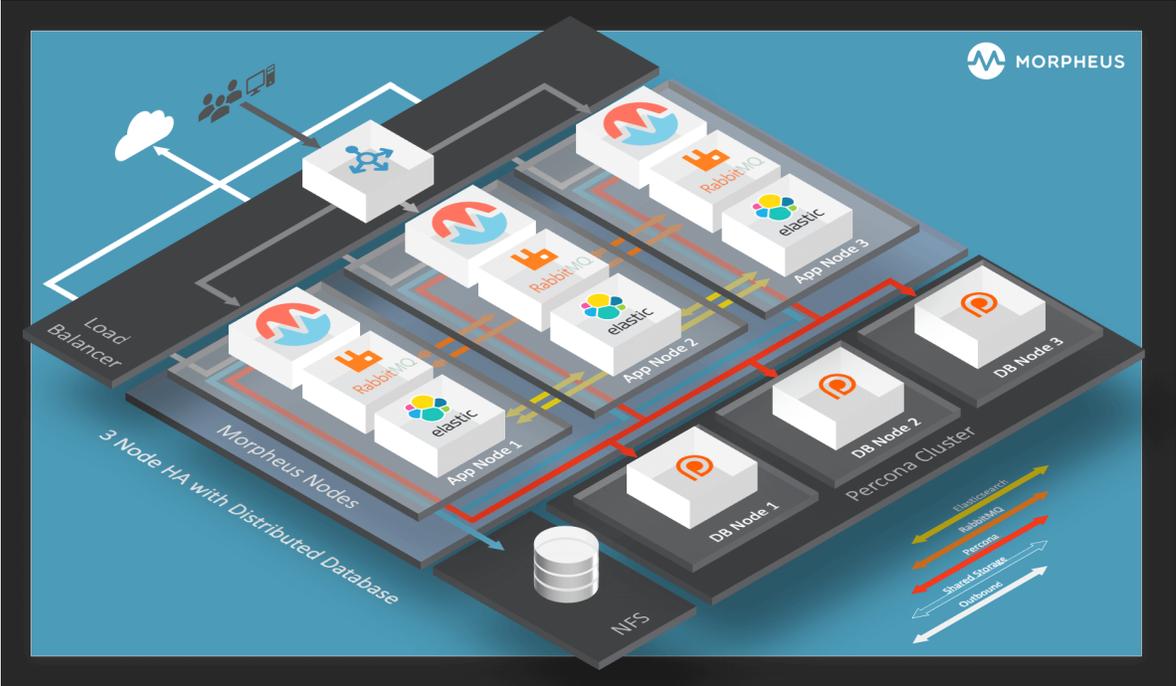
The Morpheus diagram repository contains many diagrams related to Morpheus appliance architectures, Morpheus concepts, and reference diagrams for integrating Morpheus with third-party technologies.

5.11.1 Appliance Architectures

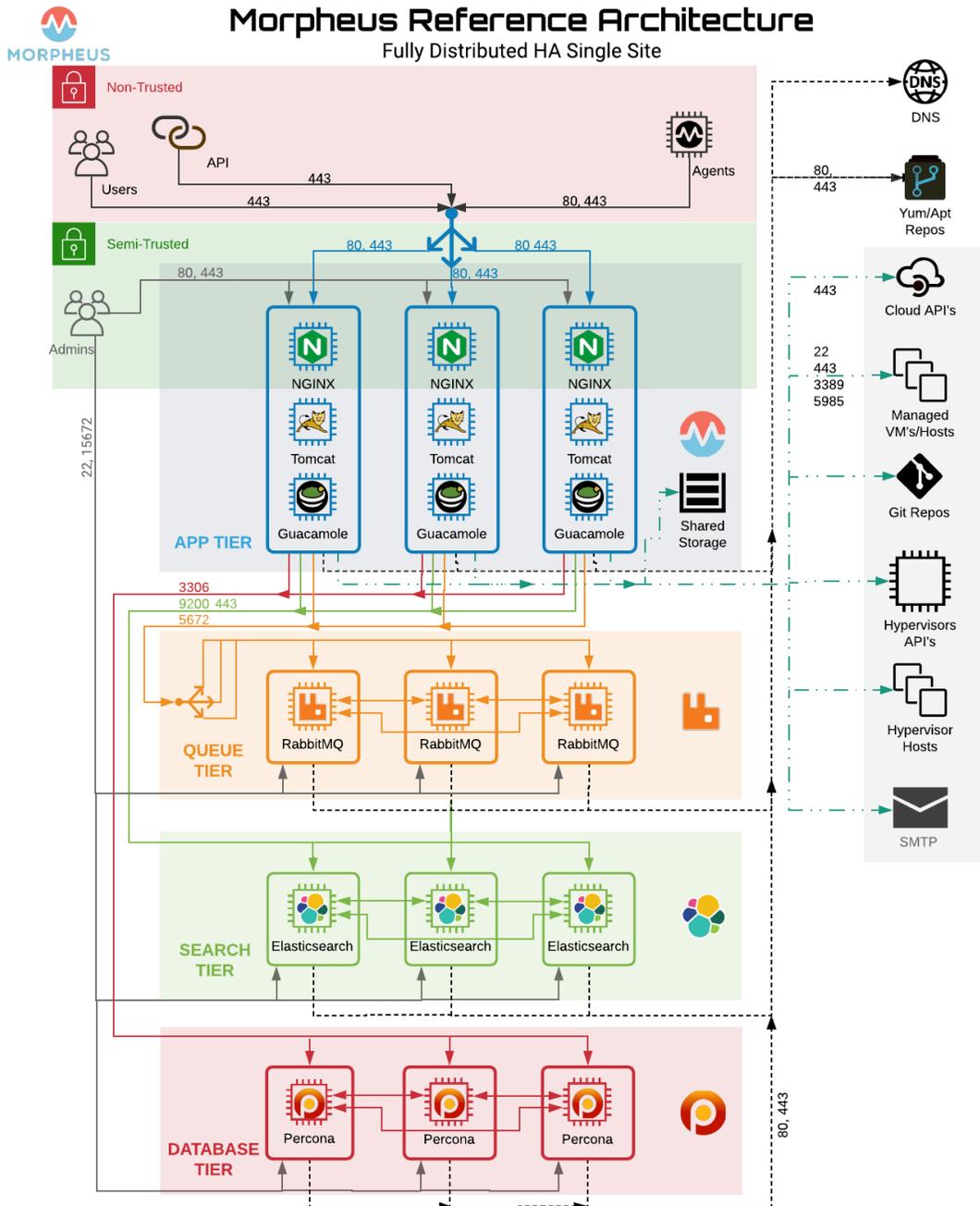
- **All-in-one (single node) Installation**



- **3-node High Availability (HA) Installation**

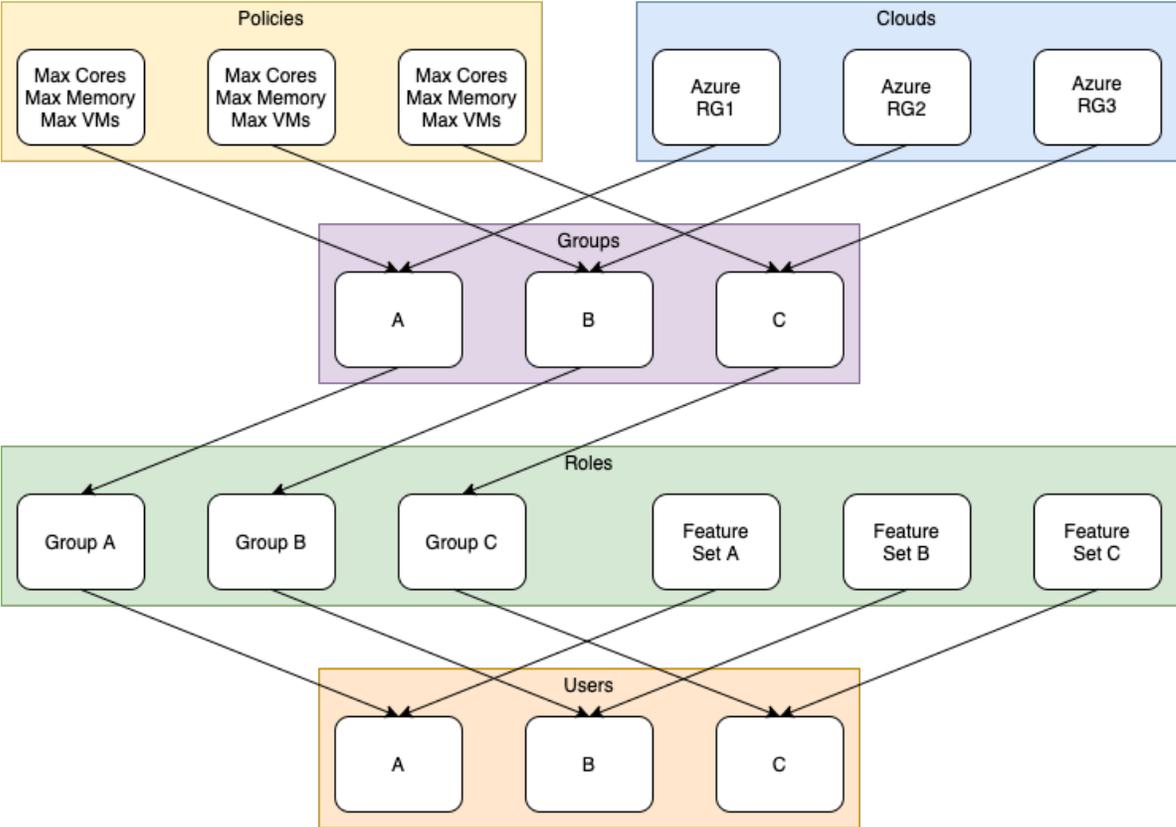


- Fully Distributed High Availability (HA) Installation



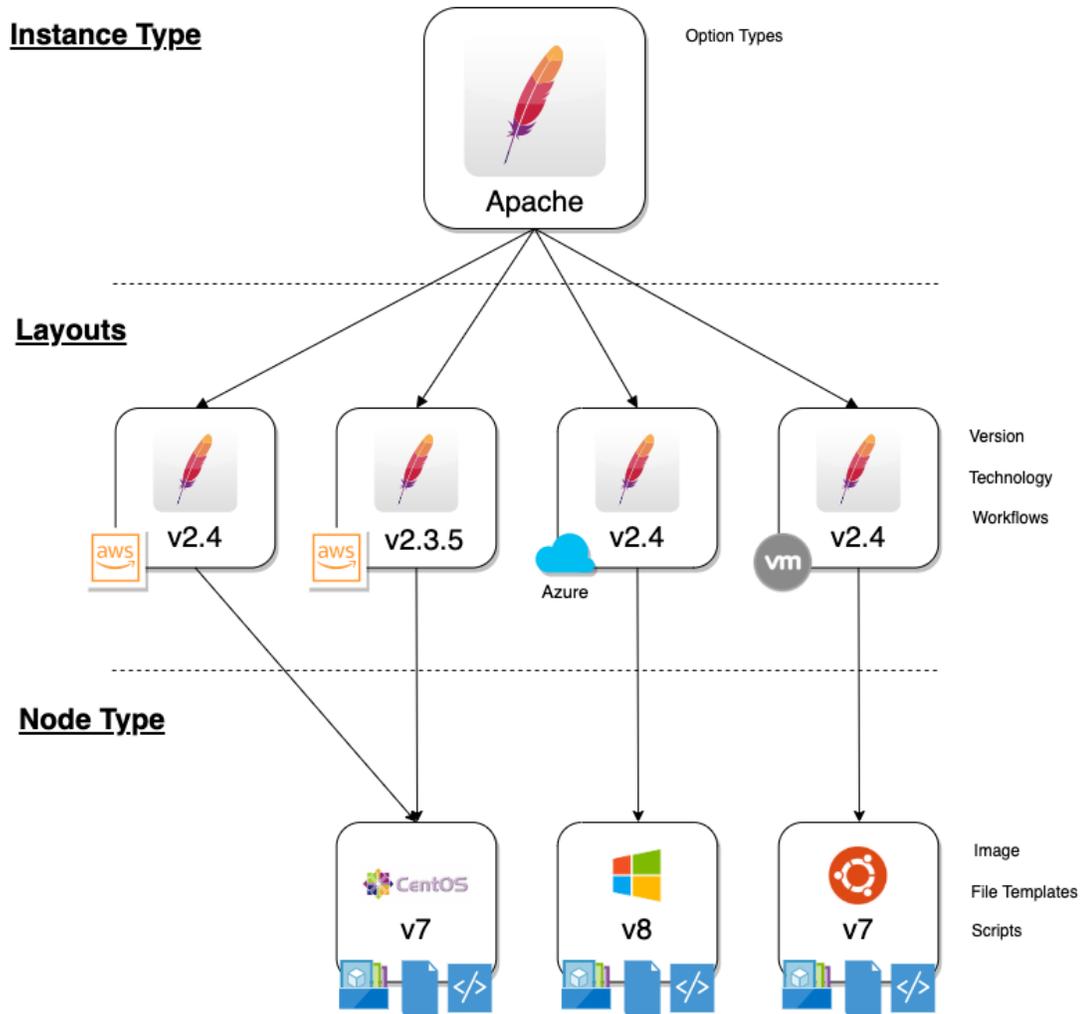
5.11.2 Morpheus Concepts

- **Cloud-Group-Role Relationship**

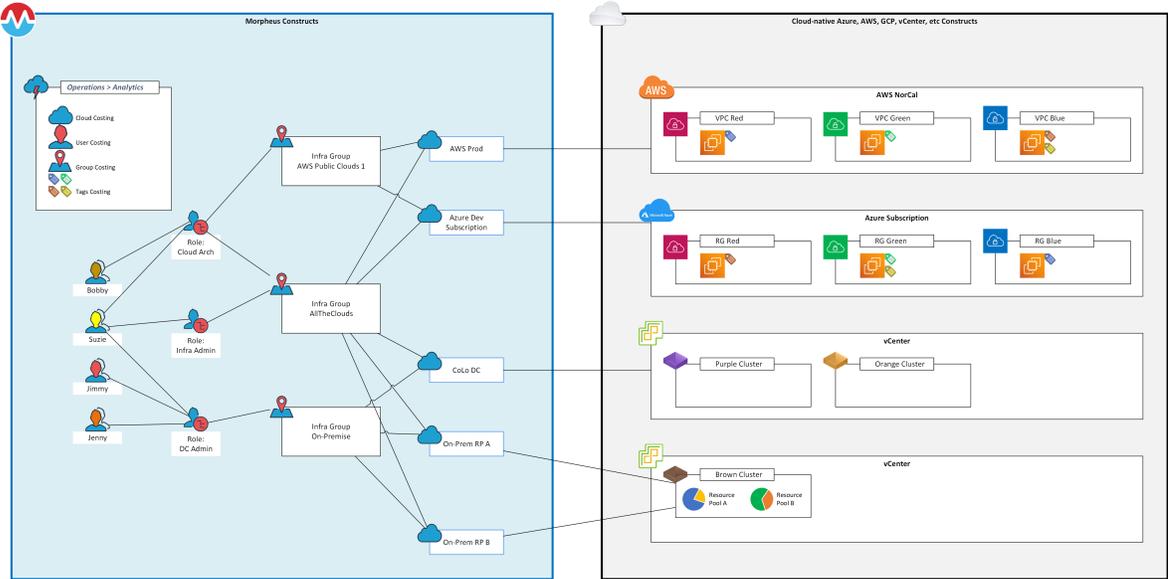


- **Library Items**

Understanding Library Items

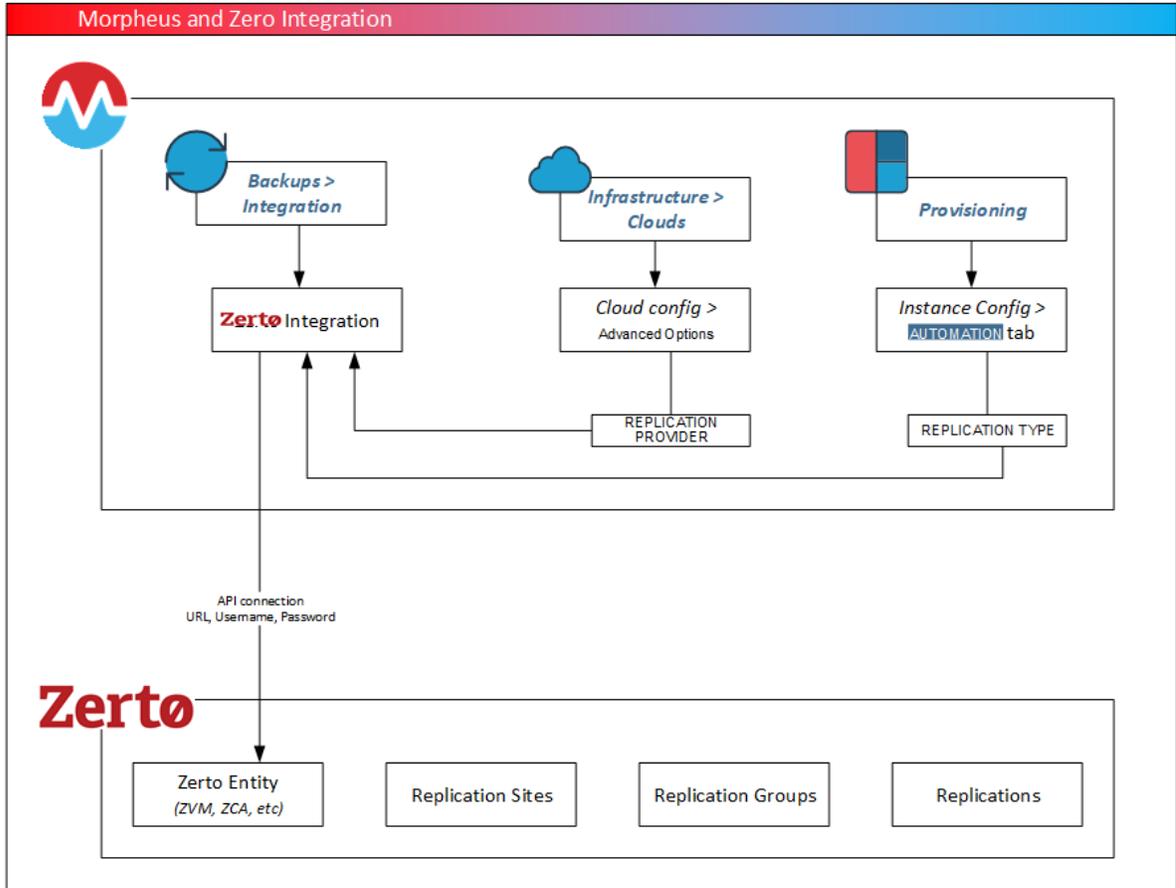


- Mapping Morpheus Constructs to Public and Private Cloud Constructs

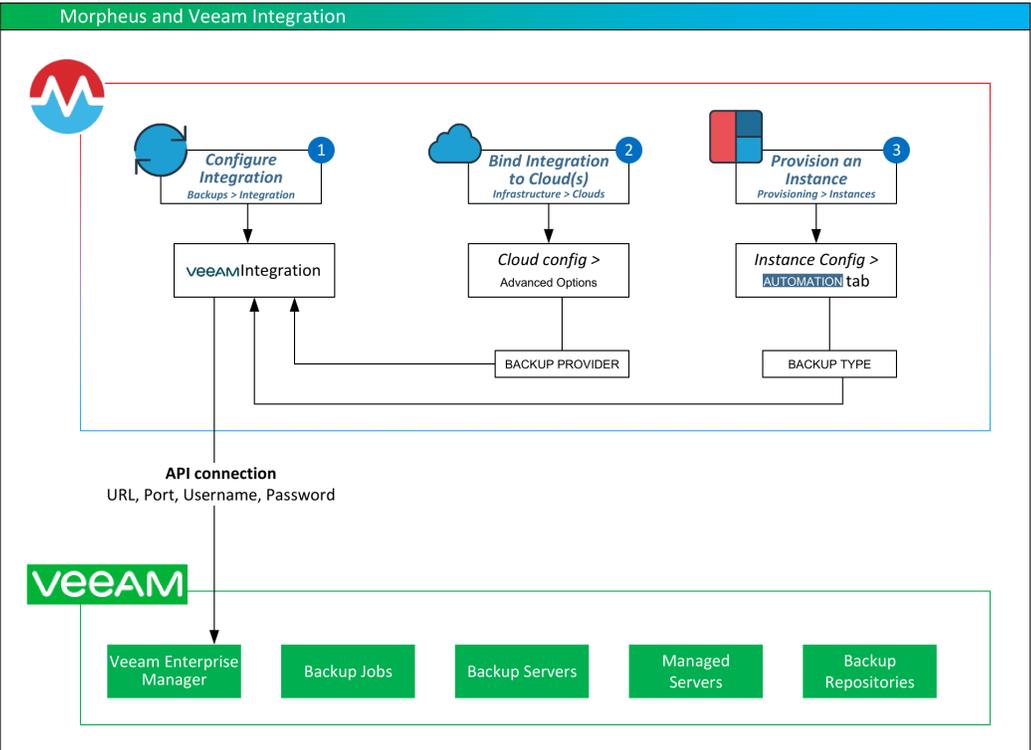


5.11.3 Third-Party Technology Integration

- Zerto

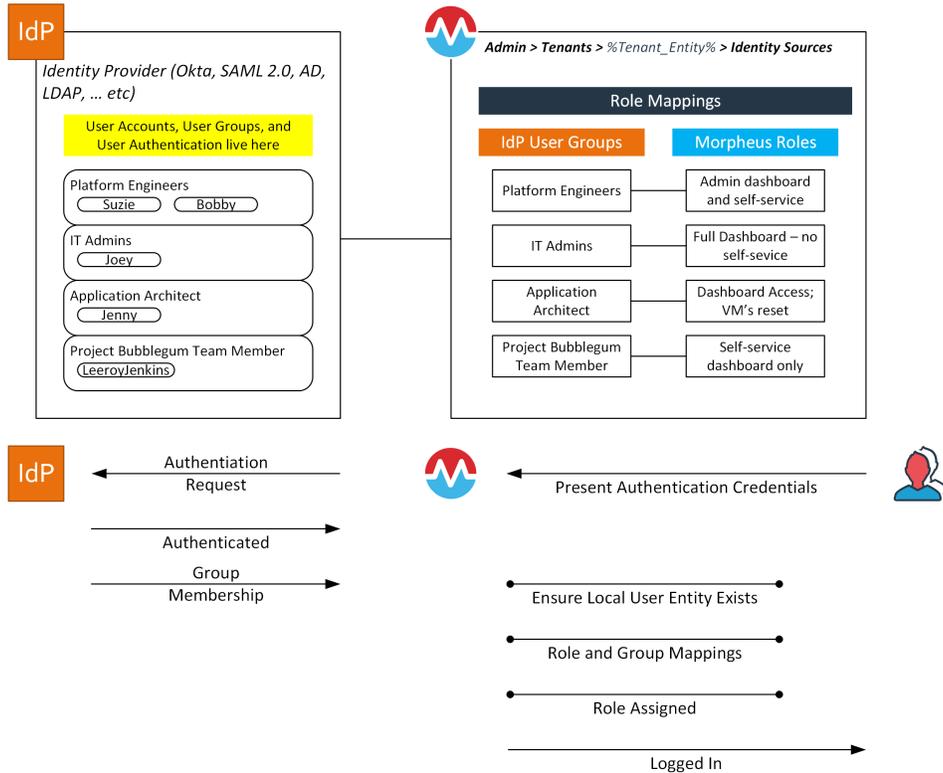


- Veeam



- Mapping Identity Source Groups to Morpheus Roles

Identity Source Group Mapping to Morpheus Roles



5.12 Troubleshooting

5.12.1 logback config

Note: This doc is for 5.4.4+ versions that use `logback.xml`. 5.4.3 and earlier versions use `logback.groovy` with a different syntax that is not compatible with this doc. Please refer to 5.4.3 and earlier documentation for `logback.groovy` configuration details.

The log output for the morpheus-ui service is configured in the `logback.xml` file. Log output levels can be updated when more or less log output is desired.

Setting Log Levels

To change a log level, edit the logback configuration file in `/opt/morpheus/conf/logback.xml` and save. The changes will be reflected within the configured `scanPeriod`, 30 seconds by default.

Levels:

- **OFF** (no log output)
- **ERROR** (includes error logs)
- **WARN** (includes warn and error logs)
- **INFO** (includes info, warn and error logs)
- **DEBUG** (includes info, warn, error and debug logs)
- **TRACE** (includes info, warn, error, debug and trace logs)

Warning: Use DEBUG and/or TRACE levels with caution. DEBUG & TRACE levels can produce many logs that can consume disk space quickly. Only use DEBUG and/or TRACE levels when needed and target them for specific services.

Example Logback Settings

Below are sample log configuration settings. This is not a complete list. Additional log names/paths can typically be determined from the standard INFO, WARN and ERROR logs.

ACI

```
<logger name="com.morpheus.integration.NetworkServersController" level=
↪ "DEBUG"/>
<logger name="com.morpheus.network.AciNetworkService" level="DEBUG"/>
<logger name="com.morpheus.network.AciUtility" level="DEBUG"/>
<logger name="com.morpheus.network.NetworkService " level="DEBUG"/>
```

Amazon

```
<logger name="com.morpheus.compute.amazon.AmazonComputeService" level=
↪ "DEBUG"/>
<logger name="com.morpheus.compute.AmazonComputeUtility" level="DEBUG"/>
<logger name="com.morpheus.costing.AmazonCostingService" level="DEBUG"/>
<logger name="com.morpheus.provision.AmazonProvisionService" level="DEBUG"/
↪ >
```

Azure

```
<logger name="com.morpheus.Azure.ServersController" level="DEBUG"/>
<logger name="com.morpheus.AzureSqlServerProvisionService" level="DEBUG"/>
<logger name="com.morpheus.compute.azure.AzureComputeService" level="DEBUG
↪ "/>
<logger name="com.morpheus.compute.AzureComputeUtility" level="DEBUG"/>
<logger name="com.morpheus.compute.AzureCostingService" level="DEBUG"/>
```

Chef

```
<logger name="com.morpheus.automation.ChefClientService" level="DEBUG"/>
<logger name="com.morpheus.automation.ChefService" level="DEBUG"/>
<logger name="com.morpheus.automation.ChefTaskService" level="DEBUG"/>
```

DNS

```
<logger name="com.morpheus.dns.MicrosoftDnsService" level="DEBUG"/>
```

General

```
<logger name="com.morpheus.InstanceService" level="DEBUG"/>
<logger name="com.morpheus.util.ApiUtility" level="DEBUG"/>
<logger name="com.morpheus.AppService" level="DEBUG"/>
<logger name="com.morpheus.MorpheusComputeService" level="DEBUG"/>
<logger name="com.morpheus.RpcService" level="DEBUG"/>
<logger name="com.morpheus.network.NetworkService " level="DEBUG"/>
<logger name="com.morpheus.provision.AbstractProvisionService" level="DEBUG"
↪"/>
<logger name="com.morpheus.provision.AbstractBoxProvisionService" level=
↪"DEBUG"/>
<logger name="com.morpheus.compute.ProgressUpdater" level="DEBUG"/>
```

Google

```
<logger name="com.morpheus.compute.google.GoogleComputeService" level=
↪"DEBUG"/>
<logger name="com.morpheus.compute.GoogleComputeUtility" level="DEBUG"/>
<logger name="com.morpheus.provision.GoogleProvisionService" level="DEBUG"/
↪>
```

Hyper-V

```
<logger name="com.morpheus.compute.hyperv.HypervComputeService" level=
↪"DEBUG" />
<logger name="com.morpheus.compute.HypervComputeUtility" level="DEBUG" />
```

IBM Cloud

```
<logger name="com.morpheus.compute.softlayer.SoftlayerComputeService" ↪
↪level="DEBUG"/>
<logger name="com.morpheus.compute.SoftlayerComputeUtility" level="DEBUG"/>
```

Kubernetes

```
<logger name="com.morpheus.app.KubernetesAppTemplateService" level="DEBUG"/
↪>
<logger name="com.morpheus.app.KubernetesResourceMappingService" level=
↪"DEBUG"/>
<logger name="com.morpheus.compute.KubernetesComputeService" level="DEBUG"/
↪>
<logger name="com.morpheus.host.KubernetesHostService" level="DEBUG"/>
<logger name="com.morpheus.provision.KubernetesProvisionService" level=
↪"DEBUG"/>
<logger name="com.morpheus.storage.KubernetesStorageService" level="DEBUG"/
↪>
```

(continues on next page)

(continued from previous page)

Monitoring

```
<logger name="com.morpheus.monitoring.MonitorCheckService" level="DEBUG"/>
```

Network

```
<logger name="com.morpheusdata.infoblox.InfobloxProvider" level="DEBUG"/>
<logger name="com.morpheus.network.InfobloxNetworkPoolService" level="DEBUG"
↪"/>
<logger name="com.morpheus.network.NetworkService " level="DEBUG"/>
<logger name="com.morpheus.network.PluginNetworkPoolService" level="DEBUG"/
↪>
```

Nutanix

```
<logger name="com.morpheus.compute.nutanix.NutanixComputeService" level=
↪"DEBUG"/>
<logger name="com.morpheus.compute.NutanixComputeUtility" level="DEBUG"/>
<logger name="com.morpheus.provision.NutanixProvisionService" level="DEBUG
↪"/>
```

Openstack

```
<logger name="com.morpheus.compute.AbstractOpenStackComputeService" level=
↪"DEBUG"/>
<logger name="com.morpheus.compute.AbstractOpenStackComputeUtility" level=
↪"DEBUG"/>
<logger name="com.morpheus.provision.OpenStackProvisionService" level=
↪"DEBUG"/>
<logger name="com.morpheus.storage.OpenStackSFSSStorageService" level="DEBUG
↪"/>
```

Option Types

```
<logger name="com.morpheus.OptionSourceService" level="DEBUG"/>
<logger name="com.morpheus.OptionTypeListService" level="DEBUG"/>
<logger name="com.morpheus.OptionTypeService" level="DEBUG"/>
```

Remote Console

```
<logger name="com.morpheus.remote.MorpheusGuacamoleWebsocketHandler" level=
↪"DEBUG"/>
```

SCVMM

```
<logger name="com.morpheus.compute.scvmm.ScvmmComputeService" level="DEBUG
↪"/>
<logger name="com.morpheus.compute.ScvmmComputeUtility" level="DEBUG"/>
<logger name="com.morpheus.provision.ScvmmProvisionService" level="DEBUG"/>
```

ServiceNow

```
<logger name="com.morpheus.cmdb.ServiceNowCmdbService" level="DEBUG"/>
<logger name="com.morpheus.integrations.ServiceNowUtility" level="DEBUG"/>
```

Tasks

```
<logger name="com.morpheus.task.AnsibleTowerTaskService" level="DEBUG"/>
<logger name="com.morpheus.task.TaskService" level="DEBUG"/>
<logger name="com.morpheus.task.WinrmTaskService" level="DEBUG"/>
```

Terraform

```
<logger name="com.morpheus.app.AbstractResourceMappingService" level="DEBUG"
↳"/>
<logger name="com.morpheus.app.TerraformAppTemplateService" level="DEBUG"/>
<logger name="com.morpheus.app.TerraformAwsResourceMappingService" level=
↳"DEBUG"/>
<logger name="com.morpheus.app.TerraformResourceMappingService" level=
↳"DEBUG"/>
<logger name="com.morpheus.provision.TerraformProvisionService" level=
↳"DEBUG"/>
```

Usage

```
<logger name="com.morpheus.AccountPriceService" level="DEBUG"/>
```

vCloud

```
<logger name="com.morpheus.compute.vmware.VcloudDirectorComputeService"
↳level="DEBUG"/>
<logger name="com.morpheus.provision.VcloudDirectorProvisionService" level=
↳"DEBUG"/>
<logger name="com.morpheus.compute.VcdComputeUtility" level="DEBUG"/>
```

Veem

```
<logger name="com.morpheus.backup.VeeamBackupService" level="DEBUG"/>
```

Vmware

```
<logger name="com.morpheus.compute.VmwareComputeUtility" level="DEBUG"/>
<logger name="com.morpheus.compute.vmware.VmwareComputeService" level=
↳"DEBUG"/>
<logger name="com.morpheus.provision.VmwareProvisionService" level="DEBUG"/
↳>
```

vRO

```
<logger name="com.morpheus.automation.VroService" level="DEBUG"/>
```

All core logger paths

Expand below to see all core Morpheus logger paths set to INFO level.

All core logger paths [Click to Expand/Hide](#)

```

<logger name="com.bertramlabs.plugins.AccountsAuthService" level="INFO"/>
<logger name="com.bertramlabs.plugins.AccountsService" level="INFO"/>
<logger name="com.bertramlabs.plugins.ActiveDirectoryUserService" level="INFO"/>
↪>
<logger name="com.bertramlabs.plugins.AzureSamlUserService" level="INFO"/>
<logger name="com.bertramlabs.plugins.CustomApiUserService" level="INFO"/>
<logger name="com.bertramlabs.plugins.CustomExternalUserService" level="INFO"/>
<logger name="com.bertramlabs.plugins.DefaultUserService" level="INFO"/>
<logger name="com.bertramlabs.plugins.JumpCloudUserService" level="INFO"/>
<logger name="com.bertramlabs.plugins.LdapUserService" level="INFO"/>
<logger name="com.bertramlabs.plugins.OktaUserService" level="INFO"/>
<logger name="com.bertramlabs.plugins.OneLoginUserService" level="INFO"/>
<logger name="com.bertramlabs.plugins.PingUserService" level="INFO"/>
<logger name="com.bertramlabs.plugins.SamlUserService" level="INFO"/>
<logger name="com.bertramlabs.plugins.UserSourceAuthenticationProvider" level=
↪ "INFO"/>
<logger name="com.morpheus.AbstractComputeService" level="INFO"/>
<logger name="com.morpheus.AbstractPriceManagerService" level="INFO"/>
<logger name="com.morpheus.AccountBudgetService" level="INFO"/>
<logger name="com.morpheus.AccountIntegrationObjectService" level="INFO"/>
<logger name="com.morpheus.AccountIntegrationService" level="INFO"/>
<logger name="com.morpheus.AccountInvoiceService" level="INFO"/>
<logger name="com.morpheus.AccountPriceService" level="INFO"/>
<logger name="com.morpheus.AccountResourceService" level="INFO"/>
<logger name="com.morpheus.AccountUsageService" level="INFO"/>
<logger name="com.morpheus.ActivityService" level="INFO"/>
<logger name="com.morpheus.analytics.AbstractAnalyticsService" level="INFO"/>
<logger name="com.morpheus.analytics.AmazonConvertibleRiAnalyticsService"
↪ level="INFO"/>
<logger name="com.morpheus.analytics.CostAnalyticsService" level="INFO"/>
<logger name="com.morpheus.analytics.UtilizationAnalyticsService" level="INFO"/>
↪>
<logger name="com.morpheus.analytics.WorkloadAnalyticsService" level="INFO"/>
<logger name="com.morpheus.AnalyticsService" level="INFO"/>
<logger name="com.morpheus.api.AbstractApiService" level="INFO"/>
<logger name="com.morpheus.api.agent.CommandService" level="INFO"/>
<logger name="com.morpheus.api.agent.DownloadService" level="INFO"/>
<logger name="com.morpheus.api.agent.UploadService" level="INFO"/>
<logger name="com.morpheus.app.AbstractAppTemplateService" level="INFO"/>
<logger name="com.morpheus.app.AbstractResourceMappingService" level="INFO"/>
<logger name="com.morpheus.app.AppTemplateService" level="INFO"/>
<logger name="com.morpheus.app.HelmAppTemplateService" level="INFO"/>
<logger name="com.morpheus.app.KubernetesAppTemplateService" level="INFO"/>
<logger name="com.morpheus.app.KubernetesResourceMappingService" level="INFO"/>
<logger name="com.morpheus.app.MorpheusAppTemplateService" level="INFO"/>
<logger name="com.morpheus.app.ScribeResourceMappingService" level="INFO"/>
<logger name="com.morpheus.app.TerraformAppTemplateService" level="INFO"/>
<logger name="com.morpheus.app.TerraformAwsResourceMappingService" level="INFO
↪"/>

```

(continues on next page)

(continued from previous page)

```
<logger name="com.morpheus.app.TerraformAzurermResourceMappingService" level=
↳ "INFO"/>
<logger name="com.morpheus.app.TerraformGoogleResourceMappingService" level=
↳ "INFO"/>
<logger name="com.morpheus.app.TerraformResourceMappingService" level="INFO"/>
<logger name="com.morpheus.app.TerraformVsphereResourceMappingService" level=
↳ "INFO"/>
<logger name="com.morpheus.ApplianceClientService" level="INFO"/>
<logger name="com.morpheus.ApplianceDelayedJobService" level="INFO"/>
<logger name="com.morpheus.ApplianceHealthService" level="INFO"/>
<logger name="com.morpheus.ApplianceJobService" level="INFO"/>
<logger name="com.morpheus.ApplianceLicenseService" level="INFO"/>
<logger name="com.morpheus.ApplianceService" level="INFO"/>
<logger name="com.morpheus.ApplianceStorageService" level="INFO"/>
<logger name="com.morpheus.approval.ApprovalService" level="INFO"/>
<logger name="com.morpheus.approval.RemedyApprovalService" level="INFO"/>
<logger name="com.morpheus.approval.ServiceNowApprovalService" level="INFO"/>
<logger name="com.morpheus.AppService" level="INFO"/>
<logger name="com.morpheus.ArchiveService" level="INFO"/>
<logger name="com.morpheus.AsyncService" level="INFO"/>
<logger name="com.morpheus.AuditLogService" level="INFO"/>
<logger name="com.morpheus.automation.AbstractConfigManagementService" level=
↳ "INFO"/>
<logger name="com.morpheus.automation.AnsibleService" level="INFO"/>
<logger name="com.morpheus.automation.AnsibleTowerService" level="INFO"/>
<logger name="com.morpheus.automation.ChefService" level="INFO"/>
<logger name="com.morpheus.automation.ConfigManagementService" level="INFO"/>
<logger name="com.morpheus.automation.HelmService" level="INFO"/>
<logger name="com.morpheus.automation.PuppetService" level="INFO"/>
<logger name="com.morpheus.automation.SaltStackService" level="INFO"/>
<logger name="com.morpheus.automation.VroService" level="INFO"/>
<logger name="com.morpheus.backup.AbstractBackupExecutionService" level="INFO"/
↳ >
<logger name="com.morpheus.backup.AbstractBackupJobService" level="INFO"/>
<logger name="com.morpheus.backup.AbstractBackupProviderService" level="INFO"/>
<logger name="com.morpheus.backup.AbstractBackupRestoreService" level="INFO"/>
<logger name="com.morpheus.backup.AbstractBackupService" level="INFO"/>
<logger name="com.morpheus.backup.AbstractReplicationService" level="INFO"/>
<logger name="com.morpheus.backup.BackupExecutionInterface" level="INFO"/>
<logger name="com.morpheus.backup.BackupInterface" level="INFO"/>
<logger name="com.morpheus.backup.BackupJobInterface" level="INFO"/>
<logger name="com.morpheus.backup.BackupJobService" level="INFO"/>
<logger name="com.morpheus.backup.BackupProviderInterface" level="INFO"/>
<logger name="com.morpheus.backup.BackupProviderService" level="INFO"/>
<logger name="com.morpheus.backup.BackupRestoreInterface" level="INFO"/>
<logger name="com.morpheus.backup.BackupRestoreService" level="INFO"/>
<logger name="com.morpheus.backup.BackupService" level="INFO"/>
<logger name="com.morpheus.backup.BackupStatus" level="INFO"/>
<logger name="com.morpheus.backup.BackupStorageService" level="INFO"/>
<logger name="com.morpheus.backup.DirectoryBackupService" level="INFO"/>
<logger name="com.morpheus.backup.FileBackupService" level="INFO"/>
<logger name="com.morpheus.backup.KarmanStorageProviderBackupService" level=
↳ "INFO"/>
```

(continues on next page)

(continued from previous page)

```

<logger name="com.morpheus.backup.LvmImageBackupService" level="INFO"/>
<logger name="com.morpheus.backup.LvmSnapshotBackupService" level="INFO"/>
<logger name="com.morpheus.backup.MorpheusApplianceBackupService" level="INFO"/
↪>
<logger name="com.morpheus.backup.MorpheusBackupService" level="INFO"/>
<logger name="com.morpheus.backup.MorpheusContainerBackupService" level="INFO"/
↪>
<logger name="com.morpheus.backup.MysqlBackupService" level="INFO"/>
<logger name="com.morpheus.backup.PluginBackupExecutionService" level="INFO"/>
<logger name="com.morpheus.backup.PluginBackupJobService" level="INFO"/>
<logger name="com.morpheus.backup.PluginBackupProviderService" level="INFO"/>
<logger name="com.morpheus.backup.PluginBackupRestoreService" level="INFO"/>
<logger name="com.morpheus.backup.PluginReplicationService" level="INFO"/>
<logger name="com.morpheus.backup.ReplicationInterface" level="INFO"/>
<logger name="com.morpheus.backup.ReplicationService" level="INFO"/>
<logger name="com.morpheus.backup.SqlserverBackupService" level="INFO"/>
<logger name="com.morpheus.backup.TarDirectoryBackupService" level="INFO"/>
<logger name="com.morpheus.BootMacService" level="INFO"/>
<logger name="com.morpheus.builds.AbstractBuildsService" level="INFO"/>
<logger name="com.morpheus.builds.BuildsService" level="INFO"/>
<logger name="com.morpheus.builds.JenkinsBuildsService" level="INFO"/>
<logger name="com.morpheus.CapacityService" level="INFO"/>
<logger name="com.morpheus.CatalogCartService" level="INFO"/>
<logger name="com.morpheus.CatalogItemService" level="INFO"/>
<logger name="com.morpheus.CatalogItemTypeService" level="INFO"/>
<logger name="com.morpheus.certificate.AbstractCertificateService" level="INFO
↪"/>
<logger name="com.morpheus.certificate.MorpheusCertificateService" level="INFO
↪"/>
<logger name="com.morpheus.CertificateService" level="INFO"/>
<logger name="com.morpheus.ChefClientService" level="INFO"/>
<logger name="com.morpheus.cm.ChangeManagementService" level="INFO"/>
<logger name="com.morpheus.cm.CherwellCmService" level="INFO"/>
<logger name="com.morpheus.cmdb.AbstractCmdbService" level="INFO"/>
<logger name="com.morpheus.cmdb.CmdbService" level="INFO"/>
<logger name="com.morpheus.cmdb.RemedyCmdbService" level="INFO"/>
<logger name="com.morpheus.cmdb.ServiceNowCmdbService" level="INFO"/>
<logger name="com.morpheus.code.AbstractCodeService" level="INFO"/>
<logger name="com.morpheus.code.CodeService" level="INFO"/>
<logger name="com.morpheus.code.GitCodeService" level="INFO"/>
<logger name="com.morpheus.code.GithubCodeService" level="INFO"/>
<logger name="com.morpheus.compliance.NVDSyncService" level="INFO"/>
<logger name="com.morpheus.compliance.PackageManagementService" level="INFO"/>
<logger name="com.morpheus.compute.cisco.UcsComputeService" level="INFO"/>
<logger name="com.morpheus.compute.CloudPluginComputeService" level="INFO"/>
<logger name="com.morpheus.compute.ComputeApiService" level="INFO"/>
<logger name="com.morpheus.compute.ComputeServiceInterface" level="INFO"/>
<logger name="com.morpheus.compute.IpmiService" level="INFO"/>
<logger name="com.morpheus.compute.KubernetesComputeService" level="INFO"/>
<logger name="com.morpheus.compute.MaasComputeService" level="INFO"/>
<logger name="com.morpheus.compute.ManualComputeService" level="INFO"/>
<logger name="com.morpheus.compute.OneviewComputeService" level="INFO"/>

```

(continues on next page)

(continued from previous page)

```
<logger name="com.morpheus.compute.SelfManagedComputeService" level="INFO"/>
<logger name="com.morpheus.compute.standard.StandardComputeService" level="INFO
↪"/>
<logger name="com.morpheus.compute.unmanaged.UnmanagedComputeService" level=
↪"INFO"/>
<logger name="com.morpheus.compute.vmware.VcloudDirectorComputeService" level=
↪"INFO"/>
<logger name="com.morpheus.compute.vmware.VmwareComputeService" level="INFO"/>
<logger name="com.morpheus.ComputeService" level="INFO"/>
<logger name="com.morpheus.container.ActivemqContainerService" level="INFO"/>
<logger name="com.morpheus.container.DockerContainerService" level="INFO"/>
<logger name="com.morpheus.container.DockerContainerUpgradeService" level="INFO
↪"/>
<logger name="com.morpheus.container.ElasticsearchContainerService" level="INFO
↪"/>
<logger name="com.morpheus.container.JavaContainerService" level="INFO"/>
<logger name="com.morpheus.container.MysqlContainerService" level="INFO"/>
<logger name="com.morpheus.container.NodeContainerService" level="INFO"/>
<logger name="com.morpheus.container.PostgresContainerService" level="INFO"/>
<logger name="com.morpheus.container.RedisContainerService" level="INFO"/>
<logger name="com.morpheus.container.SqlserverContainerService" level="INFO"/>
<logger name="com.morpheus.ContainerScriptService" level="INFO"/>
<logger name="com.morpheus.ContainerService" level="INFO"/>
<logger name="com.morpheus.costing.AbstractCostingService" level="INFO"/>
<logger name="com.morpheus.costing.CostingInterface" level="INFO"/>
<logger name="com.morpheus.costing.CostingService" level="INFO"/>
<logger name="com.morpheus.costing.StandardCostingService" level="INFO"/>
<logger name="com.morpheus.CurrencyConversionService" level="INFO"/>
<logger name="com.morpheus.cypher.CypherGORMDatastoreService" level="INFO"/>
<logger name="com.morpheus.cypher.CypherService" level="INFO"/>
<logger name="com.morpheus.DashboardService" level="INFO"/>
<logger name="com.morpheus.DatastoreService" level="INFO"/>
<logger name="com.morpheus.DataViewService" level="INFO"/>
<logger name="com.morpheus.DbSchedulerService" level="INFO"/>
<logger name="com.morpheus.deploy.AbstractDeployService" level="INFO"/>
<logger name="com.morpheus.deploy.AbstractDeployTargetService" level="INFO"/>
<logger name="com.morpheus.deploy.CloudFoundryAppDeployService" level="INFO"/>
<logger name="com.morpheus.deploy.DefaultDeployService" level="INFO"/>
<logger name="com.morpheus.deploy.DockerDeployTargetService" level="INFO"/>
<logger name="com.morpheus.deploy.GrailsDeployService" level="INFO"/>
<logger name="com.morpheus.deploy.IisDeployService" level="INFO"/>
<logger name="com.morpheus.deploy.JbossDeployService" level="INFO"/>
<logger name="com.morpheus.deploy.KubernetesDeployTargetService" level="INFO"/>
<logger name="com.morpheus.deploy.NodeDeployService" level="INFO"/>
<logger name="com.morpheus.deploy.ServerDeployTargetService" level="INFO"/>
<logger name="com.morpheus.deploy.VmDeployTargetService" level="INFO"/>
<logger name="com.morpheus.DeploymentService" level="INFO"/>
<logger name="com.morpheus.discovery.AbstractDiscoveryService" level="INFO"/>
<logger name="com.morpheus.discovery.DatastoreCapacityDiscoveryService" level=
↪"INFO"/>
<logger name="com.morpheus.discovery.DiscoveryService" level="INFO"/>
<logger name="com.morpheus.discovery.HostBalancingDiscoveryService" level="INFO
↪"/>
```

(continues on next page)

(continued from previous page)

```

<logger name="com.morpheus.discovery.HostCapacityDiscoveryService" level="INFO
↳"/>
<logger name="com.morpheus.discovery.ReservationRecommendationDiscoveryService
↳" level="INFO"/>
<logger name="com.morpheus.discovery.ShutdownDiscoveryService" level="INFO"/>
<logger name="com.morpheus.discovery.SizeDiscoveryService" level="INFO"/>
<logger name="com.morpheus.dns.AbstractDnsService" level="INFO"/>
<logger name="com.morpheus.dns.BindDnsService" level="INFO"/>
<logger name="com.morpheus.dns.ConsulDnsService" level="INFO"/>
<logger name="com.morpheus.dns.DNSProvider" level="INFO"/>
<logger name="com.morpheus.dns.DnsService" level="INFO"/>
<logger name="com.morpheus.dns.MicrosoftDnsService" level="INFO"/>
<logger name="com.morpheus.dns.PluginDnsService" level="INFO"/>
<logger name="com.morpheus.dns.PowerDnsService" level="INFO"/>
<logger name="com.morpheus.ElasticCleanupService" level="INFO"/>
<logger name="com.morpheus.EnvironmentService" level="INFO"/>
<logger name="com.morpheus.EnvironmentVariableService" level="INFO"/>
<logger name="com.morpheus.ExecuteScheduleTypeService" level="INFO"/>
<logger name="com.morpheus.ExecutionRequestService" level="INFO"/>
<logger name="com.morpheus.export.AccountInvoiceExportService" level="INFO"/>
<logger name="com.morpheus.export.CodeRepositoryExportService" level="INFO"/>
<logger name="com.morpheus.export.DeploymentExportService" level="INFO"/>
<logger name="com.morpheus.export.ExecuteScheduleTypeExportService" level="INFO
↳"/>
<logger name="com.morpheus.export.ExportService" level="INFO"/>
<logger name="com.morpheus.export.InstanceExportService" level="INFO"/>
<logger name="com.morpheus.export.integrations.AdminIntegrationExportService"
↳
↳level="INFO"/>
<logger name="com.morpheus.export.integrations.
↳AutomationIntegrationExportService" level="INFO"/>
<logger name="com.morpheus.export.integrations.BackupIntegrationExportService"
↳
↳level="INFO"/>
<logger name="com.morpheus.export.integrations.
↳CertificateIntegrationExportService" level="INFO"/>
<logger name="com.morpheus.export.integrations.DeployIntegrationExportService"
↳
↳level="INFO"/>
<logger name="com.morpheus.export.integrations.NetworkIntegrationExportService
↳" level="INFO"/>
<logger name="com.morpheus.export.LoadBalancerExpertService" level="INFO"/>
<logger name="com.morpheus.export.LoadBalancerInstancesExportService" level=
↳
↳"INFO"/>
<logger name="com.morpheus.export.NetworkDomainExportService" level="INFO"/>
<logger name="com.morpheus.export.NetworkExportService" level="INFO"/>
<logger name="com.morpheus.export.NetworkGroupExportService" level="INFO"/>
<logger name="com.morpheus.export.NetworkPoolExportService" level="INFO"/>
<logger name="com.morpheus.export.NetworkRouterExportService" level="INFO"/>
<logger name="com.morpheus.export.NetworkSecurityGroupExportService" level=
↳
↳"INFO"/>
<logger name="com.morpheus.export.PowerScheduleTypeExportService" level="INFO"/
↳
↳>
<logger name="com.morpheus.export.ServerExportService" level="INFO"/>
<logger name="com.morpheus.export.ServerGroupExportService" level="INFO"/>

```

(continues on next page)

(continued from previous page)

```
<logger name="com.morpheus.export.ServicePlanExportService" level="INFO"/>
<logger name="com.morpheus.export.TaskExportService" level="INFO"/>
<logger name="com.morpheus.export.ThresholdExportService" level="INFO"/>
<logger name="com.morpheus.export.UserExportService" level="INFO"/>
<logger name="com.morpheus.export.UserGroupExportService" level="INFO"/>
<logger name="com.morpheus.export.WorkflowExportService" level="INFO"/>
<logger name="com.morpheus.FileCopyRequestService" level="INFO"/>
<logger name="com.morpheus.GlobalSearchService" level="INFO"/>
<logger name="com.morpheus.host.AbstractHostService" level="INFO"/>
<logger name="com.morpheus.host.DockerHostService" level="INFO"/>
<logger name="com.morpheus.host.ExternalKubernetesHostService" level="INFO"/>
<logger name="com.morpheus.host.KubernetesHostService" level="INFO"/>
<logger name="com.morpheus.host.SwarmHostService" level="INFO"/>
<logger name="com.morpheus.HttpClientService" level="INFO"/>
<logger name="com.morpheus.hub.MorpheusHubQueueService" level="INFO"/>
<logger name="com.morpheus.hub.MorpheusHubService" level="INFO"/>
<logger name="com.morpheus.hub.MorpheusHubSyncService" level="INFO"/>
<logger name="com.morpheus.imagebuild.ImageBuildService" level="INFO"/>
<logger name="com.morpheus.ImageCacheService" level="INFO"/>
<logger name="com.morpheus.instance.InstanceUpgradeService" level="INFO"/>
<logger name="com.morpheus.InstanceService" level="INFO"/>
<logger name="com.morpheus.InstanceTypeService" level="INFO"/>
<logger name="com.morpheus.integration.AbstractIntegrationService" level="INFO
↵"/>
<logger name="com.morpheus.integration.CherwellIntegrationService" level="INFO
↵"/>
<logger name="com.morpheus.integration.GitRepoService" level="INFO"/>
<logger name="com.morpheus.integration.RemedyIntegrationService" level="INFO"/>
<logger name="com.morpheus.integration.RunDeckIntegrationService" level="INFO"/
↵>
<logger name="com.morpheus.integration.SalesForceIntegrationService" level=
↵"INFO"/>
<logger name="com.morpheus.integration.ScribeService" level="INFO"/>
<logger name="com.morpheus.integration.ServiceNowIntegrationService" level=
↵"INFO"/>
<logger name="com.morpheus.integration.TerraformService" level="INFO"/>
<logger name="com.morpheus.jobs.AbstractJobExecutorService" level="INFO"/>
<logger name="com.morpheus.jobs.JobExecutor" level="INFO"/>
<logger name="com.morpheus.jobs.KubernetesJobExecutorService" level="INFO"/>
<logger name="com.morpheus.jobs.SecurityScanExecutorService" level="INFO"/>
<logger name="com.morpheus.jobs.TaskJobExecutorService" level="INFO"/>
<logger name="com.morpheus.jobs.WorkflowJobExecutorService" level="INFO"/>
<logger name="com.morpheus.JobService" level="INFO"/>
<logger name="com.morpheus.KeyPairService" level="INFO"/>
<logger name="com.morpheus.library.LayoutService" level="INFO"/>
<logger name="com.morpheus.LicenseService" level="INFO"/>
<logger name="com.morpheus.LoadBalancerPriceManagerService" level="INFO"/>
<logger name="com.morpheus.LocalizationService" level="INFO"/>
<logger name="com.morpheus.LocalRepoService" level="INFO"/>
<logger name="com.morpheus.log.AbstractLogService" level="INFO"/>
<logger name="com.morpheus.log.LogRhythmLogService" level="INFO"/>
<logger name="com.morpheus.log.SplunkLogService" level="INFO"/>
```

(continues on next page)

(continued from previous page)

```

<logger name="com.morpheus.log.SyslogLogService" level="INFO"/>
<logger name="com.morpheus.LogService" level="INFO"/>
<logger name="com.morpheus.maint.UpdateService" level="INFO"/>
<logger name="com.morpheus.MarketplaceClientService" level="INFO"/>
<logger name="com.morpheus.MarshallService" level="INFO"/>
<logger name="com.morpheus.MetadataTagService" level="INFO"/>
<logger name="com.morpheus.migration.AbstractMigrationService" level="INFO"/>
<logger name="com.morpheus.migration.HypervisorMigrationService" level="INFO"/>
<logger name="com.morpheus.migration.LvmMigrationService" level="INFO"/>
<logger name="com.morpheus.migration.MigrationService" level="INFO"/>
<logger name="com.morpheus.migration.WindowsMigrationService" level="INFO"/>
<logger name="com.morpheus.monitoring.AlerterService" level="INFO"/>
<logger name="com.morpheus.monitoring.AlertRuleService" level="INFO"/>
<logger name="com.morpheus.monitoring.AvailabilityService" level="INFO"/>
<logger name="com.morpheus.monitoring.CheckAgentService" level="INFO"/>
<logger name="com.morpheus.monitoring.IncidentService" level="INFO"/>
<logger name="com.morpheus.monitoring.MonitorAppService" level="INFO"/>
<logger name="com.morpheus.monitoring.MonitorChartingService" level="INFO"/>
<logger name="com.morpheus.monitoring.MonitorCheckManagementService" level=
↪ "INFO"/>
<logger name="com.morpheus.monitoring.MonitorCheckService" level="INFO"/>
<logger name="com.morpheus.monitoring.MonitoringService" level="INFO"/>
<logger name="com.morpheus.monitoring.MonitorService" level="INFO"/>
<logger name="com.morpheus.monitoring.MorpheusMonitorService" level="INFO"/>
<logger name="com.morpheus.monitoring.NewRelicService" level="INFO"/>
<logger name="com.morpheus.monitoring.ServiceNowService" level="INFO"/>
<logger name="com.morpheus.MorpheusComputeService" level="INFO"/>
<logger name="com.morpheus.MorpheusPackageService" level="INFO"/>
<logger name="com.morpheus.MorpheusSecurityService" level="INFO"/>
<logger name="com.morpheus.MotdService" level="INFO"/>
<logger name="com.morpheus.network.A10LoadBalancerService" level="INFO"/>
<logger name="com.morpheus.network.AbstractLoadBalancerService" level="INFO"/>
<logger name="com.morpheus.network.AbstractNetworkPoolService" level="INFO"/>
<logger name="com.morpheus.network.AbstractNetworkRegistryService" level="INFO
↪ "/>
<logger name="com.morpheus.network.AbstractNetworkSecurityService" level="INFO
↪ "/>
<logger name="com.morpheus.network.AbstractNetworkService" level="INFO"/>
<logger name="com.morpheus.network.AciNetworkSecurityService" level="INFO"/>
<logger name="com.morpheus.network.AciNetworkService" level="INFO"/>
<logger name="com.morpheus.network.AviLoadBalancerService" level="INFO"/>
<logger name="com.morpheus.network.BluecatNetworkPoolService" level="INFO"/>
<logger name="com.morpheus.network.BootService" level="INFO"/>
<logger name="com.morpheus.network.CitrixNetScalerLoadBalancerService" level=
↪ "INFO"/>
<logger name="com.morpheus.network.CloudPluginNetworkService" level="INFO"/>
<logger name="com.morpheus.network.ConsulRegistryService" level="INFO"/>
<logger name="com.morpheus.network.ConsulService" level="INFO"/>
<logger name="com.morpheus.network.F5BigIpLoadBalancerService" level="INFO"/>
<logger name="com.morpheus.network.F5LineRateLoadBalancerService" level="INFO"/
↪ >
<logger name="com.morpheus.network.FirewallService" level="INFO"/>

```

(continues on next page)

(continued from previous page)

```
<logger name="com.morpheus.network.FortiADCLoadBalancerService" level="INFO"/>
<logger name="com.morpheus.network.HaproxyLoadBalancerService" level="INFO"/>
<logger name="com.morpheus.network.InfobloxNetworkPoolService" level="INFO"/>
<logger name="com.morpheus.network.InternalLoadBalancerService" level="INFO"/>
<logger name="com.morpheus.network.InternalNetworkSecurityService" level="INFO"
↳"/>
<logger name="com.morpheus.network.InternalNetworkService" level="INFO"/>
<logger name="com.morpheus.network.IPAMProvider" level="INFO"/>
<logger name="com.morpheus.network.KubernetesRegistryService" level="INFO"/>
<logger name="com.morpheus.network.LoadBalancerService" level="INFO"/>
<logger name="com.morpheus.network.LocalFirewallService" level="INFO"/>
<logger name="com.morpheus.network.MorpheusNetworkPoolService" level="INFO"/>
<logger name="com.morpheus.network.MorpheusRegistryService" level="INFO"/>
<logger name="com.morpheus.network.NetScalerLoadBalancerService" level="INFO"/>
<logger name="com.morpheus.network.NetworkConfigService" level="INFO"/>
<logger name="com.morpheus.network.NetworkPoolService" level="INFO"/>
<logger name="com.morpheus.network.NetworkRegistryService" level="INFO"/>
<logger name="com.morpheus.network.NetworkSecurityService" level="INFO"/>
<logger name="com.morpheus.network.NetworkService" level="INFO"/>
<logger name="com.morpheus.network.NetworkServicesService" level="INFO"/>
<logger name="com.morpheus.network.NutanixNetworkPoolService" level="INFO"/>
<logger name="com.morpheus.network.PaloAltoNetworkService" level="INFO"/>
<logger name="com.morpheus.network.PhipamNetworkPoolService" level="INFO"/>
<logger name="com.morpheus.network.PluginNetworkPoolService" level="INFO"/>
<logger name="com.morpheus.network.PxeService" level="INFO"/>
<logger name="com.morpheus.network.SolarWindsNetworkPoolService" level="INFO"/>
<logger name="com.morpheus.network.StealthNetworkSecurityService" level="INFO"/
↳>
<logger name="com.morpheus.NetworkDomainService" level="INFO"/>
<logger name="com.morpheus.OauthProviderService" level="INFO"/>
<logger name="com.morpheus.OperationEventService" level="INFO"/>
<logger name="com.morpheus.OptionSourcePluginService" level="INFO"/>
<logger name="com.morpheus.OptionSourceService" level="INFO"/>
<logger name="com.morpheus.OptionTypeListService" level="INFO"/>
<logger name="com.morpheus.OptionTypeService" level="INFO"/>
<logger name="com.morpheus.os.LinuxOsService" level="INFO"/>
<logger name="com.morpheus.os.WindowsOsService" level="INFO"/>
<logger name="com.morpheus.PermissionService" level="INFO"/>
<logger name="com.morpheus.plugin.AbstractPluginProviderManagerService" level=
↳"INFO"/>
<logger name="com.morpheus.plugin.backup.BackupProviderPluginManagerService"
↳
↳level="INFO"/>
<logger name="com.morpheus.plugin.backup.MorpheusBackupImplService" level="INFO"
↳"/>
<logger name="com.morpheus.plugin.backup.MorpheusBackupJobImplService" level=
↳"INFO"/>
<logger name="com.morpheus.plugin.backup.MorpheusBackupRestoreImplService"
↳
↳level="INFO"/>
<logger name="com.morpheus.plugin.backup.MorpheusBackupResultImplService"
↳
↳level="INFO"/>
<logger name="com.morpheus.plugin.backup.MorpheusBackupTypeImplService" level=
↳"INFO"/>
```

(continues on next page)

(continued from previous page)

```

<logger name="com.morpheus.plugin.backup.MorpheusReplicationGroupImplService" ↵
↵ level="INFO"/>
<logger name="com.morpheus.plugin.backup.MorpheusReplicationImplService" level=
↵ "INFO"/>
<logger name="com.morpheus.plugin.backup.MorpheusReplicationSiteImplService" ↵
↵ level="INFO"/>
<logger name="com.morpheus.plugin.backup.MorpheusReplicationTypeImplService" ↵
↵ level="INFO"/>
<logger name="com.morpheus.plugin.compute.
↵MorpheusComputeServerInterfaceImplService" level="INFO"/>
<logger name="com.morpheus.plugin.compute.MorpheusComputeZoneFolderImplService
↵ " level="INFO"/>
<logger name="com.morpheus.plugin.compute.MorpheusDatastoreImplService" level=
↵ "INFO"/>
<logger name="com.morpheus.plugin.costing.MorpheusAccountInvoiceImplService" ↵
↵ level="INFO"/>
<logger name="com.morpheus.plugin.costing.MorpheusCostingImplService" level=
↵ "INFO"/>
<logger name="com.morpheus.plugin.cypher.MorpheusCypherImplService" level="INFO
↵ "/>
<logger name="com.morpheus.plugin.integration.
↵MorpheusAccountInventoryImplService" level="INFO"/>
<logger name="com.morpheus.plugin.integration.MorpheusIntegrationImplService" ↵
↵ level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusAccountCredentialImplService" level=
↵ "INFO"/>
<logger name="com.morpheus.plugin.MorpheusAccountCredentialTypeImplService" ↵
↵ level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusCloudImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusComputePortImplService" level="INFO"/
↵
↵
<logger name="com.morpheus.plugin.MorpheusComputeServerImplService" level="INFO
↵ "/>
<logger name="com.morpheus.plugin.MorpheusComputeTypeLayoutFactoryImplService" ↵
↵ level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusComputeTypeSetImplService" level=
↵ "INFO"/>
<logger name="com.morpheus.plugin.MorpheusComputeZonePoolImplService" level=
↵ "INFO"/>
<logger name="com.morpheus.plugin.MorpheusContainerTypeImplService" level="INFO
↵ "/>
<logger name="com.morpheus.plugin.MorpheusContextImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusInstanceImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusMetadataTagImplService" level="INFO"/
↵
↵
<logger name="com.morpheus.plugin.MorpheusMetadataTagTypeImplService" level=
↵ "INFO"/>
<logger name="com.morpheus.plugin.MorpheusOperationNotificationImplService" ↵
↵ level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusOsTypeImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusPermissionImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusProcessImplService" level="INFO"/>

```

(continues on next page)

(continued from previous page)

```
<logger name="com.morpheus.plugin.MorpheusReportImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusServicePlanImplService" level="INFO"/>
↪>
<logger name="com.morpheus.plugin.MorpheusSnapshotImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusStatsImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusStorageControllerImplService" level=
↪"INFO"/>
<logger name="com.morpheus.plugin.MorpheusStorageControllerTypeImplService" ↪
↪level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusStorageVolumeImplService" level="INFO
↪"/>
<logger name="com.morpheus.plugin.MorpheusStorageVolumeTypeImplService" level=
↪"INFO"/>
<logger name="com.morpheus.plugin.MorpheusTaskImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusUsageImplService" level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusVirtualImageImplService" level="INFO
↪"/>
<logger name="com.morpheus.plugin.MorpheusVirtualImageLocationImplService" ↪
↪level="INFO"/>
<logger name="com.morpheus.plugin.MorpheusWikiPageImplService" level="INFO"/>
<logger name="com.morpheus.plugin.network.MorpheusNetworkDomainImplService" ↪
↪level="INFO"/>
<logger name="com.morpheus.plugin.network.
↪MorpheusNetworkDomainRecordImplService" level="INFO"/>
<logger name="com.morpheus.plugin.network.MorpheusNetworkImplService" level=
↪"INFO"/>
<logger name="com.morpheus.plugin.network.MorpheusNetworkPoolImplService" ↪
↪level="INFO"/>
<logger name="com.morpheus.plugin.network.MorpheusNetworkPoolIpImplService" ↪
↪level="INFO"/>
<logger name="com.morpheus.plugin.network.MorpheusNetworkPoolRangeImplService" ↪
↪level="INFO"/>
<logger name="com.morpheus.plugin.network.MorpheusNetworkSubnetImplService" ↪
↪level="INFO"/>
<logger name="com.morpheus.plugin.network.MorpheusNetworkTypeImplService" ↪
↪level="INFO"/>
<logger name="com.morpheus.plugin.PluginManagerService" level="INFO"/>
<logger name="com.morpheus.plugin.PluginProviderManagerService" level="INFO"/>
<logger name="com.morpheus.plugin.policy.MorpheusPolicyImplService" level="INFO
↪"/>
<logger name="com.morpheus.plugin.policy.MorpheusPolicyTypeImplService" level=
↪"INFO"/>
<logger name="com.morpheus.plugin.provisioning.MorpheusProvisionImplService" ↪
↪level="INFO"/>
<logger name="com.morpheus.plugin.web.MorpheusWebRequestImplService" level=
↪"INFO"/>
<logger name="com.morpheus.policy.AbstractPolicyService" level="INFO"/>
<logger name="com.morpheus.policy.BackupStoragePolicyService" level="INFO"/>
<logger name="com.morpheus.policy.MotdPolicyService" level="INFO"/>
<logger name="com.morpheus.policy.NetworkPolicyService" level="INFO"/>
<logger name="com.morpheus.policy.PolicyServiceInterface" level="INFO"/>
<logger name="com.morpheus.policy.StorageBucketQuotaPolicyService" level="INFO
↪"/>
```

(continues on next page)

(continued from previous page)

```

<logger name="com.morpheus.policy.StorageServerQuotaPolicyService" level="INFO"
↪"/>
<logger name="com.morpheus.policy.StorageShareQuotaPolicyService" level="INFO"/
↪>
<logger name="com.morpheus.policy.TagCompliancePolicyService" level="INFO"/>
<logger name="com.morpheus.policy.WorkflowPolicyService" level="INFO"/>
<logger name="com.morpheus.PolicyService" level="INFO"/>
<logger name="com.morpheus.PowerScheduleService" level="INFO"/>
<logger name="com.morpheus.PowerScheduleTypeService" level="INFO"/>
<logger name="com.morpheus.PriceManagerService" level="INFO"/>
<logger name="com.morpheus.PricePlanService" level="INFO"/>
<logger name="com.morpheus.ProcessService" level="INFO"/>
<logger name="com.morpheus.ProfileService" level="INFO"/>
<logger name="com.morpheus.project.ProjectService" level="INFO"/>
<logger name="com.morpheus.provision.AbstractBoxProvisionService" level="INFO"/
↪>
<logger name="com.morpheus.provision.AbstractProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.CloudPluginProvisioningService" level=
↪"INFO"/>
<logger name="com.morpheus.provision.DockerEngineProvisionService" level="INFO"
↪"/>
<logger name="com.morpheus.provision.DockerProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.ExternalProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.HelmProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.IProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.KubernetesProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.MaasProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.ManualProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.OneviewProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.ScribeProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.SelfManagedProvisionService" level="INFO"/
↪>
<logger name="com.morpheus.provision.StandardProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.SwarmProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.TerraformProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.UcsProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.UnmanagedProvisionService" level="INFO"/>
<logger name="com.morpheus.provision.WorkflowProvisionService" level="INFO"/>
<logger name="com.morpheus.ProvisioningService" level="INFO"/>
<logger name="com.morpheus.ProxyService" level="INFO"/>
<logger name="com.morpheus.ReferenceService" level="INFO"/>
<logger name="com.morpheus.report.AbstractReportService" level="INFO"/>
<logger name="com.morpheus.report.AmazonCoverageReportService" level="INFO"/>
<logger name="com.morpheus.report.AmazonSavingsReportService" level="INFO"/>
<logger name="com.morpheus.report.AmazonUtilizationReportService" level="INFO"/
↪>
<logger name="com.morpheus.report.CloudAppCapacityReportService" level="INFO"/>
<logger name="com.morpheus.report.CloudAppUsageReportService" level="INFO"/>
<logger name="com.morpheus.report.CloudCapacityReportService" level="INFO"/>
<logger name="com.morpheus.report.CloudInstanceTypeCapacityReportService"
↪level="INFO"/>
<logger name="com.morpheus.report.CloudInstanceTypeUsageReportService" level=
↪"INFO"/>

```

(continues on next page)

(continued from previous page)

```
<logger name="com.morpheus.report.CloudInventoryReportService" level="INFO"/>
<logger name="com.morpheus.report.CloudUsageReportService" level="INFO"/>
<logger name="com.morpheus.report.CostReportService" level="INFO"/>
<logger name="com.morpheus.report.InventoryReportService" level="INFO"/>
<logger name="com.morpheus.report.InvoiceReportService" level="INFO"/>
<logger name="com.morpheus.report.MigrationReportService" level="INFO"/>
<logger name="com.morpheus.report.PluginReportService" level="INFO"/>
<logger name="com.morpheus.report.ReportService" level="INFO"/>
<logger name="com.morpheus.report.TenantUsageReportService" level="INFO"/>
<logger name="com.morpheus.report.TimeSeriesCostReportService" level="INFO"/>
<logger name="com.morpheus.RoleService" level="INFO"/>
<logger name="com.morpheus.RpcService" level="INFO"/>
<logger name="com.morpheus.scale.AbstractScaleService" level="INFO"/>
<logger name="com.morpheus.scale.MorpheusScaleService" level="INFO"/>
<logger name="com.morpheus.ScaleService" level="INFO"/>
<logger name="com.morpheus.scribe.ScribeLibraryService" level="INFO"/>
<logger name="com.morpheus.ScriptConfigService" level="INFO"/>
<logger name="com.morpheus.sdn.AbstractSdnService" level="INFO"/>
<logger name="com.morpheus.sdn.MorpheusSdnService" level="INFO"/>
<logger name="com.morpheus.sdn.OvsService" level="INFO"/>
<logger name="com.morpheus.sdn.VethSdnService" level="INFO"/>
<logger name="com.morpheus.security.AbstractSecurityScanService" level="INFO"/>
<logger name="com.morpheus.security.ScapScanService" level="INFO"/>
<logger name="com.morpheus.security.SecurityScanService" level="INFO"/>
<logger name="com.morpheus.SecurityGroupService" level="INFO"/>
<logger name="com.morpheus.SequenceService" level="INFO"/>
<logger name="com.morpheus.ServerScriptService" level="INFO"/>
<logger name="com.morpheus.ServerService" level="INFO"/>
<logger name="com.morpheus.ServicePlanService" level="INFO"/>
<logger name="com.morpheus.SettingsService" level="INFO"/>
<logger name="com.morpheus.SetupService" level="INFO"/>
<logger name="com.morpheus.SiteService" level="INFO"/>
<logger name="com.morpheus.SnapshotPriceManagerService" level="INFO"/>
<logger name="com.morpheus.SnapshotService" level="INFO"/>
<logger name="com.morpheus.StatsService" level="INFO"/>
<logger name="com.morpheus.StatusService" level="INFO"/>
<logger name="com.morpheus.storage.AbstractStorageServerService" level="INFO"/>
<logger name="com.morpheus.storage.AbstractStorageService" level="INFO"/>
<logger name="com.morpheus.storage.BasicStorageService" level="INFO"/>
<logger name="com.morpheus.storage.CephStorageService" level="INFO"/>
<logger name="com.morpheus.storage.EcsStorageService" level="INFO"/>
<logger name="com.morpheus.storage.IsilonStorageService" level="INFO"/>
<logger name="com.morpheus.storage.KubernetesStorageService" level="INFO"/>
<logger name="com.morpheus.storage.NfsStorageService" level="INFO"/>
<logger name="com.morpheus.storage.QnapFileStationService" level="INFO"/>
<logger name="com.morpheus.storage.StorageServerService" level="INFO"/>
<logger name="com.morpheus.storage.StorageVolumeService" level="INFO"/>
<logger name="com.morpheus.storage.ThreeParStorageService" level="INFO"/>
<logger name="com.morpheus.StorageProviderService" level="INFO"/>
<logger name="com.morpheus.SubAccountService" level="INFO"/>
<logger name="com.morpheus.task.AbstractTaskService" level="INFO"/>
<logger name="com.morpheus.task.AnsibleTaskService" level="INFO"/>
```

(continues on next page)

(continued from previous page)

```
<logger name="com.morpheus.task.AnsibleTowerTaskService" level="INFO"/>
<logger name="com.morpheus.task.ChefTaskService" level="INFO"/>
<logger name="com.morpheus.task.ContainerScriptTaskService" level="INFO"/>
<logger name="com.morpheus.task.ContainerTemplateTaskService" level="INFO"/>
<logger name="com.morpheus.task.EmailTaskService" level="INFO"/>
<logger name="com.morpheus.task.ExecutableTaskInterface" level="INFO"/>
<logger name="com.morpheus.task.GroovyTaskService" level="INFO"/>
<logger name="com.morpheus.task.HttpTaskService" level="INFO"/>
<logger name="com.morpheus.task.JavascriptTaskService" level="INFO"/>
<logger name="com.morpheus.task.JRubyTaskService" level="INFO"/>
<logger name="com.morpheus.task.LocalScriptTaskService" level="INFO"/>
<logger name="com.morpheus.task.PuppetTaskService" level="INFO"/>
<logger name="com.morpheus.task.PythonTaskService" level="INFO"/>
<logger name="com.morpheus.task.RestartTaskService" level="INFO"/>
<logger name="com.morpheus.task.ShellTaskService" level="INFO"/>
<logger name="com.morpheus.task.TaskConfigService" level="INFO"/>
<logger name="com.morpheus.task.TaskService" level="INFO"/>
<logger name="com.morpheus.task.VroTaskService" level="INFO"/>
<logger name="com.morpheus.task.WinrmTaskService" level="INFO"/>
<logger name="com.morpheus.task.WriteAttributesTaskService" level="INFO"/>
<logger name="com.morpheus.trust.AbstractCredentialService" level="INFO"/>
<logger name="com.morpheus.trust.CredentialProvider" level="INFO"/>
<logger name="com.morpheus.trust.CredentialService" level="INFO"/>
<logger name="com.morpheus.trust.CypherCredentialService" level="INFO"/>
<logger name="com.morpheus.trust.InternalCredentialService" level="INFO"/>
<logger name="com.morpheus.trust.PluginCredentialService" level="INFO"/>
<logger name="com.morpheus.UsageLimitService" level="INFO"/>
<logger name="com.morpheus.UserGroupService" level="INFO"/>
<logger name="com.morpheus.UserManagementService" level="INFO"/>
<logger name="com.morpheus.vdi.VdiAppService" level="INFO"/>
<logger name="com.morpheus.vdi.VdiGatewayService" level="INFO"/>
<logger name="com.morpheus.vdi.VdiPoolService" level="INFO"/>
<logger name="com.morpheus.VirtualImagePriceManagerService" level="INFO"/>
<logger name="com.morpheus.VirtualImageService" level="INFO"/>
<logger name="com.morpheus.WikiPageService" level="INFO"/>
<logger name="com.morpheus.worker.DistributedWorkerService" level="INFO"/>
<logger name="com.morpheus.ZoneFolderService" level="INFO"/>
<logger name="com.morpheus.ZoneMarketplaceService" level="INFO"/>
<logger name="com.morpheus.ZonePoolService" level="INFO"/>
<logger name="com.morpheus.ZoneRegionService" level="INFO"/>
<logger name="com.morpheus.ZoneService" level="INFO"/>
```

Audit logs

1. To set up CEF/SIEM auditing export, add the below appender above or below the other appenders in the logback.xml configuration file:

```
<appender name="AUDIT" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>/var/log/morpheus/morpheus-ui/audit.log</file>
  <rollingPolicy class="ch.qos.logback.core.rolling.
↪SizeAndTimeBasedRollingPolicy">
    <fileNamePattern>audit.%d{yyyy-MM-dd}.%i.log</fileNamePattern>
    <maxFileSize>50MB</maxFileSize>
    <maxHistory>30</maxHistory>
  </rollingPolicy>
  <encoder>
    <pattern>[%d] [%thread] %-5level %logger{15} - %maskedMsg %n</pattern>
  </encoder>
</appender>
```

.. note:: ``maxFileSize`` and ``maxHistory`` values can be updated as needed.

2. Add the below logger above or below the other loggers in the logback.xml configuration file (make sure it is below, not above, the appender from the previous step or an error will occur):

```
<logger name="com.morpheus.AuditLogService" level="INFO" additivity="false">
  <appender-ref ref="AUDIT" />
</logger>
```

3. Once you have done this, you need to restart the Morpheus Application server:

```
morpheus-ctl stop morpheus-ui
```

Note: Please be aware this will stop the web interface for Morpheus.

4. Once the service has stopped enter the following at the xml prompt to restart (if the service does not stop, replace stop with graceful-kill and retry)

```
morpheus-ctl start morpheus-ui
```

5. To know when the UI is up and running you can run the following command

```
morpheus-ctl tail morpheus-ui
```

Once you see the ASCII art show up you will be able to log back into the User Interface. A new audit file will have been created called audit.log and will be found in the default Morpheus log path which is /var/log/morpheus/morpheus-ui/

This is only an example and other configurations are possible, such as creating an appender definition for your SIEM audit database product.

5.12.2 Ansible Troubleshooting

When a workflow is executed manually, the Ansible run output is available in the Instance History tab. Select the bubble next to the Ansible task to see the output. You can also see the run output in UI logs at `/var/log/morpheus/morpheus-ui/current`. These can be tailed by running `morpheus-ctl tail morpheus-ui`.

Verify Ansible is installed on the Morpheus Appliance

Ansible should be automatically installed but certain OS or network conditions can prevent the automated install. You can confirm installation by running `ansible --version` in the Morpheus appliance, or by viewing the Ansible integration details page (Administration > Integrations > Select Ansible Integration). We also see it in the Ansible tab of a Group or Cloud scoped to Ansible, just run `--version` as `ansible` is already included in the command.

If Ansible is not installed

Follow these instructions to install, or use your preferred installation method

Ubuntu:

```
sudo apt-get install software-properties-common
sudo apt-add-repository ppa:ansible/ansible
sudo apt-get update
sudo apt-get install ansible python-requests
```

CentOS:

```
sudo yum install epel-release
sudo yum install ansible python-requests
```

Then create the working Ansible directory for Morpheus:

```
sudo mkdir /opt/morpheus/.ansible
sudo chown morpheus-app.morpheus-app /opt/morpheus/.ansible
```

Validate Git repo authorization and the configured paths

The public and private SSH keys need to be added to the Morpheus appliance via Infrastructure > Keys & Certs and the public key needs to be added to the Git repo via user settings. If both are set up correctly, you will see the playbooks and roles populate in the Ansible Integration details page.

The Git Ref field on playbook tasks is to specify a different git branch than default. It can be left to use the default branch. If your playbooks are in a different branch you can add the branch name in the Git Ref field.

When running a playbook that is in a workflow, the additional playbooks fields do not need to be populated, they are for running a different playbook than the one set in the Ansible task in the Workflow, or using a different Git Ref.

Note: If you are manually running Workflows with Ansible tasks on existing Instances through Actions > Run Workflow and not seeing results, set the Provision Phase on the Ansible task to Provision as there may be issues with executing tasks on other phases when executing manually.

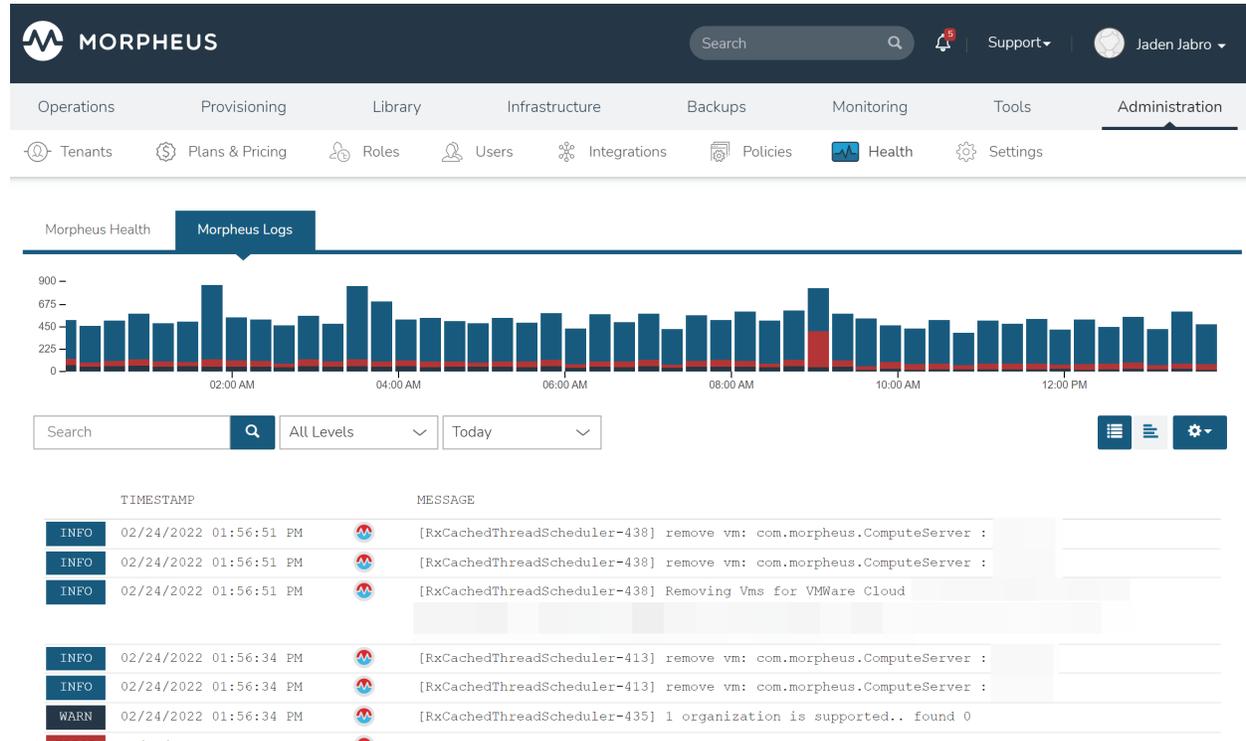
5.12.3 Attaching Logs to Case

When submitting a case it is critical to attach the relevant logs. The logs can be found at `/var/log/morpheus/morpheus-ui/current`. Logs can be attached to the case at anytime.

When submitting logs please reproduce the error right before capturing and sending the log file. This will ensure the activity that took place and resulted in an error is contained in the logs.

Log rotation takes the current file each night or after it's a certain size and compresses them. The `*.s` files in the current directory are rotated and zipped logs that can be sent as is.

The logs can also be captured from the Morpheus UI. Under Administration > Health > Morpheus Logs. Please copy relevant logs and add to case as an attachment.



5.12.4 Blank Dashboard

Problem A blank dashboard or 500 error after installing morpheus

Note: A blank or 500 error on just the dashboard is different than the entire morpheus-ui not loading. Please see UI note loading article for troubleshooting the ui not loading after an upgrade.

Cause Elasticsearch restarting prior to being fully bootstrapped during the initial install.

Solution To fix, purge elasticsearch by running the following on the Morpheus Appliance:

```
curl -XDELETE http://localhost:9200/*
morpheus-ctl restart elasticsearch
morpheus-ctl restart morpheus-ui
```

Another option is:

```
sudo rm -rf /var/opt/|morpheus| /elasticsearch/data/morpheus
morpheus-ctl restart elasticsearch
morpheus-ctl restart morpheus-ui
```

If you get a term/timeout on ui restart, run

```
morpheus-ctl kill morpheus-ui
morpheus-ctl start morpheus-ui
```

Note: The morpheus-ui may take a few minutes to load and be available after being restarted

5.12.5 CLI Troubleshooting

If you have installed the Morpheus CLI successfully and get a successful login but see this error `Error Communicating with the Appliance. SSL_connect returned=1 errno=0 state=error: certificate verify failed`

run the command

```
morpheus remote update {appliancename} --insecure
```

5.12.6 Cannot Login

Forgot password

If a user forgets their password, they can use the *FORGOT PASSWORD?* link on the login page. They can then enter their username or email address to send a reset password email to the email address defined on the user.

If the default or user added SMTP server is not functioning or blocked, a System Admin user can impersonate that user and update their password.

If the System Admin user password needs to be reset and the default or user added SMTP server is not functioning or blocked, please contact Morpheus support for assistance.

Sub-Tenant user cannot login after 3.4.0 upgrade

Morpheus v3.4.0 added support for all subtenant users to login via the main tenant url using subtenant id or subdomain prefix, ie `tenantId\username` or `subdomain\username`.

Note: Tenant subdomains can be defined by editing Tenant settings and updating the *SUBDOMAIN* field.

Important: Subtenant local users will no longer be able to login from main login url without using their subtenant id or subdomain prefix.

The login requirements were added in v3.4.0 to allow subtenant users with identity source integration generated user accounts to be able to login to the master tenant, gain API and CLI access, and remove the requirement for usernames to be unique across all tenants.

Previously subtenant users that had local/morpheus generated user accounts could login to their tenant via the master tenant url, while subtenant users that had identity source integration generated user accounts had to use the subtenant specific login url.

In v3.4.0+ all subtenant users can login via the master tenant url by specifying their tenant id or subdomain prefix, \, then username. Subtenants can still use the tenant specific login url as well.

Example: I have a username `subuser` that belongs to a tenant with the subdomain `acme` and tenant id `58`. When logging in from the main login url, I now need to enter in: `acme\subuser` and the password. Alternatively the tenant ID can be used, ie `58\subuser`

Active Directory user suddenly cannot Login

In Morpheus v3.4.0 and prior, OU changes in Active Directory can disable logins for AD users who had previously authenticated/have existing user accounts in Morpheus. If an Active Directory user cannot login to Morpheus after their OU was changed in AD, please contact Morpheus support for a resolution. The OU association for the user(s) can also be manually updated in the database. This issue is resolved in Morpheus versions 3.4.1 and higher.

5.12.7 Common Ports & Requirements

The following chart is useful for troubleshooting Agent install, Static IP assignment, Remote Console connectivity, and Image transfers.

Table 21: Common Ports & Requirements

Feature	Method	OS	Source	Destination	Port	Requirement
Agent Communication	All	All	Node	Appliance	443	DNS Resolution from node to appliance url
Agent Install	All	Linux	Node	Appliance	80	Used for appliance yum and apt repos
	SSH	Linux	Appliance	Node	22	DNS Resolution from node to appliance url Virtual Images configured SSH Enabled on Virtual Image
	WinRM	Windows	Appliance	Node	5985	DNS Resolution from node to appliance url Virtual Images configured WinRM Enabled on Virtual Image(<i>winrm quickconfig</i>)
	Cloud-init	Linux				Cloud-init installed on template/image Cloud-init settings populated in User Settings or in <i>Admin -> Provisioning</i> Agent install mode set to Cloud-Init in Cloud Settings
	Cloudbase-init	Windows				Cloudbase-init installed on template/image Cloud-init settings populated in User Settings or in <i>Admin -> Provisioning</i> Agent install mode set to Cloud-Init in Cloud Settings
	VMtools	All				VMtools installed on template Cloud-init settings populated in Morpheus user settings or in <i>Administration -> Provisioning</i> when using Static IP's Existing User credentials entered on Virtual Image when using DHCP RPC mode set to VMtools in VMware cloud settings.
Static IP Assignment & IP Pools	Cloud-Init	All				Network configured in Morpheus (Gateway, Primary and Secondary DNS, CIDR populated, DHCP disabled)
5.12. Troubleshooting						Cloud-init/Cloudbase-init installed on template/image Cloud-init settings populated in Morpheus user settings or in <i>Administration -> Provisioning</i>

5.12.8 How to un-manage an Instance/VM/Host

Description

A managed VM (and associated Instance) needs to be unmanaged and returned to Discovered type.

Solution

Delete the record from the **Infrastructure > Compute** (! not from Provisioning - Instances) selection with the following configuration in the Delete modal:

- **Remove Infrastructure** UNCHECKED
- **Remove Associated Instances** Must be checked if the server has an associated Instance, as deleting the VM but not the Instance would result in an abandoned Instance thus not allowed.
- **Force Delete** UNCHECKED

The most important items to be aware of when “un-managing” an Instance/VM/Host are:

1. The “Remove from Infrastructure” flag when deleting a VM or Host in Morpheus determines if the actual VM is deleted from the target Infrastructure.
 - Checking “Remove Infrastructure” means you **WANT TO DELETE THE ACTUAL VM**. Typing “DELETE” in the confirmation field is required when “Remove From Infrastructure” is enabled.
 - Unchecking “Remove Infrastructure” means you only want to delete the record in Morpheus but leave the actual VM untouched.
2. Deleting an Instance will always remove Infrastructure.

Important: REPEAT: Deleting an Instance from the Provisioning section will always remove the VM aka Infrastructure.

3. After removing the record from Morpheus, the VM must be in a Cloud with Inventory enabled to automatically be re-discovered.

Process

Steps to delete a managed VM from Morpheus and, when necessary, remove the associated Instance:

1. Navigate to the VM (not Instance) detail page at **Infrastructure > Compute - VMs**

Note: VM's inside an Instance can be navigated to inside the Instance Details page by selecting the VM in the VM's section on the Instance Details page.

2. Select **DELETE**
3. Configure the DELETE HOST modal with the following settings:

DELETE HOST?
✕

Warning! Deleting this Host with "Remove Infrastructure" enabled will permanently delete it from the Cloud. To only delete the record but leave in the Cloud, uncheck "Remove from Infrastructure". If "Inventory Existing Instances" is enabled on the Cloud, the host will be re-synced as discovered.

Remove Infrastructure
 Remove Associated Instances
 Force Delete

CANCEL
DELETE

- Remove Infrastructure UNCHECKED
- Remove Associated Instances Must be checked if the server has an associated Instance, as deleting the VM but not the Instance would result in an abandoned Instance thus not allowed.
- Force Delete UNCHECKED

Important: If you have to type DELETE that means the Remove Infrastructure flag is selected and you are confirming deletion of the actual VM. Ensure Remove Infrastructure is UNCHECKED when you want to leave the VM intact!

4. Select *DELETE*
5. The VM and associated Instance will be removed from Morpheus but the actual VM will remain.
6. Wait up to 5 min or click *REFRESH* on the associated Clouds details page to force a cloud sync.

Note: Inventory must be enabled on the associated cloud for the VM to automatically be re-discovered by Morpheus.

7. The VM is now back in Morpheus as discovered/unmanaged. To managed and create a new Instance from the VM, select *ACTIONS* : Convert To Managed.

5.12.9 MySQL Too many connections error

If you see the following error in the Morpheus UI logs:

```
SQLExceptionHelper - Data source rejected establishment of connection, message from ↵  
↵server: "Too many connections"
```

it means the number connections between Morpheus application and mysql have reached the *max_connections* limit set in mysql (default is 151), or the *max_active* setting, which limits the number of connections on the Morpheus end (default is 150), and the limit needs to be raised, either in Morpheus or mysql, or both depending on the number of connections and configuration.

Note: The *max_connections* setting in mysql and the maximum used connections between an app node and mysql can be viewed in the Morpheus ui in the *Administration - Health* section under Database.

Important: In Single Morpheus app node configurations, the *max_active* setting on the app node must be less than the *max_connections* setting in mysql.

Important: In HA configurations, the *max_active* setting is per app node, and the *max_connections* setting in mysql must be greater than all app nodes *max_active* values combined, ie $(\text{max_active} * \text{\$number_of_app_nodes}) \leq \text{max_connections}$.

Morpheus *max_active* setting

Edit `/etc/morpheus/morpheus.rb` and add `mysql['max_active'] = $value` replacing `$value` with desired desired number of maximum connections allowed by Morpheus to mysql. For example, to set *max_active* at 150:

```
mysql['max_active'] = 150
```

Replacing 100 with the desired number of maximum connections allowed by Morpheus to mysql.

Run `morpheus-ctl reconfigure` for the setting to be applied. Reconfigure will not restart the ui unless additional ram has been added to the appliance host since the previous reconfigure. To edit the *max_active* without a reconfigure, update the *max_active* setting in `/opt/morpheus/conf/application.yml`. Please note the default setting of 150 will be applied upon the next reconfigure unless *max_active* is defined as instructed above in the `morpheus.rb` file.

mysql *max_connections* setting

Important: Customers are responsible for configuring and maintaining external databases used by Morpheus. This explains how to set the *max_connections* setting, but the value for the setting needs to be established by a customers qualified db admin.

In mysql prompt,

```
run mysql> SET GLOBAL max_connections = $value;
```

This will immediately write the variable, however it is only a temporary setting that will be overwritten upon restart of the mysql service.

To persist the `max_connections` setting, edit `my.cnf`, and add `max_connections = $value` replacing `$value` with desired value, ie to set `max_connections` at 300 in `my.cnf`, add:

```
max_connections = 300
```

5.12.10 Morpheus Agent Install Troubleshooting

When provisioning an Instance, there are network and configuration requirements to consider in order to successfully install the Morpheus Agent. Typically, when a VM Instance is still in the provisioning phase long after the VM is up, the Instance is unable to reach Morpheus. Depending on the Agent install mode, it could also mean Morpheus is unable to reach the Instance.

The most common reason an Agent install fails is the provisioned Instance cannot reach the Morpheus Appliance via the Appliance URL set in Administration > Settings over port 443. When an Instance is provisioned from Morpheus, it must be able to reach the Morpheus appliance via the Appliance URL or the Agent will not be installed.

The screenshot shows the Morpheus Administration interface. The top navigation bar includes the Morpheus logo, a search bar, a notification bell with 5 alerts, a 'Support' dropdown, and the user profile 'Jaden Jabro'. The main navigation menu is divided into several sections: Operations, Provisioning, Library, Infrastructure, Backups, Monitoring, Tools, and Administration. Under the Administration section, there are sub-menus for Appliance, Whitelabel, Provisioning, Monitoring, Backups, Logs, Guidance, Environments, Software Licenses, and License. The 'Appliance' sub-menu is selected, showing 'Appliance Settings'. There are two input fields: 'Appliance URL' with the value 'https://appliance.url.com' and 'Internal Appliance URL (PXE)' which is empty. Below the fields is a note: 'For PXE-Boot your appliance needs to be routable directly with minimal NAT masquerading. This allows one to override the default appliance url endpoint for use by the PXE Server. If this is unset, the default appliance url will be used instead.'

In addition to the main Appliance URL in Administration > Settings, additional Appliance URLs can be set per Cloud in the Advanced Options section of the Cloud configuration modal when creating or editing a Cloud. When this field is populated, it will override the main Appliance URL for anything provisioned into that Cloud.

Tip: The Morpheus UI current log, located at `/var/log/morpheus/morpheus-ui/current`, is very helpful when troubleshooting Agent installations.

Agent Install Methods

Morpheus Agent installation supports multiple install methods.

- SSH/WinRM
- VM Tools
- Cloud-Init & Cloudbase-Init
- Windows Unattended
- Manual

For All Agent Install Methods

When an Instance is provisioned and the Agent does not install, verify the following for any Agent install mode:

- The Morpheus Appliance URL (Administration > Settings) is both reachable and resolvable from the provisioned node
- The Appliance URL begins with <https://>, not <http://>

Note: Be sure to use <https://> even when using an IP address for the appliance.

- Inbound connectivity access to the Morpheus appliance from provisioned VMs and container hosts on port 443 (needed for Agent communication)
- Private (non-Morpheus provided) VM images and templates must have their credentials stored. These can be entered or edited in the Library > Virtual Images section by clicking the Actions dropdown on an image detail page and selecting Edit.

Note: Administrator user is required for Windows Agent install.

- The Instance does not have an IP address assigned. For scenarios without a DHCP server, static IP information must be entered by selecting the Network Type: Static in the Advanced Options section during provisioning. IP Pools can also be created in the Infrastructure > Networks > IP Pools section and added to Cloud network sections for IPAM
- DNS is not configured and the node cannot resolve the appliance. If DNS cannot be configured, the IP address of the Morpheus appliance can be used as the main or Cloud appliance

SSH

- Port 22 is open for Linux images, and SSH is enabled
- Credentials set on the image if using a custom or synced image. Credentials can be entered on images in the Library > Virtual Images section

WinRM

- Port 5985 must be open and WinRM enabled for Windows images
- Credentials have been entered on the image if using a custom or synced image. Credentials can be entered on images in the Library > Virtual Images section

Note: Administrator user is required for Windows Agent install.

VMware Tools (vmtools)

- VMware Tools is installed on the template(s)
- Credentials have been entered on the image if using custom or synced image. Credentials can be entered on images in the Library > Virtual Images section
- Sudo privileges required for Linux
- Administrator User required for Windows (SID 500)

Cloud-Init

- Cloud-Init settings configured in Administration > Settings > Provisioning section
- Cloud-Init installed on Virtual Image
- Cloud-Init enabled on Virtual Image config

Cloudbase-Init

- Windows Administrator Password defined in Administration > Settings > Provisioning section
- Cloudbase-Init installed on Virtual Image
- Cloud-Init enabled on Virtual Image config
- Cloudbase-Init is only required for OpenStack Cloud types

Note: Unattend Agent Installation and customizations are recommended over Cloudbase-Init

Windows Unattended

- Windows Administrator Password defined in Administration > Settings > Provisioning section
- VMware: Force Guest Customizations set to forced on Virtual Image config when using DHCP (Static Assignment will already force Guest Customizations)
- Nutanix & SCVMM: Virtual Image is sysprepped and shutdown, Sysprep Enabled flagged on Virtual Image config

Manual

Agent Install scripts can be downloaded from Morpheus by selecting **Actions > Download Agent Script** from an Server detail page, then run manually on the target host when required for a given managed resource. Please note the script will be unique per managed resource and should not be saved to run as needed on any arbitrary resources in the future.

When installing on Windows, continue with the steps below to complete manual installation:

- Open powershell as an administrator
- Run the `unblock-file cmdlet` against the download agent installation script:

```
Unblock-File -Path C:\Users\User01\Documents\Downloads\agentInstall.ps1

Get-ExecutionPolicy

Set-ExecutionPolicy -ExecutionPolicy Unrestricted -Scope CurrentUser
```

- After running the powershell script, ensure the script downloaded the msi and the Agent service started correctly:

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

Following setup, verify that the Agent is reporting back to the Morpheus appliance.

Restarting the Morpheus Agent

In some situations, it may necessary to restart the Morpheus Agent on the host to re-sync communication from the Agent to the Morpheus appliance.

Linux

On the target host, run `sudo morpheus-node-ctl restart morphd` and the Morpheus agent will restart. `morpheus-node-ctl status` will also show the agent status.

Windows

The Morpheus Windows Agent service can be restarted in Administrative Tools > Services.

Tip: The Morpheus Remote Console is not dependent on Agent communication and can be used to install or restart the Morpheus agent on an Instance.

Uninstall Morpheus Agent

Linux Agents

You can use the following to uninstall the linux agent (contains commands for both rpm and deb agents)

```
sudo rm /etc/apt/sources.list.d/morpheus.list \
sudo morpheus-node-ctl kill \
sudo apt-get -y purge morpheus-node \
sudo apt-get -y purge morpheus-vm-node \
sudo yum -y remove morpheus-node \
sudo yum -y remove morpheus-vm-node \
sudo yum clean all \
sudo systemctl stop morpheus-node-runsvdir \
sudo rm -f /etc/systemd/system/morpheus-node-runsvdir.service \
sudo systemctl daemon-reload \
sudo rm -rf /var/run/morpheus-node \
sudo rm -rf /opt/morpheus-node \
sudo rm -rf /etc/morpheus \
```

(continues on next page)

(continued from previous page)

```
sudo rm -rf /var/log/morpheus-node \  
sudo pkill runsv \  
sudo pkill runsvdir \  
sudo pkill morphd \  
sudo usermod -l morpheus-old morpheus-node \  

```

Windows Agents

```
$app = Get-WmiObject -Class Win32_Product -Filter "Name = 'Morpheus Windows Agent'"  
$app.Uninstall()
```

CentOS/RHEL 7 Images

For custom CentOS 7 images we highly recommend setting up Cloud-Init and fixing the network device names. More information for custom CentOS images can be found in the CentOS 7 image guide.

5.12.11 Morpheus UI not loading after upgrade or reconfigure

Problem: The Morpheus ui does not load after performing an upgrade.

Common Causes:

1. The morpheus-ui has not finished loading
2. The morpheus-ui was not fully stopped before reconfigure, or not started after reconfigure
3. Morpheus was forced to restart or shut down while the database schema was being migrated during an upgrade

Solutions:

1. The morpheus-ui has not finished loading.

An easy way to see when the ui is finished loading and running is to tail the ui current file and look for the morpheus logo with version and start time

```
morpheus-ctl tail morpheus-ui
```

Note: After running *morpheus-ctl start morpheus-ui*, the Morpheus ui takes around 3 minutes to run depending on hardware.

1. The morpheus-ui was not fully stopped before reconfigure, or not started after reconfigure

The morpheus ui must be stopped prior to running *morpheus-ctl reconfigure* when upgrading. Sometimes running *morpheus-ctl stop morpheus-ui* will timeout and the ui is not actually stopped. If stopping the ui does timeout, run *morpheus-ctl kill morpheus-ui* prior to reconfigure, and be sure to run *morpheus-ctl start morpheus-ui* after reconfigure is completed.

If you ran a reconfigure before stopping the ui, run:

```
sudo morpheus-ctl kill morpheus-ui
sudo morpheus-ctl reconfigure
sudo morpheus-ctl start morpheus-ui
```

Wait for the ui to come up.

2. Morpheus was forced to restart or shut down while the database schema was being migrated during an upgrade

If the ui fails to start and you see the error `Invocation of init method failed; nested exception is liquibase.exception.LockException: Could not acquire change log lock. Currently locked by morpheus` it likely means morpheus was forced to restart or shut down while the database schema was being migrated during an upgrade, and the lock was not released.

To release the lock, you will need to run a mysql query. You will need to install `mysql-client` on the morpheus appliance, and grab the password for morpheus mysql. The username and db name are both morpheus. The password to login to mysql can be found in the `application.yml` file located at `/opt/morpheus/conf/application.yml`

Then run the following:

```
mysql -u morpheus -p -h 127.0.0.1 morpheus
```

At the prompt, enter the mysql password from the `application.yml`

Then run:

```
DELETE FROM DATABASECHANGELOGLOCK;
```

Then restart morpheus-ui:

```
sudo morpheus-ctl restart morpheus-ui
```

If the restart timesout, run:

```
sudo morpheus-ctl kill morpheus-ui
sudo morpheus-ctl start morpheus-ui
```

5.12.12 Remote Console

Morpheus has a built in Remote Console for Instances, Hosts, Virtual Machines and Bare Metal. The following information reviews the Roles Settings, Protocols, and Requirements necessary to configure and troubleshoot Remote Console access.

Role Settings

User Role settings determine if the Console tab or Open Console Action appear for a user, and if a login prompt is presented or the user is automatically logged in when using the Console.

- **Remote Console (None, Provisioned, Full)**

None The user will not have access to remote console.

Provisioned The user will only have remote console access for Instances they provisioned.

Full The user will have remote console access for all instances they have access to.

- **Remote Console: Auto Login (No, Yes)**

No A login prompt will be present in the console for Linux platforms, and the main login screen will present for Windows platforms.

Yes Morpheus will automatically login to the remote console using the credentials defined on the VM or Host. For provisioned Instances, the credentials are defined either from the credentials defined on the Virtual Image used, added via cloud-init or VMware Tools using the global cloud-init settings (Administration - Provisioning) or the Linux or Windows settings defined in User Settings. For Instances created when converting a VM or Host to managed, the credentials are entered when converting to managed. These credentials can be changed by editing the underlying VM or Host of the Instance.

Note: If the credentials defined on the VM or Host are not valid, and the `Remote Console: Auto Login Role` setting is set to `Yes`, the console will not be able to connect and no console window or login prompt will be presented. The credentials on the underlying VM or Host must be edited or `Remote Console: Auto Login Role` setting can be set to `No` for a login prompt to present in the console. Credentials cannot be changed from an Instance view, only in the Infrastructure VM or Host view.

Protocols

Platform Type and Cloud Settings determines the protocol and port used for Remote Console connections.

- **SSH** The SSH protocol will be used for Linux and OSX platform types, and 22 is the default port used.
- **RDP** The RDP (Remote Desktop) protocol will be used for Windows platform types over port 3389 by default.
- **VNC** The VNC protocol will be used for all platform types in Clouds with the `Hypervisor Console` option enabled in cloud settings. VNC connection are made directly to the Hypervisor Host over port 443.

Note: Alternative ports can be configured per VM or Host by editing the VM or Host and editing the Port field in the RPC host section.

SSH

For all Linux and OSX platform types, Morpheus will use the SSH protocol via port 22 by default for Remote Console connections, unless the `Hypervisor Console` option is enabled for VMware type clouds.

Morpheus will SSH using the username, password, RPC Host IP address and Port defined in the VM or Host record.

Default Requirements for SSH Connectivity

- SSH Enabled on the target VM or Host
- Port 22 incoming open on the target VM or Host firewalls and security groups from the Morpheus Appliance (not from the users IP address)
- An IP address defined on the VM or Host record that is routable from the Morpheus Appliance.
- Valid credentials defined on the VM or Host record in the RPC host field.
- `Remote Console` Role Permissions set to `Provisioned` or `Full` if the User provisioned the instance, or `Full` if the user did not provision the instance.

RDP

For all Windows platform types, Morpheus will use the RDP protocol via port 3389 by default for Remote Console connections, unless the *Hypervisor Console* option is enabled for VMware type clouds.

Morpheus will RDP using the username, password, RPC Host IP address and Port defined in the VM or Host record.

Default Requirements for RDP Connectivity

- Remote Access enabled on the target VM or Host and Remote Desktop enabled in the Windows Firewall settings. If the VM or Host is on a different network than the Morpheus appliance, public access for Remote Desktop must be enabled in the Firewall settings.
- Port 3389 incoming open on the target VM or Host firewalls and security groups from the Morpheus Appliance (not from the users IP address)
- An IP address defined on the VM or Host record that is routable from the Morpheus Appliance.
- Valid credentials defined on the VM or Host record in the RPC host field.
- *Remote Console* Role Permissions set to *Provisioned* or *Full* if the User provisioned the instance, or *Full* if the user did not provision the instance.

Note: If *Remote Console: Auto Login* is set to *No* in a users Role permissions, *Allow connections only from computers running Remote Desktop with Network Level Authentication* in the *Windows System Properties > Remote* settings must be **DISABLED** for Remote Console to connect.

VNC (VMware Hypervisor Console)

When the *Hypervisor Console* option is enabled in cloud settings, the VNC protocol will be used for all platform types that Cloud.

When using VNC Hypervisor Console, the Morpheus Appliance connects directly to the host the VM is on, not directly to the VM.

Morpheus features Remote Console support directly to hypervisors. To enable this feature a few prerequisites must be met:

- The Morpheus Appliance must have network access to the host the VM is on over 443.
- The Morpheus Appliance must be able to resolve the hypervisor hostnames.

Note: VNC connections for VMs and Hosts in VMware type clouds are made directly to the ESXi hosts, not vCenter.

Unlike SSH and RDP, valid credentials do not need to be set on the VM or Host records in Morpheus for VNC hypervisor console connections. An IP address is also not required on the VM or Host for VNC hypervisor console connections. Morpheus will be able to connect to the VM or Host as soon as the *Host (Hypervisor)* record is set, which can be viewed in the Info section on the VM or Host detail page.

Note:

- Auto-login is not supported for Hypervisor Console. Auto-login role settings do not apply to console connecting when using Hypervisor Console. Please note Hypervisor Console sessions persist on the ESXi host and once a user manually logs in to the VM they will continue to be logged in, even if the console tab/window in Morpheus is closed, until they manually log out.

- Copy and Paste and Text selection in Linux terminals is not supported when using VNC (VMware Hypervisor Console).
 - In Morpheus versions 3.2.0 and higher, a newer Guacamole version is installed that is not compatible with MacOS Platform Types over VNC.
-

Copy and Paste

Note: Copy and Paste for Text is supported for SSH and RDP protocols only. Use of this functionality requires the Role permission of “VDI: Copy/Paste” set to Full.

To Copy text from the console:

1. Select text in the Console window.
2. Click the COPY button at the top of the Console window.
3. The selected text is copied to the users clipboard.

To Paste text into console:

1. Copy text on the local computer to you clipboard
2. Right click into the “Paste Text Here” field at the top of the Console window. The field will the display “Text Copied, Use Console to Paste.”
3. Right click into the console window.
4. The text is pasted into the VM.

Guacamole

Overview

Morpheus uses Apache Guacamole, a clientless remote console. Guacamole is installed on the Morpheus Appliance during the initial reconfigure. In Morpheus versions 3.2.0 and higher, Guacamole 0.9.14 is automatically installed. On Morpheus versions older than 3.2.0, 0.9.9 is installed. The 0.9.14 version is required for VNC Hypervisor Console functionality on ESXi v6.5 and later.

The Guacamole proxy daemon, `guacd`, is used for all Remote Console connections and must be running for Remote Console functionality.

Troubleshooting `guacd`

If all console connections are not functioning, the Guacamole proxy daemon (`guacd`) process may not be running or have a stuck process preventing console connections. This is evident when only the header appears in the console tab/window, and no console window appears below the header and no connection status is show in the console header. The following commands can be used on the Morpheus Appliance to restore console functionality.

`morpheus-ctl status` Lists all local Morpheus services including `guacd` and their states. If `guacd` is stopped, it will need to be started again for Remote Console to function.

`morpheus-ctl start guacd` Starts the `guacd` process

`morpheus-ctl stop guacd` Stops the `guacd` process

morpheus-ctl kill guacd Forcefully kills the guacd process

morpheus-ctl restarts guacd Restarts the guacd process

morpheus-ctl tail guacd Tails the guacd current and state logs, located by default at `/var/log/morpheus/guacd/`. This log is useful when troubleshooting console connections, guacamole service status, and to determine the protocol being used for the Remote Console connection.

If guacd continues to stop even after being started, or if guacd is running and no properly configured console connections are functioning, there may be a stuck guacd or multiple guacd processes running, which will need to be killed and guacd started again.

To kill all guacd processes on the Morpheus Appliance and start guacd again:

1. Kill the morpheus guacd process: `morpheus-ctl kill guacd`
2. Grep for all running guacd processes: `sudo ps -aux | grep guacd` and note the guacd pid(s) (minus the process from the grep)
3. Kill all running guacd processes: `kill -9 pid` replacing *pid* with the pid(s) of the target processes
4. Start guacd again: `morpheus-ctl start guacd`
5. Tail the guacd logs to verify guacd is started and listening: `morpheus-ctl tail guacd` The log output will resemble below when guacd is properly running:

```
guacd[16899]: INFO:      Guacamole proxy daemon (guacd) version 0.9.14 started
guacd[16899]: INFO:      Listening on host 127.0.0.1, port 4822
```

6. Additional information in the guacd logs appears when Morpheus is making a console connection. A successful connection will resemble:

```
guacd[24725]: INFO: Creating new client for protocol "ssh"
guacd[24725]: INFO: Connection ID is "$24f67856-f050-4a17-83eb-9101g0cd8869"
guacd[24743]: INFO: Current locale does not use UTF-8. Some characters may not
↳render correctly.
guacd[24743]: INFO: User "@63102f19-eff4-412e-b1f9-718405f55782" joined connection "
↳$24f67856-f050-4a17-83eb-9101g0cd8869" (1 users now present)
guacd[24743]: INFO: Auth key successfully imported.
guacd[24743]: INFO: SSH connection successful.
```

Guacamole Version

In Morpheus versions 3.2.0 and higher, Guacamole version 0.9.14 is automatically installed. On Morpheus versions older than 3.2.0, 0.9.9 is installed. The 0.9.14 version is required for VNC Hypervisor Console functionality on ESXi v6.5 and later.

Note Guacamole version 0.9.14 is not compatible with MacOS Platform Types over VNC on ESXi v6.0 or prior (6.5 is supported). If necessary, the guacamole version can be reverted to 0.9.9.

To revert the guacamole version from 0.9.14 to 0.9.9.

1. Kill guacd - `morpheus-ctl kill guacd`
2. Check if any guacd processes are still running `ps -aux | grep guac`
3. If so, kill the processes `kill -9 pid` with *id* being the actual process id, like 16101.
4. Go to the guac 0.9.9 directory: `cd /var/opt/morpheus/guacamole-server-0.9.9`
5. Run: `make install`

6. Start guacd: `morpheus-ctl start guacd`

5.12.13 SSL Self-signed Certificate Regeneration

When Morpheus is deployed it generates a 10 year self-signed non-trusted SSL certificate. Below details the process to regenerate this certificate and key.

Replacing both the certificate and private key

1. Delete the certificate and key files in `/etc/morpheus/ssl/` that end in `.cert` and `.key`
2. Run Reconfigure `morpheus-ctl reconfigure`
3. Restart NGINX `morpheus-ctl restart nginx`

Replacing only the certificate

1. Delete the certificate file in `/etc/morpheus/ssl/` it ends in `.cert`
2. Run Reconfigure `morpheus-ctl reconfigure`
3. Restart NGINX `morpheus-ctl restart nginx`

5.12.14 Unable to Delete Tenant

Problem When trying to delete a tenant, a message stating managed resources must be removed or other error occurs and the tenant is not deleted. The tenant may be stuck in a deleting status or return to OK status after delete attempt.

Cause All managed resources must be removed from a tenant in order for that tenant to be deleted. This includes instances and their underlying managed vm's

Solution

1. Login or impersonate that an Admin user inside the tenant
2. Navigate to Infrastructure > Hosts
3. Under Hosts and VM's, delete any managed resources
 - Uncheck `remove infrastructure` when deleting a VM to only remove it from Morpheus but not from the underlying hypervisor/cloud
 - You must check `remove associated instances` if the VM has an associated instance
 - If the VM no longer exists but there is still a record in Morpheus, uncheck `remove infrastructure` and check `force delete`
4. Once all managed resources are removed from the tenant, the tenant can then be deleted
5. In certain situations other components may prevent a tenant from being deleted. If you have removed all managed resources from a tenant and the tenant still cannot be deleted, please contact Morpheus support

Warning: Managed resources can also be removed by deleting instances, but be aware this will delete VM's associated with the instance from the underlying hypervisor/cloud

5.12.15 Unable to Provision a Custom Image

Prior to provisioning an custom image, the image must be configured in the Library > Virtual Images section by selecting Edit on the Actions dropdown of the Virtual Image.

In the Edit Virtual Image pane:

1. Select “Cloud Init Enabled?” only if the Virtual Image is a linux image with cloud init installed.
2. Enter the username and password that are set on the Virtual Image.

Note: When using Static IP’s or IP Pools in VMware, VMware tools must also be installed on the template in order for Morpheus to set the static IP address when provisioning.

Note: Morpheus agents only support 64-bit vm’s prior to versions 2.12.3 and 3.0.2

5.13 Guides

5.13.1 Getting started with Morpheus and AWS

Introduction

This guide is designed to help you get started and quickly get the most out of Morpheus with AWS. By the end, you will integrate your first cloud, configure networking, prepare and consume images, provision instances, and get started with automation. We will briefly discuss installation and account setup but will provide links to additional resources for those very first steps. For the most part, this guide assumes you are able to get Morpheus installed and are ready to move forward from that point. There is a lot more to see and do in Morpheus that is beyond the scope of this guide. For more, consult the complete Morpheus documentation or take part in our user community forum.

Installation & Setup

In the simplest configuration, Morpheus needs one appliance server which will contain all the components necessary to orchestrate virtual machines and containers. Full requirements, including storage and networking considerations, can be found in *Requirements*. In order to provision any new instances, hosts, or applications, (or convert any discovered resources to managed resources) you will need a valid license. If you don’t have one, Morpheus will automatically set up a lab license on installation. A lab license is a time-unlimited license for Morpheus that limits you to 25 managed and discovered workloads. If you have a timed trial or a paid license, the license can be applied in Administration > Settings > License.

Groups

Groups in Morpheus define which resources a user has access to. Clouds are added to groups and a user can only access clouds that are in the groups to which their roles give them access. More information on Morpheus groups is available in the *Groups* section. A deep dive into groups goes beyond the scope of this guide but it’s often useful to create a group that contains all clouds for testing purposes. We will create that group now so that we can add our first cloud into this group in the next section.

Navigate to *Infrastructure > Groups*. Here we will see a list of all configured groups but, of course, this will be empty immediately after installation. Click “+CREATE”. Give your group a name, such as “All Clouds”. The “CODE” field

is used when calling Morpheus through Morpheus API or Morpheus CLI. It's useful in most cases to have an "All Clouds" group for testing purposes so this will likely help you down the road.

NEW GROUP
✕

Configuration

NAME

CODE

LOCATION

▶ **Advanced Options**

SAVE CHANGES

Click "SAVE CHANGES". Your group is now ready to accept clouds.

Integrating Your First Cloud

Clouds in Morpheus consist of any consumable endpoint whether that be On-Prem, Public clouds, or even bare metal. In this guide, we will focus on integrating and working with AWS.

To get started, we will navigate to *Infrastructure > Clouds*. This is the cloud detail page which lists all configured clouds. It will be empty if you've just completed installation and setup of Morpheus but soon we will see our integrated AWS cloud here.

Click the "+ADD" button to pop the "CREATE CLOUD" wizard. Select "AMAZON" and click the "NEXT" button.

On the "CONFIGURE" tab, give your cloud a "NAME". Select your Amazon region and enter your credentials in the "ACCESS KEY" and "SECRET KEY" fields. We can also set a number of advanced options here that may be relevant to your AWS use case.

Once you're satisfied with your selections, click "NEXT"

We have now arrived at the "GROUP" tab. In this case, we will mark the radio button to "USE EXISTING" groups if you wish to use the group we configured earlier.

Once you've selected the group, click "NEXT"

On the final tab of the "CREATE CLOUD" wizard, you'll confirm your selections and click "COMPLETE". The new cloud is now listed on the cloud detail page. After a short time, Morpheus will provide summary information and statistics on existing virtual machines, networks, and other resources available in the cloud.

Viewing Cloud Inventory

Now that we've integrated our first AWS cloud, we can stop for a moment to review what Morpheus gives us from the cloud detail page. We can see that Morpheus gives us estimated costs and cost histories, metrics on used resources, and also lists out resource counts in various categories including container hosts, hypervisors, and virtual machines. We can drill into these categories to see lists of resources in the various categories individual resources within them by clicking on the category tabs. We can link to the detail page for any specific resource by clicking on it from its resource category list.

Configuring Resource Pools

With our AWS cloud configured, Morpheus will automatically sync in available resource pools and data stores.

Resource pools, once Morpheus has had time to ingest them, will be visible from the cloud detail page. Navigate to *Infrastructure > Clouds > (your AWS cloud) > RESOURCES tab*. In here, we are able to see and control access to the various resource pools that have been configured in Amazon. For example, we can restrict access to a specific resource pool within Morpheus completely by clicking on the "ACTIONS" button, then clicking "Edit". If we unmark the "ACTIVE" button and then click "SAVE CHANGES" we will see that the resource pool is now grayed out in the list. The resources contained in that pool will not be accessible for provisioning within Morpheus.

The screenshot shows the Morpheus interface with the 'RESOURCES' tab selected. At the top, there are navigation tabs: SUMMARY, CLUSTERS, HOSTS, VMS, CONTAINERS, LOAD BALANCERS, NETWORKS, RESOURCES (highlighted), POLICIES, and WIKI. Below the tabs is a section titled 'POOLS'. There is a search bar with a magnifying glass icon and a '+ ADD RESOURCE POOL' button. Below this is a table with the following columns: NAME, DESCRIPTION, VISIBILITY, DEFAULT, TENANT, and ACTIONS. The table contains one row with the following data: NAME: My Resource Pool, DESCRIPTION: (empty), VISIBILITY: Private, DEFAULT: (empty), TENANT: morpheus, ACTIONS: ACTIONS (with a dropdown arrow). Below the table, there is a partially visible row with the following data: NAME: My Resource Pool, DESCRIPTION: My Resource Pool, VISIBILITY: Private, DEFAULT: (empty), TENANT: morpheus, ACTIONS: ACTIONS (with a dropdown arrow).

Often our clients will want to make specific blocks of resources available to certain groups. This can be easily and conveniently controlled through the same "EDIT RESOURCE POOL" dialog box we were just working in. If we expand the "Group Access" drawer, we are able to give or remove access to each pool to any group we'd like. We can also choose to make some or all of our resource pools available to every group. Specific resource pools can also be defined as the default for each group if needed.

EDIT RESOURCE POOL
✕

NAME

MOVE SERVERS

ACTIVE

DEFAULT

▼ Group Access

GROUP	ACCESS	DEFAULT
all	<input checked="" type="checkbox"/>	
AA	<input type="checkbox"/>	<input type="checkbox"/>
AA-DB	<input type="checkbox"/>	<input type="checkbox"/>

Additionally, we may choose to allow only certain service plans to be provisioned into a specific pool of resources. For example, perhaps a specific cluster is my SQL cluster and only specific services plans should be consumable within it. We can control that through this same dialog box.

Configuring Network for Provisioning

When configuring networking, we can set global defaults by going to *Infrastructure > Network > NETWORKS tab*. Here we can add or configure networks from all clouds integrated into Morpheus. Depending on the number of clouds Morpheus has ingested, this list may be quite large and may also be paginated across a large number of pages. In such a case, it may be easier to view or configure networks from the specific cloud detail page so that networks from other clouds are not shown.

NETWORKS

NETWORKS
NETWORK GROUPS
ROUTERS
IP POOLS
DOMAINS
PROXIES
SECURITY GROUPS
INTEGRATIONS

+
ADD

NAME	TYPE	CLOUD	CIDR	POOL DHCP	VISIBILITY	TENANTS	
10.30.20.0	OracleVM Network	Morpheus Oracle VM	10.30.20.0/22	✓	Public	morpheus	ACTIONS ▾
172.31.0.0/20 (subnet-63dff13b)	Amazon Subnet	ah-only	172.31.0.0/20	✓	Private	morpheus	ACTIONS ▾
172.31.16.0/20 (subnet-22110e46)	Amazon Subnet	ah-only	172.31.16.0/20	✓	Private	morpheus	ACTIONS ▾

Still in *Infrastructure > Network*, make note of the “INTEGRATIONS” tab. It’s here that we can set up any integrations

that may be relevant, such as IPAM integrations. Generally speaking, when adding IPAM integrations, we simply need to name our new integration, give the API URL, and provide credentials. There's more information in the [Networking Integrations](#) section of Morpheus Docs.

ADD IPAM INTEGRATION ✕

NAME

ENABLED

URL

USERNAME

PASSWORD

THROTTLE RATE ms

In *Infrastructure > Networking* we can also set up IP address pools from the IP Pools tab. These pools can be manually defined, known as a Morpheus-type IP pool, or they can come from any IPAM integrations you've configured. As instances are provisioned, Morpheus will assign IP addresses from the pool chosen during provisioning. When the instance is later dissolved, Morpheus will automatically release the IP address to be used by another instance when needed. When adding or editing a network, we can opt to scope the network to one of these configured IP address pools. Edit an existing network by clicking the pencil icon on the Networks List Page (*Infrastructure > Networks > Networks Tab*) and fill in the "Network Pool" field to associate the IP Pool with the network.

CREATE NETWORK POOL



NAME

POOL TYPE

Morpheus



IP Ranges

STARTING ADDRESS

ENDING ADDRESS

192.168.0.2

-

192.168.0.255

+

SAVE CHANGES

Since this guide is focused on working within the AWS cloud that we integrated at the start, we will take a look at our network configurations on the cloud detail page as well. Navigate to *Infrastructure > Clouds > (your AWS cloud) > NETWORKS tab*. Just as with resource pools, we have the ability to make certain networks inactive in Morpheus, or scope them to be usable only for certain groups or tenants.

SUMMARY CLUSTERS HOSTS VMS CONTAINERS LOAD BALANCERS **NETWORKS** RESOURCES POLICIES WIKI

NETWORKS

Search



ACTIONS ▾

<input type="checkbox"/> NAME	TYPE	CIDR	POOL	DHCP VISIBILITY	TENANT	ACTIONS ▾
<input type="checkbox"/> labs_ib_1a (subnet-21dd1a44)	Amazon Subnet	10.100.5.0/24		✓ Private	morpheus	ACTIONS ▾
<input type="checkbox"/> labs_ib_1c (subnet-dcb0b49a)	Amazon Subnet	10.100.4.0/24		✓ Private	morpheus	ACTIONS ▾

Prepping an Image

As we'll discuss and try out in the next section, Morpheus comes out of the box with a default set of blueprints that are relevant to many modern deployment scenarios. For the most part, these are base operating system images with a few additional adjustments. However, in many on-premise deployments, there are often custom image and networking requirements. We will work with the images included in Morpheus by default but have guides in Morpheus Docs for vmware-templates which are consumable in Morpheus.

Provisioning Your First Instance

At this point, we are ready to provision our first image. As a first instance, we'll provision an Apache web server to our AWS cloud.

Navigate to `|ProIns|`. If any instances are currently provisioned, we will see them listed here. To start a new instance we click the "+ADD" button to pop the "CREATE INSTANCE" wizard. We'll scroll down to and select the Apache instance type and click "NEXT".

The screenshot shows the 'CREATE INSTANCE' wizard with the following details:

- WIZARD STEPS:** TYPE (selected), GROUP, CONFIGURE, AUTOMATION, REVIEW.
- SEARCH:** A search bar is present.
- TECHNOLOGY:** A dropdown menu labeled 'TECHNOLOGY'.
- INSTANCE TYPES:**
 - ACTIVE MQ:** A fast and popular messaging server with support for many platforms and protocols including AMQP and STOMP.
 - ALIBABA Cloud:** Easily provision any Alibaba Image tied to your Alibaba cloud account just as if it were any other instance type.
 - APACHE:** An open-source modern web server common on many websites today. A secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards. (Highlighted with a red arrow)
 - AZURE:** Spin up any VM Image in the Azure Cloud.
 - CASSANDRA:** An open source distributed database management system designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure.
 - CENTOS:** A popular Linux flavor operating system. Easily provision CentOS vms. (Highlighted with a red arrow)
- NAVIGATION:** 'PREVIOUS' and 'NEXT' buttons are at the bottom right.

First, we'll specify the group to provision into which determines the clouds available. If you've followed this guide to this point, you should at least have a group that houses all of your clouds which you can select here. This will allow us to select the AWS cloud from the "CLOUD" dropdown menu. Provide a unique name to this instance and then click "NEXT"

From the "CONFIGURE" tab, we're presented with a number of options. The options are cloud and layout-specific, more generalized information on creating instances and available options is [Morpheus Agent](#). For our purposes, we'll select the following options:

- **LAYOUT:** Includes options such as the base OS, custom layouts will also be here when available
- **PLAN:** Select the resource plan for your instance. Some plans have minimum resource limits, Morpheus will only show plans at or above these limits. User-defined plans can also be created in `|AdmPla|`.
- **VOLUMES:** The minimum disk space is set by the plan, this value may be locked if you've selected a custom plan that defines the volume size
- **NETWORKS:** Select a network, note that IP pools must be linked with the networks defined in VMware in order to assign static IP addresses
- **SECURITY GROUPS**

Under the “User Config” drawer, mark the box to “CREATE YOUR USER”. Click *NEXT*.

Note: “CREATE YOUR USER” will seed a user account into the VM with credentials set in your Morpheus user account settings. If you’ve not yet defined these credentials, you can do so by clicking on your username in the upper-right corner of the application window and selecting “USER SETTINGS”.

For now, we’ll simply click “NEXT” to move through the “AUTOMATION” tab but feel free to stop and take a look at the available selections here. There is more information later in this guide on automation and even more beyond that in the rest of Morpheus docs.

Review the settings for your first instance and click *COMPLETE*.

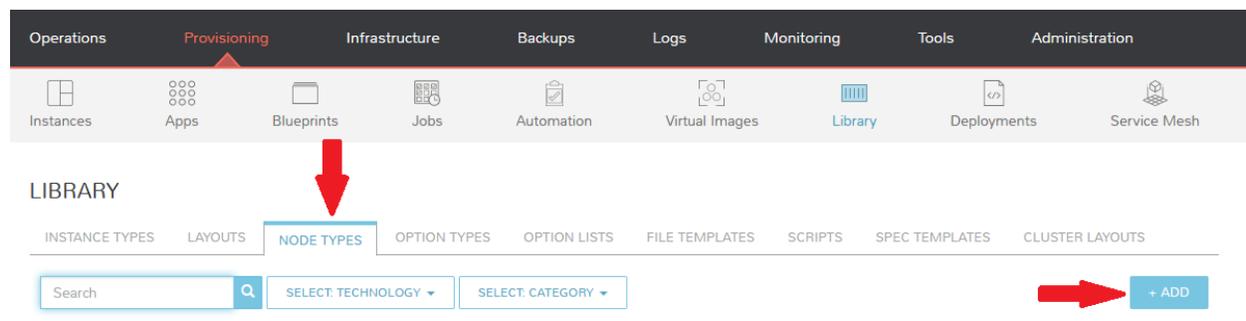
We are now dropped back onto the instances list page. We can see a new entry in the list at this point with a status indicator that the new machine is being launched (rocket icon in the status field). We can double click on the instance in the list to move to the instance detail page. For now we will see a progress bar indicating that the instance is being created and is starting up. The exact amount of time this process will take depends selections made when provisioning the instance. For more detailed information on the status of various provisioning processes, we can scroll down and select the “HISTORY” tab. The “STATUS” icon will change from the blue rocket to a green play button when the instance is fully ready. Furthermore, we can click on the hyperlinked IP address in the “VMS” section of this page to view a default page in a web browser to confirm success.

Creating Your First Library Item

In the prior section, we manually provisioned our first instance. However, Morpheus allows you to build a catalog of custom provisionable items to simplify and speed provisioning in the future. In this section, we’ll build a catalog item and show how that can translate into quick instance provisioning after configuration.

Note: Before starting this process, it’s important to decide which virtual image you plan to use. If you’re not using a Morpheus-provided image, you’ll want to ensure it’s uploaded. You will not be able to complete this section without selecting an available image. In this example we will use Morpheus Redis 3.0 on Ubuntu 14.04.3 v2.

Navigate to Library > Blueprints > Node Types and click *+ADD*.



In this example, I am going to set the following options in the “NEW NODE TYPE” wizard:

- **NAME**
- **SHORT NAME**
- **VERSION:** 1 (In this particular case, the version is not important)
- **TECHNOLOGY:** Amazon
- **AMI IMAGE:** Morpheus Redis 3.0 on Ubuntu 14.04.3 v2

NEW NODE TYPE
✕

NAME

SHORT NAME
The short name is a name with no spaces used for display in your container list.

VERSION

TECHNOLOGY

ENVIRONMENT VARIABLES

Name	Value

⚙️ +

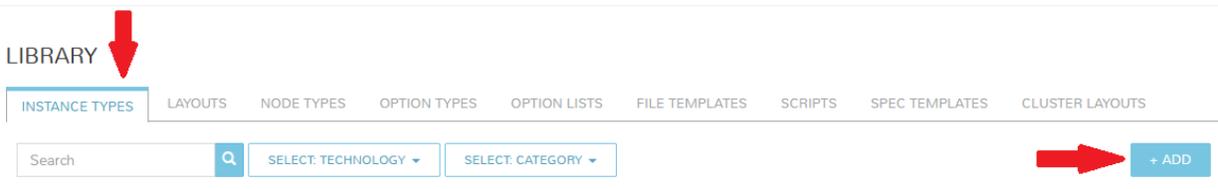
Amazon Options

AMI IMAGE

LOG FOLDER

DEPLOY FOLDER

With the new node created, we'll now add a new instance type which will be accessible from the provisioning wizard once created. Move from the "NODE TYPES" tab to the "INSTANCE TYPES" tab and click "+ADD".



In the "NEW INSTANCE TYPE" wizard, I'll simply enter a **NAME** and **CODE** value. Click "SAVE CHANGES".

NEW INSTANCE TYPE
✕

NAME

CODE

Useful shortcode for provisioning naming schemes and export reference.

DESCRIPTION

255 Characters Remaining

CATEGORY

ICON Browse

Suggested Dimensions: 150 x 51

Now that we've created a new instance type, access it by clicking on the name in the list of custom instances you've created. In my case, I've given the name "NewInstanceType".

LIBRARY

INSTANCES
LAYOUTS
NODE TYPES
OPTION TYPES
OPTION LISTS
FILE TEMPLATES
SCRIPTS
SPEC TEMPLATES
CLUSTER LAYOUTS

+ ADD

	NAME	TECHNOLOGY	CATEGORY	FEATURED	
	NewInstanceType ←		Web		ACTIONS ▾
	ActiveMQ	Mixed	Messaging		ACTIONS ▾
	Alibaba	Alibaba	Cloud		ACTIONS ▾
	Amazon Api		Apps		ACTIONS ▾
	AmazonMQ		Messaging		ACTIONS ▾

Once we've opened the new instance type, by default, we should be on the "LAYOUTS" tab. Click "+ADD LAYOUT". I've set the following fields on my example layout:

- **NAME**
- **VERSION:** This is the version number of the layout itself, which is labeled 1.0 in the example
- **TECHNOLOGY:** Amazon
- **Nodes:** Select the node we created earlier, if desired you can specify multiple nodes

Click “SAVE CHANGES”.

At this point we’ve completed the setup work and can now provision the instance we’ve created to our specifications. Navigate to `|ProIns|` and click “+ADD”. From the search bar we can search for the new instance type we’ve created. In the example case, we called it “newinstancetype”. Click “NEXT”.

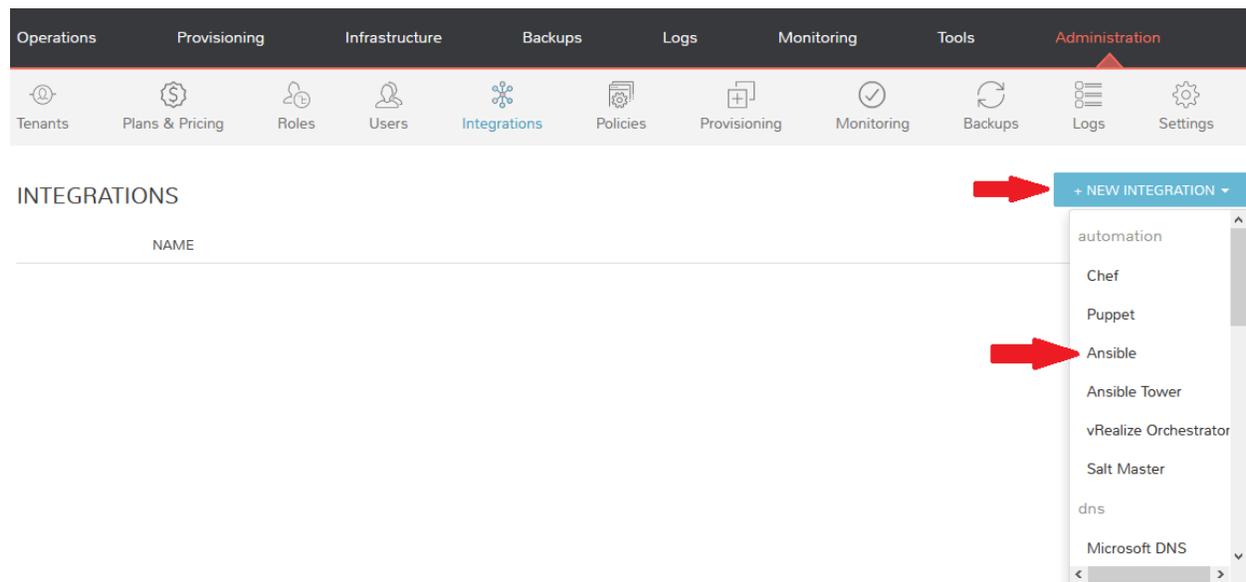
As before, we can select a group and cloud to provision this new instance. Click “NEXT”. On the “CONFIGURE” tab, make note that the layout and plan are already selected because they were configured as part of creating the new instance type. Select a network and click “NEXT”. Once again we will also click “NEXT” through the “AUTOMATION” tab. Finally, click “COMPLETE”.

As before when we manually provisioned an instance, Morpheus will now begin to spin up the new VM. Once the provisioning process has completed, open the instance detail page in Morpheus and click on the “CONSOLE” tab. You’ll be logged in with your user account and are then able to confirm the machine is ready and available.

Automation and Configuration Management

Morpheus automation is composed of Tasks and Workflows. A task could be a script added directly, scripts or blueprints pulled from the Morpheus Library, playbooks, recipes, or a number of other things. The complete list of task types can be found in *Automation*. Tasks can be executed individually but they are often combined into workflows. We can opt to run a workflow at provision time or they can be executed on existing instances through the Actions menu.

In this guide we will set up an Ansible integration, create a task, add the task to a workflow, and run the workflow against a new and existing instance. If you’ve worked through this guide to this point, you should already have an Apache instance running. If you don’t yet have that, provision one before continuing with this guide and ensure it’s reachable on port 80.

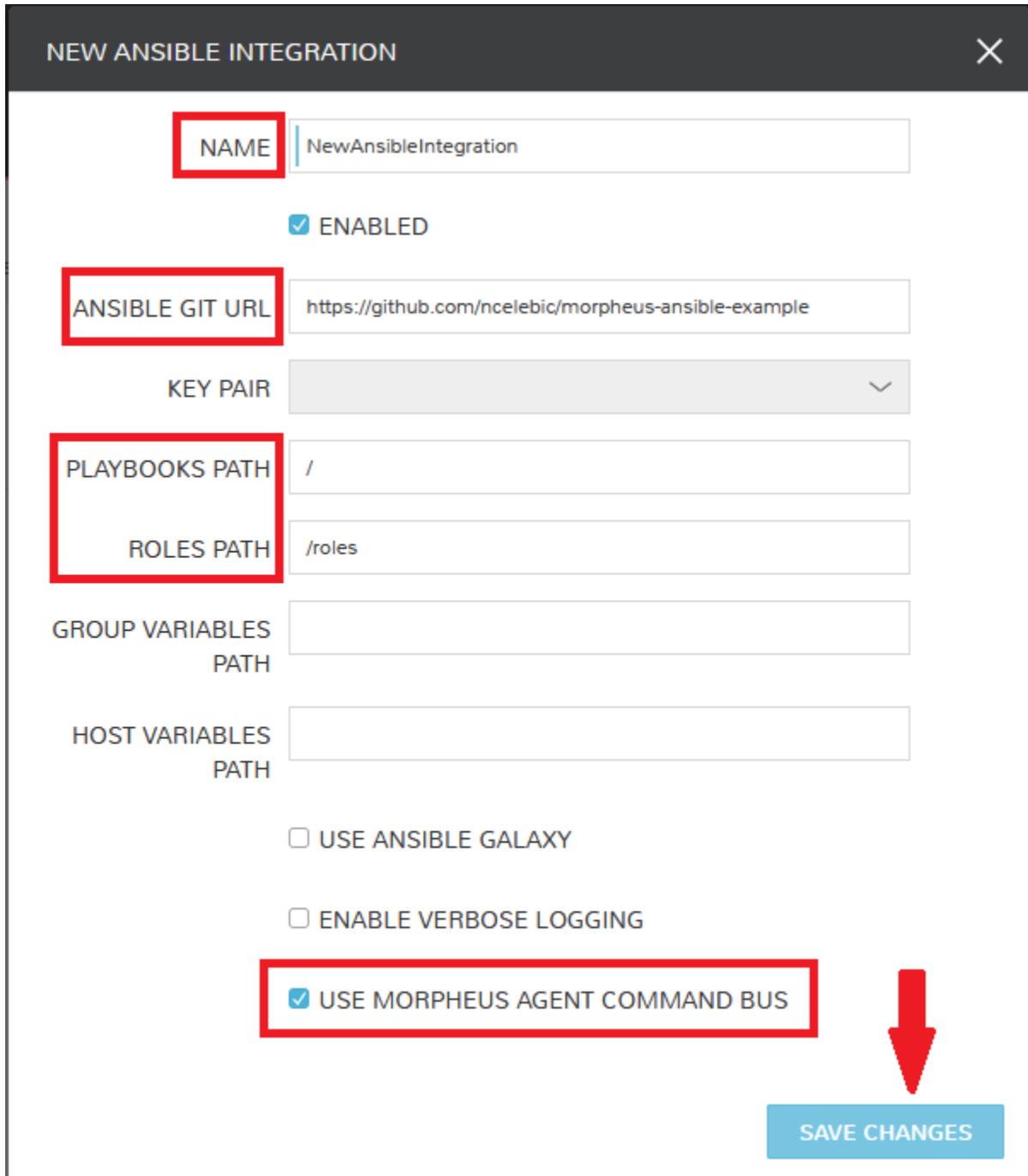


We’ll first set up the Ansible integration, you can integrate with the sample repository referenced here or integrate with your own. Go to ‘Administration > Integrations’. Click “+ NEW INTEGRATION” and select Ansible from the dropdown menu. Fill in the following details:

- **NAME**
- **ANSIBLE GIT URL:** <https://github.com/ncelebic/morpheus-ansible-example>, or enter the URL for your own Ansible git repository
- **PLAYBOOKS PATH**
- **ROLES PATH**

- Mark the box to “USE MORPHEUS AGENT COMMAND BUS”

Note: If your git repository requires authentication, you should create a keypair and use the following URL format: `git@github.com:ncelebic/morpheus-ansible-example.git`.



The screenshot shows a form titled "NEW ANSIBLE INTEGRATION" with a close button (X) in the top right corner. The form contains the following fields and options:

- NAME:** A text input field containing "NewAnsibleIntegration".
- ENABLED:** A checked checkbox.
- ANSIBLE GIT URL:** A text input field containing "https://github.com/ncelebic/morpheus-ansible-example".
- KEY PAIR:** A dropdown menu.
- PLAYBOOKS PATH:** A text input field containing "/".
- ROLES PATH:** A text input field containing "/roles".
- GROUP VARIABLES PATH:** An empty text input field.
- HOST VARIABLES PATH:** An empty text input field.
- USE ANSIBLE GALAXY:** An unchecked checkbox.
- ENABLE VERBOSE LOGGING:** An unchecked checkbox.
- USE MORPHEUS AGENT COMMAND BUS:** A checked checkbox, highlighted with a red box.

A red arrow points from the "USE MORPHEUS AGENT COMMAND BUS" checkbox down to a blue "SAVE CHANGES" button.

Click “SAVE CHANGES”. You’ll now see our new Ansible integration listed among any other configured integrations. If we click on this new integration to view detail, a green checkmark icon indicates the git repository has been fully synced.

With the Ansible integration set up, we can now create a task that includes our playbook. Go to [|LibAut|](#), click “+ADD”.

We'll first set our "TYPE" value to Ansible Playbook so that the correct set of fields appear in the "NEW TASK" wizard. Set the following options:

- **NAME**
- **ANSIBLE REPO:** Here we will choose the Ansible integration that we just created
- **PLAYBOOK:** In our example case, enter 'playbook.yml'

NEW TASK

NAME New Ansible Task

CODE

TYPE Ansible Playbook

ANSIBLE REPO NewAnsibleIntegration

GIT REF

PLAYBOOK playbook.yml

TAGS

SKIP TAGS

COMMAND OPTIONS

Target Options

EXECUTE TARGET Resource

▶ Execute Options

SAVE CHANGES

Click "SAVE CHANGES" to save our new task. We can test the new task on our Apache VM now by going to *[ProIns]* and clicking into our VM. From the "ACTIONS" menu select "Run Task". From the "TASK" dropdown menu, select the task we just added and click "EXECUTE".

EXECUTE TASK?

Are you sure you would like to perform this task operation?

TASK New Ansible Task

CANCEL EXECUTE

To see the progress of the task, click on the "HISTORY" tab and click on the (i) button to the right of each entry in the list. In this case, we can also see the results of the task by clicking on the link in the "LOCATION" column of the "VMS" section.

Now that our task is created, we can put it into a workflow. Back in *[LibAut]* we will click on the "WORKFLOWS" tab. Click "+ADD" and select Provisioning Workflow. We'll give the new workflow a name and expand the Post Provision section. As we begin to type in the name of the task we've created, it should appear as a selection. Click "SAVE CHANGES".

NEW WORKFLOW ✕

NAME

DESCRIPTION

PLATFORM

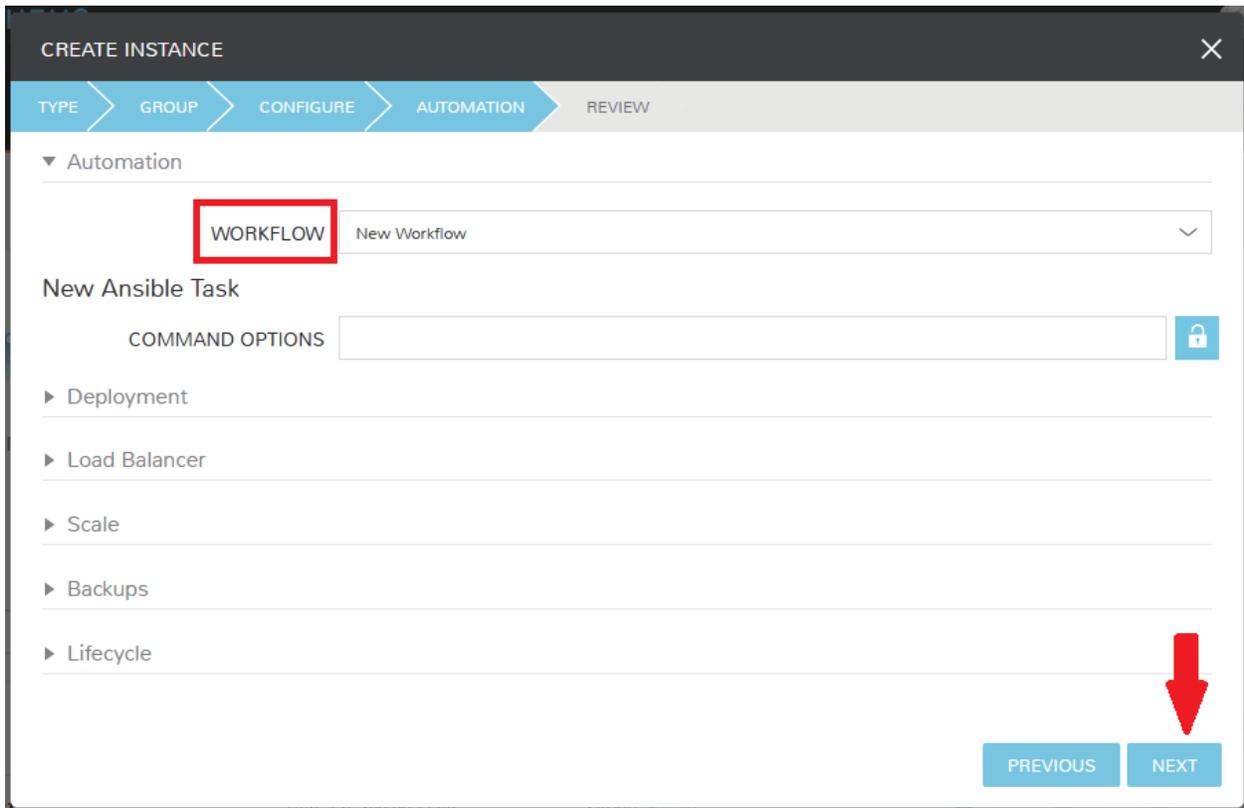
Tasks

- ▶ Pre Provision
- ▼ Provision
 -
 - ▼ Post Provision
 -
 - ▶ **New Ansible Task**
 - ▶ Stop Service
 - ▶ Pre Deploy
 - ▶ Deploy
 - ▶ Reconfigure
 - ▶ Teardown

SAVE CHANGES

Now that we have a workflow, return to `|ProIns|` and begin to provision another Apache instance. More detailed instructions on provisioning a new Apache instance are included earlier in this guide if needed. Now, when you reach the “AUTOMATION” section of the “CREATE INSTANCE” wizard, we have a workflow to select. From the “WORK-

FLOW” dropdown menu, select the workflow we just created and complete provisioning of the new instance.



The screenshot shows the 'CREATE INSTANCE' wizard in Morpheus, currently on the 'AUTOMATION' step. The 'WORKFLOW' dropdown menu is highlighted with a red box and shows 'New Workflow'. Below it is the 'New Ansible Task' section with a 'COMMAND OPTIONS' input field and a lock icon. Further down are sections for 'Deployment', 'Load Balancer', 'Scale', 'Backups', and 'Lifecycle'. At the bottom right, there are 'PREVIOUS' and 'NEXT' buttons, with a red arrow pointing to the 'NEXT' button.

As the instance is provisioning, we can go to the “HISTORY” tab and see Morpheus executing the tasks that were contained in our workflow.

This is just one example of using Morpheus to automate the process of configuring and instance to your needs. There are a number of other automation types that can be built into your workflows as well. For further information, take a look at the Automation Integrations guide in Morpheus docs.

Conclusion

At this point you should be up and running in Morpheus, ready to consume AWS. This guide only scratches the surface, there is a lot more to see and do in Morpheus. Take a look at the rest of [Morpheus Docs](#) for more information on supported integrations and other things possible.

5.13.2 Getting started with Morpheus and VMware

Introduction

This guide is designed to help you get started and quickly get the most out of Morpheus with VMWare. By the end, you will integrate your first cloud, configure networking, prepare and consume images, provision instances, and get started with automation. We will briefly discuss installation and account setup but will provide links to additional resources for those very first steps. For the most part, this guide assumes you are able to get Morpheus installed and are ready to move forward from that point. There is a lot more to see and do in Morpheus that is beyond the scope of this guide. For more, consult the complete Morpheus documentation or take part in our user community forum.

Installation & Setup

In the simplest configuration, Morpheus needs one appliance server which will contain all the components necessary to orchestrate virtual machines and containers. Full requirements, including storage and networking considerations, can be found in Morpheus documentation [here](#). In order to provision any new instances, hosts, or applications, (or convert any discovered resources to managed resources) you will need a valid license. If you don't have one, you can request a lab license for free at [Morpheus Hub](#). Once obtained, the license can be applied in Administration > Settings > License.

Groups

Groups in Morpheus define which resources a user has access to. Clouds are added to groups and a user can only access clouds that are in the groups to which their roles give them access. More information on Morpheus groups is [here](#). A deep dive into groups goes beyond the scope of this guide but it's often useful to create a group that contains all clouds for testing purposes. We will create that group now so that we can add our first cloud into this group in the next section.

Navigate to *Infrastructure > Groups*. Here we will see a list of all configured groups but, of course, this will be empty immediately after installation. Click "+CREATE". Give your group a name, such as "All Clouds". The "CODE" field is used when calling Morpheus through Morpheus API or Morpheus CLI. It's useful in most cases to have an "All Clouds" group for testing purposes so this will likely help you down the road.

NEW GROUP ✕

Configuration

NAME

CODE

LOCATION

▶ Advanced Options

SAVE CHANGES

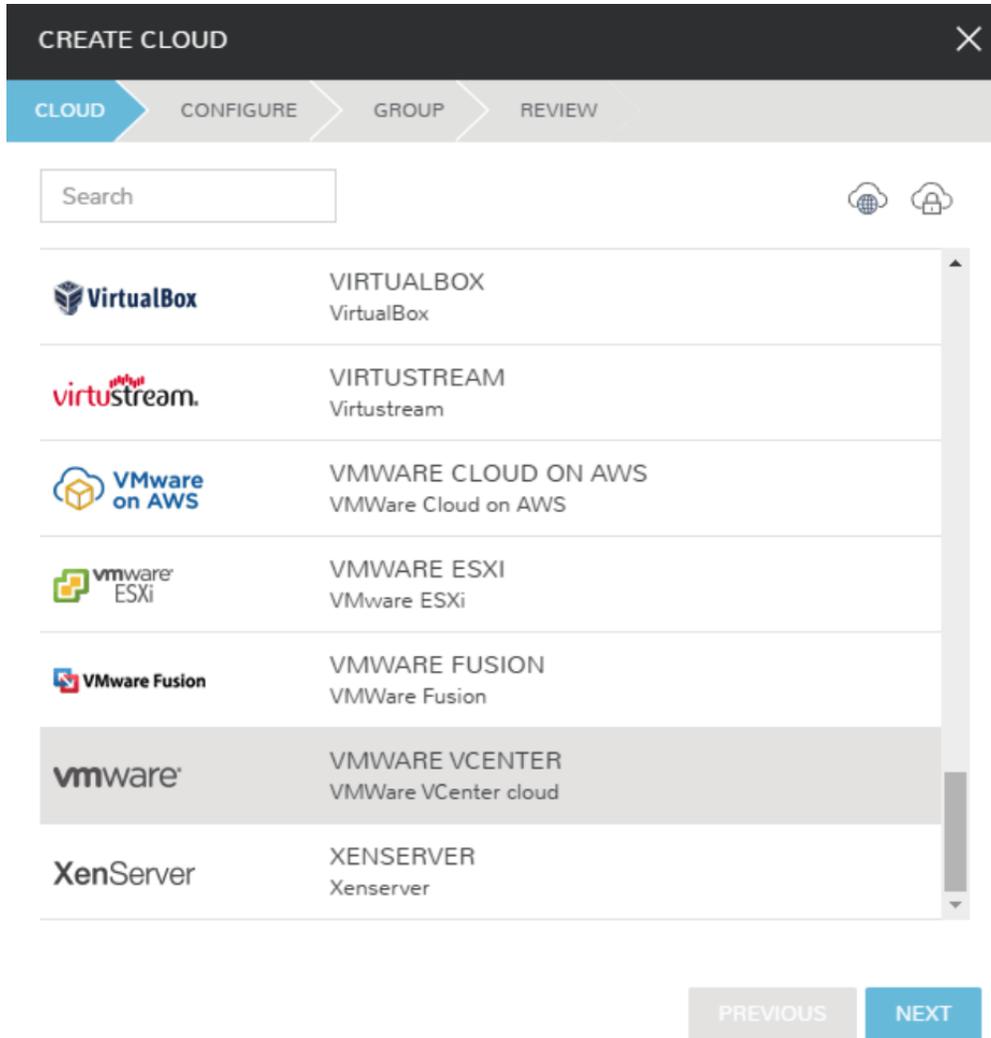
Click "SAVE CHANGES". Your group is now ready to accept clouds.

Integrating Your First Cloud

Clouds in Morpheus consist of any consumable endpoint whether that be On-Prem, Public clouds, or even bare metal. In this guide, we will focus on integrating and working with VMWare vCenter.

To get started, we will navigate to *Infrastructure > Clouds*. This is the cloud detail page which lists all configured clouds. It will be empty if you've just completed installation and setup of Morpheus but soon we will see our integrated vCenter cloud here.

Click the "+ADD" button to pop the "CREATE CLOUD" wizard. Select "VMWARE VCENTER" and click the "NEXT" button.



On the “CONFIGURE” tab, we’re asked to set the initial connection strings into vSphere. The **API URL** should be in the following format: <https://<URL>/sdk>. The **USERNAME** should be in user@domain format.

CREATE CLOUD
✕

CLOUD
CONFIGURE
GROUP
REVIEW

NAME

CODE

LOCATION

Details

API URL

USERNAME

PASSWORD

VERSION ▼

VDC ▼

CLUSTER ▼

Morpheus allows vCenter clouds to be scoped to the **VDC** and **CLUSTER** or even the specific **RESOURCE POOL** if you choose. Once you've entered your URL and credentials, these dropdown menus will become populated.

The **RPC MODE** setting determines how Morpheus will connect to VMs and make configuration and scripting calls if **Morpheus Agent** is not installed. In a VMware environment, it is recommended to select **Vmware Tools** but WinRM/SSH are able to be used to configure the guest directly. If using WinRM, ensure it is configured on the virtual image, discussed below.

Additionally, we can opt to **INVENTORY EXISTING INSTANCES** to begin polling VMs for statistics and rightsizing recommendations as well as **ENABLE HYPERVISOR CONSOLE** to use native vSphere console with port 443 connectivity between Morpheus and ESXi hosts.

To move on, expand the “Advanced Options” section.

Within the “Advanced Options” drawer are additional configurations to consider for your first cloud. Some of these won't usable until they reference additional configured integrations. Common settings to consider are **DOMAIN**, **STORAGE TYPE**, **APPLIANCE URL** (overrides the Morpheus URL for external systems), **GUIDANCE** (setting “Manual” will make recommendations for rightsizing), and **AGENT INSTALL MODE**.

▼ Advanced Options

ENABLE DISK TYPE SELECTION

ENABLE NETWORK INTERFACE TYPE SELECTION

STORAGE TYPE

DOMAIN

SCALE PRIORITY

Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL

TIME ZONE

Once you're satisfied with your selections, click "NEXT"

We have now arrived at the "GROUP" tab. In this case, we will mark the radio button to "USE EXISTING" groups if you wish to use the group we configured earlier.

CREATE CLOUD ✕

CLOUD > CONFIGURE > GROUP > REVIEW

USE EXISTING CREATE NEW

GROUP

PREVIOUS NEXT

Once you've selected the group, click "NEXT"

On the final tab of the "CREATE CLOUD" wizard, you'll confirm your selections and click "COMPLETE". The new cloud is now listed on the cloud detail page. After a short time, Morpheus will provide summary information and statistics on existing virtual machines, networks, and other resources available in the cloud.

Viewing Cloud Inventory

Now that we've integrated our first VMware cloud, we can stop for a moment to review what Morpheus gives us from the cloud detail page. We can see that Morpheus gives us estimated costs and cost histories, metrics on used resources, and also lists out resource counts in various categories including container hosts, hypervisors, and virtual machines. We can drill into these categories to see lists of resources in the various categories individual resources within them by clicking on the category tabs. We can link to the detail page for any specific resource by clicking on it from its resource category list.

Configuring Resource Pools

With our VMware cloud configured, Morpheus will automatically sync in available resource pools and data stores.

For resource pools, once Morpheus has had time to ingest them, then will be visible from the cloud detail page. Navigate to *Infrastructure > Clouds > (your VMware cloud) > RESOURCES tab*. In here, we are able to see and control access to the various resource pools that have been configured in vCenter. For example, we can restrict access to a specific resource pool within Morpheus completely by clicking on the "ACTIONS" button, then clicking "Edit". If we unmark the "ACTIVE" button and then click "SAVE CHANGES" we will see that the resource pool is now grayed out in the list. The resources contained in that pool will not be accessible for provisioning within Morpheus.

POOLS

NAME	DESCRIPTION	VISIBILITY	DEFAULT	TENANT	
Demo-vSAN		Private		morpheus	ACTIONS ▾
Demo		Private		morpheus	ACTIONS ▾
Pied Piper		Private		morpheus	ACTIONS ▾
Prod		Private		morpheus	ACTIONS ▾
Sandbox		Private		morpheus	ACTIONS ▾

Often our clients will want to make specific blocks of resources available to their own customers. This can be easily and conveniently controlled through the same "EDIT RESOURCE POOL" dialog box we were just working in. If we expand the "Group Access" drawer, we are able to give or remove access to each pool to any group we'd like. We can also choose to make some or all of our resource pools available to every group. Specific resource pools can also be defined as the default for each group if needed.

EDIT RESOURCE POOL
✕

NAME

MOVE SERVERS

ACTIVE

DEFAULT

▼ Group Access

GROUP	ACCESS	DEFAULT
all	<input checked="" type="checkbox"/>	
AA	<input type="checkbox"/>	<input type="checkbox"/>
AA-DB	<input type="checkbox"/>	<input type="checkbox"/>

Additionally, we may choose to allow only certain service plans to be provisioned into a specific pool of resources. For example, perhaps a specific cluster is my SQL cluster and only specific services plans should be consumable within it. We can control that through this same dialog box.

Configuring Data Stores

To take a look at data stores, we'll move from the "RESOURCES" tab to the "DATA STORES" tab on our cloud detail page.

Morpheus gives the user similar control with data stores to what we saw with our resources pools earlier. Just like with resource pools, we can disable access within Morpheus completely by clicking on "ACTIONS" and then "Edit". If we unmark the "ACTIVE" checkbox and click "SAVE CHANGES", you will see that specific data store has been grayed out.

SUMMARY CLUSTERS HOSTS VMS CONTAINERS LOAD BALANCERS NETWORKS DATA STORES RESOURCES POLICIES WIKI						
NAME	TYPE	CAPACITY	ONLINE	VISIBILITY	TENANT	ACTIONS
<input type="checkbox"/> ds-65-root	Vmfs	461.1GiB	Yes	Private	morpheus	ACTIONS ▼

Just like with resource pools, we are also able to scope data stores to specific groups. This ensures that the members of each group are only able to consume the data stores they should have access to.

EDIT DATA STORE
✕

NAME

ACTIVE

▼ Group Access

GROUP	ACCESS
all	<input checked="" type="checkbox"/>
AA	<input type="checkbox"/>
AA-DB	<input type="checkbox"/>

Configuring Folders

Still within the “RESOURCES” tab, within the “FOLDERS” subtab we see the folders discovered from the vCenter Cloud. Edit folder configurations by selecting “ACTIONS” from the end of the row, then clicking “Edit”. Consider the following configurations for specific folders:

- **DEFAULT:** If selected, this folder will be pre-selected when provisioning new Instances to this Cloud (See the Folder option on the CONFIGURE tab of the Instance provisioning wizard)
- **IMAGE TARGET:** Morpheus will look in the image target folder(s) to onboard VMware images

After saving the changes, you’ll see any folders set as default or image target indicated in the folders list.

Configuring Network for Provisioning

When configuring networking, we can set global defaults by going to *Infrastructure > Network > NETWORKS tab*. Here we can add or configure networks from all clouds integrated into Morpheus. Depending on the number of clouds Morpheus has ingested, this list may be quite large and may also be paginated across a large number of pages. In such a case, it may be easier to view or configure networks from the specific cloud detail page so that networks from other clouds are not shown.

NETWORKS

[NETWORKS](#)
[NETWORK GROUPS](#)
[ROUTERS](#)
[IP POOLS](#)
[DOMAINS](#)
[PROXIES](#)
[SECURITY GROUPS](#)
[INTEGRATIONS](#)

Search

NAME	TYPE	CLOUD	CIDR	POOL	DHCP	VISIBILITY	TENANTS	ACTIONS
10.30.20.0	OracleVM Network	Morpheus Oracle VM	10.30.20.0/22	✓	Public	morpheus		ACTIONS ▼
172.31.0.0/20 (subnet-63dff13b)	Amazon Subnet	ah-only	172.31.0.0/20	✓	Private	morpheus		ACTIONS ▼
172.31.16.0/20 (subnet-72110ed6)	Amazon Subnet	ah-only	172.31.16.0/20	✓	Private	morpheus		ACTIONS ▼

Still in *Infrastructure > Network*, make note of the “INTEGRATIONS” tab. It’s here that we can set up any integrations that may be relevant, such as IPAM integrations. Generally speaking, when adding IPAM integrations, we simply need to name our new integration, give the API URL, and provide credentials. There’s more information in the [IPAM integration](#) section of Morpheus Docs.

ADD IPAM INTEGRATION
✕

NAME

ENABLED

URL

USERNAME

PASSWORD

THROTTLE RATE ms

In *Infrastructure > Networking* we can also set up IP address pools from the IP Pools tab. These pools can be manually defined, known as a Morpheus-type IP pool, or they can come from any IPAM integrations you’ve configured. As instances are provisioned, Morpheus will assign IP addresses from the pool chosen during provisioning. When the instance is later dissolved, Morpheus will automatically release the IP address to be used by another instance when needed. When adding or editing a network, we can opt to scope the network to one of these configured IP address pools. Edit an existing network by clicking the pencil icon on the Networks List Page (*Infrastructure > Networks > Networks Tab*) and fill in the “Network Pool” field to associate the IP Pool with the network.

CREATE NETWORK POOL
✕

NAME

POOL TYPE

IP Ranges

STARTING ADDRESS	-	ENDING ADDRESS	
192.168.0.2	-	192.168.0.255	+

SAVE CHANGES

Since this guide is focused on working within a VMware cloud that we integrated at the start, we will take a look at our network configurations on the cloud detail page as well. Navigate to *Infrastructure > Clouds > (your VMware cloud) > NETWORKS tab*. Just as with resource pools and data stores, we have the ability to make certain networks inactive in Morpheus, or scope them to be usable only for certain groups or tenants.

NETWORKS						
NAME	TYPE	CIDR	POOL	DHCP VISIBILITY	TENANT	ACTIONS
<input type="checkbox"/> default	KVM Host Bridge	192.168.122.1/24		✓ Private	morpheus	ACTIONS ▾
<input type="checkbox"/> docker bridge	Docker Bridge			✓ Private	morpheus	ACTIONS ▾
<input type="checkbox"/> Garf-Network	Overlay	0.0.0.0/10	Vanishing-IP-Test	Private	morpheus	ACTIONS ▾

Prepping an Image

As we'll discuss and try out in the next section, Morpheus comes out of the box with a default set of blueprints that are relevant to many modern deployment scenarios. For the most part, these are base operating system images with a few additional adjustments. However, in many on-premise deployments, there are often custom image and networking requirements. We will work with images included in Morpheus by default in this guide but it's important to discuss how to prep custom images as well.

Creating a Windows Image

The following versions of Windows Server are supported:

- 2008 R2
- 2012
- 2012 R2
- 2016
- 2019
- 2022

To start, create a new Windows VM in vCenter using a base version of your selected Windows build.

Note: It's recommended to make the VMDK drive as small as possible for your purposes as this generally speeds cloning and deploy times. Morpheus provisioning and post-deploy scripts allow to expand the drive to any size that you need.

Once the VM is created, ensure VMware Tools is installed on the operating system. This will ensure the virtual machine can communicate correctly with vCenter but also if the cloud was configured to use `RPC MODE` of VMware Tools, and the Morpheus agent is not installed, Morpheus can communicate with the virtual machine directly for tasks and other processes.

Note: If the cloud was configured to use `RPC MODE` of SSH/WinRM, it will be necessary to configure WinRM on the virtual machine. Ensure WinRM is allowed from your Morpheus appliance through the local Windows firewall. Also, run the following command to enable WinRM in the guest: `winrm quickconfig`

Next, install .NET 4.5.2 or higher as a [requirement](#) for the Morpheus agent.

Finally, choose a method that will be used to customize the operating system:

- Guest Customization

- Sysprep
- Cloudbase-Init

Using Guest Customization (Recommended)

Morpheus can utilize the vCenter guest customizations feature, creating its own customization specification and applying it at deployment. There are no other changes needed in the guest operating system when using this method.

Turn off the virtual machine and convert it to a template

Refresh your cloud (Infrastructure > Clouds > Click Cloud > *Refresh* > Short). Once the Virtual Image has been synced into Library > Virtual Images, edit it and ensure **VMware Guest Customization?** is checked.

Using Windows Sysprep

Morpheus can inject an unattend file via CD-ROM to override the default sysprep process when preparing a virtual machine. Run the following command from the guest operating system to sysprep the operating system:

```
C:\Windows\System32\sysprep /oobe /generalize /shutdown
```

The unattend file created on the virtual machine can be ignored, no changes will be needed. Once the virtual machine has completed shutting down, convert it to a template.

Refresh your cloud (Infrastructure > Clouds > Click Cloud > *Refresh* > Short). Once the Virtual Image has been synced into Library > Virtual Images, edit it and ensure **Sysprepped/Generalized Image?** is checked.

Note: If **VMware Guest Customization?** is enabled on the virtual image or if using static IP addresses or IP pools when provisioning, ensure a Sysprep has not been performed in the guest. In such cases, a guest customization will always be performed.

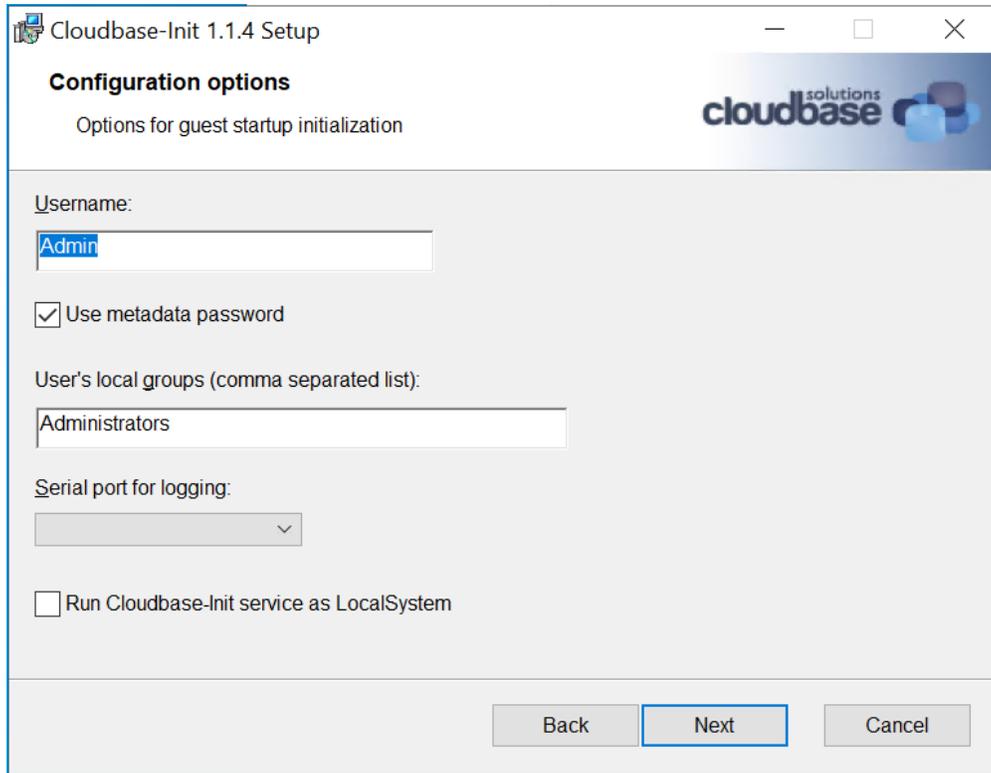
Using Cloudbase-init

Important: Some admins may struggle with using Cloudbase-Init, especially if they are not familiar with cloud-init (used in Linux). It is recommended to use either **Guest Customization** or **Windows Sysprep** if the admins are more familiar with those tools.

Morpheus can inject a cloud-init file via CD-ROM to override the default sysprep process when preparing a virtual machine, if the Cloudbase-Init service is installed.

Download Cloudbase-Init from <https://cloudbase.it> for your appropriate architecture.

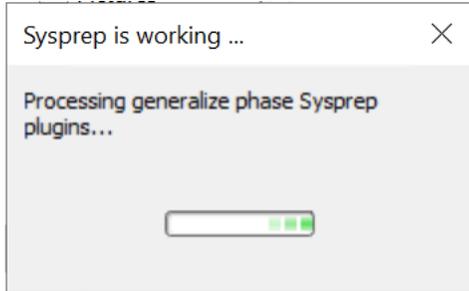
Run the installer and keep the default settings. Below is an example of the final settings page:



At the end of the installer, options will be available to sysprep and shutdown the virtual machine. Checkmark both boxes and click **Finish**:



Once started, the sysprep process should continue until the virtual machine shutdown.



Once the virtual machine has completed shutting down, convert it to a template.

Refresh your cloud (Infrastructure > Clouds > Click Cloud > *Refresh* > Short). Once the Virtual Image has been synced into Library > Virtual Images, edit it and ensure **Is Cloud Init Enabled?** is checked.

Creating a CentOS/RHEL Image

Create a new machine in vCenter and install a base version of your preferred Linux distro.

Note: If you are using cloud-init as part of your image, you will need to ensure your virtual machine has a cdrom.

Before installing the operating system, set up a single ext or xfs partition without a swap disk. Next, install the distro applying any updates to the operating system or security updates. Once the operating system is running and updated, install the following:

```
yum install cloud-init
yum install cloud-utils-growpart
yum install open-vm-tools
yum install git
yum install epel-release
```

Set selinux to permissive as the enforced setting can cause problems with cloud-init:

```
sudo vi /etc/selinux/config
```

Cloud-Init

We'll get started by installing cloud-init using the following command:

```
yum -y install epel-release
yum -y install git wget ntp curl cloud-init dracut-modules-growroot
rpm -qa kernel | sed 's/^kernel-//' | xargs -I {} dracut -f /boot/initramfs-{}.img {}
```

Note: The above command will install some core dependencies for cloud-init and automation later as you work with your provisioned instances. For example, we install Git here as it is used for Ansible automation. If you had no plans to use Ansible, this installation could be skipped. The dracut-modules-growroot is responsible for resizing the root partition upon initial boot which was potentially adjusted during provisioning.

One key benefit of using cloud-init is that we don't have to lock credentials into the blueprint. We recommend configuring a default cloud-init user that will get created automatically when the VM is booted by cloud-init. We can define that default user in Administration > Settings > Provisioning > Cloud-Init

Network Interfaces

As of CentOS 7, network interface naming conventions have changed. You can check this by running *ifconfig* and noting that the primary network interface has some value similar to “ens2344”. The naming is dynamic and typically set based on hardware ID. We don’t want this to fluctuate when provisioning this blueprint in our VMware environments. To accomplish this end, we will rename the interface back to “eth0” using the steps below.

First, adjust the bootloader to disable interface naming:

```
sed -i -e 's/quiet/quiet net.ifnames=0 biosdevname=0/' /etc/default/grub
grub2-mkconfig -o /boot/grub2/grub.cfg
```

The next step is to adjust network scripts in CentOS. Start by confirming the presence of a file called */etc/sysconfig/network-scripts/ifcfg-eth0*. Once confirmed, run the following script:

```
export iface_file=$(basename "$(find /etc/sysconfig/network-scripts/ -name 'ifcfg*' -not_
↵-name 'ifcfg-lo' | head -n 1)")
export iface_name=${iface_file:6}
echo $iface_file
echo $iface_name
sudo mv /etc/sysconfig/network-scripts/$iface_file /etc/sysconfig/network-scripts/ifcfg-
↵eth0
sudo sed -i -e "s/$iface_name/eth0/" /etc/sysconfig/network-scripts/ifcfg-eth0
sudo bash -c 'echo NM_CONTROLLED="no" >> /etc/sysconfig/network-scripts/ifcfg-eth0'
```

This script tries to confirm there is a new *ifcfg-eth0* config created to replace the old config file. Confirm this config exists after running and if not you will have to build your own:

```
TYPE=Ethernet
DEVICE=eth0
NAME=eth0
ONBOOT=yes
NM_CONTROLLED="no"
BOOTPROTO="dhcp"
DEFROUTE=yes
```

For more on CentOS/RHEL image prep, including additional configurations for specific scenarios, take a look at the [VMware image prep](#) page in Morpheus Docs.

Creating an Ubuntu 20.04 Image

Download the Ubuntu 20.04 ISO from Canonical, and upload the base image to vCenter. Then, create a new virtual machine in vCenter.

Note: Since we’ll include cloud-init with our image, we will need to ensure the virtual machine has a cdrom. Select the Ubuntu 20.04 ISO we just downloaded from the CD/DVD drive dropdown menu when creating the new virtual machine.

Before installing the operating system, set up a single ext partition without a swap disk. Then, continue on installing Ubuntu making the following selections during the setup process:

- Update to the latest installer if a later version is available
- Use the entire disk and deselect the option to set up the disk as an LVM group
- Configure an account and set a password
- Opt to install OpenSSH Server
- Other optional packages aren’t needed for this basic Ubuntu image

Complete the installation process and reboot the machine. Update the package list and apply any upgrades:

```
apt-get update
apt-get upgrade
```

Disable assignment of new styled names for network interfaces (instead of ens### they will be eth#):

```
sudo sed -i -e 's/GRUB_CMDLINE_LINUX=""/GRUB_CMDLINE_LINUX="net.ifnames=0 biosdevname=0"/
↪' /etc/default/grub
```

Update GRUB:

```
update-grub
```

Update the 70-persistent-net.rules file:

```
cat << EOF > /etc/udev/rules.d/70-persistent-net.rules
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{dev_id}=="0x0", ATTR{type}=="1",
↪NAME="eth0"
EOF
```

Remove subiquity-disable-cloudinit-networking.cfg as cloud-init will skip network configuration if it's present:

```
rm -f /etc/cloud/cloud.cfg.d/subiquity-disable-cloudinit-networking.cfg
```

Update 99-pve.cfg:

```
cat << EOF > /etc/cloud/cloud.cfg.d/99-pve.cfg
datasource_list: [ConfigDrive, NoCloud]
EOF
```

Remove Netplan files, they will not be regenerated if they exist:

```
rm -f /etc/netplan/00-installer-config.yaml
rm -f /etc/netplan/50-cloud-init.yaml
```

Run cloud-init clean:

```
cloud-init clean
```

Next, reboot the system and confirm the network interface is labeled eth0 once the machine comes back up. Then, clear BASH history for root. The history entry has a copy in the memory and it will flush back to the file when you log out. You can avoid this with the following command:

```
cat /dev/null > ~/.bash_history && history -c && exit
```

Shutdown the system:

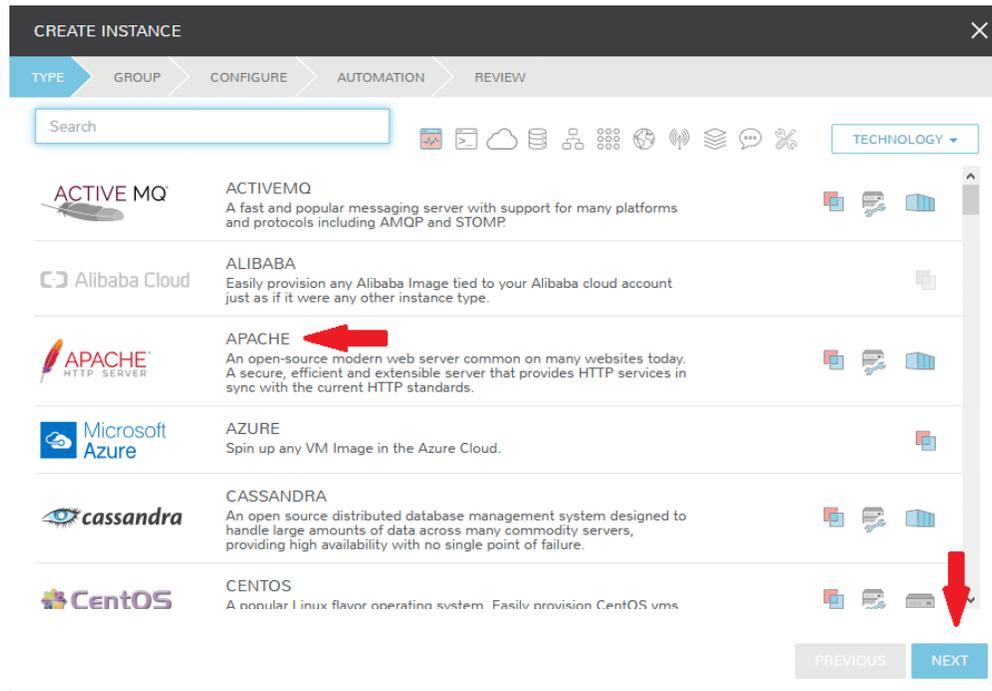
```
shutdown -h now
```

Convert the VM to a template in vCenter before moving back to Morpheus to onboard the image and use it to begin building your provisioning library.

Provisioning Your First Instance

At this point, we are ready to provision our first image. As a first instance, we'll provision an Apache web server to our vCenter cloud.

Navigate to Provisioning > Instances. If any instances are currently provisioned, we will see them listed here. To start a new instance we click the "+ADD" button to pop the "CREATE INSTANCE" wizard. We'll scroll down to and select the Apache instance type and click "NEXT".



First, we'll specify the group to provision into which determines the clouds available. If you've followed this guide to this point, you should at least have a group that houses all of your clouds which you can select here. This will allow us to select the vCenter cloud from the "CLOUD" dropdown menu. Provide a unique name to this instance and then click "NEXT"

From the "CONFIGURE" tab, we're presented with a number of options. The options are cloud and layout-specific, more generalized information on creating instances and available options is [here](#). For our purposes, we'll select the following options:

- **LAYOUT:** Includes options such as the base OS, custom layouts will also be here when available
- **PLAN:** Select the resource plan for your instance. Some plans have minimum resource limits, Morpheus will only show plans at or above these limits. User-defined plans can also be created in Administration > Plans & Pricing.
- **VOLUMES and DATASTORES:** The minimum disk space is set by the plan, this value may be locked if you've selected a custom plan that defines the volume size
- **NETWORKS:** Select a network, note that IP pools must be linked with the networks defined in VMware in order to assign static IP addresses

Under the "User Config" drawer, mark the box to "CREATE YOUR USER". Click "NEXT".

CREATE INSTANCE

TYPE > GROUP > CONFIGURE > AUTOMATION > REVIEW

Configuration Options

LAYOUT ESXi Apache on Ubuntu 14.04

PLAN 1 Core, 512MB Memory
Cores: 1 Memory: 512 MB

VOLUMES root 10 GB Auto - Datastore +

NETWORKS VM Network DHCP +

▼ User Config

CREATE YOUR USER

USER GROUP Select

▶ Network Options

▶ Advanced Options

▶ Metadata

▶ Environment

PREVIOUS NEXT

Note: “CREATE YOUR USER” will seed a user account into the VM with credentials set in your Morpheus user account settings. If you’ve not yet defined these credentials, you can do so by clicking on your username in the upper-right corner of the application window and selecting “USER SETTINGS”.

For now, we’ll simply click “NEXT” to move through the “AUTOMATION” tab but feel free to stop and take a look at the available selections here. There is more information later in this guide on automation and even more beyond that in the rest of Morpheus docs.

Review the settings for your first instance and click “COMPLETE”.

CREATE INSTANCE

TYPE GROUP CONFIGURE AUTOMATION REVIEW

APACHE HTTP SERVER apachetest admin (esxtank)

Summary

INSTANCE OPTIONS

NAME: apachetest
GROUP: admin
CLOUD: esxtank
TYPE: Apache
PLAN: 1 Core, 512MB Memory
Cores: 1 Memory: 512MB
VERSION: 2.4
LAYOUT: ESXi Apache on Ubuntu 14.04

VOLUMES

ROOT: 10 GB Auto - Datastore SCSI 0

NETWORKS

VM NETWORK: DHCP

Options

PREVIOUS COMPLETE

We are now dropped back onto the instances list page. We can see a new entry in the list at this point with a status indicator that the new machine is being launched (rocket icon in the status field). We can double click on the instance in the list to move to the instance detail page. For now we will see a progress bar indicating that the instance is being created and is starting up. The exact amount of time this process will take depends on your environment and selections made when provisioning the instance. Initially, Morpheus will guess as to how long this will take and the progress bar may not be accurate. Over time, Morpheus will learn how long these processes take and progress bar accuracy will improve. For more detailed information on the status of various provisioning processes, we can scroll down and select the “HISTORY” tab. The “STATUS” icon will change from the blue rocket to a green play button when the instance is fully ready. Furthermore, we can click on the hyperlinked IP address in the “VMS” section of this page to view a default page in a web browser to confirm success.

INFO

APACHE HTTP SERVER

Group: admin
Created By: Nick Celebic
Cores: 1
Source Image: Morpheus Apache 2.4 on Ubuntu 14.04.3

Cloud: esxtank
Layout: ESXi Apache on Ubuntu 14.04
Memory: 512.0MiB

Date Created: 11/13/2019 01:44 PM
Version: 2.4
Total Storage: 10.0GiB

VMS

STATUS	NAME	TYPE	CLOUD	LOCATION	COMPUTE	MEMORY	STORAGE	ACTIONS
	apachetest	Apache 2.4	esxtank		0	0	0	

SUMMARY WIKI DEPLOY STORAGE NETWORK LOGS BACKUPS ENVIRONMENT SCALE HISTORY

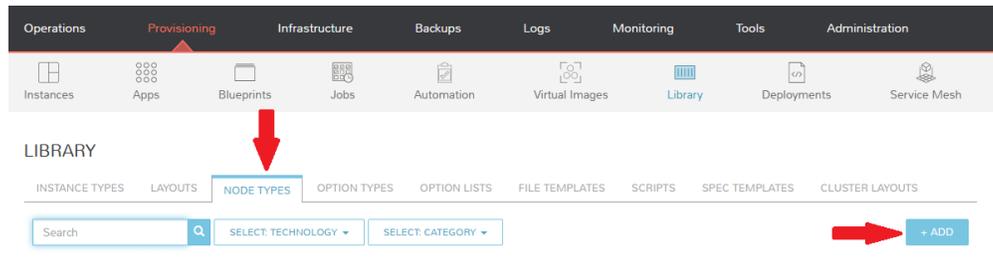
NAME	DESCRIPTION	CREATED BY	START DATE	ETA/DURATION	STATUS	ERROR
apachetest	Provision	Nick Celebic	11/13/2019 01:44 PM	00:18:44		
	Prepare Resources		11/13/2019 01:44 PM	00:18:44	Complete	
	Prepare Image		11/13/2019 01:44 PM	00:00:01	Complete	
	Configure Instance		11/13/2019 01:44 PM	260ms	Complete	
	Deploy Instance		11/13/2019 01:44 PM	00:18:44		

Creating Your First Library Item

In the prior section, we manually provisioned our first instance. However, Morpheus allows you to build a catalog of custom provisionable items to simplify and speed provisioning in the future. In this section, we'll build a catalog item and show how that can translate into quick instance provisioning after configuration.

Note: Before starting this process, it's important to decide which virtual image you plan to use. If you're not using a Morpheus-provided image, you'll want to ensure it's uploaded. You will not be able to complete this section without selecting an available image. In this example we will use Morpheus Redis 3.0 on Ubuntu 14.04.3 v2.

Navigate to Library > Blueprints > Node Types and click "+ADD".



In this example, I am going to set the following options in the "NEW NODE TYPE" wizard:

- **NAME**
- **SHORT NAME**
- **VERSION:** 1 (In this particular case, the version is not important)
- **TECHNOLOGY:** VMware
- **VM IMAGE:** Morpheus Redis 3.0 on Ubuntu 14.04.3 v2

Note: Within the "VMware VM Options" section you should add anything that will always be used for this node, regardless of the specific deployment details. This can include LDAP Authentication, bash scripts that should run on installation, among other things.

NAME

SHORT NAME
The short name is a name with no spaces used for display in your container list.

VERSION

TECHNOLOGY

ENVIRONMENT VARIABLES

Name	Value	
		⚙️ +

VMware VM Options

VM IMAGE

LOG FOLDER

CONFIG FOLDER

DEPLOY FOLDER
(Optional) If using deployment services, this mount point will be replaced with the contents of said deployments.

EXTRA OPTIONS

Name	Value	
		🗑️ +

SERVICE PORTS

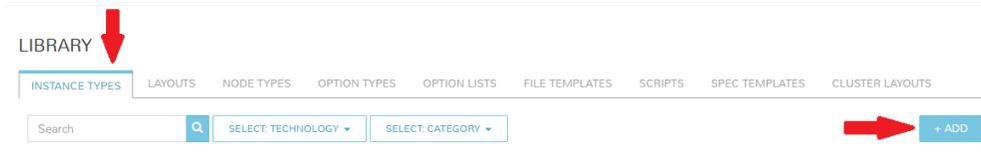
SCRIPTS

FILE TEMPLATES

▶ **Advanced Options**

SAVE CHANGES

With the new node created, we'll now add a new instance type which will be accessible from the provisioning wizard once created. Move from the "NODE TYPES" tab to the "INSTANCE TYPES" tab and click "+ADD".



In the “NEW INSTANCE TYPE” wizard, I’ll simply enter a **NAME** and **CODE** value. Click “SAVE CHANGES”.

NEW INSTANCE TYPE

NAME NewInstanceType

CODE newtype

Useful shortcode for provisioning naming schemes and export reference.

DESCRIPTION

255 Characters Remaining

CATEGORY Web

ICON Browse

Suggested Dimensions: 150 x 51

Now that we’ve created a new instance type, access it by clicking on the name in the list of custom instances you’ve created. In my case, I’ve given the name “NewInstanceType”.

LIBRARY

INSTANCE TYPES LAYOUTS NODE TYPES OPTION TYPES OPTION LISTS FILE TEMPLATES SCRIPTS SPEC TEMPLATES CLUSTER LAYOUTS

Search [magnifying glass icon] SELECT. TECHNOLOGY SELECT. CATEGORY + ADD

	NAME	TECHNOLOGY	CATEGORY	FEATURED	
	NewInstanceType		Web		ACTIONS
	ActiveMQ	Mixed	Messaging		ACTIONS
	Alibaba	Alibaba	Cloud		ACTIONS
	Amazon Api		Apps		ACTIONS
	AmazonMQ		Messaging		ACTIONS

Once we’ve opened the new instance type, by default, we should be on the “LAYOUTS” tab. Click “+ADD LAYOUT”.

I’ve set the following fields on my example layout:

- **NAME**
- **VERSION:** This is the version number of the layout itself, which is labeled 1.0 in the example
- **TECHNOLOGY:** VMware

- **Nodes:** Select the node we created earlier, if desired you can specify multiple nodes

Click “SAVE CHANGES”.

NEW LAYOUT
✕

NAME

VERSION

DESCRIPTION

CREATABLE

TECHNOLOGY

MINIMUM MEMORY

This will override any memory requirement set on the virtual image

WORKFLOW

SUPPORTS CONVERT TO MANAGED

ENVIRONMENT VARIABLES

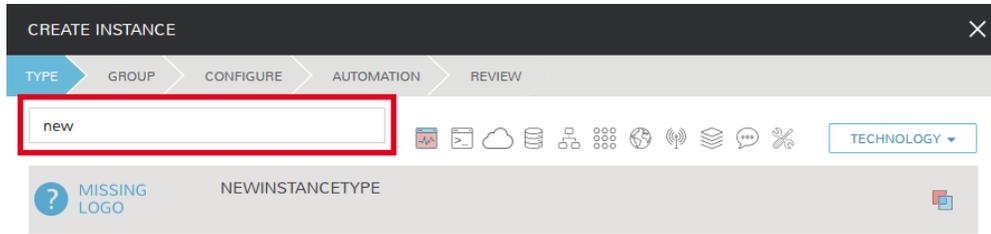
Name	Value	
		⚙️ +

Option Types

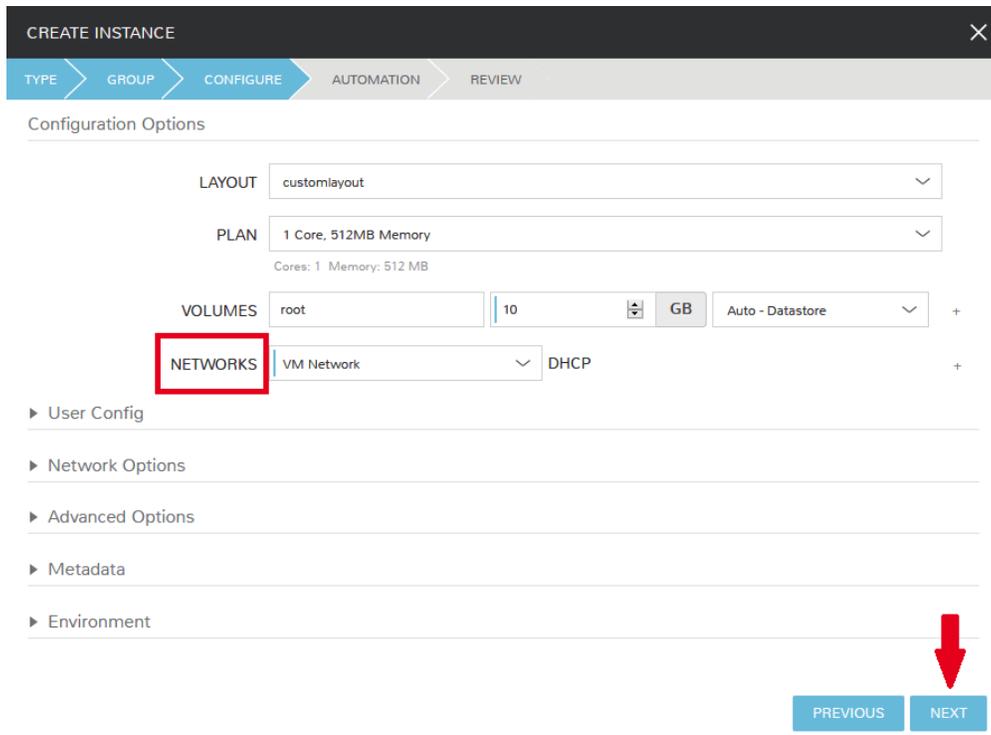
Nodes

customnodetypevm (1)

At this point we’ve completed the setup work and can now provision the instance we’ve created to our specifications. Navigate to Provisioning > Instances and click “+ADD”. From the search bar we can search for the new instance type we’ve created. In the example case, we called it “newinstancetype”. Click “NEXT”.



As before, we can select a group and cloud to provision this new instance. Click “NEXT”. On the “CONFIGURE” tab, make note that the layout and plan are already selected because they were configured as part of creating the new instance type. Select a network and click “NEXT”. Once again we will also click “NEXT” through the “AUTOMATION” tab. Finally, click “COMPLETE”.



As before when we manually provisioned an instance, Morpheus will now begin to spin up the new VM. How long this will take depends on your environment but Morpheus will predict how long this process will take and represent that on a progress bar. Over time, Morpheus begins to learn how long these processes take and becomes more accurate in

predicting spin-up time.

Once the provisioning process has completed, open the instance detail page in Morpheus and click on the “CONSOLE” tab. You’ll be logged in with your user account and are then able to confirm the machine is ready and available.

The screenshot displays the Morpheus interface for an instance named 'newinstance'. At the top, there are buttons for 'EDIT', 'ACTIONS', and 'DELETE'. Below this, a row of metrics includes Status (Running), Health (OK), Last Backup, Availability (100.00%), Response Time (N/A), Max CPU (1%), Memory (39%), and Storage (27%). The 'INFO' section provides details such as Group (admin), Cloud (esxtank), and Date Created (11/14/2019 12:21 PM). A table under 'VMS' lists the instance with a red arrow pointing to the 'CONSOLE' tab. The console output shows a terminal session on Ubuntu 14.04.3 LTS.

Automation and Configuration Management

Morpheus automation is composed of Tasks and Workflows. A task could be a script added directly, scripts or blueprints pulled from the Morpheus Library, playbooks, recipes, or a number of other things. The complete list of task types can be found in the [Automation](#) section of Morpheus docs. Tasks can be executed individually but they are often combined into workflows. We can opt to run a workflow at provision time or they can be executed on existing instances through the Actions menu.

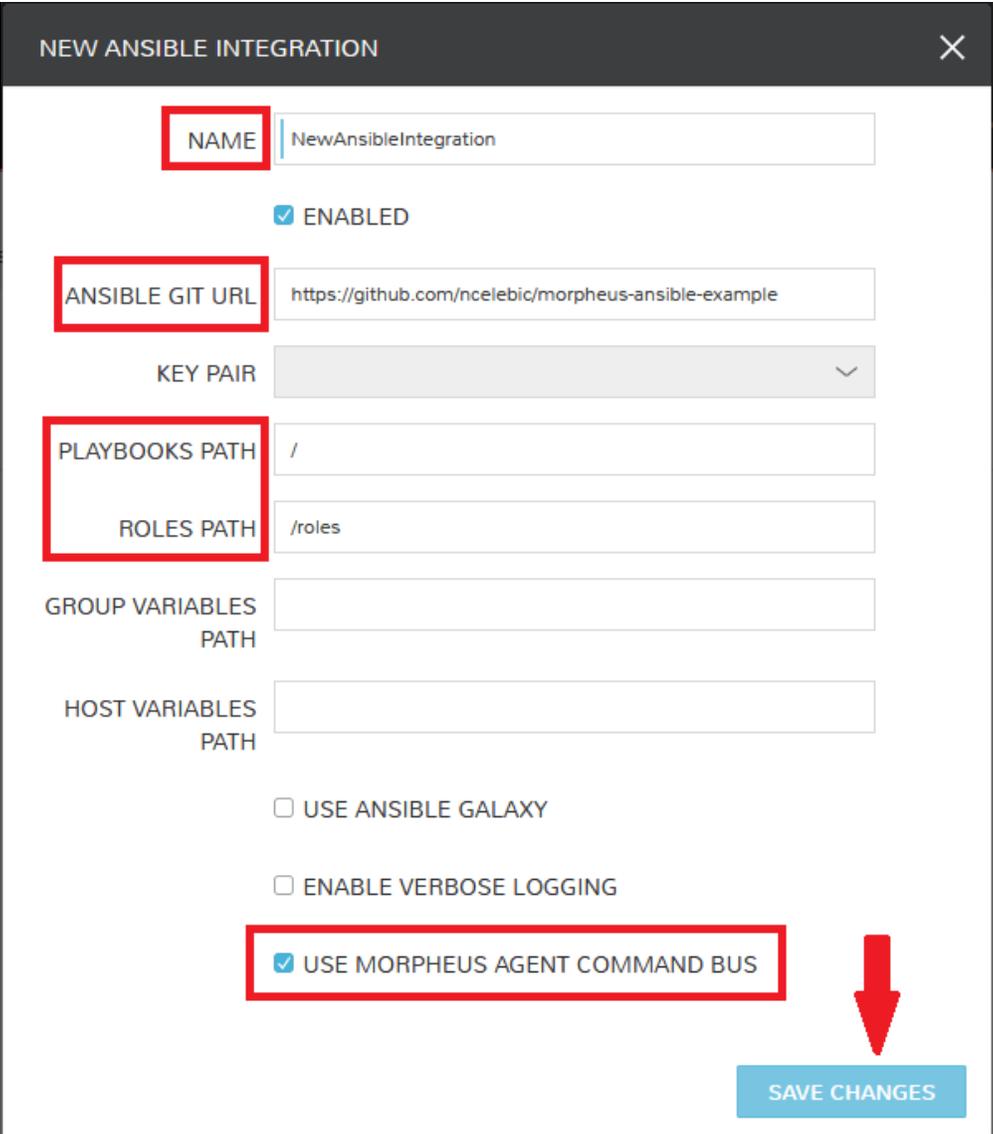
In this guide we will set up an Ansible integration, create a task, add the task to a workflow, and run the workflow against a new and existing instance. If you’ve worked through this guide to this point, you should already have an Apache instance running. If you don’t yet have that, provision one before continuing with this guide and ensure it’s reachable on port 80.

The screenshot shows the 'Administration' section of the Morpheus interface. Under the 'INTEGRATIONS' heading, there is a '+ NEW INTEGRATION +' button. A dropdown menu is open, showing various integration options: automation, Chef, Puppet, Ansible, Ansible Tower, vRealize Orchestrator, Salt Master, dns, and Microsoft DNS. Red arrows point to the '+ NEW INTEGRATION +' button and the 'Ansible' option in the dropdown.

We'll first set up the Ansible integration, you can integrate with the sample repository referenced here or integrate with your own. Go to 'Administration > Integrations'. Click "+NEW INTEGRATION" and select Ansible from the dropdown menu. Fill in the following details:

- **NAME**
- **ANSIBLE GIT URL:** <https://github.com/ncelebic/morpheus-ansible-example>, or enter the URL for your own Ansible git repository
- **PLAYBOOKS PATH**
- **ROLES PATH**
- Mark the box to "USE MORPHEUS AGENT COMMAND BUS"

Note: If your git repository requires authentication, you should create a keypair and use the following URL format: [git@github.com:ncelebic/morpheus-ansible-example.git](https://github.com:ncelebic/morpheus-ansible-example.git).



The screenshot shows a form titled "NEW ANSIBLE INTEGRATION" with a close button (X) in the top right corner. The form contains the following fields and options:

- NAME:** A text input field containing "NewAnsibleIntegration", highlighted with a red box.
- ENABLED:** A checked checkbox.
- ANSIBLE GIT URL:** A text input field containing "https://github.com/ncelebic/morpheus-ansible-example", highlighted with a red box.
- KEY PAIR:** A dropdown menu.
- PLAYBOOKS PATH:** A text input field containing "/", highlighted with a red box.
- ROLES PATH:** A text input field containing "/roles", highlighted with a red box.
- GROUP VARIABLES PATH:** An empty text input field.
- HOST VARIABLES PATH:** An empty text input field.
- USE ANSIBLE GALAXY:** An unchecked checkbox.
- ENABLE VERBOSE LOGGING:** An unchecked checkbox.
- USE MORPHEUS AGENT COMMAND BUS:** A checked checkbox, highlighted with a red box.

A red arrow points down from the "USE MORPHEUS AGENT COMMAND BUS" checkbox to a blue "SAVE CHANGES" button at the bottom right of the form.

Click "SAVE CHANGES". You'll now see our new Ansible integration listed among any other configured integrations.

tions. If we click on this new integration to view detail, a green checkmark icon indicates the git repository has been fully synced.

With the Ansible integration set up, we can now create a task that includes our playbook. Go to Library > Automation, click “+ADD”. We’ll first set our “TYPE” value to Ansible Playbook so that the correct set of fields appear in the “NEW TASK” wizard. Set the following options:

- **NAME**
- **ANSIBLE REPO:** Here we will choose the Ansible integration that we just created
- **PLAYBOOK:** In our example case, enter ‘playbook.yml’

Click “SAVE CHANGES” to save our new task. We can test the new task on our Apache VM now by going to Provisioning > Instances and clicking into our VM. From the “ACTIONS” menu select “Run Task”. From the “TASK” dropdown menu, select the task we just added and click “EXECUTE”.

To see the progress of the task, click on the “HISTORY” tab and click on the (i) button to the right of each entry in the list. In this case, we can also see the results of the task by clicking on the link in the “LOCATION” column of the “VMS” section.

Now that our task is created, we can put it into a workflow. Back in Library > Automation we will click on the “WORKFLOWS” tab. Click “+ADD” and select Provisioning Workflow. We’ll give the new workflow a name and expand the Post Provision section. As we begin to type in the name of the task we’ve created, it should appear as a selection. Click “SAVE CHANGES”.

NEW WORKFLOW ✕

NAME

DESCRIPTION

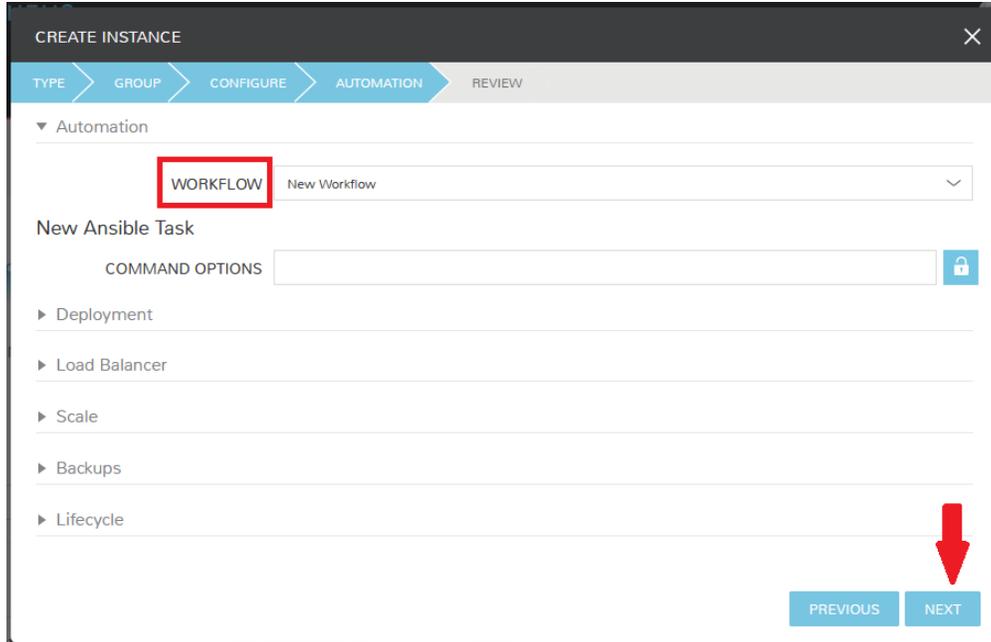
PLATFORM ▾

Tasks

- ▶ Pre Provision
- ▼ Provision
-
- ▼ Post Provision
-
- ▶ **New Ansible Task**
- ▶ Stop Service
- ▶ Pre Deploy
- ▶ Deploy
- ▶ Reconfigure
- ▶ Teardown

SAVE CHANGES

Now that we have a workflow, return to Provisioning > Instances and begin to provision another Apache instance. More detailed instructions on provisioning a new Apache instance are included earlier in this guide if needed. Now, when you reach the “AUTOMATION” section of the “CREATE INSTANCE” wizard, we have a workflow to select. From the “WORKFLOW” dropdown menu, select the workflow we just created and complete provisioning of the new instance.



As the instance is provisioning, we can go to the “HISTORY” tab and see Morpheus executing the tasks that were contained in our workflow.

This is just one example of using Morpheus to automate the process of configuring and instance to your needs. There are a number of other automation types that can be built into your workflows as well. For further information, take a look at the [automation integrations](#) guide in Morpheus docs.

Conclusion

At this point you should be up and running in Morpheus, ready to consume VMware. This guide only scratches the surface, there is a lot more to see and do in Morpheus. Take a look at the rest of [Morpheus Docs](#) for more information on supported integrations and other things possible.

5.13.3 Getting started with Morpheus and Azure

Introduction

This guide is designed to help you get started and quickly get the most out of Morpheus with Microsoft Azure public cloud. By the end, you will integrate your first cloud with Morpheus, configure networking, prepare and consume images, provision instances, and get started with automation. We will briefly discuss installation and account setup but will provide links to additional resources for those very first steps. For the most part, this guide assumes you are able to get Morpheus installed and are ready to move forward from that point. There is a lot more to see and do in Morpheus that is beyond the scope of this guide. For more, consult the complete Morpheus documentation or take part in our [Reddit user community forum](#).

Installation & Setup

In the simplest configuration, Morpheus needs one appliance server which will contain all the components necessary to orchestrate virtual machines and containers. Full requirements, including storage and networking considerations, can be found in Morpheus documentation [here](#). In order to provision any new Instances, hosts, or applications (or convert any discovered resources to managed resources) you will need a valid license. If you don't have one, you can request a community edition license for free at [Morpheus Hub](#). Once obtained, the license can be applied in Administration > Settings > License. For more, take a look at our community edition [welcome package](#).

Groups

Groups in Morpheus define which resources a user has access to. Clouds are added to Groups and a user can only access Clouds that are in the Groups to which their roles give them access. More information on Morpheus Groups is [here](#). A deep dive into Groups goes beyond the scope of this guide but it's often useful to create a Group that contains all Clouds for testing purposes. We will create that group now so that we can add our first Cloud into this Group in the next section.

Navigate to *Infrastructure > Groups*. Here we will see a list of all configured groups but, of course, this will be empty immediately after installation. Click "+CREATE". Give your group a name, such as "All Clouds". The "CODE" field is used when calling Morpheus through Morpheus API or Morpheus CLI. It's useful in most cases to have an "All Clouds" group for testing purposes so this will likely help you down the road.

Click *SAVE CHANGES*. Your Group is now ready to accept Clouds.

NEW GROUP ✕

Configuration

NAME

CODE

LOCATION

▶ Advanced Options

SAVE CHANGES

Integrating Your First Cloud

Clouds in Morpheus consist of any consumable endpoint whether that be on-prem, public clouds, or even bare metal. In this guide, we will focus on integrating and working with Microsoft Azure public cloud.

To get started, we will navigate to *Infrastructure > Clouds*. This is the Cloud list page which lists all configured Clouds. It will be empty if you've just completed installation and setup of Morpheus but soon we will see our integrated Azure cloud here.

Click the "+ADD" button to pop the "CREATE CLOUD" wizard. Select "AZURE (PUBLIC)" and click the "NEXT" button.

On the "CONFIGURE" tab, we're asked to provide Azure-specific details to connect to the cloud. Morpheus Azure integration requires Owner or Contributor access to subscription via App Registration. Adding an Azure Cloud or Clouds to Morpheus will require the following:

- Azure Subscription ID
- Directory (tenant) ID
- Application (client) ID
- Application (client) Secret
- Application (client) must be Owner or Contributor of Subscription

CSP Accounts require the additional following input:

- CSP Directory (tenant) ID
- CSP Application (client) ID
- CSP Application (client) SECRET

CREATE CLOUD ✕

CLOUD > **CONFIGURE** > GROUP > REVIEW

NAME

CODE

LOCATION

VISIBILITY ▾

TENANT ▾

ENABLED

AUTOMATICALLY POWER ON VMS

Details

CLOUD TYPE ▾

SUBSCRIPTION ID

TENANT ID

CLIENT ID

CLIENT SECRET

LOCATION ▾

RESOURCE GROUP ▾

INVENTORY EXISTING INSTANCES

INVENTORY LEVEL ▾

ACCOUNT TYPE ▾

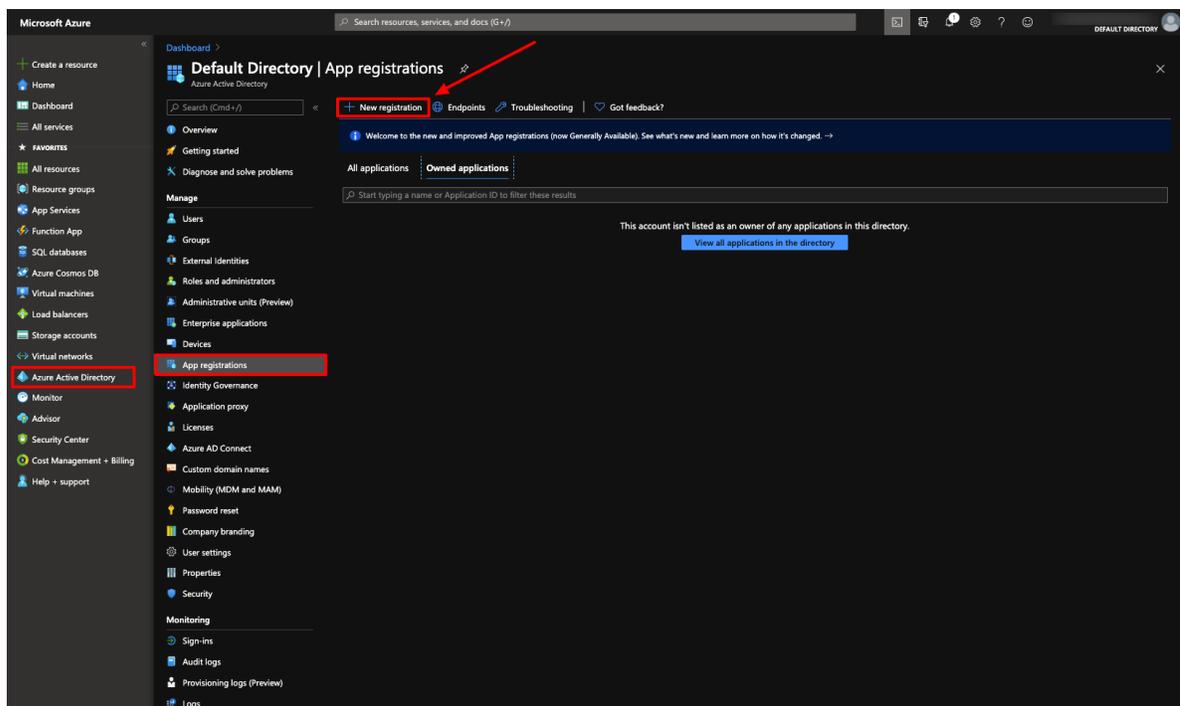
Create App Registration

Morpheus authenticates with Azure via an App Registration with an Owner or Contributor Role on a Subscription. Use the steps below to create and collect the required credentials and assign the required permissions to integrate Azure with Morpheus.

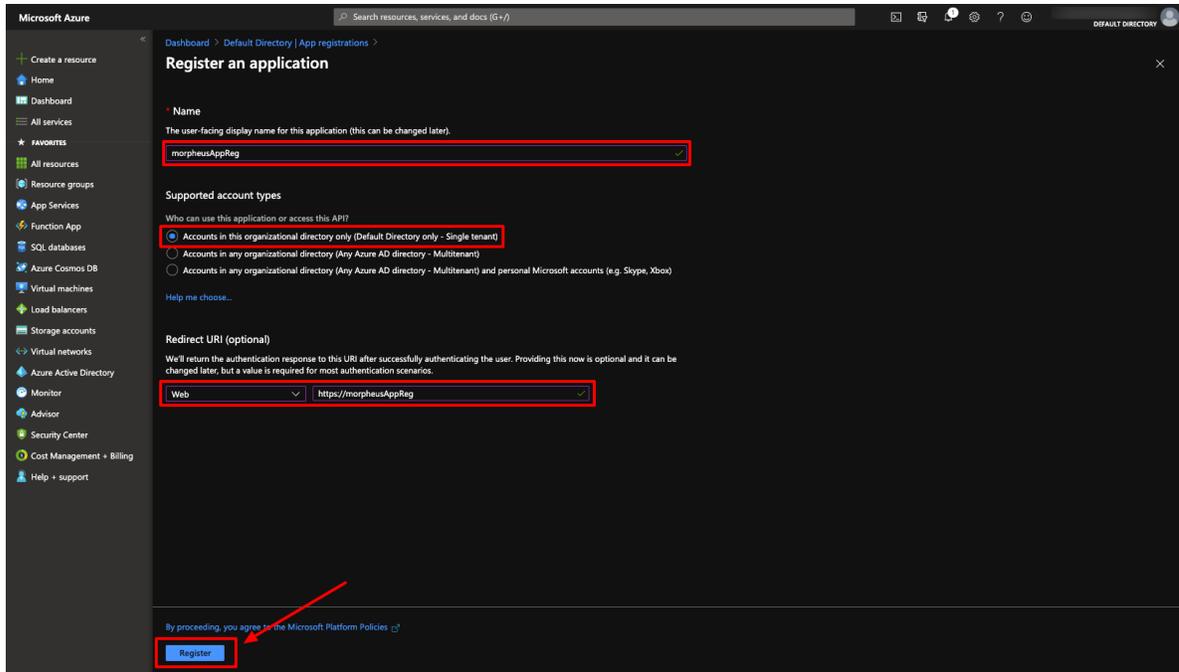
Warning: Using an App Registration (service principal) that has selective resource permissions and is not an Owner or Contributor of the Subscription is not supported and will cause failures/issues. Please confirm the App Registration you use to integrate Azure with Morpheus has Owner or Contributor permissions on the specified Subscription.

If you do not have an existing Azure Active Directory App Registration, or you wish to use a new one for Morpheus, you will need to create one using the steps below. If you already have one you wish to use, continue to the next section.

1. Log into the Azure portal
2. Select “Azure Active Directory”
3. Select “App Registrations”
4. Select “New Registration”



5. Next, give the app a name, specify Web app / API for the type (default) and enter any URL for the Sign-on URL:
6. Click Create and your new App Registration will be created.

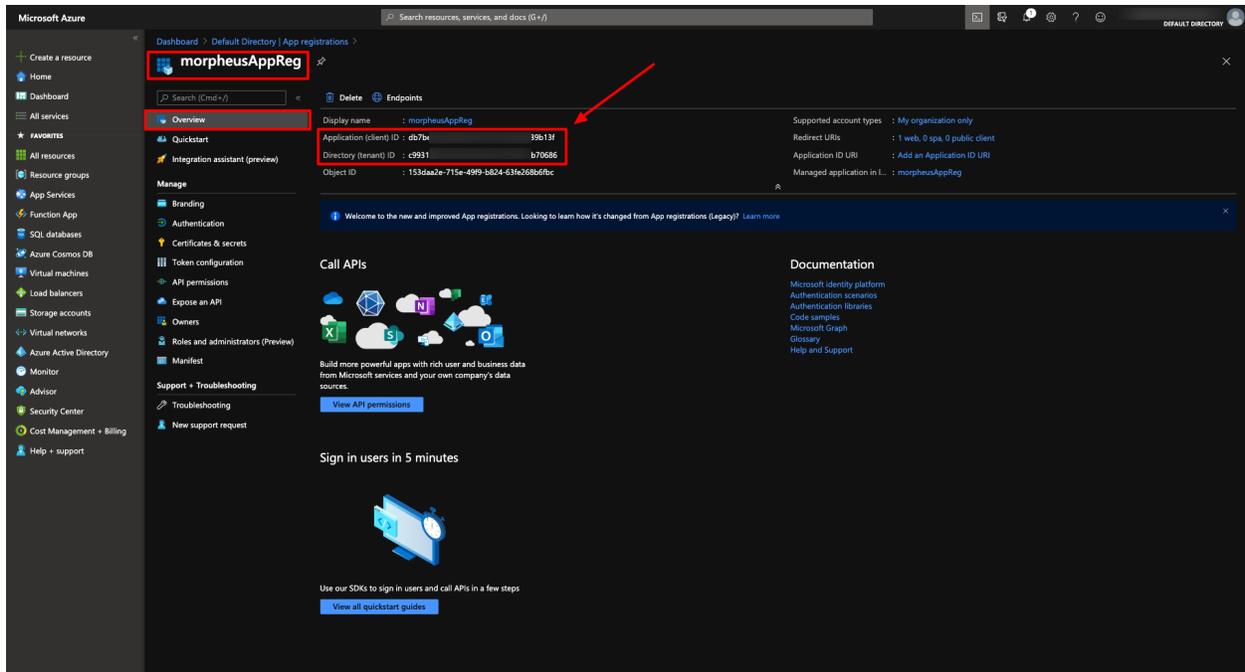


Now that we have our App Registration, we will gather the credentials required for the Morpheus Azure integration in the next section.

Copy Directory (tenant) and Application (client) IDs

The App Registration Directory (tenant) and Application (client) ID are required for the Morpheus Azure integration. Both can be found in the overview section of the App Registration.

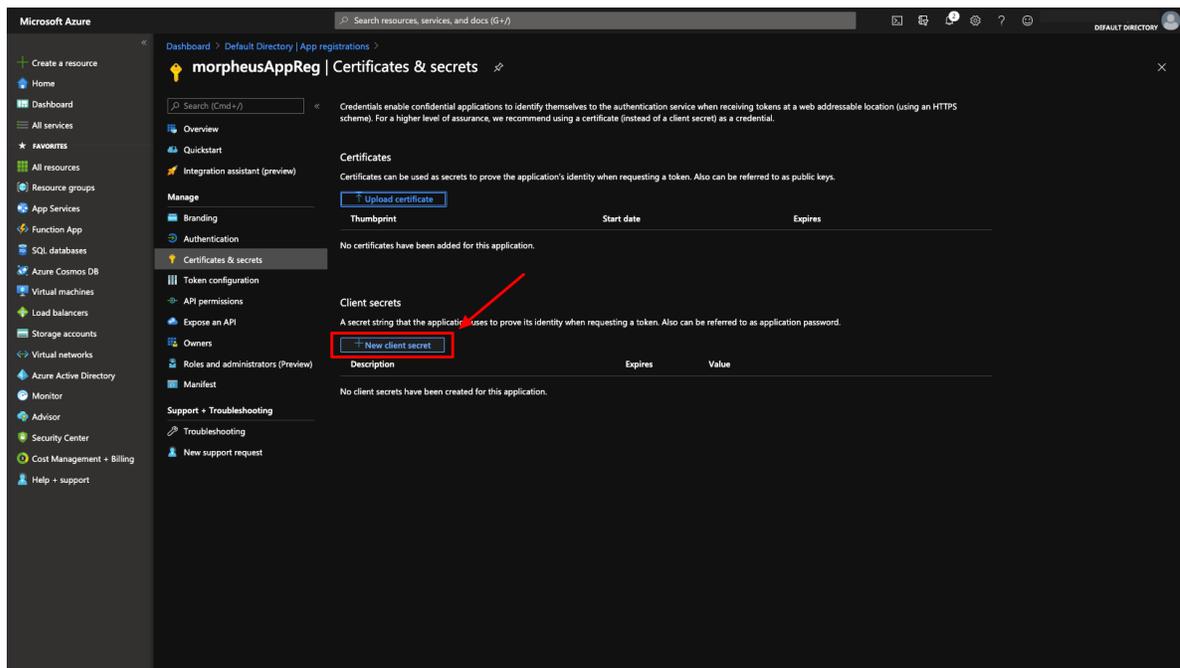
1. Go to the Overview section of your App Registration
2. Copy the Directory (tenant) ID
3. Store/Paste for use as the Tenant ID when adding your Azure cloud in Morpheus
4. Copy the Application (client) ID
5. Store/Paste for use as the Client ID when adding your Azure cloud in Morpheus



Generate a Client Secret

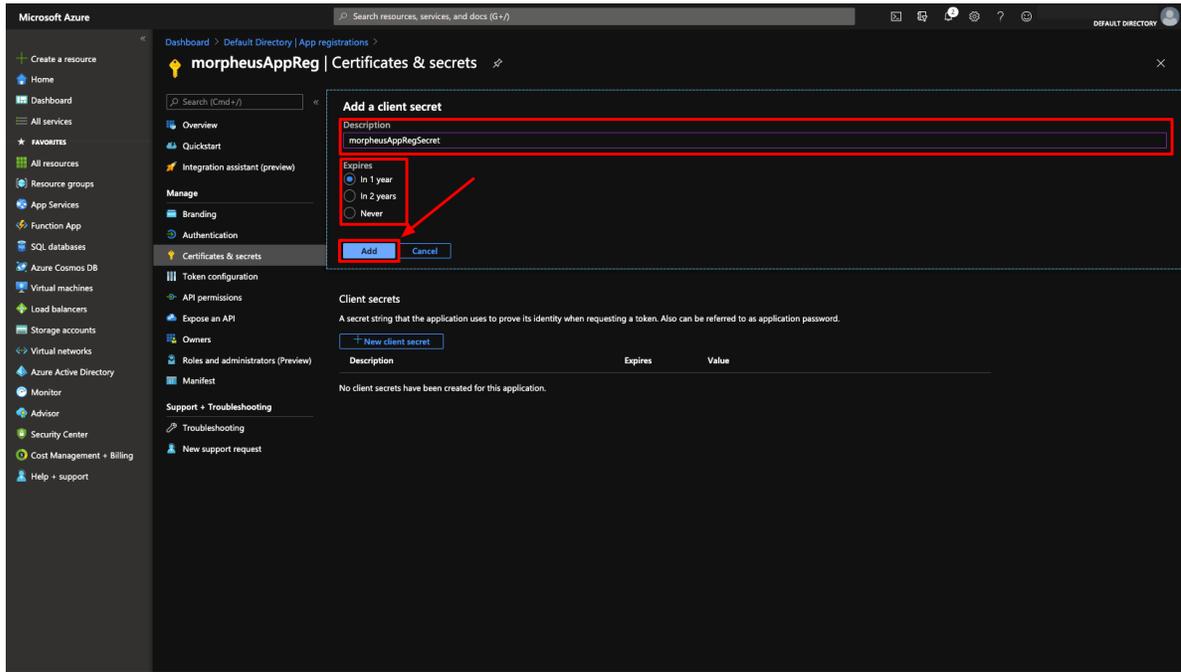
While still in your App Registration:

1. Select “Certificates & secrets” in the Manage section
2. Select + New client secret



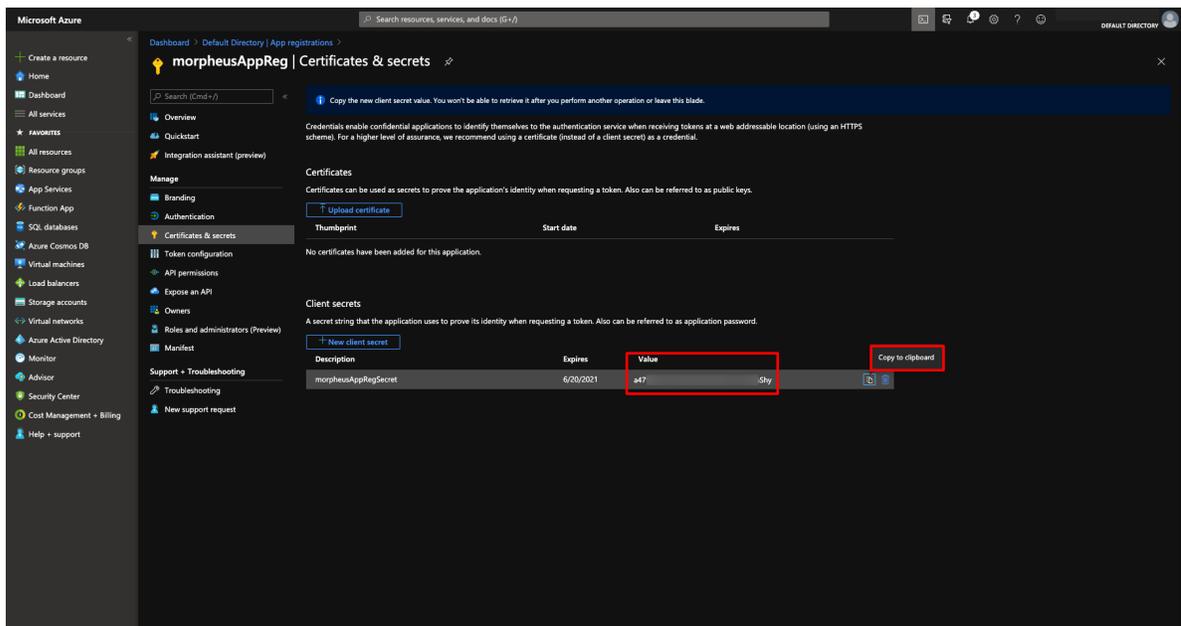
3. The “Add a client secret” modal will come up
4. Add a description to help identify the secret in the future

- 5. Select an expiration duration
- 6. Click *Add*



- 7. Copy the newly-generated client secret value.

Important: Copy the client secret value before continuing as it will not be viewable again later.



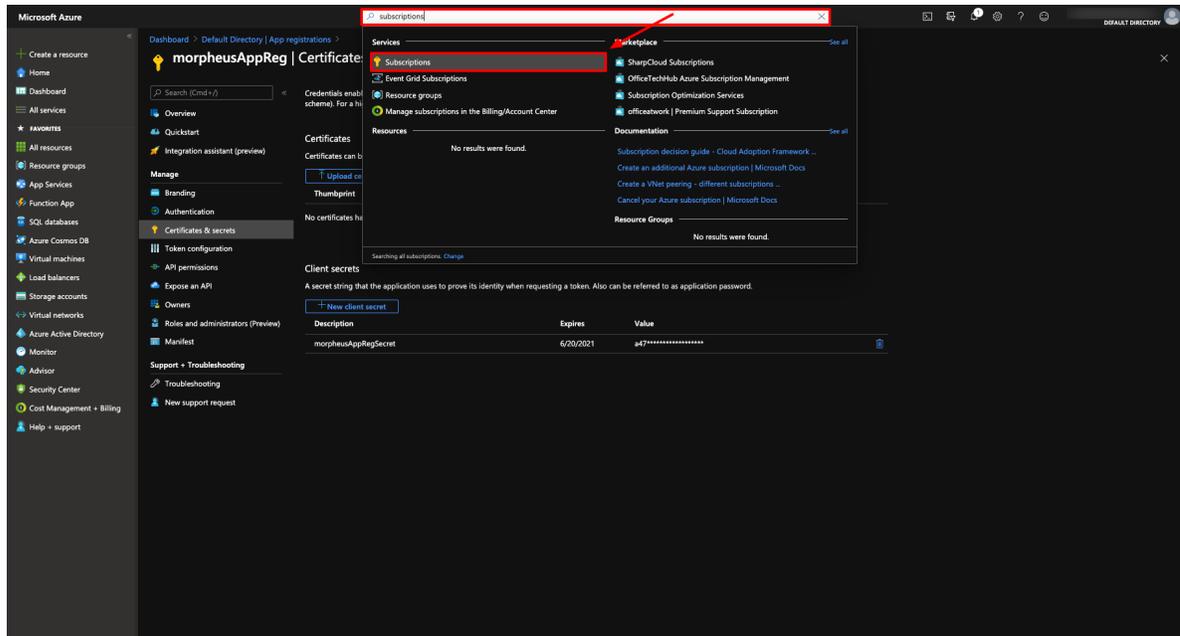
- 8. Store/Paste client secret for use later when adding your Azure cloud in Morpheus

You now have three of the four credentials required for Morpheus Azure cloud integration. The last credential required is the Azure Subscription ID which we will gather in the next section.

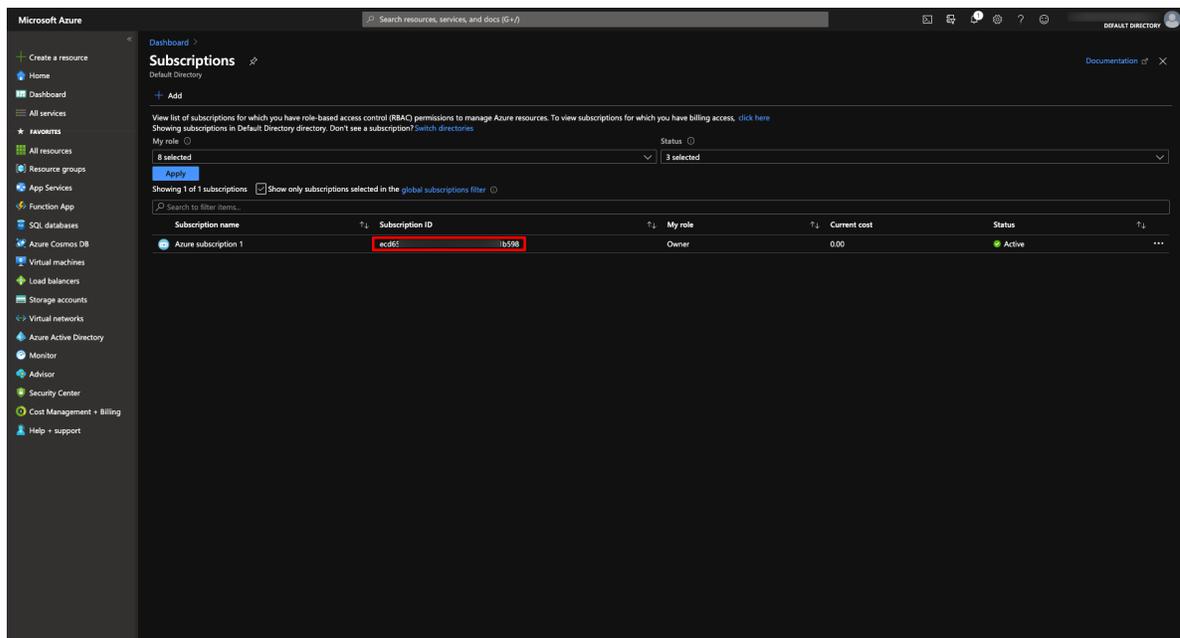
Subscription ID

To get the Azure Subscription ID:

1. Navigate to the Subscriptions section. The search function can help to locate these sections if they aren't immediately apparent in the UI menu



2. In the Subscriptions section, copy the Subscription ID

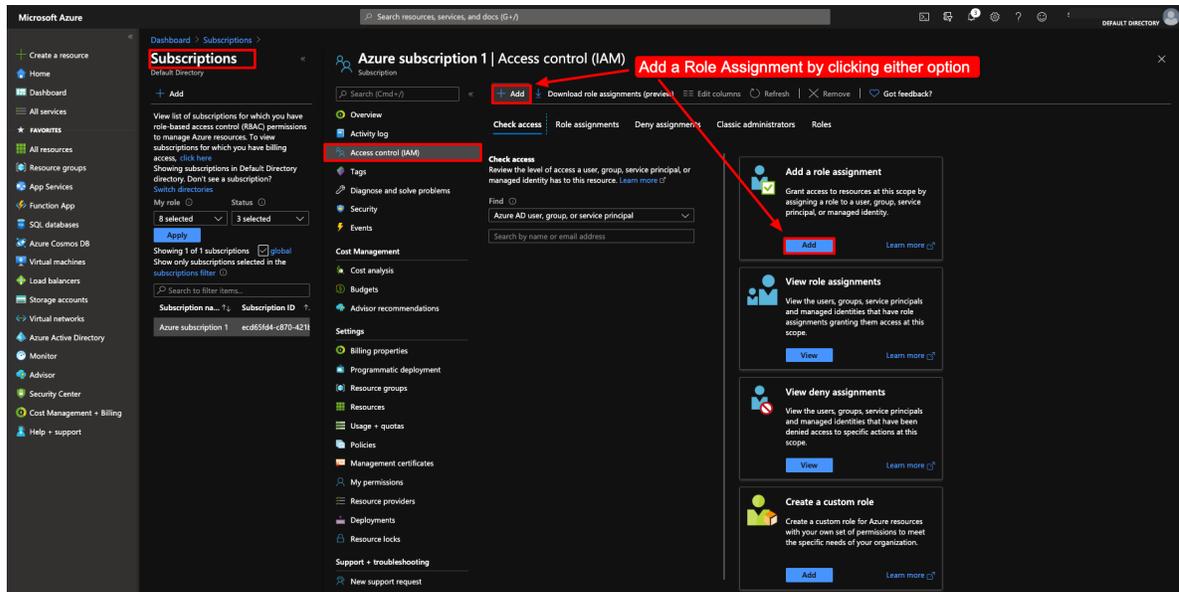


3. Store/Paste for use as the Subscription ID when adding your Azure cloud in Morpheus

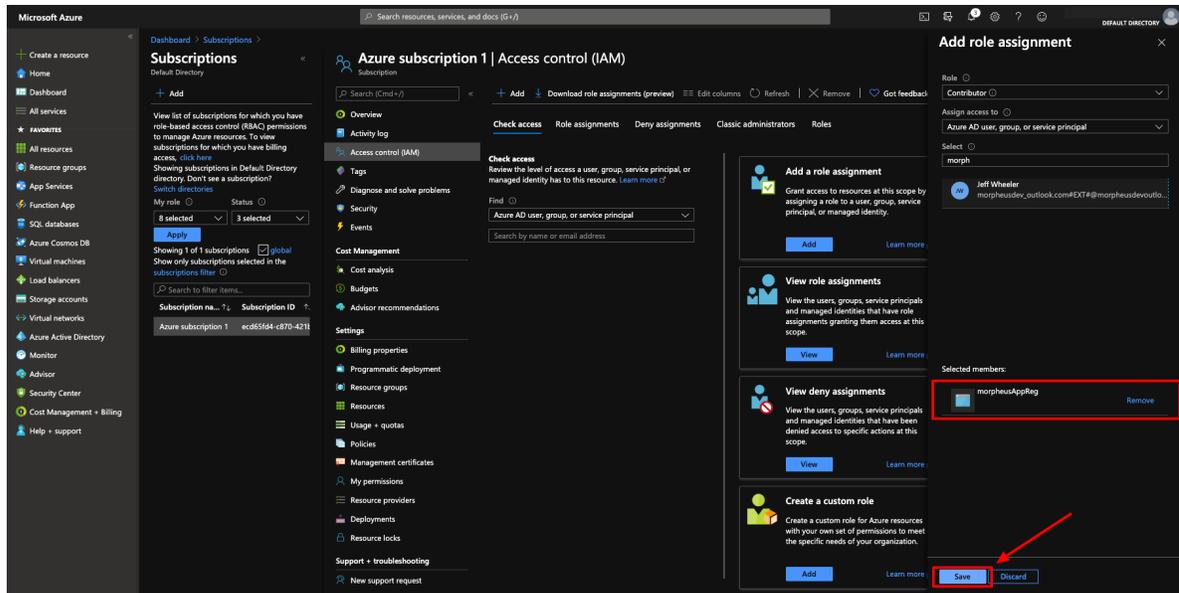
Make App Registration owner or contributor of Subscription

The App Registration used needs to be an owner of the Azure Subscription used for the Morpheus cloud integration. If lesser permissions are given or permissions are assigned at individual resource levels, Morpheus will not be able to properly inventory existing cloud resources, create resources or remove them.

1. In the Subscriptions section in Azure, select the Subscription
2. In the Subscription pane, select “Access Control (IAM)”
3. Either Click :guilabel`+ Add`, and then “Add Role Assignment” OR simply select “Add a role assignment”



4. In the right pane, select “Owner” or “Contributor” Role type
5. Search for the name of the App Registration used for the Morpheus integration
6. Select the App Registration in the search results
7. Select “Save”



You now have the required credentials and permissions to add an Azure Cloud integration into Morpheus. Continue on with the next sections of this guide to complete the integration from the Morpheus side.

Complete the Add Cloud Process in Morpheus

If you've followed this guide from the start, you will already have a Cloud integration modal open in Morpheus UI. If you still need to open that wizard, navigate to Infrastructure > Clouds > + ADD > Azure (Public) and click *NEXT*. Fill in the following fields with the information gathered in the steps above:

- Subscription ID
- Tenant ID
- Client ID
- Client Secret
- Location
- Resource Group
- Inventory Existing Instances
- Inventory Level
- Account Type

Once valid credentials are populated in the appropriate fields, the LOCATION dropdown menu will be populated. Select an available region, this is also a helpful check to ensure you've correctly provided working credentials. In addition, we can scope the cloud integration to all resource groups in the region (All) or can select a specific resource group to limit Morpheus resource inventorying and creation to just that resource group.

By checking INVENTORY EXISTING INSTANCES, Morpheus will automatically onboard existing cloud resources which are scoped to the region and resource group indicated. If this box is checked, we will also need to select either basic inventorying, which syncs name, IP addresses, platform types, power status, and sizing data (storage, CPU, and RAM) OR full (API heavy) inventorying which syncs resource utilization metrics (storage, CPU, and RAM) when available in addition to what we get with basic inventorying.

To move on, expand the "Advanced Options" section.

Note: CSP accounts will also need to enter CSP TENANT ID, CSP CLIENT ID, and CSP CLIENT SECRET in the Advanced Options section.

Within the “Advanced Options” drawer are additional configurations to consider for your first Cloud. Some of these won’t usable until they reference additional configured integrations. Common settings to consider are **DOMAIN**, **STORAGE TYPE**, **APPLIANCE URL** (overrides the Morpheus URL for external systems), **GUIDANCE** (setting “Manual” will make recommendations for rightsizing), **COSTING**, **DNS INTEGRATION**, **CMDB**, and **AGENT INSTALL MODE**.

Once you’re satisfied with your selections, click “NEXT”

We have now arrived at the “GROUP” tab. In this case, we will mark the radio button to “USE EXISTING” Groups if you wish to use the Group we configured earlier. Alternatively, you can create a new one here.

The screenshot shows the 'CREATE CLOUD' wizard interface. At the top, there is a dark blue header with the text 'CREATE CLOUD' and a close button (X). Below the header is a progress bar with four steps: 'CLOUD', 'CONFIGURE', 'GROUP', and 'REVIEW'. The 'GROUP' step is currently active and highlighted in a darker blue. Below the progress bar, there are two radio buttons: 'USE EXISTING' (which is selected) and 'CREATE NEW'. Underneath the radio buttons is a dropdown menu labeled 'GROUP' with the text 'All Clouds' and a downward arrow. At the bottom right of the form, there are two buttons: 'PREVIOUS' and 'NEXT'.

Once you’ve selected or created the Group, click “NEXT”

On the final tab of the “CREATE CLOUD” wizard, you’ll confirm your selections and click “COMPLETE”. The new Cloud is now listed on the Cloud list page. After a short time, Morpheus will provide summary information and statistics on existing virtual machines, networks, and other resources available in the Cloud.

Viewing Cloud Inventory

Now that we've integrated our first Azure cloud, we can stop for a moment to review what Morpheus gives us from the Cloud detail page. We can see that Morpheus gives us estimated costs and cost histories, metrics on used resources, and also lists out resource counts in various categories including container hosts, hypervisors, and virtual machines. We can drill into these categories to see lists of resources in the various categories by clicking on the category tabs. We can link to the detail page for any specific resource by clicking on it from its resource category list.

Configuring Resource Pools

With our Azure Cloud configured, Morpheus will automatically sync in available resource pools and data stores.

For resource pools, once Morpheus has had time to ingest them, then will be visible from the cloud detail page. Navigate to *Infrastructure > Clouds > (your Azure cloud) > Resources tab*. In here, we are able to see and control access to the various resource pools that have been configured in Azure. For example, we can restrict access to a specific resource pool within Morpheus completely by clicking on the "ACTIONS" button, then clicking "Edit". If we unmark the "ACTIVE" button and then click "SAVE CHANGES" we will see that the resource pool is now grayed out in the list. The resources contained in that pool will not be accessible for provisioning within Morpheus if it is not configured as active.

The screenshot displays the Morpheus web interface. At the top, the navigation bar includes the Morpheus logo, a search bar, a notification bell, a 'Support' dropdown, and a user profile icon. Below this is a secondary navigation bar with tabs for Operations, Provisioning, Infrastructure (selected), Backups, Logs, Monitoring, Tools, and Administration. A third navigation bar contains icons for Groups, Clouds, Clusters, Hosts, Network, Load Balancers, Storage, Keys & Certs, and Boot.

The main content area shows the 'Clouds' section for 'AdamAzure'. A green checkmark indicates the cloud is active. There are three buttons: 'EDIT', 'REFRESH', and 'DELETE'. Below these, it shows 'Last Sync: 07/28/2020 02:39 PM' and 'Sync Duration: 6 seconds'.

Five performance metrics are displayed as circular gauges:

- COST THIS MONTH: \$0
- AVG MONTHLY COST: 0.0 (with a line graph showing values for Feb, Mar, Apr, May, Jun, Jul)
- MAX CPU: 0%
- MEMORY: 0%
- STORAGE: 0%

Resource counts are shown below the metrics:

- CONTAINER HOSTS: 0
- HYPERVISORS: 0
- BARE METAL: 0
- VIRTUAL MACHINES: 1
- DISCOVERED: 1

A secondary navigation bar at the bottom of the main content area includes tabs for Summary, Clusters, Hosts, VMs, Containers, Load Balancers, Networks, Data Stores, Resources (selected), Policies, and Wiki.

The 'RESOURCES' section is titled 'POOLS' and features a search bar and a '+ ADD RESOURCE POOL' button. Below is a table with the following columns: NAME, DESCRIPTION, VISIBILITY, DEFAULT, TENANT, and ACTIONS.

NAME	DESCRIPTION	VISIBILITY	DEFAULT	TENANT	ACTIONS
		Private		morpheus	ACTIONS ▾

Often our clients will want to make specific blocks of resources available to their own customers. This can be easily and conveniently controlled through the same "EDIT RESOURCE POOL" dialog box we were just working in. If we

expand the “Group Access” drawer, we are able to give or remove access to each pool to any Group we’d like. We can also choose to make some or all of our resource pools available to every Group. Specific resource pools can also be defined as the default for each Group when needed.

EDIT RESOURCE POOL ✕

NAME

MOVE SERVERS

ACTIVE

DEFAULT

▼ Group Access

GROUP	ACCESS	DEFAULT
all	<input checked="" type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>

Additionally, we may choose to allow only certain service plans to be provisioned into a specific pool of resources. For example, perhaps a specific cluster is my SQL cluster and only specific services plans should be consumable within it. We can control that through this same dialog box.

Configuring Data Stores

To take a look at data stores, we’ll move from the “Resources” tab to the “Data Stores” tab on our Cloud detail page. Morpheus gives the user similar control with data stores to what we saw with our resources pools earlier. Just like with resource pools, we can disable access within Morpheus completely by clicking on “ACTIONS” and then “Edit”. If we unmark the “ACTIVE” checkbox and click “SAVE CHANGES”, you will see that specific data store has been grayed out.

Operations Provisioning **Infrastructure** Backups Logs Monitoring Tools Administration

Groups Clouds Clusters Hosts Network Load Balancers Storage Keys & Certs Boot

Clouds > AdamAzure

EDIT REFRESH DELETE

Last Sync: 07/28/2020 02:44 PM Sync Duration: 33 seconds

\$0 COST THIS MONTH

0.0 AVG MONTHLY COST

0% MAX CPU

0% MEMORY

0% STORAGE

0 CONTAINER HOSTS

0 HYPERVISORS

0 BARE METAL

1 VIRTUAL MACHINES

1 DISCOVERED

Summary Clusters Hosts VMs Containers Load Balancers Networks **Data Stores** Resources Policies Wiki

Search

ACTIONS

NAME	TYPE	CAPACITY	ONLINE	VISIBILITY	TENANT	ACTIONS
	Generic	Unknown	Yes	Private	morpheus	ACTIONS

Just like with resource pools, we are also able to scope data stores to specific Groups. This ensures that the members of each Group are only able to consume the data stores they should have access to.

Configuring Network for Provisioning

When configuring networking, we can set global defaults by going to *Infrastructure > Network > NETWORKS* tab. Here we can add or configure networks from all Clouds integrated into Morpheus. Depending on the number of clouds Morpheus has ingested, this list may be quite large and may also be paginated across a large number of pages. In such a case, it may be easier to view or configure networks from the specific Cloud detail page so that networks from other Clouds are not shown.

Still in *Infrastructure > Network*, make note of the “INTEGRATIONS” tab. It’s here that we can set up any integrations that may be relevant, such as IPAM integrations. Generally speaking, when adding IPAM integrations, we simply need to name our new integration, give the API URL, and provide credentials. There’s more information in the [IPAM integration](#) section of Morpheus Docs.

In *Infrastructure > Networking* we can also set up IP address pools from the IP Pools tab. These pools can be manually defined, known as a Morpheus-type IP pool, or they can come from any IPAM integrations you’ve configured. As instances are provisioned, Morpheus will assign IP addresses from the pool chosen during provisioning. When the instance is later dissolved, Morpheus will automatically release the IP address to be used by another instance when needed. When adding or editing a network, we can opt to scope the network to one of these configured IP address pools. Edit an existing network by clicking the pencil icon on the Networks List Page (*Infrastructure > Networks > Networks* Tab) and fill in the “Network Pool” field to associate the IP Pool with the network.

Since this guide is focused on working within an Azure cloud that we integrated at the start, we will take a look at our

network configurations on the cloud detail page as well. Navigate to *Infrastructure > Clouds > (your Azure cloud) > NETWORKS tab*. Just as with resource pools and data stores, we have the ability to make certain networks inactive in Morpheus, or scope them to be usable only for certain Groups or Tenants.

The screenshot shows the Morpheus web interface. At the top, there is a navigation bar with the Morpheus logo and a search bar. Below the navigation bar, there are tabs for 'Operations', 'Provisioning', 'Infrastructure', 'Backups', 'Logs', 'Monitoring', 'Tools', and 'Administration'. Under 'Infrastructure', there are sub-tabs for 'Groups', 'Clouds', 'Clusters', 'Hosts', 'Network', 'Load Balancers', 'Storage', 'Keys & Certs', and 'Boot'. The 'Clouds' tab is selected, and the 'AdamAzure' cloud is chosen. The dashboard shows several metrics: 'COST THIS MONTH' at \$0, 'AVG MONTHLY COST' at 0.0, 'MAX CPU' at 0%, 'MEMORY' at 0%, and 'STORAGE' at 0%. Below these are counts for 'CONTAINER HOSTS' (0), 'HYPERVISORS' (0), 'BARE METAL' (0), 'VIRTUAL MACHINES' (1), and 'DISCOVERED' (1). The 'Networks' tab is selected in the sub-navigation bar. The 'NETWORKS' section has a search bar and an 'ACTIONS' dropdown. A table lists two networks:

NAME	TYPE	CIDR	POOL	DHCP VISIBILITY	TENANT	ACTIONS
[redacted]	Azure Network	[redacted]		✓ Private	morpheus, [redacted]	ACTIONS ▾
default	Subnet	[redacted]		✓ Private	morpheus	ACTIONS ▾

Below the table is the 'HOST SECURITY GROUPS' section, which includes a warning message: 'The Host Level Firewall is not currently enabled. These Security Groups will not be applied unless this setting is turned on in the cloud settings host firewall.' and an 'EDIT SECURITY GROUPS' button. At the bottom, there is a table with columns for 'NAME' and 'DESCRIPTION'.

Provisioning Your First Instance

At this point, the groundwork is laid and we are ready to attempt our first new provisioning. As a first Instance, we'll provision an Apache web server to our Azure cloud. Morpheus includes a very robust catalog of pre-configured Instance types. We'll use one of these included catalog items for this guide but you'll likely also need to prep your own custom images and Instance types to make available to your users. Much more on this can be found elsewhere in Morpheus documentation.

Navigate to `|ProIns|`. If any Instances are currently provisioned, we will see them listed here. To start a new Instance we click + *ADD* to open the "CREATE INSTANCE" wizard. We'll scroll down to and select the Apache instance type and click "NEXT".

The screenshot shows the 'CREATE INSTANCE' wizard interface. At the top, there's a dark header with 'CREATE INSTANCE' and a close button. Below it, a navigation bar shows steps: TYPE (selected), GROUP, CONFIGURE, AUTOMATION, and REVIEW. A search bar is present with the text 'as if it were any other instance type.' To the right of the search bar are several icons representing different technologies. A 'TECHNOLOGY' dropdown menu is also visible. The main area displays a list of instance types. The 'APACHE' instance type is highlighted in a light blue box. It features the Apache logo and the text: 'APACHE HTTP SERVER' and 'An open-source modern web server common on many websites today. A secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards.' Other instance types like 'AMAZON LINUX 2' are also visible. At the bottom right, there are 'PREVIOUS' and 'NEXT' buttons.

First, we'll specify the Group to provision into which determines the Clouds available. If you've followed this guide to this point, you should at least have a Group that houses all of your Clouds which you can select here. This will allow us to select the Azure cloud from the "CLOUD" dropdown menu. Provide a unique name to this instance and then click "NEXT"

From the "CONFIGURE" tab, we're presented with a number of options. The options are cloud and layout-specific, more generalized information on creating Instances and available options is [here](#). For our purposes, we'll select the following options:

- **LAYOUT:** Includes options such as the base OS, custom layouts will also be here when available
- **PLAN:** Select the resource plan for your instance. Some plans have minimum resource limits, Morpheus will only show plans at or above these limits. User-defined plans can also be created in `|AdmPla|`.
- **VOLUMES:** The minimum disk space is set by the plan, this value may be locked if you've selected a custom plan that defines the volume size

- **NETWORKS:** Select a network

Under the “User Config” drawer, mark the box to “CREATE YOUR USER”. Click *NEXT*.

CREATE INSTANCE ✕

TYPE GROUP CONFIGURE AUTOMATION REVIEW

Configuration Options

LAYOUT	Azure Apache			▼
PLAN	Standard_B2ms (2 Core, 8GB Memory) (westus)			▼
	Cores: 2 Memory: 8 GB Price: \$73.0816 / Month			💰
RESOURCE POOL				▼
VOLUMES	root	Page Blob	morpheusfuorhvcynyobjtro	+
NETWORKS		DHCP		+
SECURITY GROUP	Select			▼
AVAILABILITY SET	No Availability Set			▼

ASSIGN PUBLIC IP

▶ User Config

▶ Advanced Options

PREVIOUS NEXT

Note: “CREATE YOUR USER” will seed a user account into the VM with credentials set in your Morpheus user account settings. If you’ve not yet defined these credentials, you can do so by clicking on your username in the upper-right corner of the application window and selecting “USER SETTINGS”.

For now, we’ll simply click *NEXT* to move through the “AUTOMATION” tab but feel free to stop and take a look at the available selections here. There is more information later in this guide on automation and even more beyond that in the rest of Morpheus docs.

Review the settings for your first instance and click *COMPLETE*.

We are now dropped back onto the Instances list page. We can see a new entry in the list at this point with a status indicator that the new machine is being launched (rocket icon in the status field). We can double click on the Instance in the list to move to the Instance detail page. For now we will see a progress bar indicating that the Instance is being created and is starting up. The exact amount of time this process will take depends on selections made when provisioning the Instance. Initially, Morpheus will guess as to how long this will take and the progress bar may not be accurate. Over time, Morpheus will learn how long these processes take and progress bar accuracy will improve. For more detailed

information on the status of various provisioning processes, we can scroll down and select the “HISTORY” tab. The “STATUS” icon will change from the blue rocket to a green play button when the Instance is fully ready. Furthermore, we can click on the hyperlinked IP address in the “VMS” section of this page to view a default page in a web browser to confirm success.

Creating Your First Library Item

In the prior section, we manually provisioned our first Instance. However, Morpheus allows you to build a catalog of custom provisionable items to simplify and speed provisioning in the future. In this section, we’ll build a catalog item and show how that can translate into quick Instance provisioning after configuration.

Note: Before starting this process, it’s important to decide which virtual image you plan to use. If you’re not using a Morpheus-provided image, you’ll want to ensure it’s configured. You will not be able to complete this section without selecting an available image. In this example we will use a CentOS image that was previously configured in the Morpheus library. If you need to configure your own images prior to starting this section, navigate to Library > Virtual Images and click + *ADD*. A deeper dive into image prep and virtual image configuration goes beyond the scope of this guide.

Provisionable elements in Morpheus combine a Node Type(s), Layout(s), and an Instance Type. The [Overview section](#) of Morpheus docs discusses these objects and how they work together in greater detail. Our first step here will be to create a Node Type which wrap the image itself with additional configuration, templates, and scripts. While not strictly required, creating the Node Type, Instance Type, and then the Layout is often a good workflow for creating Library items. That is the order we will follow in this guide.

Navigate to Library > Blueprints > Node Types and click + *ADD*

In this example, I am going to set the following options in the “NEW NODE TYPE” wizard:

- **NAME:** *Example Azure CentOS 7*
- **SHORT NAME:** *eac7* (Identifies the Node Type in Morpheus API/CLI)
- **VERSION:** *7* (Ensures the correct Node Types are used when tying Layouts with multiple images to the same Instance Type)
- **TECHNOLOGY:** *Azure*
- **VM IMAGE:** *Azure-Centos-7*

Click *SAVE CHANGES*

EDIT NODE TYPE

NAME SHORT NAME

The short name is a name with no spaces used for display in your container list.

VERSION ENVIRONMENT
VARIABLES

Name

Value



Azure Options

VM IMAGE LOG FOLDER DEPLOY FOLDER

(Optional) If using deployment services, this mount point will be replaced with the contents of said deployments.

SERVICE PORTS

port

name

No LB



SCRIPTS

FILE TEMPLATES

▶ Advanced Options

SAVE CHANGES

With the new Node Type created, we'll now add a new Instance type which will be accessible from the provisioning wizard once created. Move from the "NODE TYPES" tab to the "INSTANCE TYPES" tab and click + *ADD*.

In the "NEW INSTANCE TYPE" wizard, I'll simply enter a **NAME** and **CODE** value. Click *SAVE CHANGES*. You could also provide a description, icon, and category for easier identification from the provisioning wizard later.

EDIT INSTANCE TYPE
✕

NAME

CODE
 Useful shortcode for provisioning naming schemes and export reference.

DESCRIPTION

CATEGORY

ICON Browse

Suggested Dimensions: 150 x 51

Now that we've created a new Instance type, access it by clicking on the name in the list of custom Instances you've created. In my case, I've given the name "*Example Azure CentOS 7*".

Once we've opened the new Instance type, by default, we should be on the "LAYOUTS" tab. Click + *ADD LAYOUT*. I've set the following fields on my example layout:

- **NAME:** *Example Azure CentOS 7*
- **VERSION:** 7 (This is the version number of the layout itself, which is labeled 7 in the example)
- **TECHNOLOGY:** Azure
- **Nodes:** Select the Node Type we created earlier, if desired you can specify multiple nodes

Click *SAVE CHANGES*.

EDIT LAYOUT ✕

NAME

VERSION

DESCRIPTION

CREATABLE

TECHNOLOGY ▼

MINIMUM MEMORY ▼

This will override any memory requirement set on the virtual image

WORKFLOW ▼

SUPPORTS CONVERT TO MANAGED

ENVIRONMENT VARIABLES

Name	Value
------	-------

⚙️ +

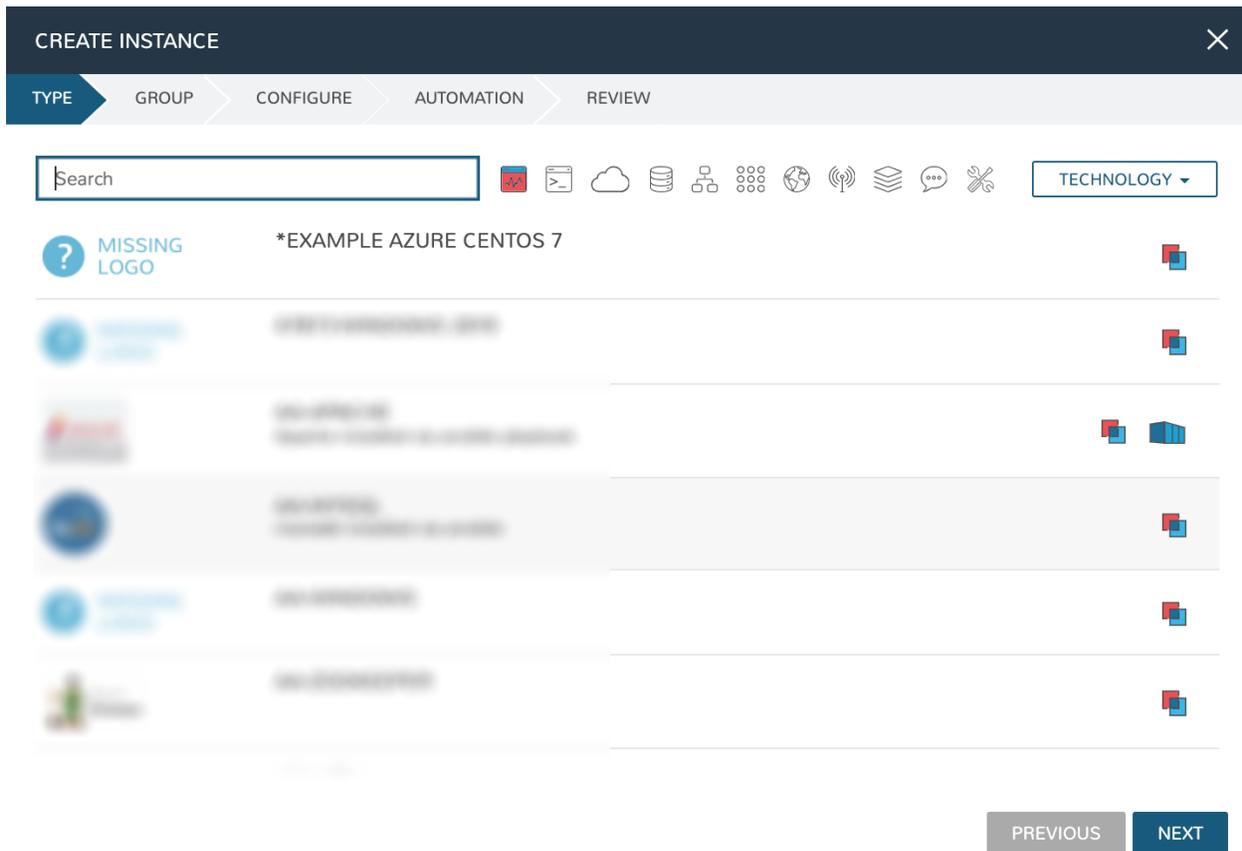
Option Types

Nodes

*Example Azure CentOS 7 (7) ✕

SAVE CHANGES

At this point we've completed the setup work and can now provision the Instance we've created to our specifications. Navigate to *[ProIns]* and click + *ADD*. From the search bar we can search for the new Instance type we've created.



As before, we can select a Group and Cloud to provision this new Instance. Click *NEXT*. On the “CONFIGURE” tab, make note that the layout and plan are already selected because they were configured as part of creating the new Instance type. Select a network and click *NEXT*. Once again we will also click *NEXT* through the “AUTOMATION” tab. Finally, click *COMPLETE*.

As before when we provisioned a pre-existing Instance from the default catalog, Morpheus will now begin to spin up the new VM. How long this will take depends on the configuration and environmental factors but Morpheus will predict how long this process will take and represent that on a progress bar. Over time, Morpheus begins to learn how long these processes take and becomes more accurate in predicting spin-up time.

Once the provisioning process has completed, open the Instance detail page in Morpheus and click on the “CONSOLE” tab. You’ll be logged in with your user account and are then able to confirm the machine is ready and available, assuming the image and your custom catalog item were configured to seed user accounts and connect back to the Morpheus appliance.

Automation and Configuration Management

Morpheus automation is composed of Tasks and Workflows. A Task could be a script added directly, scripts or Blueprints pulled from the Morpheus Library, playbooks, recipes, or a number of other things. The complete list of Task types can be found in the [Automation section](#) of Morpheus docs. Tasks can be executed individually but they are often combined into workflows. We can opt to run a workflow at provision time or they can be executed on existing instances through the Actions menu.

In this guide we will set up an Ansible integration, create a Task, add the Task to a Workflow, and run the Workflow against a new and existing Instance. If you've worked through this guide to this point, you should already have an Apache instance running. If you don't yet have that, provision one before continuing with this guide and ensure it's reachable on port 80.

We'll first set up the Ansible integration, you can integrate with the sample repository referenced here or integrate with your own. Go to 'Administration > Integrations'. Click **+NEW INTEGRATION** and select Ansible from the dropdown menu. Fill in the following details:

- **NAME**
- **ANSIBLE GIT URL:** <https://github.com/ncelebic/morpheus/-ansible-example>, or enter the URL for your own Ansible git repository
- **PLAYBOOKS PATH**
- **ROLES PATH**
- Mark the box to "USE Morpheus AGENT COMMAND BUS"

Note: If your git repository requires authentication, you should create a keypair and use the following URL format: <git@github.com:ncelebic/morpheus/-ansible-example.git>.

Click **SAVE CHANGES**. You'll now see our new Ansible integration listed among any other configured integrations. If we click on this new integration to view detail, a green checkmark icon indicates the git repository has been fully synced.

With the Ansible integration set up, we can now create a task that includes our playbook. Go to [|LibAut|](#), click **+ ADD**. We'll first set our "TYPE" value to Ansible Playbook so that the correct set of fields appear in the "NEW TASK" wizard. Set the following options:

- **NAME**
- **ANSIBLE REPO:** Here we will choose the Ansible integration that we just created
- **PLAYBOOK:** In our example case, enter 'playbook.yml'

Click "SAVE CHANGES" to save our new task. We can test the new task on our Apache VM now by going to [|ProIns|](#) and clicking into our VM. From the "ACTIONS" menu select "Run Task". From the "TASK" dropdown menu, select the task we just added and click "EXECUTE".

To see the progress of the task, click on the "HISTORY" tab and click on the (i) button to the right of each entry in the list. In this case, we can also see the results of the task by clicking on the link in the "LOCATION" column of the "VMS" section.

Now that our task is created, we can put it into a workflow. Back in [|LibAut|](#) we will click on the "WORKFLOWS" tab. Click **+ADD** and select Provisioning Workflow. We'll give the new workflow a name and expand the Post Provision section. As we begin to type in the name of the task we've created, it should appear as a selection. Click "SAVE CHANGES".

Now that we have a Workflow, return to [|ProIns|](#) and begin to provision another Apache instance. More detailed instructions on provisioning a new Apache instance are included earlier in this guide if needed. Now, when you reach the

“AUTOMATION” section of the “CREATE INSTANCE” wizard, we have a workflow to select. From the “WORKFLOW” dropdown menu, select the workflow we just created and complete provisioning of the new instance.

As the instance is provisioning, we can go to the “HISTORY” tab and see Morpheus executing the tasks that were contained in our workflow.

This is just one example of using Morpheus to automate the process of configuring an instance to your needs. There are a number of other automation types that can be built into your Workflows as well. For further information, take a look at the [automation integrations](#) guide in Morpheus docs.

Conclusion

At this point you should be up and running in Morpheus, ready to consume Azure public cloud. This guide only scratches the surface, there is a lot more to see and do in Morpheus. Take a look at the rest of [Morpheus Docs](#) for more information on supported integrations and other things possible.

5.13.4 Automated Single-Node Application Deployment with Morpheus

This document provides a complete step-by-step guide to automating the configuration, installation, and deployment of the open source CRM application SuiteCRM using Morpheus. In this guide, we will create a custom Instance Type for SuiteCRM which can then be selected during provisioning. Once configured, the provisioning process is completely automated and results in SuiteCRM and a database installed on a single node. There is also an accompanying guide which automates the process for deployed SuiteCRM as a multi-tiered application consisting of a MySQL database on one node and the SuiteCRM application on a second node.

In this guide, we'll use the following Morpheus constructs to fully automate the provisioning process of a single-node application:

- Instance Types
- Layouts
- Node Types
- Cypher
- Inputs
- File Templates
- Tasks
- Provisioning Workflows

Each of these constructs can be explored more deeply in their own specific sections of Morpheus docs but this guide will illustrate how these pieces can be pulled together to automate deployment, ensure consistency and security, and enable self-service. Additionally, while this could be done on many different Cloud types, I'm setting up this Instance Type for provisioning on a VMware vCenter Cloud. You would need to have a VMware Cloud already integrated with Morpheus in order to follow the guide exactly but I will not go through the process of creating new Clouds here. If you do not have a vCenter Cloud available to you, the concepts in the guide will translate to other Cloud types, including public clouds like Amazon AWS. You may have to make slight modifications in spots in order to make a fully working Instance Type for other Clouds.

Create Cypher

Cypher is a secure key/value store in Morpheus. Using Cypher, we can securely store passwords and other secret values (such as API keys) which can then be called into automation Tasks and templates. Here we will store the MySQL root user password as a Cypher entry. In the Morpheus UI, go to Tools > Cypher and click + *ADD*.

There are a number of different types of Cypher keys, which are useful in different contexts. Here's we'll use the "secret" type which allows us to enter some known value which can be securely accessed later. Enter the following:

- **KEY:** secret/mysql_root
- **VALUE:** Enter the MySQL root account password here
- **LEASE:** 0 (Lease time is given in seconds, with "0" being unlimited)

Click *SAVE CHANGES*

ADD KEY
✕

KEY

Keys can have different behaviors depending on the specified mountpoint.

Available Mountpoints:

- **password** - Generates a secure password of specified character length in the key pattern (or 15) with symbols, numbers, upper case, and lower case letters (i.e. password/15/mypass generates a 15 character password).
- **tfvars** - This is a module to store a tfvars file for terraform.
- **secret** - This is the standard secret module that stores a key/value in encrypted form.
- **uuid** - Returns a new UUID by key name when requested and stores the generated UUID by key name for a given lease timeout period.
- **key** - Generates a Base 64 encoded AES Key of specified bit length in the key pattern (i.e. key/128/mykey generates a 128-bit key)

VALUE

1		
---	--	--

LEASE

Quick S Time Reference:

- Day: 86400
- Week: 604800
- Month (30 days): 2592000
- Year: 31536000
- Unlimited: 0

SAVE CHANGES

Create Inputs

Inputs are custom input fields which can be added to Layouts, Instance Types, and other constructs in Morpheus. The input can be consumed as variables within templates and scripts. We'll create three inputs in this case for the database name, the database username, and the database user password.

In Morpheus UI, navigate to Library > Options > Inputs. Click + *ADD*. Complete the following fields:

- **NAME:** SuiteCRM DB Name (The name for the Input object in Morpheus)
- **FIELD NAME:** databaseNameSCRM (The internal property which the input value is assigned to)
- **TYPE:** Text (The input type, in this case an open text field for the user)

- **LABEL:** SuiteCRM DB Name (The label the user sees next to the input field)

- **Database Name Input**

NEW OPTION TYPE ×

NAME

DESCRIPTION

FIELD NAME
This is the input fieldName property that the value gets assigned to.

EXPORT AS TAG

TYPE

LABEL
This is the input label that shows typically to the left of a custom option.

PLACEHOLDER

DEFAULT VALUE

REQUIRED

[SAVE CHANGES](#)

Click *SAVE CHANGES*. Then, create two additional Inputs:

- **NAME:** SuiteCRM DB User
- **FIELD NAME:** databaseUserSCRM
- **TYPE:** Text
- **LABEL:** SuiteCRM DB User
- **Database Username Input**

NEW OPTION TYPE ✕

NAME

DESCRIPTION

FIELD NAME
This is the input fieldName property that the value gets assigned to.

EXPORT AS TAG

TYPE

LABEL
This is the input label that shows typically to the left of a custom option.

PLACEHOLDER

DEFAULT VALUE

REQUIRED

[SAVE CHANGES](#)

- **NAME:** SuiteCRM DB Password
- **FIELD NAME:** databasePassSCRM
- **TYPE:** Password (Entries in a password field are not shown in plaintext on screen when entered and internally are passed securely as well)
- **LABEL:** SuiteCRM DB Password
- **Database Password Input**

NEW OPTION TYPE
✕

NAME

DESCRIPTION

FIELD NAME
This is the input fieldName property that the value gets assigned to.

EXPORT AS TAG

TYPE

LABEL
This is the input label that shows typically to the left of a custom option.

PLACEHOLDER

DEFAULT VALUE

REQUIRED

SAVE CHANGES

Create File Templates

For our SuiteCRM application, we'll need to create an Apache config file. We can create a File Template in Morpheus and the config file will be generated dynamically at provision time with the appropriate values. Navigate to Library > Templates > File Templates and click + *ADD*. Enter the following:

- **NAME:** suitecrm - conf
- **FILE NAME:** suitecrm.conf
- **FILE PATH:** /etc/apache2/sites-available
- **PHASE:** Provision
- **TEMPLATE:** See below for the complete template, note how we're able to dynamically resolve Morpheus variables within the template
- **FILE OWNER:** root
- **SETTING NAME:** suitecrm
- **SETTING CATEGORY:** App

```
<VirtualHost *:80>
  ServerAdmin admin@localhost
  ServerAlias "<%=server.externalIp%>"
  DocumentRoot /var/www/html/suitecrm

  <Directory /var/www/html/suitecrm/>
    Options FollowSymlinks
    AllowOverride All
    Require all granted
```

(continues on next page)

(continued from previous page)

```

</Directory>

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

<Directory /var/www/html/suitecrm/>
    RewriteEngine on
    RewriteBase /
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteRule ^(.*) index.php [PT,L]
</Directory>
</VirtualHost>

```

NEW FILE TEMPLATE
✕

NAME

FILE NAME

FILE PATH

PHASE

TEMPLATE

```
<VirtualHost *:80>
ServerAdmin admin@localhost
ServerAlias ".*server.externalip*"
DocumentRoot /var/www/html/suitecrm

<Directory /var/www/html/suitecrm/>
    Options FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

<Directory /var/www/html/suitecrm/>
    RewriteEngine on
```

FILE OWNER

SETTING NAME

SETTING CATEGORY

[SAVE CHANGES](#)

Create Tasks

At this point, we need to create three automation Tasks. One will set the Apache config file we just created, another will be a Bash script Task to actually install and configure SuiteCRM on the box, and the third will be another Bash script Task which will restart the Apache service.

To create a Library Template Task, navigate to Library > Automation > Tasks. Click + *ADD*. Enter the following:

- **NAME:** suitecrm file template
- **CODE:** suitecrmfiletemplate
- **TYPE:** Library Template (The proper fields will appear once the Type is set)
- **TEMPLATE:** suitecrm - conf (Select the File Template we already created from this dropdown menu)
- **EXECUTE TARGET:** Resource

NEW TASK
✕

NAME

CODE

TYPE

TEMPLATE

Target Options

EXECUTE TARGET

▶ Execute Options

SAVE CHANGES

Now create the first Bash Task which will install and configure SuiteCRM on a newly-provisioned box:

- **NAME:** suitecrm - single node
- **TYPE:** Shell Script (The proper fields will appear once the Type is set)
- **RESULT TYPE:** None
- **SUDO:** Checked
- **SOURCE:** Local (We will enter the script locally in this case but if version control repositories are integrated, such as Github, script content can be dynamically pulled from the repository at the time the Task is invoked. This ensures the code is always current without ever manually updating Tasks)
- **CONTENT:** Expand the section below to see the script content. Note how Cypher secrets and custom option (Input) values are invoked in this script
- **EXECUTE TARGET:** Resource

Install Task Content

```

RPass="

```

(continues on next page)

(continued from previous page)

```

apt update
apt-get install mariadb-server mariadb-client -y
systemctl stop mariadb.service
systemctl start mariadb.service
systemctl enable mariadb.service

#The following commands are from the mysql secure installation guidance
mysql -u root -e "UPDATE mysql.user SET Password=PASSWORD('$RPass') WHERE User='root';"
mysql -u root -e "flush privileges"
mysql -u root -p$RPass -e "DELETE FROM mysql.user WHERE User='';"
mysql -u root -p$RPass -e "DELETE FROM mysql.user WHERE User='root' AND Host NOT IN (
↳ 'localhost', '127.0.0.1', '::1');"
mysql -u root -p$RPass -e "DROP DATABASE IF EXISTS test;"
mysql -u root -p$RPass -e "DELETE FROM mysql.db WHERE Db='test' OR Db='test\_%';"
mysql -u root -p$RPass -e "FLUSH PRIVILEGES;"

#Create the SuiteCRM database
mysql -u root -p$RPass -e "CREATE DATABASE $SCRMDb;"
mysql -u root -p$RPass -e "GRANT ALL ON $SCRMDb.* TO $SCRMUser@localhost IDENTIFIED BY '
↳ $SCRMPass';"
mysql -u root -p$RPass -e "FLUSH PRIVILEGES;"

#Install required software for SuiteCRM
add-apt-repository ppa:ondrej/php -y
apt-get update
apt-get install php7.3 libapache2-mod-php7.3 php7.3-common php7.3-mysql php7.3-gmp php7.
↳ 3-curl php7.3-intl php7.3-mbstring php7.3-xmlrpc php7.3-gd php7.3-bcmath php7.3-imap
↳ php7.3-xml php7.3-cli php7.3-zip -y

#Update php.ini file with required settings
short_open_tag=On
memory_limit=256M
upload_max_filesize=100M
max_execution_time=360

for key in short_open_tag memory_limit upload_max_filesize max_execution_time
do
    sed -i "s/^\($key\).*\/\1 $(eval echo = \${$key})/" /etc/php/7.3/apache2/php.ini
done

#Restart apache
systemctl restart apache2.service

#Test file created for debugging
echo "<?php phpinfo( ); ?>" | sudo tee /var/www/html/phpinfo.php

#Download and install latest SuiteCRM. Composer v2 does not work with Suitecrm.
curl -sS https://getcomposer.org/installer | sudo php -- --version=1.10.9 --install-dir=/
↳ usr/local/bin --filename=composer
git clone https://github.com/salesagility/SuiteCRM.git /var/www/html/suitecrm

cd /var/www/html/suitecrm

```

(continues on next page)

(continued from previous page)

```
composer install --no-dev
chown -R www-data:www-data /var/www/html/suitecrm/
chmod -R 755 /var/www/html/suitecrm/
```

NEW TASK
✕

NAME

CODE

TYPE

RESULT TYPE

SOURCE

CONTENT

```
RPass=<%=cypher.read('secret/mysql_root')%>
SCRMDb=<%=customOptions.databaseNameSCR%>
SCRUser=<%=customOptions.databaseUserSCR%>
SCRMPass=<%=customOptions.databasePassSCR%>

until [ `ps -ef | grep [a]pt-get | wc -l` = 0 ]
do
    sleep 60
done

sudo apt-get install apache2 -y
sudo systemctl stop apache2.service
sudo systemctl start apache2.service
sudo systemctl enable apache2.service

sudo apt-get install mariadb-server mariadb-client -y
```

Target Options

EXECUTE TARGET

Execute Options

SAVE CHANGES

Finally, we'll add the Apache restart Task. Configure a new Task as shown below:

- **NAME:** suitecrm apache restart
- **TYPE:** Shell Script (The proper fields will appear once the Type is set)
- **RESULT TYPE:** None
- **SUDO:** Checked
- **SOURCE:** Local
- **CONTENT:** Expand the section below to see the script content
- **EXECUTE TARGET:** Resource

Restart Task Content

```
a2ensite suitecrm.conf
a2enmod rewrite
systemctl restart apache2.service
```

NEW TASK
✕

NAME

CODE

TYPE

RESULT TYPE

SOURCE

CONTENT

```
sudo a2ensite suitecrm.conf
sudo a2enmod rewrite
sudo systemctl restart apache2.service
```

Target Options

EXECUTE TARGET

▶ Execute Options

SAVE CHANGES

Create the Provisioning Workflow

Morpheus Workflows pull multiple Tasks together into a logical group. There are two types of Workflows: Operational and Provisioning. We won't use any Operational Workflows here but these Workflows can be run on-demand as needed or set to run on a recurring time schedule (like a cronjob). Provisioning Workflows are associated with an Instance at provision time and will automatically run when the Instance reaches certain phases of its lifecycle, such as during provisioning, teardown, startup, or shutdown. In this case, we can create a Provisioning Workflow with our Tasks in the provisioning phase so that SuiteCRM will be installed, the Apache config file will be set, and the Apache service will be restarted automatically when the Instance is provisioned.

Navigate to Library > Automation > Workflows and click + *ADD*. Set the following configurations:

- **NAME:** SuiteCRM - single node
- **PLATFORM:** Linux
- **TASKS:** Expand the Provision section and begin typing the names of our Tasks in the Search field. After adding them, they can be reordered but they should be set such that the install script is run first, the file template is set second, and the Apache restart is run last

Click *SAVE CHANGES*

NEW WORKFLOW ✕

NAME 👤

DESCRIPTION

PLATFORM ▾

Tasks

▶ Pre Provision

▼ Provision 3

1	 BASH	suitecrm - single node	☰	✕
2	 Library Template	suitecrm file template	☰	✕
3	 BASH	suitecrm apache restart	☰	✕

▶ Post Provision

▶ Start Service

▶ Stop Service

▶ Pre Deploy

▶ Deploy

▶ Reconfigure

▶ Teardown

Permissions

VISIBILITY ▾

SAVE CHANGES

Create a Custom Library Item

Having created Cypher entries, Inputs, and Tasks, we're ready to put them all together into a custom Instance Type for our Morpheus Library. We'll create a new SuiteCRM Library entry that will be available to some or all users (depending on Role permissions) in the provisioning wizard. This will allow them to stand up single node SuiteCRM appliances with just a few clicks. In Morpheus there are three layers to such Library items: Instance Types, Layouts, and Node Types. We'll create the Instance Type first:

Navigate to Library > Blueprints > Instance Types and click + *ADD*. Enter the following configurations:

- **NAME:** Suite_CRM
- **CODE:** SuiteCRM
- **CATEGORY:** Apps
- **ICON:** If desired, search the file system on your local computer for a SuiteCRM logo icon for easier identification of this Instance Type at provision time
- **ENVIRONMENT PREFIX:** SUITE_CRM

NEW INSTANCE TYPE
✕

NAME

CODE
Useful shortcode for provisioning naming schemes and export reference.

DESCRIPTION

CATEGORY

ICON Browse



Suggested Dimensions: 150 x 51

Permissions

VISIBILITY

Option Types

▼ Advanced

ENVIRONMENT PREFIX
Used for exportable environment variables when tying instance types together in app contexts. If not specified a name will be generated.

ENVIRONMENT VARIABLES

Name	Value	⚙	+
------	-------	---	---

ENABLE SETTINGS

ENABLE SCALING (HORIZONTAL)

SUPPORT DEPLOYMENTS
Requires a data volume be configured on each version. Files will be copied into this location.

SAVE CHANGES

Click *SAVE CHANGES*. After creating the Instance Type, click into it and then click + *ADD LAYOUT* from the Instance Type Detail Page. A Layout specifies the technology the Instance will run on, in this case VMware. It's possible to have multiple Layouts associated with an Instance Type which can be selected depending on the chosen Cloud the user might be provisioning on. Configure the Layout as follows:

- **NAME:** Single Node SuiteCRM
- **VERSION:** Latest
- **CREATABLE:** Checked (If unchecked, this Layout won't be an available option at provision time)
- **TECHNOLOGY:** VMware
- **MINIMUM MEMORY:** 2 GB (If entered, this value will override any memory requirement set on the virtual image to ensure your Instance service will run properly)

- **WORKFLOW:** Select the Workflow we've already created
- **INPUTS:** Search and find the three custom Inputs we created earlier

NEW LAYOUT
✕

NAME

VERSION

DESCRIPTION

CREATABLE

TECHNOLOGY

MINIMUM MEMORY

This will override any memory requirement set on the virtual image

WORKFLOW

SUPPORTS CONVERT TO MANAGED

ENVIRONMENT VARIABLES

Option Types

SuiteCRM DB Name (text)	☰ ✕
SuiteCRM DB User (text)	☰ ✕
SuiteCRM DB Password (password)	☰ ✕

Nodes

Once the configurations are entered, click *SAVE CHANGES*. After creating the Layout, we need to associate a Node Type. From the Layout Detail Page, click + *ADD* within the “VM Types” section. The term VM Types is sometimes used in place of Node Types in Morpheus but they refer to the same thing and are fully interchangeable. In this case, we’re simply going to point to a default Ubuntu image which is supplied by Morpheus though you can associate Node Types with your own custom virtual images when needed. Set the following configurations on the new Node Type:

- **NAME:** SuiteCRM on Ubuntu
- **SHORT NAME:** SuiteCRMUbuntu
- **VERSION:** Latest
- **VM IMAGE:** Select the included Ubuntu 18.04 image
- **COPIES:** 1

Click *SAVE CHANGES*

NEW NODE TYPE
✕

NAME

SHORT NAME
The short name is a name with no spaces used for display in your container list.

VERSION

TECHNOLOGY

ENVIRONMENT VARIABLES ⚙️ +

VMware VM Options

VM IMAGE

LOG FOLDER

CONFIG FOLDER

DEPLOY FOLDER
(Optional) If using deployment services, this mount point will be replaced with the contents of said deployments.

EXTRA OPTIONS 🗑️ +

SERVICE PORTS ⌵ +

SCRIPTS

FILE TEMPLATES

▼ Layout Specific Settings

COPIES

SCALE FACTOR APPLIES

► Advanced Options

SAVE CHANGES

Provision the SuiteCRM Instance Type

At this point, the setup is finished and SuiteCRM will be available as an Instance Type option for your users. We'll go ahead and walk through the provisioning process at this point just to take a look.

To begin provisioning, navigate to Provisioning > Instances and click + *ADD*. From the list of Instance Types, select the "SUITE_CRM" Instance Type we just created, click *NEXT*. From the Group tab, select a Group which contains a VMware Cloud and then select the VMware Cloud you'd like to provision the app onto. Click *NEXT*. From the Configuration Tab, select the Layout we created and configure a plan, Resource Pool, and network which makes sense for your specific vCenter. You'll then notice the Input fields we created where you'll need to enter a SuiteCRM database name, Username, and Password. Click *NEXT*. On the Automation tab, we do not need to select a Workflow as our

Workflow is already set on the Layout. Click *NEXT* and click *COMPLETE*.

CREATE INSTANCE
✕

TYPE
GROUP
CONFIGURE
AUTOMATION
REVIEW

Configuration Options

LAYOUT ▼
Single Node SuiteCRM

PLAN ▼
2 CPU, 8GB Memory
Cores: 2 Memory: 8 GB

RESOURCE POOL ▼
Demo-vSAN / Sandbox

VOLUMES +
root | 80 | GB | vsanDatastore - 3.1TB Fre-

NETWORKS +
VLAN0002 - Internal Server | DHCP

SUITECRM DB NAME 🗑
suitecrm

SUITECRM DB USER 🗑
suitecrm

SUITECRM DB PASSWORD 🔒

FOLDER ▼
Select

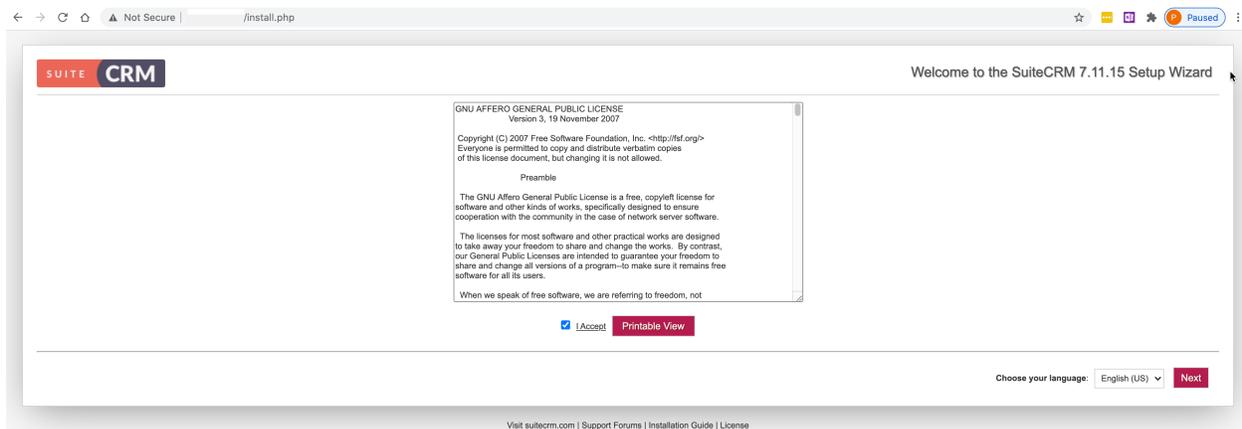
▶ User Config

▶ Advanced Options

PREVIOUS
NEXT

Configure SuiteCRM

SuiteCRM is now ready for its initial setup. In a web browser, go to `http://<YOUR_INSTANCE_IP>/install.php`. You should see the license agreement page and can proceed with the setup steps. SuiteCRM is now up and running. Additional instances of SuiteCRM can be stood up in the future with just a few clicks!



5.13.5 Automated Multi-Node Application Deployment with Morpheus

This document provides a complete step-by-step guide to automating the configuration, installation, and deployment of the open source CRM application SuiteCRM using Morpheus. This is a companion guide to our [single-node application deployment guide](#) and, in some cases, will refer directly to sections of that guide rather than duplicating information here.

In this guide, we will create a custom App Blueprint for SuiteCRM consisting of a database node and an application node. Once configured, the provisioning process is completely automated (Provisioning > Apps) and results in the two-tiered app mentioned previously. We'll use the following Morpheus constructs to fully automate the provisioning process of a multi-tiered application:

- App Blueprints
- Cypher
- Inputs
- File Templates
- Tasks
- Provisioning Workflows

Each of these constructs can be explored more deeply in their own specific sections of Morpheus docs but this guide will illustrate how these pieces can be pulled together to automate deployment, ensure consistency and security, and enable self-service. Additionally, while this could be done on many different Cloud types, I'm setting up this Instance Type for provisioning on a VMware vCenter Cloud. You would need to have a VMware Cloud already integrated with Morpheus in order to follow the guide exactly but I will not go through the process of creating new Clouds here. If you do not have a vCenter Cloud available to you, the concepts in the guide will translate to other Cloud types, including public clouds like Amazon AWS. You may have to make slight modifications in spots in order to make a fully working Instance Type for other Clouds.

Create Cypher

Cypher is a secure key/value store in Morpheus. Using Cypher, we can securely store passwords and other secret values (such as API keys) which can then be called into automation Tasks and templates. In this case, we need to store two Cypher secrets: The first is for the MySQL root password and the second is for the user which we will use to run SSH commands from the app server to the database server. Refer to the [single-node deployment guide](#) for details on setting up the first Cypher entry.

Next we need to set up a Cypher for the SSH user so our application node can access the database. A handy trick here is to use the user specified in the Morpheus global Cloud-Init Setting as this will be automatically created on the Instances when they are provisioned by Morpheus. To check that the user is set up correctly, in the Morpheus UI go to Administration > Settings > Provisioning. Under the Cloud-Init Settings section, ensure that a username and password is set. After that, simply create a *secret/cloudinit* Cypher entry which stores the password for this user using the same process by which you set up the MySQL root password Cypher entry.

Create Inputs

Inputs are custom input fields in Morpheus. The input can be consumed as variables within templates and scripts. We'll create three inputs in this case for the database name, the database username, and the database user password. The steps to create these inputs have been detailed in the companion guide, if needed, you can reference them [here](#).

Create File Template and Library Template Task

For our SuiteCRM application, we'll need to create an Apache config file. We can create a File Template in Morpheus and the config file will be generated dynamically at provision time with the appropriate values. We'll also create a Task which we can use to automatically set our file template at the appropriate time during provisioning. The steps to create the [File Template](#) and [Library Template Task](#) are detailed in the companion guide so follow the links for walkthrough steps and return to this guide.

Create Database Install Task

Since the database is installed on its own node, we need a Task which installs just database. In the prior section, we created a Library Template Task but this time we will create Bash Script Tasks to handle database node installation and application node installation. We'll also need a simple third Task to restart the Apache service after setting the config file. We'll go through the steps for creating all three of these Tasks in this section.

Navigate to Library > Automation > Tasks and click + *ADD* to start a new Task. Configure the new Task as follows:

- **Name:** suitecrm install db multi node
- **Code:** suitecrminstalldbMN
- **Type:** Shell Script (The proper fields will appear once the Type is set)
- **Sudo:** Checked
- **Source:** Local (We will enter the script locally in this case but if version control repositories are integrated, such as Github, script content can be dynamically pulled from the repository at the time the Task is invoked. This ensures the code is always current without ever manually updating Tasks)
- **Execute Target:** Resource

Database Install Task Content

```
RPass="<%=cypher.read('secret/mysql_root')%>"
IP="<%=server.internalIp%>"

#Wait until any apt-get processes have finished
if [ `ps -ef | grep [a]pt-get | wc -l` != 0 ]
then
    sleep 120
fi

#Install MariaDB, start service and enable on boot
wget https://downloads.mariadb.com/MariaDB/mariadb_repo_setup
echo "fd3f41eef54ce144c932100f9e0f9b1d181e0edd86a6f6b8f2a0212100c32c mariadb_repo_setup
↵" | sha256sum -c -
chmod +x mariadb_repo_setup
./mariadb_repo_setup --mariadb-server-version="mariadb-10.6"
apt update
apt-get install mariadb-server mariadb-client -y
```

(continues on next page)

(continued from previous page)

```

systemctl stop mariadb.service
systemctl start mariadb.service
systemctl enable mariadb.service

#The following commands are from the mysql secure installation guidance
mysql -u root -e "UPDATE mysql.user SET Password=PASSWORD('$RPass') WHERE User='root';"
mysql -u root -e "flush privileges"
mysql -u root -p$RPass -e "DELETE FROM mysql.user WHERE User='';"
mysql -u root -p$RPass -e "DELETE FROM mysql.user WHERE User='root' AND Host NOT IN (
↪ 'localhost', '127.0.0.1', ':::1');"
mysql -u root -p$RPass -e "DROP DATABASE IF EXISTS test;"
mysql -u root -p$RPass -e "DELETE FROM mysql.db WHERE Db='test' OR Db='test\_%';"
mysql -u root -p$RPass -e "FLUSH PRIVILEGES;"

#Set bind-address parameter in my.cnf
sed -e '/^bind/s/^/#/g' -i /etc/mysql/mariadb.conf.d/50-server.cnf
systemctl restart mariadb.service

```

Create Application Install and Apache Restart Tasks

We now need to create a Bash script task to install the application. However, before we do this, we need to consider how these Tasks will run. We need to have the database installed before the application. We can achieve this in our App Blueprint by configuring the boot order so that the database is provisioned before the application. However, as part of the database configuration, we also need to create a database user and set up remote access to the database for the application server IP. The challenge here is that we cannot create the database user as part of the database install Task as we will not know the application server IP address. To get around this, we will run the database configuration Tasks within the application install task using *sshpass* to remotely execute the MySQL commands on the database. This is where the *cloud-init* user and Cypher we created earlier come in.

Note: In the code section below, remember to set the correct username for your cloud-init user. In the code below it is set to *pjonesci*. You will also see in the Task below we are using environment variables to pull in the IP address of the database host (*MYSQL_HOST* variable). We are able to do this because in the App Blueprint we will connect the tiers which means Instances in a tier can import the envs from Instances in connected tiers.

Once again, click + *ADD* to start a new Task. Configure the new Task as follows:

- **Name:** suitecrm install app multi node
- **Code:** suitecrminstallappMN
- **Type:** Shell Script
- **Sudo:** Checked
- **Source:** Local
- **Execute Target:** Resource

Application Install Task Content

```

RPass="<%=cypher.read('secret/mysql_root')%>"
CIPass="<%=cypher.read('secret/cloudinit')%>"
SCRMDb="<%=customOptions.databaseNameSCRM%>"

```

(continues on next page)

(continued from previous page)

```

SCRMSUser="<%=customOptions.databaseUserSCRMS%>"
SCRMPass="<%=customOptions.databasePassSCRMS%>"
MYSQL_HOST="<%=evars.SUITECRMDBMN_IP%>"
IP="<%=server.internalIp%>"

#Wait until any apt-get processes have finished
if [ `ps -ef | grep [a]pt-get | wc -l` = !0 ]
then
    sleep 120
fi

#Install sshpass and apache, start service and enable on boot
apt-get install sshpass -y
apt-get install apache2 -y
systemctl stop apache2.service
systemctl start apache2.service
systemctl enable apache2.service

#Use sshpass to remotely execute mysql commands on DB server to create database and
↳ database user
sshpass -p $CIPass ssh -o StrictHostKeyChecking=no -t pjonesci@$MYSQL_HOST <<REMOTE
sudo -S <<< "$CIPass" mysql -u root -p$RPass -e "CREATE USER '$SCRMSUser'@$IP'
↳ IDENTIFIED BY '$SCRMPass';"
sudo -S <<< "$CIPass" mysql -u root -p$RPass -e "CREATE DATABASE $SCRMDb;"
sudo -S <<< "$CIPass" mysql -u root -p$RPass -e "GRANT ALL ON $SCRMDb.* TO $SCRMSUser@$IP
↳ IDENTIFIED BY '$SCRMPass';"
sudo -S <<< "$CIPass" mysql -u root -p$RPass -e "FLUSH PRIVILEGES;"
REMOTE

#Install required software for SuiteCRM
add-apt-repository ppa:ondrej/php -y
apt-get update
apt-get install php7.3 libapache2-mod-php7.3 php7.3-common php7.3-mysql php7.3-gmp php7.
↳ 3-curl php7.3-intl php7.3-mbstring php7.3-xmlrpc php7.3-gd php7.3-bcmath php7.3-imap
↳ php7.3-xml php7.3-cli php7.3-zip -y

#Update php.ini file with required settings
short_open_tag=On
memory_limit=256M
upload_max_filesize=100M
max_execution_time=360

for key in short_open_tag memory_limit upload_max_filesize max_execution_time
do
    sed -i "s/^\($key\).*$/\1 $(eval echo = \${$key})/" /etc/php/7.3/apache2/php.ini
done

#Restart apache
systemctl restart apache2.service

#Test file created for debugging
echo "<?php phpinfo( ); ?>" | sudo tee /var/www/html/phpinfo.php

```

(continues on next page)

(continued from previous page)

```
#Download and install latest SuiteCRM. Composer v2 does not work with Suitecrm.
curl -sS https://getcomposer.org/installer | sudo php -- --version=1.10.9 --install-dir=/
↳usr/local/bin --filename=composer
git clone https://github.com/salesagility/SuiteCRM.git /var/www/html/suitecrm

cd /var/www/html/suitecrm
composer install --no-dev
chown -R www-data:www-data /var/www/html/suitecrm/
chmod -R 755 /var/www/html/suitecrm/
```

Finally, create the Apache restart Task mentioned earlier. The exact steps to create this Task are shown in the [companion single-node application guide](#).

Create Workflows

Morpheus Workflows pull multiple Tasks together into a logical group. There are two types of Workflows: Operational and Provisioning. We won't use any Operational Workflows here but these Workflows can be run on-demand as needed or set to run on a recurring time schedule (like a cronjob). Provisioning Workflows are associated with an Instance at provision time and will automatically run when the Instance reaches certain phases of its lifecycle, such as during provisioning, teardown, startup, or shutdown. In this case, we can create a Provisioning Workflow with our Tasks in the provision phase. One Workflow will primarily handle database node installation and the other will primarily handle application node installation.

Navigate to Library > Automation > Workflows and click + *ADD*. Set the following configurations:

- **NAME:** SuiteCRMDB - Multi Node
- **PLATFORM:** Linux
- **TASKS:** In the provision phase, add our “suitecrm install db multi node” Task

NEW WORKFLOW ✕

NAME

DESCRIPTION

PLATFORM

Tasks

- ▶ Pre Provision
- ▼ Provision 1
 -
 - 1  suitecrm install db multi node ✕
- ▶ Post Provision
- ▶ Start Service
- ▶ Stop Service
- ▶ Pre Deploy
- ▶ Deploy
- ▶ Reconfigure
- ▶ Teardown

[SAVE CHANGES](#)

Click *SAVE CHANGES* and then create the second Workflow:

- **NAME:** SuiteCRMApp - Multi Node
- **PLATFORM:** Linux
- **TASKS:** In the provision phase, add our “suitecrm install app multi node”, “suitecrm file template”, and “suitecrm apache restart” Tasks

Click *SAVE CHANGES*

NEW WORKFLOW ✕

NAME

DESCRIPTION

PLATFORM

Tasks

- ▶ Pre Provision
- ▼ Provision 3
 -
 - 1  suitecrm install app multi node ⋮ ✕
 - 2  suitecrm file template ⋮ ✕
 - 3  suitecrm apache restart ⋮ ✕
- ▶ Post Provision
- ▶ Start Service
- ▶ Stop Service
- ▶ Pre Deploy
- ▶ Deploy
- ▶ Reconfigure
- ▶ Teardown

Create Instance Types for the Database and Application

At this point we're ready to put our pieces together into custom Instance Types for the database and application. Instance Types can be provisioned individually (Provisioning > Instances) but in this case we want to structure multiple Instance Types into logical tiers in an App Blueprint so they can be provisioned as a Morpheus App. Each Instance Type will contain a Layout and a Node Type, I'll discuss each construct more fully as it's time to create them.

Navigate to Library > Blueprints > Instance Types and click + *ADD*. Set the following configurations:

- **NAME:** SUITE_CRM_DB_MN
- **CODE:** suitecrmdbmn
- **CATEGORY:** SQL
- **ICON:** If desired, browse your local computer for a MariaDB icon to make this Instance Type more recognizable when provisioning or creating App Blueprints
- **ENVIRONMENT PREFIX:** SUITECRMDBMN

NEW INSTANCE TYPE
✕

NAME

CODE
Useful shortcode for provisioning naming schemes and export reference.

DESCRIPTION

CATEGORY

ICON

Suggested Dimensions: 150 x 51

Permissions

VISIBILITY

Option Types

▼ Advanced

ENVIRONMENT PREFIX
Used for exportable environment variables when tying instance types together in app contexts. If not specified a name will be generated.

ENVIRONMENT VARIABLES

Name	Value
<input type="checkbox"/> ENABLE SETTINGS	
<input type="checkbox"/> ENABLE SCALING (HORIZONTAL)	
<input type="checkbox"/> SUPPORT DEPLOYMENTS	<small>Requires a data volume be configured on each version. Files will be copied into this location.</small>

Click *SAVE CHANGES*. With the Instance Type created, we're ready to add a Layout. A Layout specifies the technology the Instance will run on, in this case VMware. It's possible to have multiple Layouts associated with an Instance Type which can be selected depending on the chosen Cloud the user might be provisioning on. Configure the Layout as follows:

- **NAME:** LO_SUITE_CRM_DB_MN
- **VERSION:** Latest
- **CREATABLE:** Checked
- **TECHNOLOGY:** VMware
- **WORKFLOW:** Select the "SuiteCRM - Multi Node" Workflow that was previously created

- **INPUTS:** Include the “SuiteCRM DB Name”, “SuiteCRM DB Password”, and “SuiteCRM DB User” Inputs that were previously created

NEW LAYOUT ✕

NAME

VERSION

DESCRIPTION

CREATABLE

TECHNOLOGY

MINIMUM MEMORY
This will override any memory requirement set on the virtual image

WORKFLOW

SUPPORTS CONVERT TO MANAGED

ENVIRONMENT VARIABLES ⚙️ +

Option Types

SuiteCRM DB Name (text)	☰	✕
SuiteCRM DB Password (password)	☰	✕
SuiteCRM DB User (text)	☰	✕

Nodes

SAVE CHANGES

Click *SAVE CHANGES*. With the Layout created, we’re ready to associated a Node Type. From the Layout Detail Page, click + *ADD* within the “VM Types” section. The term VM Types is sometimes used in place of Node Types in Morpheus but they refer to the same thing and are fully interchangeable. In this case, we’re simply going to point to a default Ubuntu image which is supplied by Morpheus though you can associate Node Types with your own custom virtual images when needed. Set the following configurations on the new Node Type:

- **NAME:** NODE_SUITE_CRM_DB_MN
- **SHORT NAME:** nodesuitecrmdbmn
- **VERSION:** Latest
- **TECHNOLOGY:** VMware
- **VM IMAGE:** Choose the Morpheus Ubuntu 18 image that is included with Morpheus out of the box

Click *SAVE CHANGES*. This completes the database Library item needed to build out our App Blueprint in the next section. We now need to add a Library item representing the application node. Below I will list out the configurations

for the Instance Type, Layout, and Node Type, you can refer back to the steps above if you need to see the click-by-click instructions once again for creating these objects.

- **NAME:** SUITE_CRM_APP_MN
- **CODE:** suitecrmappmn
- **CATEGORY:** Apps
- **ICON:** If desired, browse your local computer for a SuiteCRM icon to make this Instance Type more recognizable when provisioning or creating App Blueprints
- **ENVIRONMENT PREFIX:** SUITECRMAPPMN

NEW INSTANCETYPE
✕

NAME

CODE
Useful shortcode for provisioning naming schemes and export reference.

DESCRIPTION

CATEGORY

ICON

Suggested Dimensions: 150 x 51

Permissions

VISIBILITY

Option Types

▼ **Advanced**

ENVIRONMENT PREFIX
Used for exportable environment variables when tying instance types together in app contexts. If not specified a name will be generated.

ENVIRONMENT VARIABLES

- ENABLE SETTINGS
- ENABLE SCALING (HORIZONTAL)
- SUPPORT DEPLOYMENTS
Requires a data volume be configured on each version. Files will be copied into this location.

- **NAME:** LO_SUITE_CRM_APP_MN

- **VERSION:** Latest
- **CREATABLE:** Checked
- **TECHNOLOGY:** VMware
- **MINIMUM MEMORY:** 4 GB (If entered, this value will override any memory requirement set on the virtual image to ensure your Instance service will run properly)
- **INPUTS:** Include the “SuiteCRM DB Name”, “SuiteCRM DB Password”, and “SuiteCRM DB User” Inputs that were previously created

NEW LAYOUT ✕

NAME

VERSION

DESCRIPTION

CREATABLE

TECHNOLOGY

MINIMUM MEMORY
This will override any memory requirement set on the virtual image

WORKFLOW

SUPPORTS CONVERT TO MANAGED

ENVIRONMENT VARIABLES ⚙️ +

Option Types

SuiteCRM DB Name (text)	☰	✕
SuiteCRM DB Password (password)	☰	✕
SuiteCRM DB User (text)	☰	✕

Nodes

[SAVE CHANGES](#)

- **NAME:** NODE_SUITE_CRM_APP_MN
- **SHORT NAME:** nodesuitecrmappmn
- **VERSION:** Latest
- **TECHNOLOGY:** VMware
- **VM IMAGE:** Choose the Morpheus Ubuntu 18 image that is included with Morpheus out of the box

NEW NODE TYPE
✕

NAME

SHORT NAME
The short name is a name with no spaces used for display in your container list.

VERSION

TECHNOLOGY

ENVIRONMENT VARIABLES ⚙️ +

VMware VM Options

VM IMAGE

LOG FOLDER

CONFIG FOLDER

DEPLOY FOLDER
(Optional) If using deployment services, this mount point will be replaced with the contents of said deployments.

EXTRA OPTIONS 🗑️ +

SERVICE PORTS ⌵ +

SCRIPTS

FILE TEMPLATES

▼ Layout Specific Settings

COPIES

SCALE FACTOR APPLIES

▶ Advanced Options

SAVE CHANGES

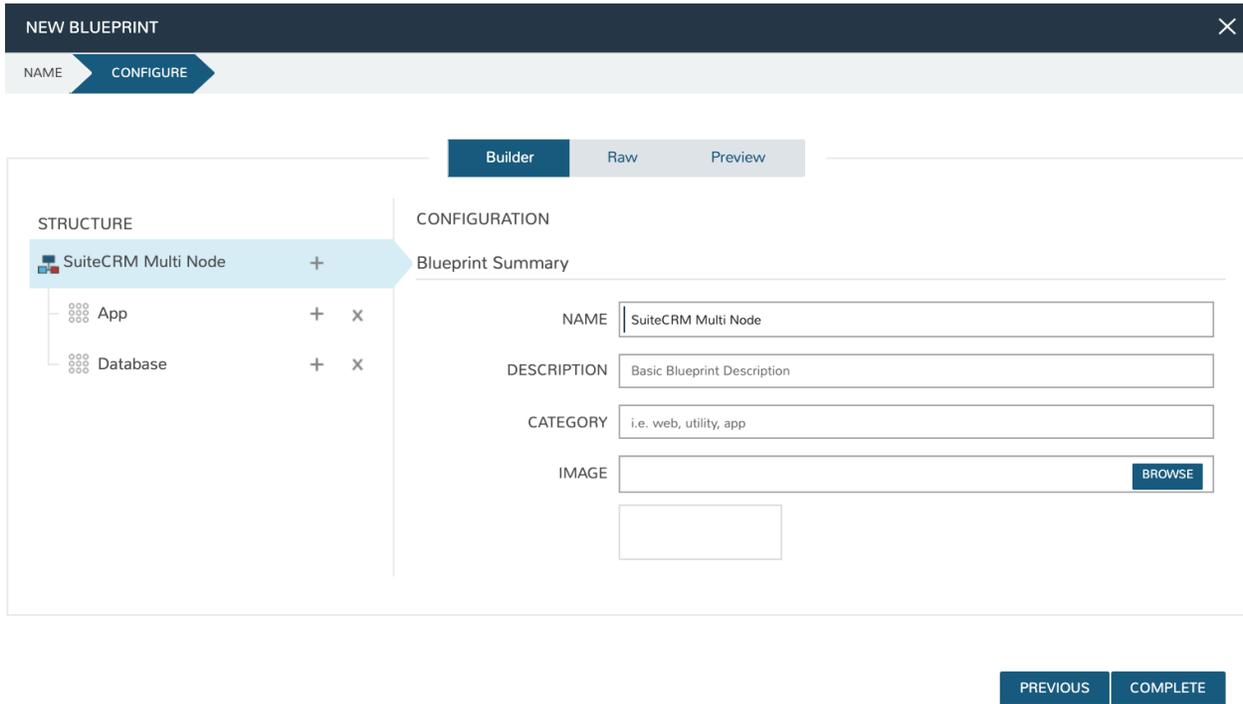
Create the App Blueprint

We are now ready create a Morpheus App Blueprint. Morpheus Blueprints allow pre-configured multi-tier application deployments for multiple environments. In this example we will use the custom Instance Types previously created to build out an App Blueprint for a two-tier SuiteCRM application with the database running on one VM and the application running on another VM.

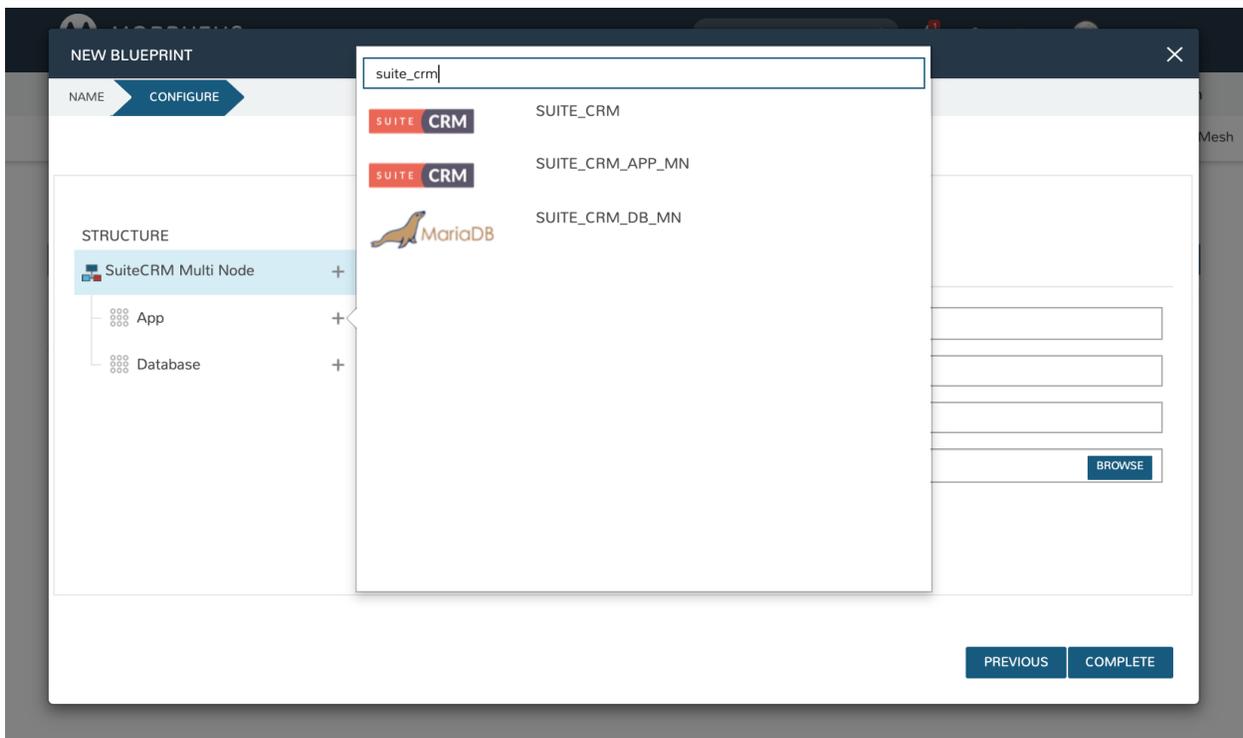
Go to Library > Blueprints > App Blueprints and click + *ADD*. Enter a name for the Blueprint (SuiteCRM Multi Node) and set type to Morpheus. Click *NEXT*.

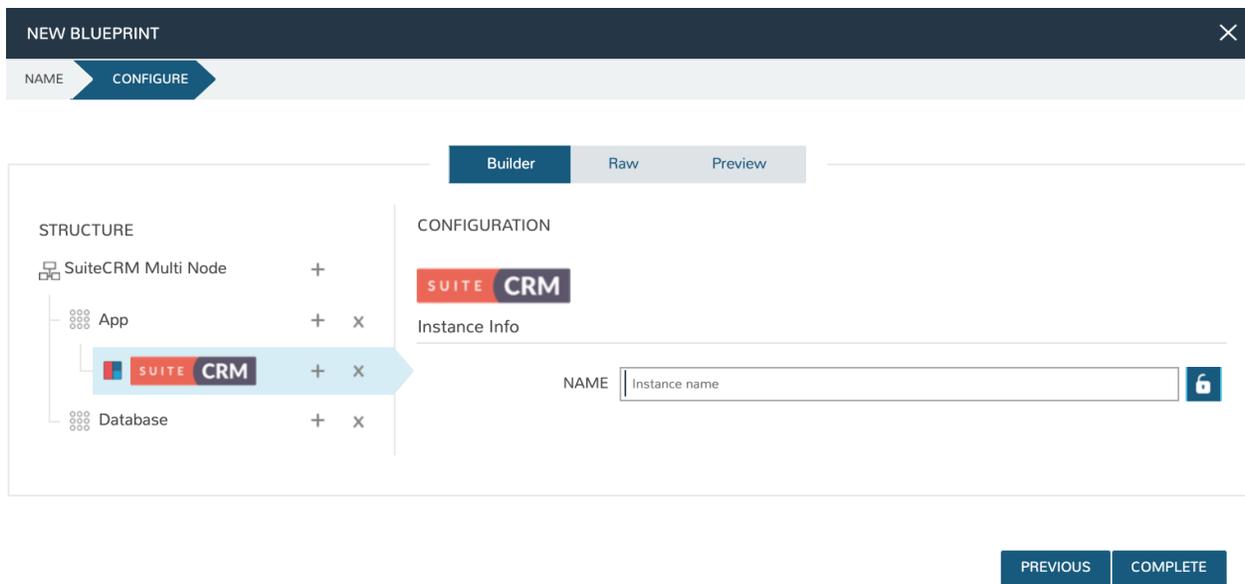
You will now be in the App Blueprint configuration screen where you can build out the structure of the App Blueprint.

In the structure section on the left of the screen, click the + to add App Tier and then click it again to add a Database tier.

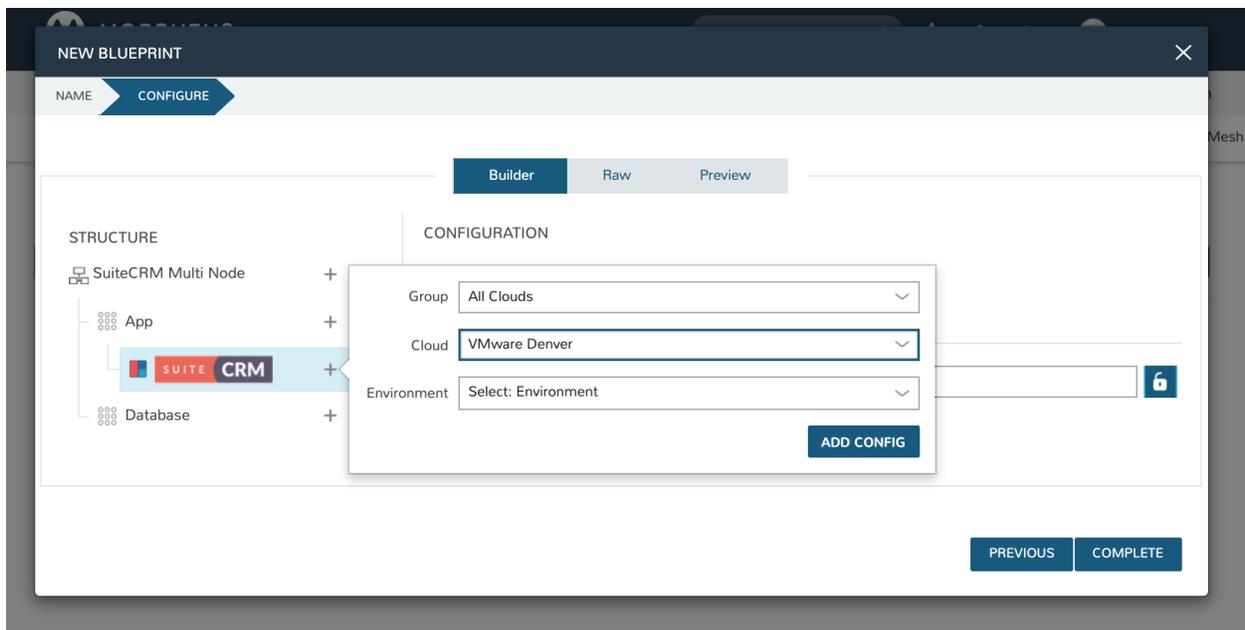


Now that we have the tiers created, we can create the configuration for each tier. Click the “+” next to the App tier and in the window that pops up select the application Instance Type created previously (SUITE_CRM_APP_MN).





Now click the “+” next to the Suite CRM Instance Type (under App). Select the Group and Cloud required (I’m selecting my VMware Cloud, in this case) and then click *ADD CONFIG*.



You will now have a configuration under the Suite CRM icon. Click on the configuration and it will appear in the right-hand pane of the window. Fill in the configuration required to provision the Instance. The locks to the right of the fields allow you to lock down entries so that they cannot be changed when provisioning the App later. Do not click *COMPLETE* yet, we still need to configure the database tier.

NEW BLUEPRINT
✕

NAME
CONFIGURE

Builder
Raw
Preview

STRUCTURE

- ☰ SuiteCRM Multi Node +
- ☰ App + X
- ☑ SUITE CRM + X
- ⚙ Group: All Clouds, Cloud: VMware Denver X
- ☰ Database + X
- 🦁 MariaDB + X
- ⚙ Group: All Clouds, Cloud: VMware Denver X

CONFIGURATION

ALLOW EXISTING INSTANCES

▼ Instance Info

NAME 🔒

DESCRIPTION 🔒

▼ Configuration Options

LAYOUT 🔒

PLAN 🔒
Cores: 1 Memory: 4 GB

RESOURCE POOL 🔒

VOLUMES GB + 🔒

NETWORKS DHCP + 🔒

SUITECRM DB NAME 🔒

SUITECRM DB PASSWORD 🔒

SUITECRM DB USER 🔒

FOLDER 🔒

▶ User Config

▶ Advanced Options

▶ Automation

▶ Backups

▶ Lifecycle

PREVIOUS
COMPLETE

Back in the left-hand pane, click the “+” next to Database and perform the same steps as before to add in the SUITE_CRM_DB_MN Instance Type to the database tier. Add in the required configuration.

NEW BLUEPRINT ✕

NAME CONFIGURE

BuilderRawPreview

STRUCTURE

- SuiteCRM Multi Node +
- App + X
- SUITE CRM + X
- Group: All Clouds, Cloud: VMware Denver X
- Database + X
- MariaDB + X
- Group: All Clouds, Cloud: VMware Denver X

CONFIGURATION

ALLOW EXISTING INSTANCES

▼ Instance Info

NAME 🔒

DESCRIPTION 🔒

▼ Configuration Options

LAYOUT 🔒

PLAN 🔒
Cores: 1 Memory: 4 GB

RESOURCE POOL 🔒

VOLUMES + 🔒

NETWORKS + 🔒

SUITECRM DB NAME 🔒

SUITECRM DB PASSWORD 🔒

SUITECRM DB USER 🔒

FOLDER 🔒

▶ User Config

▶ Advanced Options

▶ Automation

▶ Backups

▶ Lifecycle

PREVIOUS COMPLETE

Once the configuration is set up for both tiers, we need to set the boot order and connect the tiers. The boot order is used to control the order in which the tiers are built. We want the database tier to build first, so we set the boot order to 0 on the database and 1 on the application. We also need to connect the tiers by checking the box under connected tiers.

EDIT BLUEPRINT

Builder Raw Preview

STRUCTURE

- SuiteCRM Multi Node +
 - App + X
 - SUITE CRM + X
 - Group: All Clouds, Cloud: VMware Denver X
 - Database + X APP
 - MariaDB + X
 - Group: All Clouds, Cloud: VMware Denver X

CONFIGURATION

Tier Info

NAME Database

BOOT ORDER 0

Connected Tiers

APP

SAVE

Save the App Blueprint by clicking *COMPLETE*.

Deploy the App Blueprint

To deploy an App Blueprint, navigate to Provisioning > Apps and click + *ADD*. Select the App Blueprint we just created and work through the provisioning wizard, including naming the App and selecting the target Cloud.

Note: For the database hostname, specify the internal IP address of the database node

NEW APP

TEMPLATE SETUP CONFIGURE REVIEW

App Summary

NAME SuiteCRM

GROUP All Clouds

DEFAULT CLOUD VMware Denver

ENVIRONMENT Choose an Environment

PREVIOUS NEXT

NEW APP [X]

TEMPLATE > SETUP > **CONFIGURE** > REVIEW

Builder Raw Preview

STRUCTURE

- SuiteCRM
 - Database
 - MariaDB
 - App
 - SUITE CRM

CONFIGURATION

Blueprint Summary

NAME: SuiteCRM Multi Node

DESCRIPTION: Basic Blueprint Description

CATEGORY: i.e. web, utility, app

PREVIOUS NEXT

NEW APP [X]

TEMPLATE > SETUP > CONFIGURE > **REVIEW**

Review App

NAME: SuiteCRM
TEMPLATE: SuiteCRM Multi Node
GROUP: All Clouds
ESTIMATED COST: USD108.00 / Month

▼ APP

 **INSTANCE OPTIONS**

- NAME: PJSPSUI1003
- CLOUD: VMware Denver
- TYPE: suitecrmappmn
- PLAN: 1 CPU, 4GB Memory
Cores 1 Memory 4GB Price USD54.00 / Month
- VERSION: Latest
- LAYOUT: LO_SUITE_CRM_APP_MN

VOLUMES

ROOT: 40 GB vsanDatastore

NETWORKS

VLAN0002 - INTERNAL SERVER: DHCP

▼ DATABASE

 **INSTANCE OPTIONS**

- NAME: PJSPSUI1003
- CLOUD: VMware Denver
- TYPE: suitecrmdbmn
- PLAN: 1 CPU, 4GB Memory
Cores 1 Memory 4GB Price USD54.00 / Month
- VERSION: Latest
- LAYOUT: LO_SUITE_CRM_DB_MN

VOLUMES

ROOT: 40 GB vsanDatastore

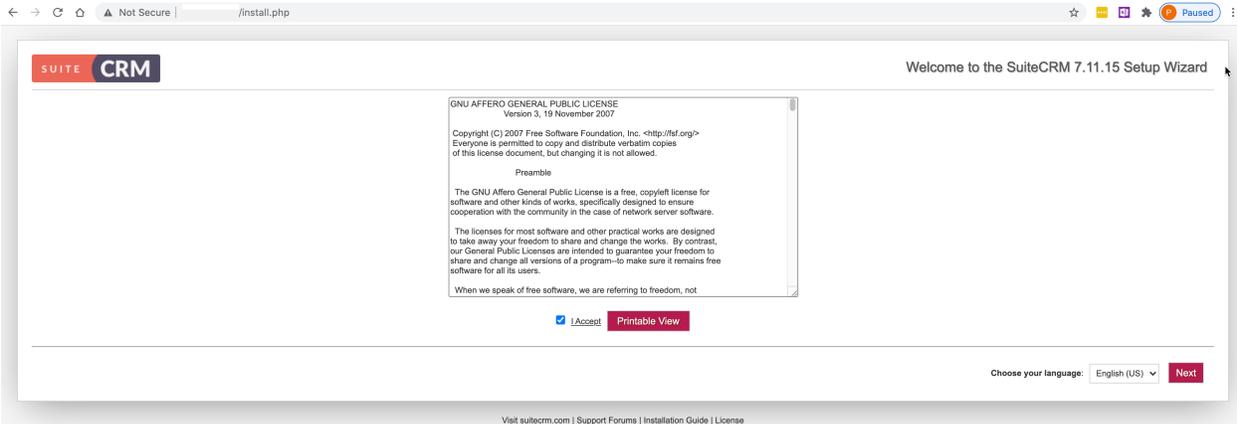
NETWORKS

VLAN0002 - INTERNAL SERVER: DHCP

PREVIOUS COMPLETE

Configure SuiteCRM

SuiteCRM is now ready for its initial setup. In a web browser, go to `http://<YOUR_APP_NODE_IP>/install.php`. You should see the license agreement page and can proceed with the setup steps. SuiteCRM is now up and running. Additional instances of SuiteCRM can be stood up in the future with just a few clicks!



5.13.6 Adding Functionality Through Operational Workflows

Operational Workflows, in Morpheus, are logically arranged sets of individual automation Tasks designed to accomplish some function. They can be executed on a one-off basis as needed or set on a timed execution schedule, similar to a cron job. While they're often used to orchestrate some facet of cloud resource management, they can also be used to speed and standardize creation of Morpheus constructs.

As an example case for this guide, we'll create an Operational Workflow which will allow Primary Tenant administrators to create Groups within Subtenants. This is not natively-available functionality as Groups are not a tenanted construct in Morpheus. They exist solely within the Tenant that created them and cannot be shared or viewed by others.

In this guide, we'll use the following Morpheus constructs:

- Groups
- Cypher
- Tasks
- Workflows
- Inputs
- Morpheus API

Initial Considerations

Since we're creating Groups in a second Tenant, you'll need to have at least one other Tenant in your appliance (Administration > Tenants). You'll also need access to a user account within the Subtenant which has rights to create Groups. You could use a pre-existing user account or create a service user for automated processes like this one.

Impersonate or log in with your service account and create an API Key. API Keys are created by clicking on the user's name in the upper-right corner of the application window, then clicking User Settings. On the User Settings page, click *API ACCESS*. Next to one of the entries (it doesn't matter which one but be careful about overwriting a key that is already in use), click *ACTIONS* and then *Generate*. Copy the Access Token into a secure place you can retrieve it from in the next step. You will not be able to see this key again after closing the modal containing the API Keys. Return to your primary Morpheus account in the original Tenant to continue with the guide.

Create Cypher

Cypher is a secure key/value store in Morpheus. Using Cypher, we can securely store passwords and other secret values (such as API keys) which can then be called into automation Tasks and templates. Here we will store the API Key for our service account user so we can call it into automation scripts in a later step. In the Morpheus UI, go to Tools > Cypher and click + *ADD*.

There are a number of different types of Cypher keys, which are useful in different contexts. Here we'll use the "secret" type which allows us to enter some known value which can be securely accessed later. Enter the following:

- **KEY:** secret/subusertoken
- **VALUE:** Enter Morpheus API Key for the Subtenant user
- **LEASE:** 0 (Lease time is given in seconds, with "0" being unlimited)

Click *SAVE CHANGES*

ADD KEY

✕

KEY

Keys can have different behaviors depending on the specified mountpoint.

Available Mountpoints:

- **password** - Generates a secure password of specified character length in the key pattern (or 15) with symbols, numbers, upper case, and lower case letters (i.e. password/15/mypass generates a 15 character password).
- **tfvars** - This is a module to store a tfvars file for terraform.
- **secret** - This is the standard secret module that stores a key/value in encrypted form.
- **sys** - This is the standard secret module that stores a key/value in encrypted form.
- **uuid** - Returns a new UUID by key name when requested and stores the generated UUID by key name for a given lease timeout period.
- **key** - Generates a Base 64 encoded AES Key of specified bit length in the key pattern (i.e. key/128/mykey generates a 128-bit key)
- **vault** - This allows secret data to be fetched from a HashiCorp Vault integration. This can be configured in the vault/config key setup

VALUE

1 xxx

LEASE

Quick S Time Reference:

- Day 86400

Create Inputs

Inputs are custom input fields which can be added to Layouts, Instance Types, Workflows, and other constructs in Morpheus. The input can be consumed as variables within templates and scripts. In this case, the Inputs will be associated with our Operational Workflow. We will create three so the user can define a name for the new Group and optionally provide a code and/or location value if desired.

In Morpheus UI, navigate to Library > Options > Inputs. Click + *ADD*. Complete the following fields:

- **NAME:** Subtenant Group Name (The friendly name for the Input object in Morpheus)
- **FIELD NAME:** subgroupname (The internal property which the input value is assigned to)
- **TYPE:** Text (The input type, in this case an open text field for the user)
- **LABEL:** Subtenant Group Name (The label the user sees next to the input field)
- **REQUIRED:** Checked (A name value is required for our Group so this Input should be marked required. The other two Inputs we'll create will not be required as they are optional fields when creating a Group)

Once done, click *SAVE CHANGES*.

EDIT INPUT
✕

NAME

DESCRIPTION

FIELD NAME
This is the input fieldName property that the value gets assigned to.

EXPORT AS TAG

DEPENDENT FIELD
A fieldName that will trigger reloading this option list

VISIBILITY FIELD
A fieldName that will trigger visibility of this option

REQUIRE FIELD
A fieldName that will trigger required attribute of this option

SHOW ON EDIT

DISPLAY VALUE ON DETAILS

TYPE

LABEL
This is the input label that shows typically to the left of a custom option.

Create two more Inputs in a similar fashion with the following configuration:

Code Input

- **NAME:** Subtenant Group Code
- **FIELD NAME:** subgroupcode
- **TYPE:** Text
- **LABEL:** Subtenant Group Code
- **HELP BLOCK:** Optional Code Value (I opted to enter help block text to make it clearer to the user that this is an optional input)
- **REQUIRED:** Unchecked

Location Input

- **NAME:** Subtenant Group Location
- **FIELD NAME:** subgrouplocation
- **TYPE:** Text
- **LABEL:** Subtenant Group Location
- **HELP BLOCK:** Optional Location Value
- **REQUIRED:** Unchecked

Create Task

Tasks, in Morpheus, are individual automation scripts. They can be pieced together into Workflows (as we'll see later) to create more comprehensive automation packages. They can be written in a number of different languages (including BASH, Powershell, Python, Javascript, and more) or to accomplish specific functions like restarting a server or sending an email notification. In this case, I've written a Python script outside Morpheus and tracked it in a Github repository. Since my Github account is integrated with Morpheus, I can simply refer to the script in a new Task. This prevents me from having to cut and paste code and also ensures the latest version of my code is executed every time the Task is invoked. Integrating with version control is optional, for simplicity you can also draft a quick script directly in Morpheus if you'd prefer. If you want to view, copy, or fork the code, you can see it [here](#). New Github integrations can be added in Administration > Integrations.

Navigate to Library > Automation > Tasks and click + *ADD*. Create a new Task with the following configuration:

- **NAME:** Subtenant Group Create - Task
- **TYPE:** Python Script (Once Type is selected, available fields will be updated to those specific to the chosen type)
- **RESULT TYPE:** None
- **SOURCE:** Repository (Select Local to draft or paste code directly into Morpheus)
- **FILE PATH:** main.py (In my case, the script is in the root of the repository so I can simply refer to it by filename)
- **COMMAND ARGUMENTS:** Optional command line arguments for the Python script. In my case, I'm passing the Morpheus API Key from Cypher as a command line argument (as seen in the screenshot) and consuming it in my code using the `sys` module, which is part of the Python standard library. There are other ways to consume Cypher secrets in Python scripts as well, which are laid out in a [Knowledge Base article](#).
- **ADDITIONAL PACKAGES:** List packages used which are not part of the standard Python library

Once done, click *SAVE CHANGES*.

EDIT TASK ✕

NAME	<input type="text" value="Subtenant Group Create - Task"/>
CODE	<input type="text"/>
RESULT TYPE	<input type="text" value="None"/>
SOURCE	<input type="text" value="Repository"/>
REPOSITORY	<input type="text" value="morpheus-group-create - Harker GitHub"/>
FILE PATH	<input type="text" value="main.py"/>
VERSION REF	<input type="text"/>
COMMAND ARGUMENTS	<input type="text" value="<%= cypher.read('secret/subusertoken') %>"/>
ADDITIONAL PACKAGES	<input type="text" value="requests"/> <small>Format expected is 'packageName==x.x.x packageName2==x.x.x'. Additional packages will be installed after requirements.txt if detected.</small>
PYTHON BINARY	<input type="text"/> <small>Optional override default python binary</small>

Create Operational Workflow

Morpheus Workflows pull multiple Tasks together into a logical group. There are two types of Workflows: Operational and Provisioning. Provisioning Workflows (which aren't used in this guide) are associated with an Instance at provision time and will automatically run the appropriate Tasks when the Instance reaches certain phases of its lifecycle, such as during provisioning, teardown, startup, or shutdown. In our case, we need an Operational Workflow which are run on-demand when needed or on timed intervals.

Navigate to Library > Automation > Workflows and click + *ADD*. Set the following configurations:

- **NAME:** Subtenant Group Create
- **PLATFORM:** All
- **TASKS:** Subtenant Group Create - Task
- **INPUTS:** Find Subtenant Group Name, Subtenant Group Code, and Subtenant Group Location using the typeahead field. I think it makes most sense to order them with Name first but it's not required

Once done, click *SAVE CHANGES*

EDIT WORKFLOW
✕

NAME

DESCRIPTION

PLATFORM

Tasks

1

Subtenant Group Create - Task

☰ ✕

Inputs

Subtenant Group Name (text)
☰ ✕

Subtenant Group Code (text)
☰ ✕

Subtenant Group Location (text)
☰ ✕

Execute Workflow

After saving the new Operational Workflow, you'll be left on the Workflows list page. From here we can try the new Workflow. Click on the gear icon at the far-right part of the row, then click Execute. Our three Inputs should be visible. We can see the Name field is marked as required (as we configured) and our help block text underneath the optional Inputs. Complete each field (Context Type can be left at None) and click *EXECUTE*.

EXECUTE WORKFLOW?
✕

EXECUTE WORKFLOW: SUBTENANT GROUP CREATE

Workflow Configuration

SUBTENANT GROUP NAME

SUBTENANT GROUP CODE
Optional Code Value

SUBTENANT GROUP LOCATION
Optional Location Value

Execution Config

CONTEXT TYPE ▾

EXECUTE

Once again login in as or impersonate your service account user within the Subtenant. Navigate to Infrastructure > Groups and inspect the list. You should now see that our Group has been created here.

Operations
Provisioning
Library
Infrastructure
Backups
Monitoring
Tools
Administration

Groups
 Clouds
 Clusters
 Compute
 Network
 Load Balancers
 Storage
 Trust
 Boot

GROUPS

+ CREATE

NAME	LOCATION	CLOUDS	INSTANCES	HOSTS	VMS	BARE METAL
Dev Group	us-east-1	0	0	0	0	0
Public Clouds	eu-east-1	0	0	0	0	0

5.13.7 Creating XaaS Instance Types with Morpheus

Morpheus version 5.4.2 and higher includes the ability to build and provision XaaS Instance Types. These are non-VM backed Layouts which allow users to represent anything as an Instance. This includes the ability to utilize Provisioning Workflows to manage all phases of the Instance lifecycles consistently and automatically. XaaS Instance Types can also be tied into service plans to track costs or bill customers.

In this specific example, we will create an XaaS Instance Type to manage the lifecycle of folders within a Dropbox account. Users will be able to visit the Morpheus Catalog and have the folder created by entering a name for the folder and clicking a single button. We'll see how each folder is then represented as an Instance. With the Instance standing, users will be able to update the folder name by simply editing a custom value on the Instance. Finally, when the Instance is deleted, the folder is deleted from Dropbox as well.

In this guide, we'll use the following Morpheus constructs:

- Clouds
- Inputs

- Tasks
- Workflows
- Instance Types
- Layouts
- Catalog Items
- Instances

Note: XaaS Instances are associated with Morpheus-type Clouds. Morpheus Clouds are generic cloud wrappers that can be used to contain manually-managed servers, VMs, and (in this case) non-VM based resources. In this guide we will go through the process of creating a Morpheus-type Cloud which can be used to hold your XaaS resources.

Create Cypher

Cypher is a secure key/value store in Morpheus. Using Cypher, we can securely store passwords and other secret values (such as API keys) which can then be called into automation Tasks and templates. Here we will store the Dropbox API token as a Cypher entry. Creating Dropbox developer accounts and obtaining API keys goes beyond the scope of this guide but Dropbox developer tools are well-documented if you want to try this out for yourself. In the Morpheus UI, go to Tools > Cypher and click + *ADD*.

There are a number of different types of Cypher keys, which are useful in different contexts. Here we'll use the "secret" type which allows us to enter some known value which can be securely accessed later. Enter the following:

- **KEY:** secret/dropboxtoken
- **VALUE:** Enter Dropbox API token here
- **LEASE:** 0 (Lease time is given in seconds, with "0" being unlimited)

Click *SAVE CHANGES*

ADD KEY

KEY

Keys can have different behaviors depending on the specified mountpoint.

Available Mountpoints:

- password - Generates a secure password of specified character length in the key pattern (or 15) with symbols, numbers, upper case, and lower case letters (i.e. password/15/mypass generates a 15 character password).
- tfvars - This is a module to store a tfvars file for terraform.
- secret - This is the standard secret module that stores a key/value in encrypted form.
- uuid - Returns a new UUID by key name when requested and stores the generated UUID by key name for a given lease timeout period.
- key - Generates a Base 64 encoded AES Key of specified bit length in the key pattern (i.e. key/128/mykey generates a 128-bit key)

VALUE

1	xxx
---	-----

LEASE

Quick S Time Reference:

- Day: 86400
- Week: 604800
- Month (30 days): 2592000
- Year: 31536000
- Unlimited: 0

One of my Tasks will also send a request to the Morpheus API so I'll create a second Cypher entry to store a Morpheus API key:

- **KEY:** secret/morphapitoken
- **VALUE:** Enter Dropbox API token here
- **LEASE:** 0 (Lease time is given in seconds, with "0" being unlimited)

Click *SAVE CHANGES*

Create Cloud

XaaS Instances are associated with Morpheus-type Clouds. Morpheus Clouds are generic cloud wrappers that can be used to contain manually-managed servers, VMs, and (in this case) non-VM based resources. For this example, I'm creating a Cloud just to organize Dropbox folders. Navigate to Infrastructure > Clouds and click + *ADD*. The only required steps to create the Cloud are to give it a name and associate it with a Group. I've given mine a generic name but you could opt to be more targeted with your naming if you will use many Morpheus-type Clouds to manage other types of resources.

- **NAME:** MorphCloud

EDIT CLOUD
×

NAME

CODE

LOCATION

ENABLED

AUTOMATICALLY POWER ON VMS

Details

INVENTORY EXISTING INSTANCES

▶ Advanced Options

▶ Provisioning Command

[SAVE CHANGES](#)

Create Inputs

Inputs are custom input fields which can be added to Layouts, Instance Types, and other constructs in Morpheus. The input can be consumed as variables within templates and scripts. We'll create two Inputs in this case, one to allow the user to enter a name for their folder on provisioning and another which will be visible when editing the Instance to allow the user to rename their folder.

In Morpheus UI, navigate to Library > Options > Inputs. Click + *ADD*. Complete the following fields:

- **NAME:** DropBox Folder Name (The name for the Input object in Morpheus)
- **FIELD NAME:** dbfoldername (The internal property which the input value is assigned to)
- **TYPE:** Text (The input type, in this case an open text field for the user)
- **SHOW ON EDIT:** Checked (When checked, this Option is visible when editing an Instance)
- **EDITABLE:** Checked (When checked, this Option is editable in addition to being visible while editing the Instance)
- **LABEL:** DropBox Folder Name (The label the user sees next to the input field)

Once done, click *SAVE CHANGES*

EDIT INPUT ×

NAME

DESCRIPTION

FIELD NAME
This is the input fieldName property that the value gets assigned to.

EXPORT AS TAG

DEPENDENT FIELD
A fieldName that will trigger reloading this option list

VISIBILITY FIELD

Next, configure a second Input with the following attributes:

- **NAME:** DropBox Folder New Name
- **FIELD NAME:** dbfoldernewname
- **TYPE:** Text
- **SHOW ON EDIT:** Checked
- **EDITABLE:** Checked
- **LABEL:**

Click *SAVE CHANGES*

Create Tasks

Tasks, in Morpheus, are individual automation scripts. They can be pieced together into Workflows (as we'll see later) to create more comprehensive automation packages. They can be written in a number of different languages (including BASH, Powershell, Python, Javascript, and more) or to accomplish specific functions like restarting a server or sending an email notification. In this case, I'll write the Task configuration directly into the Morpheus Tasks. However, Tasks can also be sourced directly from integrated version control repositories (like Github) so you never have to copy and paste code or make manual updates when your code changes.

For this example, we need to create four Tasks. One to create the folder, one to rename the folder, one to rename the Morpheus Instance, and one to delete the folder. I've used Python Tasks to interact with the Dropbox Python SDK. I won't go into how to write the individual Tasks here but the Python SDK is well-documented if you want to try things out for yourself. The same functions could be carried out using other Task types as well.

TYPE	NAME	
python	Dropbox - Create Folder	
python	Dropbox - Delete Folder	
python	Dropbox - Rename Folder	
python	Dropbox - Reset Instance Name	

Navigate to Library > Automation > Tasks and click + *ADD*. Create a new Task with the following configuration:

- **NAME:** Dropbox - Create Folder
- **TYPE:** Python Script (Once Type is selected, available fields will be updated to those specific to the chosen type)
- **RESULT TYPE:** None
- **SOURCE:** Local (Select Repository to source your code from an integrated version control repository)
- **CONTENT:** Enter Task content here
- **COMMAND ARGUMENTS:** Optional command line arguments for the Python script. In my case, I'm passing the Dropbox API token from Cypher as a command line argument (as seen in the screenshot) and consuming it in my code using the `sys` module, which is part of the Python standard library. There are other ways to consume Cypher secrets in Python scripts as well, which are laid out in a [Knowledge Base article](#)
- **ADDITIONAL PACKAGES:** List packages used which are not part of the standard Python library

Once done, click *SAVE CHANGES*

EDIT TASK✕

NAME	<input style="width: 95%;" type="text" value="Dropbox - Create Folder"/>
CODE	<input style="width: 95%;" type="text"/>
RESULT TYPE	<input style="width: 95%;" type="text" value="None"/>
SOURCE	<input style="width: 95%;" type="text" value="Local"/>
CONTENT	<pre>import dropbox import sys TOKEN = sys.argv[1] FOLDER_NAME = morpheus['customOptions']['dbfoldername'] dbx = dropbox.Dropbox(TOKEN) dbx.files_create_folder_v2(path=f'/morpheus/{FOLDER_NAME}', autorename=False)</pre>
COMMAND ARGUMENTS	<input style="width: 95%;" type="text" value="<%= cypher.read('secret/dropboxtoken') %>"/>
ADDITIONAL PACKAGES	<input style="width: 95%;" type="text" value="dropbox"/> <small>Format expected is 'packageName==x.x.x packageName2==x.x.x'. Additional packages will be installed after requirements.txt if detected.</small>
PYTHON BINARY	<input style="width: 95%;" type="text"/> <small>Optional override default python binary</small>

The process for creating the remaining three Tasks is very similar, expand the sections below to see screenshots of the Task config, if desired:

- **Dropbox - Delete Folder**

EDIT TASK
✕

NAME

CODE

RESULT TYPE

SOURCE

CONTENT

```
import dropbox
import sys

TOKEN = sys.argv[1]
FOLDER_NAME = morpheus['customOptions']['dbfoldernewname']

dbx = dropbox.Dropbox(TOKEN)
dbx.files_delete_v2(path=f'/morpheus/{FOLDER_NAME}')
```

COMMAND ARGUMENTS

ADDITIONAL PACKAGES
Format expected is 'packageName==x.x.x packageName2==x.x.x'. Additional packages will be installed after requirements.txt if detected.

PYTHON BINARY
Optional override default python binary

• Dropbox - Rename Folder

EDIT TASK
✕

NAME

CODE

RESULT TYPE

SOURCE

CONTENT

```
import dropbox
import sys

TOKEN = sys.argv[1]
FOLDER_NAME = morpheus['customOptions']['dbfoldername']
NEW_FOLDER_NAME = morpheus['customOptions']['dbfoldernewname']

dbx = dropbox.Dropbox(TOKEN)
dbx.files_move_v2(from_path=f'/morpheus/{FOLDER_NAME}', to_path=f'/morpheus/{NEW_FOLDER_NAME}',
autorename=True)
```

COMMAND ARGUMENTS

ADDITIONAL PACKAGES
Format expected is 'packageName==x.x.x packageName2==x.x.x'. Additional packages will be installed after requirements.txt if detected.

PYTHON BINARY
Optional override default python binary

• Dropbox - Reset Instance Name

EDIT TASK
✕

NAME

CODE

RESULT TYPE

SOURCE

CONTENT

```

import requests
import sys

INSTANCE_ID = morpheus['instance']['id']
TOKEN = sys.argv[1]
FOLDER_NAME = morpheus['customOptions']['dbfoldernewname']
URL = f'https://10.0.1.21/api/instances/{INSTANCE_ID}'
HEADERS = {"Authorization": f'BEARER {TOKEN}', "Content-Type": "application/json"}
PAYLOAD = {"instance": {"name": FOLDER_NAME}}

response = requests.put(URL, json=PAYLOAD, headers=HEADERS, verify=False)
print('Successfully reset Instance name')

```

COMMAND ARGUMENTS

ADDITIONAL PACKAGES

Format expected is 'packageName==x.x.x packageName2==x.x.x'. Additional packages will be installed after requirements.txt if detected.

PYTHON BINARY

Optional override default python binary

Create Provisioning Workflow

Morpheus Workflows pull multiple Tasks together into a logical group. There are two types of Workflows: Operational and Provisioning. We won't use any Operational Workflows here but these Workflows can be run on-demand as needed or set to run on a recurring time schedule (like a cronjob). Provisioning Workflows are associated with an Instance at provision time and will automatically run the appropriate Tasks when the Instance reaches certain phases of its lifecycle, such as during provisioning, teardown, startup, or shutdown. In our case, we need the following to occurring during the Instance lifecycle:

- At the provisioning phase, we want a folder to be created
- At the reconfigure phase (when the Instance is edited), we want the folder to be renamed and the Instance name to be updated
- At the teardown phase (when the Instance is deleted), we want the folder to be deleted

Navigate to Library > Automation > Workflows and click + *ADD*. Set the following configurations:

- **NAME:** XaaS - Dropbox
- **PLATFORM:** All
- **TASKS - Provision:** Dropbox - Create Folder
- **TASKS - Reconfigure:** Dropbox - Rename Folder; Dropbox - Reset Instance Name
- **TASKS - Teardown:** Dropbox - Delete Folder

Once done, click *SAVE CHANGES*

Tasks

- ▶ Configuration
- ▶ Pre Provision
- ▼ Provision 1

1 python Dropbox - Create Folder ☰ ×
- ▶ Post Provision
- ▶ Start Service
- ▶ Stop Service
- ▶ Pre Deploy
- ▶ Deploy
- ▼ Reconfigure 2

1 python Dropbox - Rename Folder ☰ ×

2 python Dropbox - Reset Instance Name ☰ ×
- ▼ Teardown 1

1 python Dropbox - Delete Folder ☰ ×

Create Instance Type

With the Workflow and Inputs complete, we're ready to put them all together into a custom Instance Type for our Morpheus Library. From this, we'll create a catalog item that our users can order in a later step.

Navigate to Library > Blueprints > Instance Types and click + *ADD*. Enter the following configurations:

- **NAME:** XaaS - Dropbox
- **CODE:** xaas
- **CATEGORY:** Utility
- **ICON:** If desired, search the file system on your local computer for a Dropbox logo icon for easier identification of this Instance Type at provision time
- **ENVIRONMENT PREFIX:** XAAS

EDIT INSTANCE TYPE ✕

NAME

CODE
Useful shortcode for provisioning naming schemes and export reference.

DESCRIPTION

CATEGORY

ICON 

Suggested Dimensions: 150 x 51

Click *SAVE CHANGES*.

Create Layout

After creating the Instance Type, click into it and then click + *ADD LAYOUT* from the Instance Type Detail Page. A Layout specifies the technology the Instance will run on, in this case Workflow. It's possible to have multiple Layouts associated with an Instance Type which can be selected depending on the chosen Cloud the user might be provisioning on (when dealing with VM-based Instance Types). Configure the Layout as follows:

- **NAME:** XaaS - Dropbox
- **VERSION:** Latest
- **CREATABLE:** Checked (If unchecked, this Layout won't be an available option at provision time)
- **TECHNOLOGY:** Workflow
- **WORKFLOW:** Select the Workflow we've just created, "XaaS - Dropbox". By selecting this, all Instances provisioned with this Layout will automatically have our chosen Tasks run during specific Instance lifecycle phases
- **INPUTS:** Search and find the two custom Inputs we created earlier, "DropBox Folder Name" and "DropBox Folder New Name"

EDIT LAYOUT
✕

NAME

VERSION

DESCRIPTION

CREATABLE

TECHNOLOGY

MINIMUM MEMORY

This will override any memory requirement set on the virtual image

WORKFLOW

SUPPORTS CONVERT TO MANAGED

ENABLE SCALING (HORIZONTAL)

ENVIRONMENT VARIABLES

Inputs

DropBox Folder Name <small>(text)</small>	☰ ✕
DropBox Folder New Name <small>(text)</small>	☰ ✕

Create Catalog Item

Catalog Items offer a simplified provisioning process. Administrators can create Catalog Items using existing Instance Types, App Blueprints, or Workflows. Most or all of the provisioning options can be pre-selected leaving fewer decisions up to the user and allowing them to easily create what they need. In this case, we'll create one based on the Instance Type that was made in the previous sections. Navigate to Library > Blueprints > Catalog Items and click + *ADD*, then Instance. Configure the following:

- **NAME:** XaaS - Dropbox Folder
- **ENABLED:** Checked (If unchecked, this Catalog Item will not be displayed in the provisioning catalog for users)
- **LOGO:** If desired, browse your local disk for a Dropbox logo to make this Catalog Item easily recognizable

Then, click *CONFIGURATION WIZARD*. On the “TYPE” tab, search for the Instance Type we created and click *NEXT*. On the “Group” tab, select the Morpheus-type Cloud and enter any name (we'll override this name value later so it's dynamic based on user input). Click *NEXT*. On the “CONFIGURE” tab, the Layout and Plan fields should default to acceptable values. Enter anything for the “Dropbox Folder Name”, we will also update this to be dynamic in the next step. Click through the final two tabs, there's no need to attach our Workflow on the “AUTOMATION TAB” since we already have it on our Layout. Finally, click *COMPLETE*.

Once finished, you'll see the JSON configuration map for the Instance loaded into the “CONFIG” field. Update the “dbfoldername” Input value and the Instance name value to dynamically take on the user-entered value on the “Dropbox Folder Name” field as I've done in my screenshot below:

EDIT CATALOG ITEM✕

NAME

DESCRIPTION

ENABLED

FEATURED

LOGO

Dropbox

 ▼

CONFIG CONFIGURATION WIZARD

```
"createUser": true,
"isEC2": false,
"isVpcSelectable": true,
"customOptions": {
  "dbfoldername": "<%= customOptions.dbfoldername %>",
  "dbfoldernewname": null
}
},
"name": "<%= customOptions.dbfoldername %>",
"volumes": [
  {
    "rootVolume": true,
    "name": "root",
    "maxStorage": 0,
    "size": 0,
```

Near the bottom of the modal window, search for and attach the “DropBox Folder Name” Input. This will be the only input our users need to make to order their folder. Then, click *SAVE CHANGES*

Ordering Catalog Item

At this point, the configuration steps are completed. As a test, we can order a folder from the provisioning Catalog. Navigate to Provisioning > Catalog and click “Order” on our “XaaS - Dropbox Folder” item. We need only provide a name for our folder and click “Order Now”.

Catalog > Order



XaaS - Dropbox Folder

DropBox Folder Name

Price: No pricing configured

[ADD TO ORDER](#) [ORDER NOW](#)

If we now head to Provisioning > Instances, we can see a new Instance entry has been created for our Dropbox folder. Note that the Instance is named for our folder name exactly as we configured earlier.

INSTANCE COUNT:  1

INSTANCE STATUS:

Running	1
Stopped	0

MAX CPU:  0%

STORAGE:  0%

MEMORY:  0%

INSTANCES

my All Groups All Clouds All Statuses [+ ADD](#) [ACTIONS v](#) [Settings](#)

NAME	SUMMARY	LOCATION	STATS
<input type="checkbox"/> my_folder	Version: Latest Virtual Machines: 0	Group: Harker Personal Clouds Clouds: MorphCloud	    

Taking a look in the Dropbox web console, we can also see a folder has been created just as we'd expect.



Dropbox / morpheus

- Home
- All files
 - morpheus
 - my_folder
- Recents
- Starred
- Photos New
- Shared
- File requests
- Deleted files

[Upload](#) [Create](#) [More](#)

Name	Modified	Who can access
my_folder	--	Only you

Managing and Deleting Instances

Back in Morpheus, we can take a look at the Instance detail page (Provisioning > Instances > Specific Instance) and perform some Day 2 actions. By clicking *EDIT*, we can update Instance details. When Input values are updated, Morpheus will automatically trigger reconfigure actions on our Instance. In our case, we've configured it to update the folder name in Dropbox and update the Instance name in Morpheus for easier identification. As you can see in the screen shot, I'm providing a new folder name value:

EDIT INSTANCE ✕

NAME

DESCRIPTION

ENVIRONMENT ▾

GROUP ▾

TAGS +

LABELS

OWNER

DROPBOX FOLDER NAME

DROPBOX FOLDER NEW NAME
This is a new name for your Dropbox folder to be entered when editing the DropBox folder Instance

As expected, our Instance name is updated and the folder is renamed on the Dropbox side:

Instances > morpheus_stuff

morpheus_stuff

EDIT ACTIONS DELETE

Plan: *XaaS Workflow Test

			100.00%	N/A			
STATUS	HEALTH	LAST BACKUP	AVAILABILITY	RESPONSE TIME	MAX CPU	MEMORY	STORAGE

INFO



Group: [Harker Personal Clouds](#) Cloud: MorphCloud Date Created: 01/26/2022 03:31 PM
 Owner: Alex Harker Layout: XaaS - Dropbox Version: Latest
 Provision Time: 7 seconds

Summary Storage Network Logs Environment Scale History Wiki

0.00 GiB

Dropbox / morpheus

Upload Create

Name	Modified	Who can access
morpheus_stuff	--	Only you

Finally, I'll delete the Instance. This has the effect of deleting the Instance object out of Morpheus and triggering our Teardown-phase action which deletes the folder from Dropbox:

Dropbox / morpheus

Upload Create

This folder is empty
 Drag and drop files onto this window to upload.

5.13.8 Morpheus Virtual Desktops

VDI Pools Overview

The VDI Pools section of Morpheus Tools provides a management area for defining VDI Pools and VDI Apps that a user can consume within the [Virtual Desktop Persona](#).

Pools can be either persistent or non-persistent and have various controls pertaining to idle pools and minimum or maximum sizes. The idea here is to make sure a server is always quickly available to accommodate user demand.

Morpheus leverages its Instance Types concept for provisioning servers within the VDI Pool. All the options available during Instance provisioning is available for setting the base server configuration. This includes Workflows, domain joins, tagging, image selection and more.

A timeout setting can also be applied to release pool allocations from a user once they have disconnected their session. For non-persistent pools, a good recommendation is ten minutes whereas, for a persistent pool, a sensible recommendation would be around one hour.

Pool behavior changes depending on the pool type. In a non-persistent pool, when a timeout period expires, the VM is destroyed and a new one is allocated for use. This functionality will change based on the cloud technology in a future update allowing for potential recycling of the VMs. In a Persistent pool, when the lease timeout expires, the Instance will shutdown until the user requests it again in the future. It is important to note that lease timeouts auto-extend for as long as the user is logged into or browsing any area of the Morpheus application. Once the user closes their browser or logs out of their session, the timeouts will no longer auto-extend.

Configuring Access to VDI Pools

Access to the Virtual Desktop Persona and individual VDI pools is handled through the user Role and, where applicable, Tenant Role. When creating a new Role, access is restricted to the Virtual Desktop Persona and all VDI pools by default. To grant access:

1. Navigate to the Role (Administration > Roles > Selected Role)
2. Access the Personas tab
3. Toggle the Virtual Desktop permission to “Full” or “None”
4. Access the VDI Pool Access tab
5. Toggle access to selected VDI pools to “Full” or “None”, you can also toggle permission on all pools to “Full” or “None” with the Global Access selection
6. Role changes are saved automatically, there is no need to manually save

Additionally, users should have a Linux and/or Windows username and password configured in their user profiles in order for virtual desktop login to be as seamless as possible. User profiles are accessed by clicking on the user’s name in the upper-right corner of the application windows and clicking USER SETTINGS. The section to enter Windows and Linux account credentials is in the right column of the page.

Creating VDI Templates

Note: The following guide focuses on VMware and Windows but is applicable to most cloud environments. Morpheus also supports Linux virtual desktops.

1. Create a thin-provisioned VM in the VMware vCenter console. It's recommended you allocate at least 50 GB of storage, at least 2 vCPU and at least 8 GB RAM on the template. Smaller VMs can be deployed from this template later.
2. Install a Windows operating system, there is no need to supply a license during deployment.
3. Supply the initial username and password
4. Install any updates, applications or optimizations. See the next section for recommended optimizations for the most performant virtual desktop experience
5. Shutdown the VM and convert to a template. Optionally, you can also use the Linked Clones (VMware) process which is described in a later section

Suggested Optimizations

Reducing display and input delays is key to providing the best virtual desktop experience for the user. Consider the following optimizations for VDI desktops and servers:

- Disable desktop wallpapers
- Implement Roaming Profiles
- Enforce WDDM remote display driver
- Re-enable local Administrator
- Delete the initial created user profile
- Clean up any unneeded installer packages

Additionally, there are a number of OS optimization tools available on the Internet which are specific to VDI use cases.

Linked Clones

Linked Clones are a feature of VMware which references snapshots of a VM to deploy from. This adds the advantage of quicker clone times and the ability to more easily share small modifications to a file system. Morpheus supports Linked Clones but recommends them for VDI workloads only.

Note: Linked Clones are not templates but rather powered down VMs.

1. Locate the VM you desire to have the Linked Clone in Morpheus. If it's not currently managed by Morpheus, navigate to the appropriate Cloud (Infrastructure > Clouds), find the VM on the "VMs" tab, and click "Convert to Managed" from the ACTIONS menu
2. In the CONVERT TO INSTANCE modal, select "No Agent Install" in the AGENT field
3. If snapshots are already on the VM, these will now be synced by Morpheus. If you have not yet created a snapshot, do so in the vCenter console (and refresh the Cloud integration in Morpheus afterward) or from the ACTIONS menu in Morpheus itself. Be sure to take a snapshot of a powered-off VM and give the snapshot a name that will be identifiable for administrators

4. From the Instance detail page in Morpheus, navigate to the Backups tab to find the snapshot
5. Select “More” and create the Linked Clone
6. The Linked Clone will now appear in the Morpheus Virtual Image repository (Library > Virtual Images), ready to use with your custom Layouts

Note: You should modify the Virtual Image to “Force Guest Customization” unless you `sysprep` your VM at shutdown time

Creating or Editing a VDI Pool

VDI pools are configured from the Tools menu (VDI Pools selection). The following information is displayed in the VDI pools list view, bear in mind some fields may be hidden depending on how you’ve configured your VDI pools list view (gear icon):

- **TYPE:** An icon indicating the machine type associated with the pool. Morpheus includes many logos out of the box and also allows users to set their own custom icons
- **NAME:** The friendly name given to the VDI pool
- **PERSISTENT:** A check mark will appear when the VDI pool is configured for persistent virtual desktops
- **ENABLED:** A check mark will appear when the VDI pool is enabled and visible to users whose Role permissions allow them access
- **POOL USAGE:** A graph representing the usage of the VDI pool. The total length of the bar represents the maximum pool size based on the configuration. Green segments represent available virtual desktops, blue segments represent reserved virtual desktops, yellow segments represent virtual desktops which are being prepared, and gray segments represent additional pool capacity which could be made available depending on how many virtual desktops are currently reserved and how many idle machines you’ve configured the pool to keep available
- **DESCRIPTION:** A description of the virtual desktop type, if provided

VDI POOLS

TYPE	NAME	PERSISTENT	ENABLED	POOL USAGE	
	Suse	✓	✓		
	Windows		✓		

Create a VDI pool by selecting **+ ADD** from the VDI Pools tab or edit an existing one by clicking on the pencil icon from the appropriate row. Configure the following, fields containing a vertical blue bar along the left edge are required:

- **NAME:** A friendly name for the VDI pool in Morpheus
- **DESCRIPTION:** A description of the virtual desktop type
- **MIN IDLE:** The minimum number of virtual desktops that should remain idle and ready
- **INITIAL POOL SIZE:** The number of virtual desktops that will be prepared when the pool is created or enabled

- **MAX IDLE:** The maximum number of virtual desktops that remain idle and ready. Machines will be shut down as necessary when this number is exceeded due to users vacating their machines
 - **MAX SIZE:** The total number of virtual desktops this pool can have. Additional users will not be able to access machines once this number is reached
 - **LEASE TIMEOUT (MINUTES):** The user lease time on a virtual desktop they've reserved. The lease will continue to auto-renew itself as long as the user is logged into Morpheus. Once the user has logged out and the lease timeout period has expired, the machine will be released as appropriate based on your configuration
 - **PERSISTENT:** Pools with persistent virtual desktops will reserve a machine for each user in order to preserve settings, installed applications, work files and more. Machines in persistent pools will be shut down rather than destroyed when they are no longer in use
 - **RECYCLABLE:** When enabled, the VDI Instance will revert back to a snapshot and become available once again after the user has logged out and the VDI session has expired. This behavior will not apply to VDI pools which are also configured to be persistent because in that configuration the Instance is merely stopped and saved for the user's next session. This feature is currently only available for Cloud types which support snapshot management (VMware, Nutanix, and vCD)
 - **ALLOW COPY** Enables or disables the ability for the VDI user to copy contents from the VDI instance to the local clipboard
 - **ALLOW PRINTER** When enabled, users local system printers can be targeted from the VDI Instance
 - **ALLOW HYPERVISOR CONSOLE:** When checked, native cloud console will be enabled (if available) rather than using Morpheus-native RDP/SSH capability
 - **AUTO CREATE LOCAL USER UPON RESERVATION:** When marked, the user configured in Morpheus user settings will be created when the machine is initially accessed. If unchecked or if there is no user configured in Morpheus user settings, ensure the machine is joining a domain or there is a known user on the machine image in order to allow access
 - **ENABLED:** When marked, the initial pool size will begin to deploy once the VDI pool is saved. The icon for this desktop environment will also be presented to Virtual Desktop Persona users
 - **CONFIGURE:** Click this button to configure the deployment configuration each system will use. The wizard is identical to the Instance provisioning wizard meaning all available Instance Types, Workflows, and more are available to virtual desktop machine creation. Consult the steps above to see an example VDI image prep walkthrough
 - **LOGO:** Upload or select a logo to represent the virtual desktop type to users
 - **VDI APPS:** Optionally select one or more frequently-used applications the user can launch directly. Users will also have the option to launch into the desktop
 - **VDI GATEWAY** Select a configure VDI Gateway for VDI sessions to be redirected to. VDI sessions will be redirected to the gateway when a gateway is specified.
- Guest Console SSH Tunnel (optional)** A Jump Host can be configured for VDI session connections. Morpheus will tunnel through the Jump Host when connecting Guest Console sessions for VDI. This is not applicable for Hypervisor Console connections.
- **GUEST CONSOLE JUMP HOST** Jump Host IP address or hostname used to connect to the Jump Host for Guest Console sessions to VDI Instances
 - **GUEST CONSOLE JUMP USERNAME** Jump Host Username used to connect to the Jump Host for Guest Console sessions to VDI Instances
 - **GUEST CONSOLE JUMP PORT** Jump Host Port used to connect to the Jump Host for Guest Console sessions to VDI Instances

- **GUEST CONSOLE JUMP PASSWORD** Jump Host Password used to connect to the Jump Host for Guest Console sessions to VDI Instances (optional if key specified)
- **GUEST CONSOLE KEYPAIR** Jump Host SSH Key used to connect to the Jump Host for Guest Console sessions to VDI Instances (optional if password specified)

Note: A Guest Console Keypair included here must be a local keypair, not a synced keypair.

EDIT VDI POOL

NAME: Windows

DESCRIPTION: Windows 10 Desktop

MIN IDLE: 1
Sets the minimum number of idle instances on standby in the pool. The pool will always try to maintain this number of available instances on standby.

INITIAL POOL SIZE: 1
The initial size of the pool to be allocated on creation.

MAX IDLE: 2
Sets the maximum number of idle instances on standby in the pool. If the number of idle instances supersedes this, the pool will start removing instances.

MAX SIZE: 2
Max limit on number of allocations and instances within the pool.

LEASETIMEOUT (MINUTES): 60

Creating or Editing a VDI Apps

VDI Apps allow users to launch directly into commonly-used apps rather than the OS desktop. Currently, VDI Apps only work with RDP Windows Instances, taking advantage of native Windows Remote Application functionality. Natively-hosted remote desktop applications can only be presented from Windows 10 Enterprise and Education. Other versions of Windows 10 can present remote applications using the procedure below:

1. Open the Windows Registry Editor
2. Locate the following entry: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Terminal Server\TSAppAllowList`
3. Navigate to `fDisabledAllowList` and set its value to “1” in the right-hand pane
4. Add a new key under `TSAppAllowList` and name it “Applications”
5. Add a new key under “Applications” using any name you’d like
6. Within this new key, create two new string values, one called “Name” and one called “Path”
7. The string value for “Name” should describe the application (ex. “Notepad”)
8. The string value for “Path” should be the absolute path to the executable for that application (ex. “C:\Windows\System32\notepad.exe”)

VDI Apps are created by selecting + *ADD* from the VDI Apps tab or edit an existing one by clicking on the pencil icon from the appropriate row. Configure the following, fields containing a vertical blue bar along the left edge are required:

- **NAME:** A friendly name for the VDI App in Morpheus
- **DESCRIPTION:** A description of the virtual app type

- **LAUNCH PREFIX:** A reference to the remote app registry prepended with two pipes (||). For example, we might create a registry “Chrome” for a Chrome browser VDI App and the associated launch prefix would be “||Chrome”
- **LOGO:** Upload or select a logo to represent the virtual app type to users

VDI Gateways

The Morpheus Worker is a light weight distributed worker daemon as well as a scalable VDI Gateway. Currently, the features center around VDI Gateway but will expand to support full plugin workloads as well as agent relay capabilities.

Adding VDI Gateways to Morpheus

VDI Gateways can be linked to a Morpheus appliance and then used in VDI Pool configurations. VDI sessions will be redirected to configured gateways instead of the Morpheus appliance when a VDI Gateway is specified for a VDI Pool.

Note: A VDI Gateway is a separate VM or container Instance used to route users to VDI Instances. The Morpheus VDI Gateway section is for configuring a connection to a VDI Gateway, not creating the gateway Instance itself.

- **NAME** Specify a name for the VDI Gateway in Morpheus. Note that the VDI Gateway Name is not used when connecting to the gateway
- **DESCRIPTION** Specify a description for the VDI Gateway in Morpheus. (optional)
- **GATEWAY URL** The url of the VDI Gateway. This url is used to connect to the gateway, and should match the the worker url of the VDI Gateway.

Upon creation, the VDI Gateway record will produce an **API KEY**. This **API KEY** needs to be specified in the `morpheus-worker.rb` file on the API Gateway itself under `worker['apikey'] = '$API_KEY'`. Once the gateway object is created you will need to configure it as the default gateway in Morpheus global settings (Administration > Settings > Appliance). Scroll down to the “Default Console Gateway” setting and select the gateway object you’ve just created. Continue on to the next section to actually install the gateway and configure it with your API key.

VDI Gateway VM Install

A VDI Gateway VM is installed and configured similarly to a Morpheus appliance via `rpm` or `deb` package.

Note: VDI Gateway Package URLs are available at <https://morpheushub.com> in the downloads section.

Requirements

Table 22: **Supported VDI Gateway Operating Systems**

OS	Version(s)
Amazon Linux	2
CentOS	7.x, 8.x
Debian	9, 10, 11
RHEL	7.x, 8.x
SUSE, SLES	12
Ubuntu	16.04, 18.04, 20.04

- **Memory:** 4 GB RAM minimum recommended for default installations supporting up to 20 concurrent sessions. Add 50 MB RAM per additional concurrent session
 - **Storage:** 10 GB storage minimum recommended. Storage is required for VDI Gateway Packages and log files
 - **CPU:** 4-core minimum recommended
 - Network connectivity to and from Morpheus appliance and from users to the VDI Gateway over TCP 443 (HTTPS)
 - Superuser privileges via the `sudo` command for the user installing the Morpheus VDI Gateway package
 - Access to base `yum` or `apt` repos. Access to Optional RPM repos may be required for RPM distros
1. Download the target distro & version package for installation in a directory of your choosing. The package can be removed after successful installation.

```
wget https://downloads.morpheusdata.com/path/to/morpheus-worker-$version.distro
```

2. Validate the package checksum matches source checksums. For example:

```
sha256sum morpheus-worker-$version.distro
```

3. Next install the package using your selected distribution's package installation command and your preferred opts. Example, for RPM:

rpm:

```
sudo rpm -ihv morpheus-worker-$version.$distro

Preparing...                               ##### [100%]
Updating / installing...
 1:morpheus-worker-5.3.1-1.$distro         ##### [100%]
Thank you for installing Morpheus Worker!
Configure and start the Worker by running the following command:

sudo morpheus-worker-ctl reconfigure
```

4. Configure the gateway by editing `/etc/morpheus/morpheus-worker.rb` and updating the following:

```
worker_url 'https://gateway_worker_url' # This is the gateway URL the |morpheus|
↪appliance can resolve and reach on 443
worker['appliance_url'] = 'https://morpheus_appliance_url' # The resolvable URL or
↪IP address of |morpheus| appliance which the gateway can reach on port 443
worker['apikey'] = 'API KEY FOR THIS GATEWAY' # VDI Gateway API Key generated from
↪|morpheus| Appliance VDI Pools > VDI Gateways configuraiton
```

Note: By default the `worker_url` uses the machine's hostname, ie `https://your_machine_name`. The default `worker_url` value can be changed by editing `/etc/morpheus/morpheus-worker.rb` and changing the value of `worker_url`. Additional appliance configuration options are available below.

5. After all configuration options have been set, run `sudo morpheus-worker-ctl reconfigure` to install and configure the worker, nginx and guacd services:

```
sudo morpheus-worker-ctl reconfigure
```

The worker reconfigure process will install and configure the worker, nginx and guacd services and dependencies.

Note: Configuration options can be updated after the initial reconfigure by editing `/etc/morpheus/morpheus-worker.rb` and running `sudo morpheus-worker-ctl reconfigure` again.

- Once the installation is complete the morpheus worker service will automatically start and open a web socket with the specified Morpheus appliance. To monitor the startup process, run `morpheus-worker-ctl tail` to tail the logs of the worker, nginx and guacd services. Individual services can be tailed by specifying the service, for example `morpheus-worker-ctl tail worker`

VDI Gateway Docker Install

To Use VDI Gateway within a Docker container, a few pieces of information are needed.

Firstly, in Morpheus, go to **Tools > VDI Pools > VDI Gateways** and create a new VDI Gateway Record. Be sure to set the HTTPS URL as Morpheus will need to be able to redirect the user's browser to that page. An API Key will be generated. Make note of this as you will need it later.

Now Simply run with:

```
docker run -d -p 8443:8443 -e MORPHEUS_SELF_SIGNED=true -e MORPHEUS_KEY=[apiKey] -e MORPHEUS_URL=https://my.morpheusAppliance.url morpheusdata/morpheus-worker:latest`
```

This will setup an HTTPS self-signed exposed port on 8443 for the vdi gateway. It is highly recommended to use valid certificates on your VDI Gateways. It could be terminated at the VIP or a p12 SSL File can be used and configured for the container.

If the docker entrypoint detects a file at `/etc/certs/cert.p12`, SSL Will be enabled on port 8443 instead. be sure to set environment variables `MORPHEUS_SSL_ALIAS` and `MORPHEUS_SSL_PASSWORD` when using p12 files.

If you wish to run in HTTP mode and SSL terminate at the VIP, you can run the container like so:

```
docker run -d -p 8080:8080 -e MORPHEUS_SELF_SIGNED=true -e MORPHEUS_KEY=[apiKey] -e MORPHEUS_URL=https://my.morpheus.url morpheusdata/morpheus-worker:latest
```

VDI Gateway Helm Chart Installation

First, configure the Helm repository:

```
helm repo add morpheusdata https://gomorpheus.github.io/helm-charts-morpheus/
```

Next, install the Morpheus worker using `helm install`. You can specify each parameter using `--set key=value[, key=value]` arguments as in the following example:

```
helm install morpheus-worker --set replicaCount="1" morpheusdata/morpheus-worker
```

Alternatively, you can create a values YAML file and pass an argument as in the following example:

```
helm install -f values.yaml morpheus-worker morpheusdata/morpheus-worker
```

Upgrading the workers node(s) is as simple as refreshing the repo and using `helm upgrade`:

```
helm repo update
helm upgrade -f values.yaml morpheus-worker morpheusdata/morpheus-worker
```

To uninstall, use one of the following:

```
helm uninstall morpheus-worker
```

or

```
helm delete morpheus-worker --purge
```

Note: `helm delete` removes all the Kubernetes components associated with the chart and deletes the release.

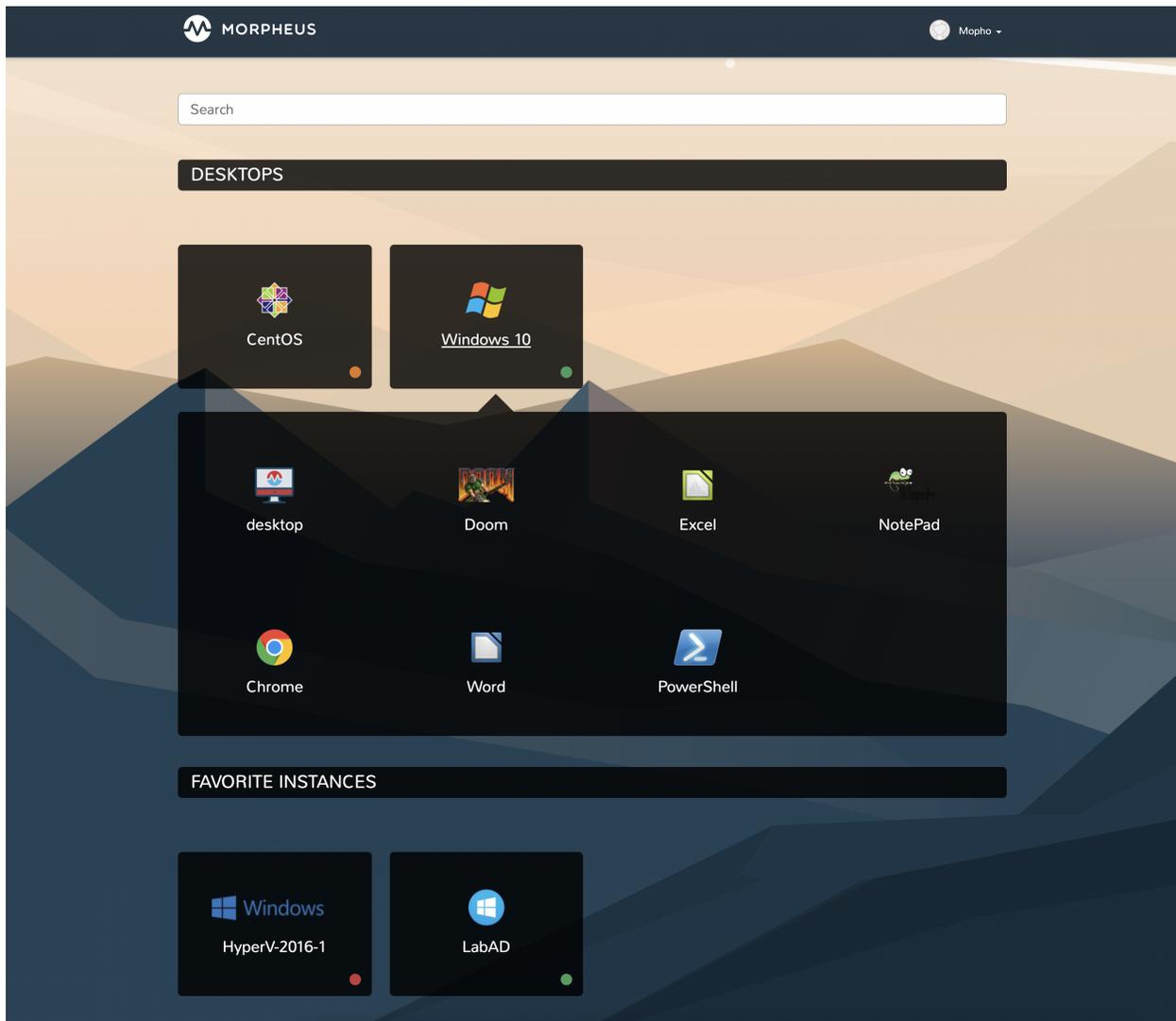
The following table lists the configurable parameters of the Sentry chart and their default values:

Parameter	Description	Default
<code>image.repository</code>	Image repository	morpheusdata/morpheus-worker
<code>image.tag</code>	Image tag. Possible values listed here.	5.3.1-4
<code>image.pullPolicy</code>	Image pull policy	IfNotPresent
<code>env.MORPHEUS_KEY</code>	API Key for Morpheus Worker	
<code>env.MORPHEUS_URL</code>	Morpheus FQDN with protocol	
<code>env.MORPHEUS_SELF_SIGNED</code>	Is Morpheus using a Self Signed Certificate	false
<code>service.type</code>	Kubernetes service type for the GUI	ClusterIP
<code>service.port</code>	Kubernetes port where the GUI is exposed	8989
<code>livenessProbe.initialDelaySeconds</code>	Initial delay (seconds) for liveness monitoring	5
<code>livenessProbe.timeoutSeconds</code>	Timeout (seconds) before health check considered unhealthy	5
<code>livenessProbe.periodSeconds</code>	Poll interval (seconds) between health checks	10
<code>livenessProbe.failureThreshold</code>	Number of failed polls before restarting service	3
<code>replicaCount</code>	Number of Replicas if AutoScaling False	1
<code>autoscaling.enabled</code>	Enable AutoScaling	false
<code>autoscaling.minReplicas</code>	Minimum number of Replicas	1
<code>autoscaling.maxReplicas</code>	Maximum number of Replicas	100
<code>autoscaling.targetCPUUtilizationPercentage</code>	CPU Threshold for AutoScaling	80
<code>autoscaling.targetMemoryUtilizationPercentage</code>	Memory Threshold for AutoScaling	
<code>ingress.enabled</code>	Enables Ingress	false
<code>ingress.annotations</code>	Ingress annotations	{}
<code>ingress.path</code>	Ingress path	/
<code>ingress.hosts</code>	Ingress accepted hostnames	chart-example.local
<code>ingress.tls</code>	Ingress TLS configuration	[]
<code>resources</code>	CPU/Memory resource requests/limits	{}
<code>nodeSelector</code>	Node labels for pod assignment	{}
<code>tolerations</code>	Toleration labels for pod assignment	[]
<code>affinity</code>	Affinity settings for pod assignment	{}

Virtual Desktop Persona Overview

The Morpheus VDI Persona provides a virtual desktop environment to grant users access to workstations and applications in a secure manner. Deploy pools of virtual machines on any supported Morpheus Cloud for users to reserve and use! Morpheus leverages open source client technologies, such as Apache Guacamole, to provide a performant and secure virtual desktop client for the end user while wrapping its frontend in a completely new framework. For information on configuring VDI pools for consumption in the Virtual Desktop Persona, see the [Tools section](#) of Morpheus docs.

Note: This is not an integration with existing VDI Pool Managers such as VMWare Horizon, Citrix VDI, or Nutanix XiFrame. It is a standalone Morpheus feature.



Key Features

- VDI Pool Management
- Virtual Desktop Persona
- RDP/SSH/VNC Console Support
- RDP Remote App Support
- Clipboard Copy/Paste
- HiDPI Support
- Auto Compression Scanning based on User Bandwidth
- Audio Playback (RDP)
- Local Printer (RDP)
- Auto-Resize
- Auto-Login based on Morpheus User Settings
- Customizable User Background

Configuring Access to the Virtual Desktop Persona

Access to the Virtual Desktop Persona and individual VDI pools is handled through the user Role and, where applicable, Tenant Role. When creating a new Role, access is restricted to the Virtual Desktop Persona and all VDI pools by default. To grant access:

1. Navigate to the Role (Administration > Roles > Selected Role)
2. Access the Personas tab
3. Toggle the Virtual Desktop permission to “Full” or “None”
4. Access the VDI Pool Access tab
5. Toggle access to selected VDI pools to “Full” or “None”, you can also toggle permission on all pools to “Full” or “None” with the Global Access selection
6. Role changes are saved automatically, there is no need to manually save

Launching a VDI Instance

VDI Instances are launched from the Virtual Desktops Persona. Depending on Role permissions, your account may default to this view or may even restrict you solely to this view. To access the Virtual Desktops Persona from the standard view or from another Persona, click on the user’s name in the upper-right corner of the application window. When available, this dropdown menu will list the standard Morpheus Persona view as well as any other Personas the user has permission to access. Click on Virtual Desktop to access the Virtual Desktop Persona.

The Virtual Desktop Persona view lists out each of the virtual desktop types they can access. Click on the desired virtual desktop type to launch it. If there are virtual apps available for any listed desktop types, they are presented in a flyout menu alongside a “desktop” option to access the base OS over an individual app. Items categorized as “Desktops” are VDI pools configured in the Tools menu of the Standard Persona. Items categorized as “Instances” are Instances favorited by the current user in the Standard Persona (if they have access and they have favorited any Instances). Clicking on an Instance tile offers quick access to the Instance console.

Important: Virtual Desktops are launched in a pop-up window. Be sure your web browser is not blocking pop-ups or create an appropriate exception for Morpheus virtual desktop pop-ups.

Changing the Virtual Desktop Persona Background

The Morpheus Virtual Desktop Persona includes default backgrounds for an elegant look out of the box. If desired, users may change this background to suit personal taste or organizational branding. This change is unique to each individual account. At this time, there is no option for appliance-wide whitelabeling for Virtual Desktop Persona backgrounds.

1. Click on the user's name in the far upper-right corner of the application window
2. Click USER SETTINGS
3. The section for "VDI Settings" is in the lower-right corner of the page
4. Mark the RESET box and click *SAVE* to reset the Virtual Desktop Persona to the default background
5. Click "Browse" and upload a local image to add a custom background
6. Click *SAVE*

5.13.9 Getting Started with Terraform Instance Types

[Terraform](#) is a common tool that allows IT administrators to map out infrastructure as code in configuration files and supports all of the popular providers used in the modern datacenter. Once configured, Terraform will plan, deploy, and manage the infrastructure as needed. Configuration files can be brought under version control so teams can easily make changes to environments. Infrastructure can also be monitored for drift and corrective action can easily be taken.

Morpheus allows users to on-board or even draft Terraform spec directly. With the configuration on-board, we can begin to piece together infrastructure constructs into the Morpheus Library as Layouts and Instance Types. With the Library items staged, users can deploy new infrastructure directly into the selected providers. Once deployed, infrastructure can be monitored for drift from within Morpheus UI. When needed, we can plan and take corrective action easily from the detail page of a Morpheus Terraform Instance.

In this section, we've discussed a high-level overview of Terraform and working with Terraform in the Morpheus context. In the next section, we'll actually onboard Terraform spec, create Library items, and deploy real infrastructure to AWS with Morpheus.

Terraform Instance Types in Action

In this example, we're going to deploy a VPC and three subnets to AWS using Terraform and Morpheus. I've created Spec Templates to onboard `.tf` configuration files which handle the AWS provider (with assume role for account flexibility), VPC creation, subnet creation, and variable injection.

We'll onboard the Terraform configuration files as modular Spec Templates, create new Instance Types with custom Layouts for the Morpheus Library, and set up Inputs to inject variable values at provision time. Once deployed, we'll take a look at the new infrastructure in Morpheus and go over the management capabilities for the new environment.

Spec Templates

Terraform configuration is stored as a Spec Template in Morpheus. You can store your configuration as one monolithic file for each Instance Type you intend to create or you can create individual Spec Templates for modular pieces which can be reused across multiple Instance Types. When added to the Layout later, we'll be able to include as many Spec Templates as we wish which enables us to reuse smaller modular pieces if desired.

Spec Templates are added in the Morpheus Library (Library > Templates > Spec Templates tab). We can pull in the template from some type of repository, such as through a Github integration, or write new spec directly into the New Spec Template modal. In most cases, the spec will be pre-existing and pulled in from a version-controlled repository but here I have my Terraform spec entered locally. Click + *ADD*, complete the fields as I've done in the example below and click *SAVE CHANGES*.

In the *VERSION* field at the bottom of the TF Spec Template modal, enter a Terraform version number to force that version to be used. This version is only honored if the Terraform Runtime setting (Administration > Settings > Provisioning) is set to "auto". When "manual" is selected as the Terraform Runtime setting, Morpheus will simply use the version installed on the appliance box.

Tip: When declaring variables, keep in mind that Morpheus expects users to follow Terraform best practices. For example, when a variable type is not defined, it defaults to string. See [Terraform Documentation](#) for additional resources on variable declaration.

EDIT SPEC TEMPLATE
✕

NAME

CATEGORY

TYPE

SOURCE

CONTENT

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = ">= 3.35.0"
    }
  }
}
```

VERSION

SAVE CHANGES

- **AWS Subnet by Count**

```
# This spec template creates AWS subnets based on the count requested utilizing the
↳vpc cidr provided in var.vpc_cidr variable

locals {
  bitCount = sum([tonumber(local.subnet_options.cidrMask), -tonumber(split("/", var.
↳vpc_cidr)[1])])
}

resource "aws_subnet" "main" {
  count = tonumber(var.subnetCount)
  vpc_id = aws_vpc.main.id
  cidr_block = cidrsubnet(var.vpc_cidr, local.bitCount, count.index)

  tags = merge(
    local.default_tags,
    {
      Name = "${var.vpc_name}-subnet-0${count.index}"
    }
  )
}

output "aws_subnet" {
  value = aws_subnet.main
  sensitive = true
}
```

- **AWS Terraform Default Vars**

```
variable "access_key" {
  type = string
}

variable "secret_key" {
  type = string
}

variable "subnetCount" {
  type = number
  default = "<%=customOptions.subnetCount%>"
}

variable "sensitive_thing" {
  type = string
  default = "this_var_is_sensitive"
  sensitive = true
}
```

- **AWS Provider Role Assume**

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
    }
  }
}
```

(continues on next page)

(continued from previous page)

```

    version = ">= 3.35.0"
  }
}

provider "aws" {
  region      = local.vpc_options.region
  access_key  = var.access_key
  secret_key  = var.secret_key

  assume_role {
    # The role ARN within Account B to AssumeRole into.
    role_arn = "arn:aws:iam::${local.vpc_options.aws_account}:role/
↪OrganizationAccountAccessRole"
  }
}

```

- **AWS Terraform Locals**

```

locals {
  # Common tags to be assigned to all resources
  default_tags = {
    Owner      = "<%=username%"
    Group      = "<%=groupName%"
    Management_Tool = "Terraform"
    Management_Platform = "Morpheus"
  }

  subnet_options = {
    cidrMask = "<%=customOptions.cidrMask%"
    subnetCount = "<%=customOptions.subnetCount%"
  }

  vpc_options = {
    region = "<%=customOptions.awsRegion%"
    aws_account = "<%=customOptions.awsAccount%"
  }
}

```

- **AWS VPC**

```

variable "vpc_cidr" {
  type      = string
  description = "CIDR for the the VPC"
  default = "172.16.0.0/24"
}

variable "vpc_name" {
  type      = string
  description = "Name for the VPC"
  default = "durka"
}

```

(continues on next page)

(continued from previous page)

```
resource "aws_vpc" "main" {
  cidr_block = var.vpc_cidr

  tags = merge(
    local.default_tags,
    {
      Name = var.vpc_name
    }
  )
}

output "aws_vpc" {
  value = aws_vpc.main
  sensitive = true
}
```

Note: In the AWS Terraform Locals example Spec Template above, pre-provision variables are used. Note the use of [pre-provision](#) variables to store the value for Owner and Group, among other things. See the variables section of Morpheus documentation (linked in the prior sentence) for a listing of other possible pre-provision variables and a complete map of variables which can be resolved after provisioning has completed.

Inputs and Option Lists

In order to create the Layout later in the guide, I need to create four Inputs so the user can make certain selections at provision time. I wrote my Terraform Configuration with this flexibility in mind so that the same Instance Type can be reused in different scenarios. In this particular case, I'm populating the Inputs with manual Option Lists but they can also be populated through REST calls or calls to the Morpheus API when needed.

Option Lists are created in the Library (Library) under the Option Lists tab. These are lists of items which will be used to create dropdown selections at provision time. Click + *ADD*, complete the fields as I've done in the example below and click *SAVE CHANGES*. I've created one each for the AWS account selection, region selection, and CIDR mask input.

EDIT OPTION LIST ✕

NAME

DESCRIPTION

TYPE

VISIBILITY

If the account assigned is not the master tenant, visibility will automatically change to private.

DATASET Create an initial JSON or CSV dataset to be used as the collection for this option list. It should be a list containing objects with properties 'name', and 'value'.

```
[{"name": "Acct1", "value": "1111111111"}, {"name": "Acct2", "value": "2222222222"}]
```

REAL TIME

Inputs are also created in the Library under the Inputs tab. In this case, I'm creating four Inputs. Three of them will display as dropdown selections and will be tied to one of the Option Lists we just made. The other will be a simple text input where the user can indicate the total number of subnets that should be created. Click + *ADD*, complete the fields as I've done in the example below and click *SAVE CHANGES*.

NEW OPTION TYPE ✕

NAME

DESCRIPTION

FIELD NAME

This is the input fieldName property that the value gets assigned to.

EXPORT AS TAG

DEPENDENT FIELD

A fieldName that will trigger reloading this option list

VISIBILITY FIELD

A fieldName that will trigger visibility of this option

DISPLAY VALUE ON DETAILS

TYPE

LABEL

This is the input label that shows typically to the left of a custom option.

Instance Type

At this point we're ready to create a new Instance Type. We'll give the Instance Type a name, which users will use to identify the Instance Type from the list in the provisioning wizard. We don't need to set much else in this case, most of the pieces we've created in previous steps will be associated with the Layout that we create next. The Layout will also be tied to the Instance Type we're creating now. Instance Types are also created in the Library (Library) under the Instance Types tab. Click + *ADD*, complete the fields as I've done in the example below and click *SAVE CHANGES*.

NEW INSTANCE TYPE ✕

NAME

CODE
Useful shortcode for provisioning naming schemes and export reference.

DESCRIPTION

CATEGORY

ICON

Suggested Dimensions: 150 x 51

Layout

The Layout will bring together everything we've made to this point, the Spec Templates, Inputs and the Instance Type. We can add a new one from the Instance Type detail page (Library > Blueprints > Instance Types > Selected Instance Type) by clicking + *ADD LAYOUT*. We can also create one from the Layouts section (Library > Blueprints > Layouts) by clicking + *ADD*.

First, change the `TECHNOLOGY` value to Terraform and the fields will change to allow proper configuration. Next, provide a name for your Layout. If you're creating the Layout through the Layout tab rather than from the Instance Type detail page, you'll need to identify the Instance Type the Layout goes with. Using the typeahead fields at bottom of the modal window, add our four Inputs and our five Spec Templates to the Layout. Finally, point the layout to a `TFVAR SECRET` from Morpheus Cypher if needed. You can see a screenshot of my Layout configuration below

Option Types

- AWS Account (select) ☰ ✕
- AWS Region (select) ☰ ✕
- Subnet Count (text) ☰ ✕
- CIDR Mask (select) ☰ ✕

Spec Templates

- AWS Subnet by Count ✕
- AWS Terraform Default Vars ✕
- AWS Provider Role Assume ✕
- AWS Terraform Locals ✕
- AWS VPC ✕

[SAVE CHANGES](#)

Provisioning

Now, we're ready to provision new infrastructure into AWS using Morpheus and Terraform. Just like any other Instance Type, we begin from the Instances list page (Provisioning > Instances) and click + *ADD*. Select the Instance Type we've just created and move on to the **GROUP** tab of the wizard. Here you'll give the new instance a name and select your Group and Cloud. Once finished, you'll move on to the **CONFIGURE** tab where we'll see the Inputs we created and associated with the Layout. Once finished with this tab, step through the rest of the wizard to complete the process. You can see the options I've selected for this configuration in the image below.

CREATE INSTANCE ✕

TYPE > GROUP > **CONFIGURE** > AUTOMATION > REVIEW

Configuration Options

LAYOUT

PLAN
Cores: 1

SELECT ACCOUNT

SUBNET CIDR
Bitmask for the subnet CIDR

SELECT REGION

SUBNET COUNT

Review the New Instance

After completing the wizard, from the History tab of the Instance detail page users can review the Terraform plan being executed and see the output while the resources are still being provisioned.

The screenshot displays the Morpheus web interface. At the top, there is a navigation bar with the Morpheus logo, a search bar, and a user profile. Below this is a secondary navigation bar with tabs for Operations, Provisioning (selected), Infrastructure, Backups, Logs, Monitoring, Tools, and Administration. Under the Provisioning tab, there are sub-tabs for Instances, Apps, Blueprints, Jobs, Automation, Virtual Images, Library, Deployments, and Service Mesh. The main content area shows a breadcrumb trail: VPC > main > vpc. Below this is another set of tabs: Summary, State, Storage, Network, Logs, Environment, History (selected), and Wiki. A search bar and an 'EXPAND ALL' button are visible. The main content area shows a list of provisioning tasks. The first task is 'provisioning - provision' with a green checkmark, a plus icon, and a dropdown arrow. Below it, details are shown: Start Date: 04/29/2021 10:24 AM, Duration: 44s, User: Sean Jabro. A sub-task 'demo - prepare resources' is also shown with a green checkmark, Start Date: 04/29/2021 10:24 AM, and Duration: 8s. Below the sub-task, there is a text box containing the following information: 'An execution plan has been generated and is shown below. Resource actions are indicated with the following symbols: + create. Terraform will perform the following actions:'.

Once the provisioning process is complete, head to the State tab. Here we can see and link through to the associated Spec Templates. If needed, you can also edit the configuration spec by clicking on the pencil icon at the end of the row for any listed Spec Template.

By clicking *APPLY STATE*, the user can once again see the Input selections which were presented during the initial provisioning and make changes when needed. After making changes and clicking *NEXT*, Morpheus will show the plan output no different than if you'd run `terraform plan`. On clicking *COMPLETE*, the plan will be executed as if you'd run `terraform apply`. Back on the State tab you will see the output from the Apply process as well as an indicator of the success or failure of the operation.

The screenshot shows the Morpheus web interface. At the top, there is a navigation bar with the Morpheus logo, a search bar, a notification bell, and a 'Support' dropdown. Below this is a main navigation menu with tabs for Operations, Provisioning (selected), Infrastructure, Backups, Logs, Monitoring, Tools, and Administration. Under Provisioning, there are sub-tabs for Instances, Apps, Blueprints, Jobs, Automation, Virtual Images, Library, Deployments, and Service Mesh. The main content area shows a 'VPC' resource with a 'main' environment and a 'vpc' resource. Below this, there is a 'Summary' tab and a 'State' tab (selected). The 'State' tab shows 'STATE DETAIL' with 'REFRESH STATE' and 'APPLY STATE' buttons. Below the buttons, it says 'DEMO' and 'Last Check: 04/29/2021 10:24 AM', 'Status: OK', and 'Drift Status: No Drift'. A terminal window is open, showing the output of an 'apply' command:

```

aws_vpc.main: Refreshing state... [id=vpc-05ffc795ad6738e44]
aws_subnet.main[1]: Refreshing state... [id=subnet-0dca56bffc339745]
aws_subnet.main[0]: Refreshing state... [id=subnet-0ad5e6d08b14375d3]
aws_subnet.main[2]: Refreshing state... [id=subnet-07fa95d0b7095ce24]

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create
~ update in-place
-/+ destroy and then create replacement

```

Morpheus will also regularly check for drift from the Terraform configuration. On the State tab near the top is a “Drift Status” indicator. This will either show Drift or No Drift depending on the situation. Morpheus will automatically check for drift every few minutes but you can perform a manual check at any time by clicking *REFRESH STATE*. Drift can be corrected when needed by reapplying state (*APPLY STATE* button).

5.13.10 Backing Up and Restoring Morpheus Appliance

Morpheus includes built-in tools for backing up managed Instances as well as the appliance itself. Use this guide to configure a location and schedule for backing up your Morpheus appliance. This guide also includes steps for restoring or migrating your appliance from the created backup. The steps are the same whether your appliance is deployed in a single node or distributed architecture.

The built-in Morpheus appliance backup functionality backs up the MySQL data. In addition to the database, it’s advisable to back up your shared storage (at `/var/opt/morpheus/morpheus-ui`) and the `morpheus.rb` configuration file.

Note: The destination Morpheus appliance must be running the same version as that which the backup was taken from.

Create A Backup Job

A Backup Job in Morpheus holds the schedule timing and retention count for automated backups. If you already have a Job configured, you can move on to the next section. By default, Morpheus includes two execution schedules: Daily at Midnight and Weekly on Sunday at Midnight. If currently-existing options do not make sense for your backup needs, create a new execution schedule:

1. Navigate to Library > Automation
2. Click on the “Execute Scheduling” tab
3. Click + *ADD*
4. Enter schedule timing using cron notation
5. Click *SAVE*

With the execution schedule created, we can move on to creating the Backup Job itself. A Backup Job includes both the backup retention count and an execution schedule (which we just created).

1. Navigate to Backups > Jobs
2. Click + *ADD*
3. Name the Job, then configure the retention count and the schedule
4. Click *SAVE*

Integrate a Bucket or File Share

When configuring a Morpheus appliance backup, a storage location is selected. If you already have the destination bucket or file share integrated with Morpheus, skip to the next section.

1. Navigate to Infrastructure > Storage
2. Click on the Buckets or File Shares tab depending on your chosen storage type
3. Click + *ADD*
4. Select the appropriate bucket or file share type
5. Complete the required fields and click *SAVE CHANGES*

Note: Additional guidance on integrating each of the supported bucket and file share types can be found elsewhere in Morpheus documentation.

Configuring Morpheus Appliance Backup

With the groundwork laid in the previous sections, we’re ready to enable and configure Morpheus appliance backup.

1. Navigate to Administration > Settings > Backups
2. Slide the switch labeled “Backup Appliance”
3. Click *SAVE*

On saving this change, a text link labeled “Backup” will be activated which will take you directly to the automatically-generated appliance backup job. Click this link to continue.

1. Click *EDIT*

2. Enter a name for the appliance backup job
3. Select an integrated storage bucket or file share
4. Choose a pre-created backup job. If you do not have an existing backup job that fits, a retention count and schedule can be manually created in this modal. If you manually configure retention counts and schedules in addition to associating a Job, the Job values will override any manual settings.
5. Click *SAVE CHANGES*

At this point, your appliance will be automatically backed up on the schedule you chose and stored in the selected location. An appliance backup will store backup copies of the appliance MySQL database. Should you need to restore or migrate your database from backup, follow the steps in the next section of this guide.

Restoring an Appliance from Backup

Begin by ensuring the Morpheus UI service is stopped on all of the application servers:

```
[root@app-server-new ~] morpheus-ctl stop morpheus-ui
```

To access the MySQL shell we will need the password for the Morpheus DB user. We can find this in the `morpheus-secrets` file:

```
[root@app-server-old ~] cat /etc/morpheus/morpheus-secrets.json | grep morpheus_password
"morpheus_password": "451e122cr5d122asw3de5e1b", <---- this one
"morpheus_password": "9b5vdj4de5awf87d",
```

Make note of the first `morpheus_password` value as indicated above.

Copy the SQL database backup from the backup bucket or file share to an appliance node at `/tmp/morpheus_backup.sql`. Then, you can import the MySQL dump into the target database using the embedded MySQL binaries, specifying the database host, and entering the password for the morpheus user when prompted:

```
[root@app-server-new ~] /opt/morpheus/embedded/mysql/bin/mysql -u morpheus -h 127.0.0.1
↪morpheus -p < /tmp/morpheus_backup.sql
Enter password:
```

The data from the old appliance is now replicated on the new appliance. Simply start the UI to complete the process:

```
[root@app-server-new ~] morpheus-ctl start morpheus-ui
```

5.13.11 Cloud Resource Tagging with Morpheus

Introduction

As organizations scale their cloud environments, they often need to devise methodologies for organizing those resources. Tags consist of key and optional value pairs which make it easier to search for or filter your cloud resources based on categories relevant to the organization. While each public or private cloud handles tagging slightly differently, Morpheus removes that complexity and differentiation by handling tags in a consistent way across clouds. In addition, the Morpheus policy engine gives administrators tools to set up guard rails and ensure cloud resource tagging is handled consistently with each provisioning.

This guide will go through Morpheus tagging features and best practices, as well as include provisioning examples from some of the most commonly integrated clouds.

Note: Tag syncing is bi-directional in Morpheus for supported clouds. Tag syncing is currently supported in Amazon, Azure, Google, VMware, and Alibaba Clouds.

Tagging on Provisioning

In the simplest use case, tags can be entered manually in most provisioning scenarios. For example, on the Configure tab of the Create Instance wizard, the user can expand the Tags section and enter as many Tags as he or she needs to comply with organization practices.

✕
CREATE INSTANCE

TYPE >
GROUP >
CONFIGURE
AUTOMATION >
REVIEW

Configuration Options

LAYOUT

PLAN
Cores: 1 Memory: 1 GB Price: \$8.4912 / Month

RESOURCE POOL

VOLUMES GB +

NETWORKS DHCP +

SECURITY GROUPS +

PUBLIC IP

▶ User Config

▶ Network Options

▶ Advanced Options

▼ Tags

TAGS 🗑️ +

▶ Environment

PREVIOUS
NEXT

Once the resource is deployed, Tags are synced and applied to the provisioned machine in the relevant cloud. Tags are shown on the Instance detail page in Morpheus and can also be confirmed in the cloud console if desired. Tag syncing is also a two-way street in that any tag updates applied within Morpheus or within the cloud console will be reflected everywhere.

Note: Tag updates made in the cloud console may take up to five minutes to be reflected in Morpheus UI following the next sync of cloud data.

Custom Instance Types and Tagging

Manually tagging resources as described in the previous section will work in some cases but many administrators will likely need to pre-seed the provisioning wizards with tagging prompts or build dropdown lists for tag values. This is accomplished in Morpheus through the use of custom Instance Types, Inputs, and Option Lists.

To get started, first create an Option List to hold dropdown or typeahead sets of available key values if needed based on organizational tagging policy. If the tag value field should be manually entered at provision time or if the value field is to be left blank, this step can be skipped. Option Lists are created and stored in the Morpheus Library (Library). They can be populated manually by entering CSV or JSON datasets as shown in the example below. They can also be dynamically populated through Morpheus API or REST calls.

EDIT OPTION LIST
✕

NAME

DESCRIPTION

TYPE

DATASET Create an initial JSON or CSV dataset to be used as the collection for this option list. It should be a list containing objects with properties 'name', and 'value'.

```
'App1 ' , 'App1 '
'App2 ' , 'App2 '
'App3 ' , 'App3 '
'App4 ' , 'App4 '
'App5 ' , 'App5 '
```

With the set of possible values defined (if needed), we next create an Input to prompt the user for tag input when provisioning relevant Instance types. Inputs are also housed in the Morpheus Library (Library).

It's important to note the value entered for the FIELD NAME on the new Input will be set as the tag key. The EXPORT AS TAG box should also be marked. By default, the TYPE value is Text. This is appropriate when the user should be prompted with a free text field at provision time to enter a tag value. To tie this Input to the Option List that was just created (if needed), change the TYPE value to Select List or Typeahead. Typeahead works best for very long lists while Select List is often a better user experience for lists of a more manageable size. Set the Option List we created in the previous step in the OPTION LIST value (if needed).

EDIT OPTION TYPE ✕

NAME

DESCRIPTION

FIELD NAME
This is the input fieldName property that the value gets assigned to.

EXPORT AS TAG

TYPE ▼

LABEL
This is the input label that shows typically to the left of a custom option.

DEFAULT VALUE

REQUIRED

OPTION LIST ▼

DEPENDENT FIELD
A fieldName that will trigger reloading this option list

[SAVE CHANGES](#)

At this point, we are ready to add this Input to any custom Instance Types or Layouts. When those Instance Types or Layouts are provisioned, the values input by the user become tags associated with the created cloud resources. By setting the Input on an Instance Type, the tag selection appears when provisioning all associated Layouts. Alternatively, if the Input is set on individual Layouts, it will only appear when those Layouts are provisioned.

Instance Types and Layouts are also stored in the Morpheus Library (Library). By opening up any custom Instance Type, we can add the Input we just created when editing the Instance Type. Additionally, we can drill into associated Layouts and apply the Input to selected Layouts if that's more appropriate.

EDIT INSTANCE TYPE
✕

NAME

CODE
 Useful shortcode for provisioning naming schemes and export reference.

DESCRIPTION

CATEGORY ▾

ICON Browse

Suggested Dimensions: 150 x 51

Option Types

*Tag Application <small>(select)</small>	☰ ✕
*Tag Compliance <small>(text)</small>	☰ ✕
*Tag Cost Center <small>(text)</small>	☰ ✕
*Tag Department <small>(text)</small>	☰ ✕
*Tag Environment <small>(text)</small>	☰ ✕
*Tag Owner <small>(text)</small>	☰ ✕

Going forward, each time the chosen Instance Type or Layout is provisioned, the user will be prompted to enter relevant

tag selections. We can even require the user input these values or govern their inputs through tagging policies which will be discussed in the next section.

Instituting Tagging Policies

If needed, Morpheus allows cloud resource tagging to be governed through its native policy engine. Like other policies, tag policies are added from Administration > Policies. By creating a new policy and setting the TYPE to Tags, the relevant fields are revealed.

Note: Tag policy scanning and enforcement is only currently supported for Azure, Amazon AWS, VMware, and Google Cloud Platform clouds. Additional Clouds may be supported in the future.

With a tag policy, we can choose to enforce the policy on a strict or passive basis by marking or unmarking the STRICT ENFORCEMENT box. Strictly enforced tagging policies will not allow provisioning to proceed in supported clouds if the policy requirements are not met. If we opt to enforce the policy passively, a warning banner will appear on the detail page of any server that does not meet policy requirements. Additionally, existing servers in supported clouds will be scanned and those which do not meet policy requirements will also receive the warning banner.

A tag policy must be given a KEY value. If we define only a KEY value, the policy will look for a tag with that key and any (or no) value. Alternatively, we can select any pre-existing Option List as the VALUE LIST to require the tag contain a value that exists in that list or even enter a specific VALUE.

Finally, like other Morpheus policies, we can choose to scope it globally, by group, by cloud, or by user. Primary Tenant administrators can also choose to scope the policy to one or more Subtenants.

EDIT POLICY ×

TYPE

NAME

DESCRIPTION

ENABLED

Config

STRICT ENFORCEMENT

KEY

VALUE

VALUE LIST ▼
Choose from any configured Option List

Filter

SCOPE ▼

TENANTS

Tagging in Action

With the prep work complete, we can take a look at our Inputs in action at provision time. In this example case, several Inputs have been created and applied to one custom Instance Type. The example Instance Type has three associated CentOS Layouts, one for AWS, one for VMware, and one for Azure. Regardless of the selected Layout, users are prompted to fill the same tag fields and our tagging remains consistent regardless of the user who is provisioning a new resource at the time.

Tagging and AWS

When provisioning my CentOS Instance Type with an Amazon Layout, the tag prompts are shown in the provisioning wizard.

The screenshot displays the 'CREATE INSTANCE' wizard in the Morpheus console, specifically the 'CONFIGURE' step. The wizard is titled 'CREATE INSTANCE' and has a close button (X) in the top right corner. The progress bar shows the following steps: TYPE, GROUP, CONFIGURE (current), AUTOMATION, and REVIEW.

The 'Configuration Options' section includes the following fields:

- LAYOUT:** A dropdown menu with the selected value 'aws_cent'.
- PLAN:** A dropdown menu with the selected value 'Amazon T2 Micro - 1 Core, 1GB Memory'. Below this, it shows 'Cores: 1 Memory: 1 GB Price: \$8.4912 / Month'.
- RESOURCE POOL:** A dropdown menu with a partially visible value.
- VOLUMES:** A table-like structure with columns for name, size, unit, and type. The first row shows 'root', '20', 'GB', and 'gp2'. There is a '+' sign to the right of the row.
- NETWORKS:** A dropdown menu with a partially visible value, followed by 'DHCP' and a '+' sign.
- SECURITY GROUPS:** A dropdown menu with the value 'Select' and a '+' sign.
- TAG APPLICATION:** A dropdown menu with the value 'Select'.
- COMPLIANCE:** A text input field with the value 'PCI'.
- COST CENTER:** A text input field with the value '10001'.
- DEPARTMENT:** A text input field with the value 'Security'.
- ENVIRONMENT:** A text input field with the value 'Dev'.
- OWNER:** A text input field with the value 'John User'.
- PUBLIC IP:** A dropdown menu with the value 'Subnet Default'.

At the bottom left, there is a section titled 'User Config' with a right-pointing arrow.

In the AWS web console, we can see the same tags are applied. We also have two-way tag sync going forward. When tags are updated in Morpheus, the changed is synced to the AWS web console. The opposite is also true.

- Description
- Status Checks
- Monitoring
- Tags**

Add/Edit Tags

Key	Value	
Morpheus ID	12345	Show Column
Morpheus Instance ID	6666	Show Column
Morpheus Server ID	7777	Show Column
Name	CentOS Instance Type 1001	Show Column
application	App1	Show Column
compliance	PCI	Show Column
costCenter	10001	Show Column
department	Security	Show Column
environment	Dev	Show Column
owner	John User	Show Column

Tagging and VMware

When provisioning my CentOS Instance Type with a VMware Layout, the tag prompts are shown in the provisioning wizard.

CREATE INSTANCE
✕

TYPE >
GROUP >
CONFIGURE
AUTOMATION >
REVIEW

Configuration Options

LAYOUT

PLAN

Cores: 1 Memory: 1 GB Price: \$5.726 / Month 📄

RESOURCE POOL

VOLUMES GB +

NETWORKS +

TAG APPLICATION

COMPLIANCE

COST CENTER

DEPARTMENT

ENVIRONMENT

OWNER

HOST

FOLDER

▶ User Config

In the VMware console, we can see the same tags are applied. We also have two-way tag sync going forward. When tags are updated in Morpheus, the changed is synced to VMware. The opposite is also true.

Tagging and Azure

When provisioning my CentOS Instance Type with an Azure Layout, the tag prompts are shown in the provisioning wizard.

CREATE INSTANCE
✕

TYPE
GROUP
CONFIGURE
AUTOMATION
REVIEW

Configuration Options

LAYOUT

_azure_cent

▼

PLAN

Basic_A1 (1 Core, 1.75GB Memory) (eastus)

▼

Cores: 1 Memory: 1.8 GB Price: \$17.4556 / Month 💰

RESOURCE POOL

Select

▼

VOLUMES

root

Page Blob

blabsuseastsa

+

NETWORKS

Select Network

DHCP

+

TAG APPLICATION

Select

▼

COMPLIANCE

PCI

COST CENTER

10001

DEPARTMENT

Security

ENVIRONMENT

Dev

OWNER

John User

SECURITY GROUP

Select

▼

AVAILABILITY SET

No Availability Set

▼

In the Azure console, we can see the same tags are applied. We also have two-way tag sync going forward. When tags are updated in Morpheus, the changed is synced to Azure. The opposite is also true.

5.13.12 Cypher Policies

Morpheus allows administrators to set robust Cypher policies, which determine global, role, and/or specific user access to configured Cypher secret mount points. A number of considerations should be made when deploying Cypher Access policies, including how user role permissions will interact with the policy and how conflicts between overlapping policies are resolved.

Role Permissions

User Role permissions (Administration > Roles) greatly affect Cypher access. Cypher access Role permissions are set from the Features tab of the selected Role under “Tools: Cypher”. The Role permission should be set based on the highest level of access to any one individual Cypher entry needed for the specific Role. For example, if the Role needs no access to any Cypher entries, set the feature permission to “None” and hide the Cypher UI from the Role completely. Alternatively, if the Role needs to use and decrypt even one Cypher entry, set the feature permission to “Full Decrypt”. The complete set of available permissions are below:

- **NONE:** Cypher UI hidden
- **READ:** Cypher UI present, entries can be listed
- **USER:** Cypher UI present, user sees and can use their own entries, user can create new entries
- **FULL:** Cypher UI present, user sees and can use all entries, user can create new entries, user cannot decrypt any entries
- **FULL DECRYPT:** Full access to Cypher features including the ability to decrypt secrets

Cypher Access Policies

Like other Policy types, Cypher Access Policies are created in Administration > Policies. Click + *ADD POLICY* to create new. Set the type to “Cypher Access” and the relevant configuration options will be displayed. In addition to the type, enter a name for the Policy in the top section.

In the next section, enter the key path to which the Policy will apply. In addition to static entries that point to one specific Cypher entry, this field supports pattern matching with regex. For example, enter “.*” to refer to all Cypher entries or “secret/.*” to refer to all entries under the secret mount point.

In addition to the path, set the privileges users in the Policy scope should have on the indicated path.

- **LIST:** See the entries on the indicated path listed in the Cypher UI
- **READ:** Decrypt the entries on the indicated path
- **WRITE:** Add new entries on the indicated path
- **UPDATE:** Edit Cypher entries on the indicated path. This is future functionality, the ability to update Cypher entries does not currently exist in the product
- **DELETE:** Delete entries on the indicated path

Config

KEY PATH

READ

WRITE

UPDATE

DELETE

LIST

Finally, set the scope for the Policy. Cypher Access Policies support Global, Role, and User scope. For example, you may want to block off sets of Cypher entries for various departments within your organization. If you have existing Roles in Morpheus for each department, you can set up Role-scoped Policies to ensure they can only list, use, and add Cypher entries which are relevant to their own department.

Filter

SCOPE

ROLE

APPLY INDIVIDUALLY TO EACH USER IN ROLE

TENANTS

Important: When Cypher Access Policies conflict, the Policy with the longest path string length (typically the most specific) takes precedence. For example, a Policy giving LIST and READ access to “secret/aws/*” would be superseded by a Policy giving NO access to “secret/aws/my-secret-key”. In such a case, the user would see everything at the “secret/aws/*” path except the one indicated in the more specific Policy. When Policies targeting the same path differ only in their scope, the following scope precedence is applied: Role > User > Global. For example, if a Role-scoped Policy targeting “.*” grants LIST and READ while a User-scoped Policy targeting the same path grants LIST, the user would be granted the rights in the Role-scoped Policy.

Cloud Profiles

Terraform Cloud Profiles are created on each Cloud detail page (Infrastructure > Clouds > Selected Cloud > Profiles Tab), encrypted in Cypher, and create a Cypher entry that is visible both on the Profile tab of the Cloud detail page and in Cypher. When added to a Cloud they create a Cypher entry at path `tfvars/profile/cloud/$cloudCode/variables`. If a Cloud profile Cypher entry is restricted by a Cypher Access policy, it will be (or will not be) listable/readable/deletable as dictated by the Policy but still will be viewable from the Cloud detail page if the user has sufficient permissions. Restricting or granting access to Cloud profiles via Policy does not affect access on the Cloud. Other Role permissions, such as “Admin: Profiles”, “Infrastructure: Clouds”, and Cloud/Group access must be used to restrict access via Cloud detail pages.

Example Policy

In my example organization, I have one department that needs access to AWS-related secrets and another department that needs access to Azure-related secrets. There are many other secrets stored in my appliance but I don't want either of these departments to access any of those.

secret/cypher_policy_test_1	DECRYPT	04/04/2022 02:16 PM	04/05/2022 02:32 PM	ACTIONS ▾
secret/cypher_policy_test_2	DECRYPT	04/04/2022 02:16 PM	04/04/2022 02:46 PM	ACTIONS ▾
secret/example/aws/access-key	DECRYPT	04/05/2022 07:28 PM	04/05/2022 07:28 PM	ACTIONS ▾
secret/example/aws/my-special-key	DECRYPT	04/05/2022 07:28 PM	04/05/2022 07:28 PM	ACTIONS ▾
secret/example/aws/secret-key	DECRYPT	04/05/2022 07:27 PM	04/05/2022 07:27 PM	ACTIONS ▾
secret/example/azure/secret1	DECRYPT	04/05/2022 07:29 PM	04/05/2022 07:29 PM	ACTIONS ▾
secret/example/azure/secret2	DECRYPT	04/05/2022 07:30 PM	04/05/2022 07:30 PM	ACTIONS ▾
secret/hello	DECRYPT	04/12/2021 02:40 PM	04/01/2022 10:06 PM	ACTIONS ▾
secret/QA Kubernetes Cluster-kube-cluster-admin-kubeconfig	DECRYPT	02/16/2022 07:11 PM	03/14/2022 12:54 PM	ACTIONS ▾

For the first department, I've set up a Policy that allows them to list and read (including use and decryption rights) AWS secrets. A second Policy specifically excludes them from seeing one specific entry. The Policy with the more specific path will supersede the more generic Policy that includes a wildcard.

POLICIES				
Search	TYPE: CYPHER ACCESS ▾	SCOPE: ALL ▾	+ ADD POLICY	
	Cypher Exclusion	Cypher: - key path: secret/example/aws/my-special-key - none	 cypher_access_test_role	Accounts: All  
	Cypher Policy - AWS	Cypher: - key path: secret/example/aws/* - read, list	 cypher_access_test_role	Accounts: All  

By impersonating the user, we see they indeed have access to just the two desired Cypher entries.

Cypher ⁱ				
Search				+ ADD
KEY	VALUE	LEASE REMAINING	LAST UPDATED	LAST ACCESSED
secret/example/aws/access-key	DECRYPT		04/05/2022 07:28 PM	04/05/2022 07:28 PM
secret/example/aws/secret-key	DECRYPT		04/05/2022 07:27 PM	04/05/2022 07:27 PM

For the second department, I have set up a Policy that allows them to list and read (including use and decryption rights) Azure secrets.

POLICIES

Search [] TYPE: CYPHER ACCESS SCOPE: ALL + ADD POLICY

Policy Name	Cypher Key Path	Role	Accounts	Actions
Cypher Exclusion	Cypher - key path: secret/example/aws/my-special-key - none	cypher_access_test_role	Accounts: All	[Edit] [Delete]
Cypher Policy - AWS	Cypher - key path: secret/example/aws/* - read, list	cypher_access_test_role	Accounts: All	[Edit] [Delete]
Cypher Policy - Azure	Cypher - key path: secret/example/azure/* - read, list	cypher_access_test_role	Accounts: All	[Edit] [Delete]
tm - all	Cypher - key path: secret/tm/vh.* - read, write, update, delete, list	Tommy Marks	Accounts: All	[Edit] [Delete]

By impersonating the user once again, we see they indeed have access only to Azure entries.

Cypher ⓘ

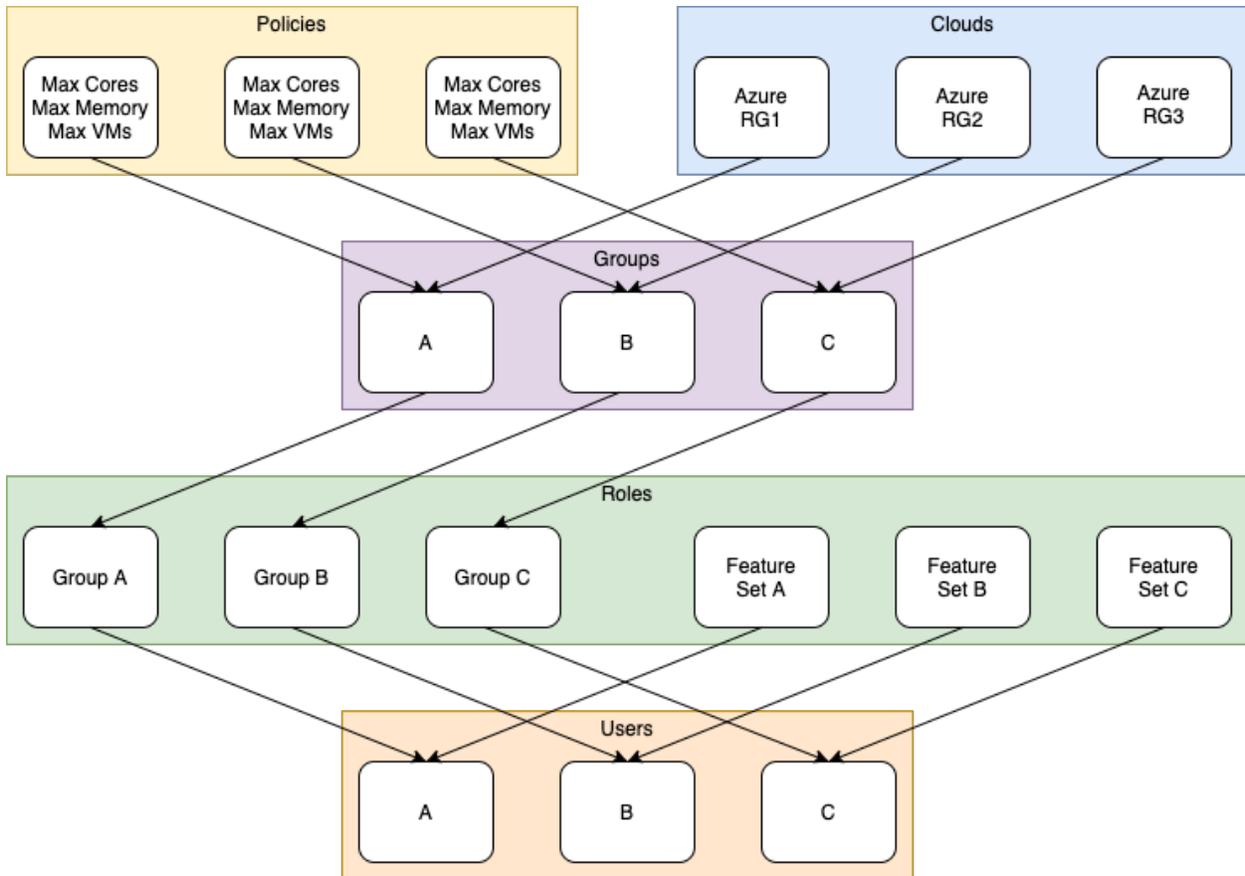
Search [] + ADD

KEY	VALUE	LEASE REMAINING	LAST UPDATED	LAST ACCESSED	ACTIONS
secret/example/azure/secret1	DECRYPT		04/05/2022 07:29 PM	04/05/2022 07:29 PM	ACTIONS ▾
secret/example/azure/secret2	DECRYPT		04/05/2022 07:30 PM	04/05/2022 07:30 PM	ACTIONS ▾

5.13.13 Configuring Access with Clouds, Groups, and Roles

As you get started, it's vital to understand constructs such as Clouds and Groups as they are implemented in Morpheus. Deploying an effective strategy for dividing organizational resources and people groups under these constructs will ensure all parties have convenient access to the resources they need. Map currently-existing strategies for splitting resources and controlling costs into Morpheus and your teams will have access only to resources allocated to their department.

This guide discusses the constructs of Clouds, Groups, Roles, Users, and Policies while also presenting an example use case for allocating resources and distributing access under these umbrellas. We'll end up with a permissions structure represented by the diagram below and step through each part of the implementation process.



Clouds

Clouds have broad meaning in Morpheus and can be integrations into public clouds, private clouds, or bare metal servers. You can read more about Clouds in Morpheus Docs [here](#).

In our example case, the organization has an Azure account and three separate departmental projects consolidated under separate resource groups. The Morpheus integration with Azure public cloud allows users to scope the Cloud to one specific resource group, if desired. We will do so in this case because it will allow us to restrict our users to the appropriate Azure resource groups in a later step. In the screenshot below, note how I am scoping the Morpheus Azure Cloud to a specific resource group when adding or editing the Cloud. This process will be repeated two additional times to create the three Morpheus Clouds we need. Complete details on integrating Morpheus with Azure public cloud are [here](#) in Morpheus Docs.

Details

CLOUD TYPE

SUBSCRIPTION ID

TENANT ID

CLIENT ID

CLIENT SECRET

LOCATION

RESOURCE GROUP

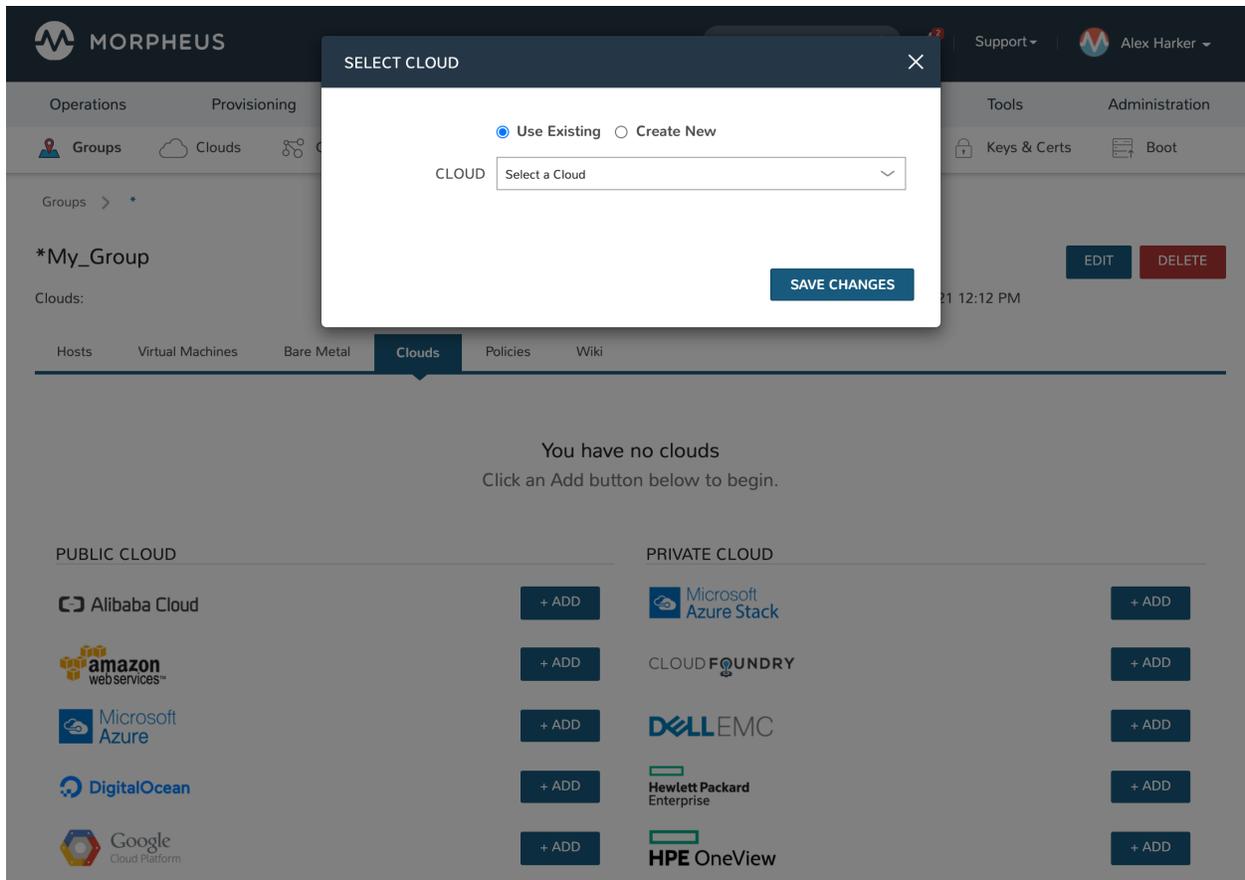
INVENTORY LEVEL

ACCOUNT TYPE

Groups

Groups in Morpheus allow administrators to determine which Clouds their users have access to. Part of defining a Role, which we'll see in the next section, is selecting which Groups are associated with the Role. One or multiple Clouds are added to each Group and passed to users through their assigned Roles.

When creating a Cloud, it must be added to a Group. Additionally, Clouds can be added to a Group at any time from the Clouds tab on the Group detail page (Infrastructure > Groups > Specific Group). In the example case diagrammed above, one Cloud (Azure cloud account scoped to a specific resource group) is associated with each Group but often Groups will contain many Clouds. In the screenshot below, I am adding an Azure Cloud to a Group that is currently empty.



Roles

Morpheus Roles determine user access to many things, including application features, Groups, configured Instance Types, configured Blueprints, Personas, and Catalog Item configurations for the Service Catalog Persona. It's important to note that a user can have multiple Roles and will take the most permissive rights across all of these Roles. In many cases, this greatly simplifies Role configuration for administrators as it prevents the need to create highly-specific individual Roles that can suit every case. To illustrate this for our example case, I will create a set of Roles to handle Group access for my users and another set of Roles to give access to the correct feature permissions.

For this example case, I'll start by creating one Role for each of the three Groups we created in the previous step. By default, a new Role will have all feature permissions set to "None" as shown below meaning we can skip directly to the Group Access tab.

Roles > *My Group Role

*My Group Role

PERMISSIONS

NAME	ACCESS
Admin: Appliance Settings	None
Admin: Backup Settings	None
Admin: Environment Settings	None
Admin: Guidance Settings	None
Admin: Health	None
Admin: Identity Source	None
Admin: Integrations	None
Admin: License Settings	None

On the Group Access tab, I will toggle the Global Access setting to Custom and give access only to one Group as shown in the next image. By doing this for each of our Groups created in the last section, we can easily control Group access for each user by applying one or more of our Group Roles to their user which we will do in the next section.

Roles > *My Group Role

*My Group Role

PERMISSIONS

NAME	ACCESS
*My_Group	Full
AA	None
Adam H	None
Adam H VSAN only	None
Adam L	None

In addition to Roles for handling Group access, we need additional Role sets which will control our feature access. For example, we might have highly permissive feature access for administrators and more restrictive ones for developers or other users who have specific responsibilities requiring access to just a few specific areas of the UI. These additional

Roles are created in the same way as the Group Roles we just created but all Group access will be set to “None” and the appropriate feature access will be configured on the “Feature Access” tab.

Users

With the groundwork laid in the previous steps, we can easily configure permissions for any user accounts we might add to Morpheus. In the screenshot below, I’m adding a user account for a developer at my organization. I’ve applied one Role to handle correct Group access for this user and I’ve applied the “Developer” role to give only the feature access needed for this user to carry out his or her responsibilities.

NEW USER ✕

FIRST NAME

LAST NAME

USERNAME

EMAIL

RECEIVE NOTIFICATIONS

Roles

<input type="checkbox"/> ADMIN	<input type="checkbox"/> GROUP ACCESS B
<input type="checkbox"/> BLANK TEST	<input type="checkbox"/> GROUP ACCESS C
<input checked="" type="checkbox"/> DEVELOPER	<input type="checkbox"/> NEW CATALOG TEST
<input checked="" type="checkbox"/> GROUP ACCESS A	<input type="checkbox"/> SAMPLE ROLE

Policies

One final consideration for this example case is policy application. Morpheus enables administrators to place fine-grained governance scoped to specific users, Roles, Groups, Clouds, Tenants, or globally. You can read more about policies in Morpheus docs [here](#) but we will also create some that apply to the example case discussed in this guide.

Policies are created from the Administration menu in the Policies section. In the screenshot below, I’ve created a new policy and chosen to scope it to a Group. In this case, I’m creating a maximum VMs policy but there are many other types which are listed in our documentation linked above.

NEW POLICY ✕

TYPE

NAME

DESCRIPTION

ENABLED

Config

MAX VMs

Filter

SCOPE

GROUP

Conclusion

At this point, we've completed implementation of the access structure. As new users are added, we can easily give them the Group and feature access they need. We can also easily add or edit the policies that are in place or add new Clouds and distribute access to existing users. The key is understanding the relationship between the Morpheus constructs discussed in this guide. Armed with that understanding, the example case presented here can be expanded to suit the needs of large-scale enterprise.

5.13.14 Tenancy

Morpheus appliances may be multi-tenanted, that is, they may contain multiple isolated environments known as Tenants. Successfully building a multi-Tenanted Morpheus environment requires creating and pulling together multiple constructs. This guide aims to describe the pieces needed to create functioning Tenants with Users, identity integrations, Roles, Groups, and whitelabeling. We will also share a Cloud and Library item from the Master Tenant which is provisionable from the new Subtenant.

On installation, a single Tenant is created. This Tenant is often referred to as the Master Tenant and has some special properties that additional created Tenants (known as Subtenants) do not have. Tenants manage their own workloads, may have their own custom whitelabeling, and also have their own users and integrations. The Master Tenant may (or may not) share down integrations, Library items, Roles, and more but Subtenants are fully isolated from each other.

- The Master Tenant is the default Tenant created during the installation of Morpheus
- All Tenants created after installation are Subtenants. Only one Master Tenant can exist
- The Master Tenant creates and controls all Subtenants.
- Tenants are isolated environments with unique users, workloads, and Groups
- The Master Tenant can share or assign its resources to Subtenants
- Subtenants cannot share their resources with other Tenants
- Subtenants cannot see resources from other Subtenants

- Subtenants can only access Master Tenant resources that have been set to Public visibility or specifically assigned to the Subtenant

Tenant Roles

Before even creating the Tenant, we need to understand what a Tenant Role is and ensure that a Tenant Role with correct rights permissions exists. Tenant Roles set the maximum feature permissions levels for any User in the Subtenant and also give (or revoke) access to Clouds, Library items, Tasks, Workflows, and other things integrated or created in and shared from the Master Tenant. Only Master Tenant administrators have the ability to create Tenant Roles.

Note: User Roles within the Tenant may be created with lesser permissions than the Tenant Role allows but we will discuss User Roles further in a later section.

In a new environment, the only Tenant Role will be the included “Tenant Admin” Role but this Role is not editable. It’s advisable to create your own Tenant Roles so they may be edited as needed. To get started, we can create a new Tenant Role and copy from the pre-existing “Tenant Admin” Role.

1. Navigate to Administration > Roles
2. Click + *CREATE ROLE*
3. Set a NAME, TYPE (Tenant Role), and COPY FROM ROLE (Tenant Admin)
4. Click *SAVE CHANGES*

NEW ROLE ×

NAME

DESCRIPTION

TYPE Tenant Role ▼

The master tenant can create two types of roles. A Tenant Role is specifically created for assignment to an Tenant as a Base Role and can be used to restrict cloud access and base features that all subroles have to adhere to.

COPY FROM ROLE Tenant Admin ▼

SAVE CHANGES

Once created, click into the new Role and edit feature permissions if you wish. Remember the Tenant Role is setting the maximum available Role permissions for each feature that Users within the Tenant can have. We’ll worry about scoping resources to the Subtenant (such as Clouds) in a later section.

With a Tenant Role created, we can move on to creating the Tenant itself.

Create a Tenant

We'll now create a Tenant and apply the Tenant Role we've just created.

1. Navigate to Administration > Tenants
2. Click *CREATE TENANT*
3. Name your Tenant and set the Tenant Role in the BASE ROLE field
4. Click *SAVE CHANGES*

The NAME and BASE ROLE are the most important settings and the only two required. See the Tenants section of Morpheus documentation to learn more about other settings which may be applied when adding or editing Tenants.

NEW TENANT ×

NAME

DESCRIPTION

ENABLED

SUBDOMAIN
The SubDomain is used for creating a direct login url in Identity Sources or as a login prefix to identify the tenant i.e. 'subdomain\username'

BASE ROLE ▼
This is the primary role of the tenant. All roles created within the tenant must inherit this role as the base role.

CURRENCY ▼

▶ Advanced Options

With the Tenant created, the rest of the guide will pertain to configuring the new Tenant and demonstrating how to provision new workloads using resources shared down from the Master Tenant.

Users and User Roles

In the prior steps we've created a new Tenant and associated our created Tenant Role with it. If you click into the new Tenant you'll notice that there are currently no users so the new Tenant is still unusable. We could go ahead and create a new User but if this is a new Morpheus environment, the new Tenant will not have any User Roles to apply to the new User we wish to create.

To remedy this problem, we can add a "Multi-Tenant User Role" from the Master Tenant. This is essentially a Role Template from which a Role is created and seeded down to each Tenant. Only Master Tenant administrators can create Multi-Tenant User Roles. After creation, edits can be made and permission changes will be filtered down to the child Role that exists within each Subtenant. However, it's important to know that should the child Role ever be edited in the Subtenant, the link back to the Multi-Tenant User Role will be broken and it will become like any other User Role created in the Subtenant. To prevent this situation, Master Tenant administrators can set a "MULTITENANT LOCKED" option to restrict such edits in the Subtenant.

New Multi-Tenant User Roles are created in Administration > Roles:

1. Click + *CREATE ROLE*
2. Provide a **NAME**, **TYPE** (User Role), **COPY FROM ROLE** (if desired)
3. Mark the boxes to set **MULTITENANT ROLE** and **MULTITENANT LOCKED**
4. Click *SAVE CHANGES*

After saving the Role, it will appear in the Master Tenant and will also be seeded down to the Tenant we've already created since it is a Multi-Tenant User Role. We've also locked the Role so that Subtenant users cannot edit it and thus unlink it from future permissions edits made in the Master Tenant. Keep in mind that no matter what permissions we set, Users in the Subtenant will not be able to exceed permissions set on the Tenant Role as discussed in a prior section. If desired, you can click into this User Role now and edit its permissions to suit your needs.

NEW ROLE
✕

NAME

DESCRIPTION

TYPE

The master tenant can create two types of roles. A Tenant Role is specifically created for assignment to an Tenant as a Base Role and can be used to restrict cloud access and base features that all subroles have to adhere to.

COPY FROM ROLE

MULTITENANT ROLE
A Multitenant role is automatically copied into all existing subtenants as well as placed into a subtenant when created. Useful for providing a set of predefined roles a Customer can use

MULTITENANT LOCKED
Prevents subtenants from branching off this role/modifying it.

SAVE CHANGES

At this point we're ready to add a User which will make the Tenant usable. We can create one in the new Tenant by navigating to the Tenant list page (Administration > Tenants) and selecting the Tenant. There are a number of configurations to make when creating a user but when selecting the User's Role you should see the previously-created Multi-Tenant User Role.

NEW USER ×

FIRST NAME

LAST NAME

USERNAME

EMAIL

RECEIVE NOTIFICATIONS

Roles

Multi-Tenant Admin ×

Credentials

PASSWORD

CONFIRM

ENABLED

After saving the new User, we can make the first test of the new Tenant by impersonating the User and entering the Tenant. In the list of Users on the Tenant detail page, click the gear () icon at the end of the User's row and select "Impersonate." You should end up inside the new Tenant operating as the new user. This can be confirmed by making note of the updated name given in the upper-right portion of the application window. To quit impersonating, click on the name and select "Quit Impersonating."

At this point we've created a User manually but it's important to point out that Morpheus can also use an existing identity integration, such as Microsoft Active Directory, to manage User creation and Role assignment. Expand the section below to see an example use case or continue on if you'd prefer to administer your Users directly in Morpheus. The complete integration guide elsewhere in Morpheus docs also includes common Active Directory troubleshooting steps should those be needed.

- **Microsoft Active Directory Example**

Active Directory is Microsoft's primary authentication service. It is widely used in enterprise organizations and even in Microsoft cloud services. While Active Directory also supports LDAP protocol support (which Morpheus can integrate with as well), Morpheus includes a dedicated identity integration type specifically for Active Directory. By integrating Active Directory, Morpheus administrators can fully offload the work of managing the user lifecycle to Active Directory. Creating new users, applying roles to users, updating basic user data, disabling users, and more can be handled in Active Directory and automatically filtered down to Morpheus. This section includes an example integration walkthrough as well as additional details on the integration feature set.

Note: Caution should be used when integrating more than one Active Directory identity source with the same Morpheus Tenant. You must ensure the users on each identity source are unique users or that the two domains

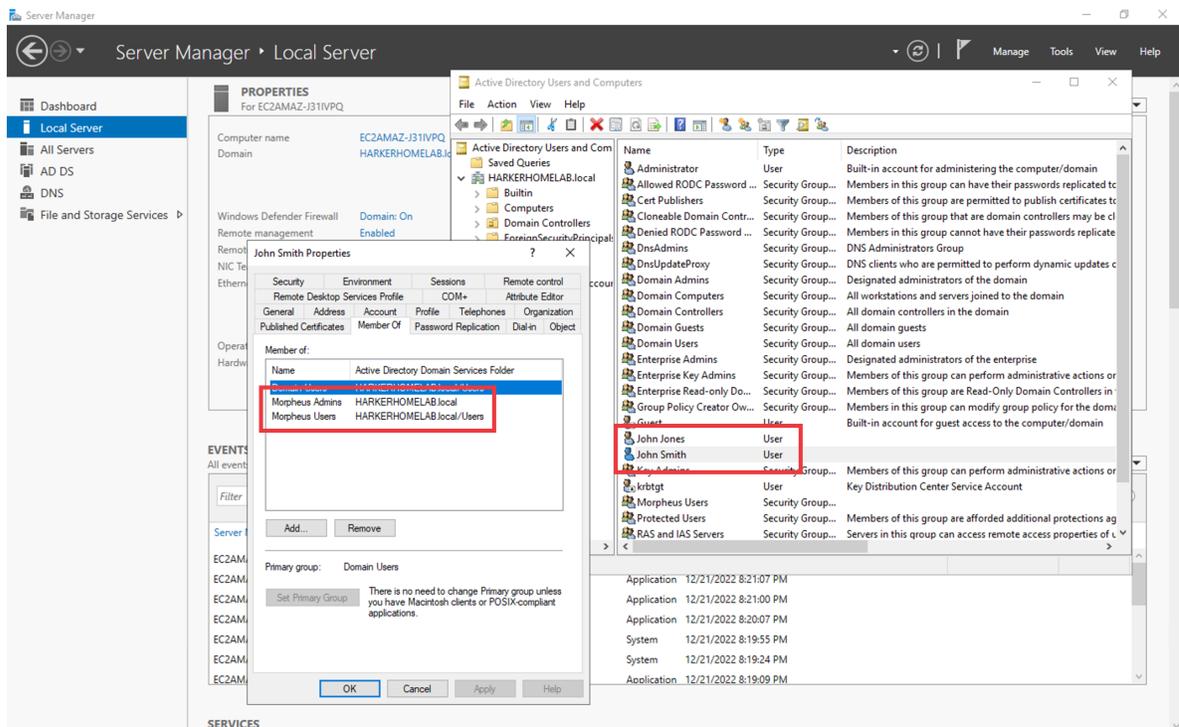
use different naming conventions for users.

In Morpheus, identity sources (if used) are configured per Tenant. In order to see an identity integration or create a new one, navigate to a Tenant detail page (Administration > Tenants > Selected Tenant) and click *IDENTITY SOURCES*. From this page we can add a new identity source or click the pencil () icon next to an existing identity source to view or edit it. Any configured identity source will apply to just one Tenant and they cannot be shared. One scenario where this is especially useful is in an MSP appliance where each Tenant is a siloed environment for a specific customer. The customer’s own existing Active Directory server and groups can be leveraged to build Morpheus user accounts with correct role mapping automatically.

When configuring an Active Directory integration in Morpheus, AD groups are mapped to Morpheus roles. When the user logs in for the first time, Morpheus adds the new user account with correct name, email address, and applies one or more roles depending on configuration. Going forward, Morpheus will sync down any changes to the user, including any role changes based on changes to the user’s associated AD groups or updated passwords. Additionally, disabling a user in AD will prevent them from accessing Morpheus.

Important: Morpheus will connect over port 389 for non-secure LDAP and port 636 for secure LDAP. Ensure that Morpheus is able to talk to the AD server on the proper port.

In a simple example, we have an Active Directory server which has two groups relevant to Morpheus, “Morpheus Users” and “Morpheus Admins.” We will configure Morpheus so that only users in the “Morpheus Users” group can access Morpheus in any capacity. Users who are also in the “Morpheus Admins” group will take on the System Admin role in Morpheus. We’ll see later when the integration is configured how Morpheus roles can be mapped to AD groups. In the same AD server, I have two users. John Smith is in groups “Morpheus Users” and “Morpheus Admins”. John Jones is only in the “Morpheus Users” group.



Knowing the AD scheme and the requirements for Morpheus user roles, we can begin the process of creating the integration. Identity integrations are specific to each Tenant so begin by navigating to the Tenant detail page (Administration > Tenants > Selected Tenant) and clicking *IDENTITY SOURCES*. On setting the type to “Active Directory,” the form will update with the needed fields. Note the following basic fields:

- **AD SERVER:** The IP address or hostname for the Active Directory Server
- **DOMAIN:** The AD domain in which the relevant users and groups reside
- **BINDING USERNAME:** A server user which has access to relevant objects on the AD server. In my example, I've used the in-built Administrator user which is the easiest option. Other users may be used depending on your organization's IT security policies but the integration with Morpheus will not work properly if the user does not have the needed access
- **BINDING PASSWORD:** The password for the user in the prior field

With the basic configurations completed, the remaining configurations will affect Morpheus user and role generation. A **REQUIRED GROUP** is optional and is a group the user *must* be in to have any Morpheus access. Here we require that users be in the “Morpheus Users” group. A **DEFAULT ROLE** is required and will be assigned to all users regardless of any additional roles they may be assigned based on their AD group membership. Beyond that, all other Morpheus roles will be listed here and an AD group name can be associated with as many as you required. In this example, we are giving users in the “Morpheus Admins” group the Morpheus System Admin role. Though we did not use them, it's worth pointing out that **ENABLE ROLE MAPPING PERMISSION** will give administrators in the Tenant the ability to update the AD role mappings (though they will not have access to the core integration fields such as **AD SERVER**, **DOMAIN**, or binding user details). **MANUAL ROLE ASSIGNMENT** allows users to manually update Morpheus roles outside of the automatic mappings created by the AD integration.

EDIT IDENTITY SOURCE✕

Identity Source

TYPE

ACTIVE

NAME

DESCRIPTION

Active Directory Connection

AD SERVER

DOMAIN

USE SSL Yes No

BINDING USERNAME

BINDING PASSWORD

REQUIRED GROUP

Include Member Groups

Role Mappings

DEFAULT ROLE

AZRA CATALOG USER

MULTI-TENANT ADMIN

SYSTEM ADMIN

TEST MULTITENANT

USER ADMIN

Role Options

ENABLE ROLE MAPPING PERMISSION

MANUAL ROLE ASSIGNMENT

DEACTIVATE DELETE SAVE CHANGES

With the above integration steps completed, users can now log into Morpheus and a user account with correct roles will automatically be created. In our example case, John Smith has logged in and we can see he is assigned the default role as well as the System Admin role based on his AD group associations. Going forward, Morpheus will continue to sync any changes to these users. For example, Morpheus roles may be updated based on changing AD groups or user access may be completely revoked by disabling the user in AD.

The screenshot shows the Morpheus Administration interface. The top navigation bar includes the Morpheus logo, a search bar, and the user's name 'Alex Harker'. Below the navigation bar are tabs for Operations, Provisioning, Library, Infrastructure, Backups, Monitoring, Tools, and Administration. The Administration tab is active, showing a sub-menu with Users and User Groups. The Users page has a search bar, buttons for IDENTITY SOURCES, + CREATE USER, and a settings icon. A table lists users with columns for DISPLAY NAME, USERNAME, EMAIL, and ROLE. The role 'Azra Catalog User, System Admin' is highlighted with a red box.

DISPLAY NAME	USERNAME	EMAIL	ROLE
Alex Harker	aharker	aharker@morpheusdata.com	System Admin
John Smith	jsmith	jsmith@example.com	Azra Catalog User, System Admin

1. Navigate to Administration > Tenants
2. Select a Tenant
3. Select *IDENTITY SOURCES*
4. Select + *ADD IDENTITY SOURCE*
5. Set the TYPE to “Active Directory”
6. Populate the following:

Name A friendly name in Morpheus for the AD integration

AD Server The Hostname or IP address of AD Server

Domain The AD domain in which the relevant user and group objects reside

USE SSL Indicates whether SSL should be used for communication with the AD server. Morpheus will connect over port 389 for non-secure LDAP and port 636 for secure LDAP, ensure Morpheus can connect to the AD server over the correct port

Binding Username A username for a service account which has access to relevant objects (users, groups, etc.). For ease, the “Administrator” user may be used

Binding Password The password for the above account

Required Group The AD group users must be in to have access (optional, see example in the prior section)

Default Role The default role a user is assigned when they are in the required group or if no specific group mapping applies to the user (see example in prior section)

ENABLE ROLE MAPPING PERMISSION When selected, Tenant users with appropriate rights to view and edit Roles will have the ability to set role mapping for the Identity Source integration. This allows the Tenant user to edit only the role mappings without viewing or potentially editing the basic Identity Source configuration (AD server, domain, binding user details, etc)

MANUAL ROLE ASSIGNMENT When selected, administrators can manually edit Roles for users created through this identity source integration from the user detail page (Administration > Users > Selected user)

Note: For more on Identity Source role mapping permissions, see the [associated guide](#) in our KnowledgeBase.

7. Select *SAVE CHANGES*.

Now allowed AD users can login to Morpheus via their Active Directory credentials and a User will be automatically generated to Morpheus with matching metadata and mapped Role permissions.

Note: Sub-tenant Morpheus API authentication for Active Directory generated users is not currently supported.

Whitelabeling

Morpheus allows each Tenant to be customized with its own look and feel through whitelabeling features. Your organization can replace the stock Morpheus logo and color scheme with your own logos and colors. Whitelabel settings are held in Administration > Settings > Whitelabel and are mostly self-explanatory. Upload logos and set custom hexadecimal colors in each field. When done, be sure to set the “Enable Whitelabel” configuration and click *SAVE* all the way at the bottom of the page. If needed, review a more detailed explanation of each [whitelabel setting](#) in Morpheus documentation.

Whitelabel Settings

Enable Whitelabel

Support Additional Themes

Appliance Name

Header Logo [Browse](#)

RESET

Preferred Image Size (500x76)

Disable Support Menu

Support Menu Links [ADD LINK](#)

Customize support links. Label Code can be used for translations and is optional. Be sure to specify fully qualified url if linking to external sites.

Security Banner

Scoping Clouds to Tenants

With the Tenant now created and seeded with a User and Roles, we're ready to look at how to set up useful provisioning resources. Of course, Tenants are complete (yet isolated) Morpheus environments which can create their own Cloud integrations or utilize integrations with other third party technologies (assuming they haven't been restricted from doing so by the Tenant Role). However, Master Tenant administrators may also choose to share integrations created in the Master Tenant down to Subtenant users.

Many constructs in Morpheus have a "Visibility" attribute when viewed from the Master Tenant. This attribute controls whether the construct is "Private" (in other words reserved for a selected Tenant) or Public (available to all Tenants). In our example, we're going to expose an Amazon AWS Cloud integrated in the Master Tenant down to our newly-created Subtenant. To set the Cloud visibility, use the following steps:

1. Navigate to the Clouds list page (Infrastructure > Clouds)
2. Edit the Cloud by clicking the pencil icon () in its row
3. Toggle the VISIBILITY attribute to "Public"
4. Click *SAVE CHANGES*

EDIT CLOUD [X]

NAME

CODE

LOCATION

VISIBILITY

TENANT

ENABLED

AUTOMATICALLY POWER ON VMS

The Cloud is now exposed to the Subtenant but we also need to expose a Resource Pool (A VPC, in Amazon parlance) for Subtenant users to provision into. We do this from the Cloud detail page (Infrastructure > Clouds) > Click on selected Cloud.

1. Select the Resources subtab
2. Click ACTIONS next to the selected Resource Pool (VPC)
3. Click Edit
4. Expand the "Tenant Permissions" section
5. Set Visibility to Public (alternatively, you could keep the Private visibility and add the new Tenant in the next field)
6. Click *SAVE CHANGES*

The screenshot shows the Morpheus Infrastructure Resources page. The navigation bar includes Operations, Provisioning, Library, Infrastructure (selected), Backups, Monitoring, Tools, and Administration. The Infrastructure sub-tab includes Groups, Clouds, Clusters, Compute, Network, Load Balancers, Storage, Trust, and Boot. The dashboard displays several metrics: COST THIS MONTH (\$0), COST HISTORY (2.89), MAX CPU (0%), MEMORY (0%), and STORAGE (12%). Below these are counts for CONTAINER HOSTS (0), HYPERVISORS (0), BARE METAL (0), VIRTUAL MACHINES (3), and DISCOVERED (0). The Resources sub-tab is active, showing a search bar and a '+ ADD RESOURCE POOL' button. A table lists resource pools with columns for NAME, DESCRIPTION, VISIBILITY, DEFAULT, and TENANT. One pool is listed with ID 'vpc-01cc4fdb5477b2fcd (us-east-2)', description 'vpc-01cc4fdb5477b2fcd - vpc-01cc4fdb5477b2fcd - 172.31.0.0/16', visibility 'Private', and default tenant 'Morpheus'. An 'ACTIONS' dropdown menu is visible with 'Edit' and 'Delete' options.

Repeat the same process for Networks and Security Groups, both of which are contained in the Networks subtab of the Cloud detail page.

Groups

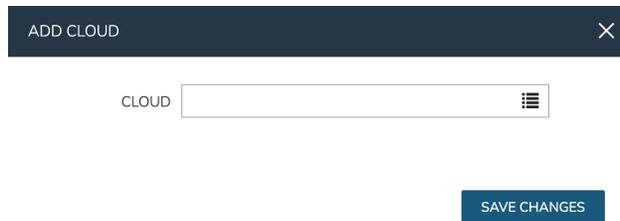
The last thing to take a look at before Library items and provisioning are Groups. Groups define which resources a User can access. Group access is defined by the User Role and many resources have a Group access component which allows administrators to control User access to things like Clouds, Networks, Datastores, Resource Pools, Folders and more.

Groups exist solely within the Tenant that creates them, they cannot be shared down from Master Tenant to Subtenants. In order to create a Group for our new Tenant, impersonate the User we've created in previous steps. Once working in the Tenant, use the following steps:

1. Navigate to Infrastructure > Groups
2. Click + *CREATE*
3. Set a **NAME** for the Group and click *SAVE CHANGES*

We'll now add the shared Cloud to the Group:

1. Click on the name of the Group to access the Group detail page
2. Click on the Clouds tab
3. Click + *ADD*
4. Use the typeahead field to select the shared Cloud
5. Click *SAVE CHANGES*



ADD CLOUD

CLOUD

SAVE CHANGES

At this point the basic configuration steps are complete and we will attempt a provisioning test using built-in Instance Types which are preinstalled with Morpheus.

Provisioning Test

At this point, you can conduct a test by attempting to provision one of the basic Instance Types shipped with Morpheus for demonstration purposes. To test, we'll provision a basic Ubuntu box to our AWS Cloud.

1. Navigate to Provisioning > Instances
2. Click + *ADD*
3. Select the Ubuntu Instance Type
4. Click *NEXT*
5. Select the Group and Cloud noting that only resources shared with the Subtenant are selectable
6. Name the Instance
7. Click *NEXT*
8. On the *CONFIGURE* tab, select a Resource Pool, Network, and Security Group once again noting that only resources shared with the Subtenant are selectable
9. Click *NEXT*
10. Click *NEXT* once again
11. Click *COMPLETE*

CREATE INSTANCE
✕

TYPE >
 GROUP >
CONFIGURE >
AUTOMATION >
REVIEW

Configuration Options

VERSION	22.04	▼					
LAYOUT	Amazon VM	▼					
PLAN	T2 Micro - 1 Core, 1GB Memory	▼					
	Cores: 1 Memory: 1 GB Price: \$10.4912 / Month ⓘ						
RESOURCE POOL	vpc-01cc4fdb5477b2fcd (us-east-2)	▼					
VOLUMES	root	20	GB	gp2	▼	+	
NETWORKS	172.31.0.0/20 (subnet-06bc579d44)	▼	DHCP				+
SECURITY GROUPS	default	▼					+
PUBLIC IP	Subnet Default	▼					

▶ User Config

▶ Advanced Options

PREVIOUS
NEXT

After a minute or two, the Instance should be fully provisioned.

Exposing Custom Instance Types to Subtenants

Master Tenant administrators can also share custom Library items down to Subtenants. Do do this, navigate to Library > Blueprints and select an Instance Type. Click the pencil () icon in the row for the selected Instance Type to edit it. As we've done before, update the VISIBILITY setting to Public. In addition, ensure the User Role for the Subtenant User grants access to the selected Instance Type. Once configurations have been made, conduct a similar test to the previous section. When you attempt to provision a new Instance, the new Instance Type should be selectable and the User should be able to provision the custom Instance Type in the same was as the default Instance Type was provisioned in the previous step.

5.13.15 Automation Integrations

Ansible

Overview

Ansible is a configuration management engine that is rapidly growing in popularity in the IT and DevOPS community. While it lacks some of the benefits at scale that solutions such as Salt, Chef, or Puppet offer. It is very easy to get started and allows engineers to develop tasks in a simplistic markup language known as YAML. Morpheus integrates with an existing repository of playbooks as the master in a master-slave Ansible architecture.

Morpheus not only supports Ansible but greatly enhances Ansible to do things that it could not do in its native form. For example, Ansible can now be configured to run over the Morpheus agent communication bus. This allows playbooks to be run against instances where SSH/WinRM access may not be feasible due to networking restrictions or other firewall constraints. Instead it can run over the Morpheus Agent which only requires port 443 access back to the Morpheus appliance URL.

This integration supports both Linux-based and Windows platforms for playbook execution and can also be configured to query secrets from Morpheus Cypher services (similar to Vault).

Requirements

- Minimum Ansible Version Requirement is 2.7.x
- For agentless non-commandbus, sshpass is required
- For Windows non-agent command bus, pywinrm is required
- **Integrations:** Ansible User Role Permission required for access to Ansible Details Pages and Ansible tabs in Groups and Clouds
- Calling Morpheus Cypher secrets into Ansible scripts requires the Python `requests` module which is not part of the Python standard library. For Ubuntu 18.04 or 20.04 run `sudo apt install python-requests` or `sudo apt install python3-requests`, respectively. For RHEL/CentOS, run `sudo yum install python2-requests`

Note: Installing Ansible on the Morpheus appliance is a requirement. In some cases, this is handled automatically but in certain situations you may have to install manually. See the section below on [troubleshooting Ansible](#) for installation steps.

Add Ansible Integration

1. Navigate to Library > Integrations and select + *New Integration*
2. Select Integration Type “Ansible”
3. Populate the following fields:

Name Name of the Ansible Integration in Morpheus

Enabled Enabled by default

Ansible Git URL https or git url format of the Ansible Git repo to use

Keypair For private Git repos, a keypair must be added to Morpheus and the public key added to the git account.

Playbooks Path Path of the Playbooks relative to the Git url.

Roles Path Path of the Roles relative to the Git url.

Group Variable Path Path of the Group Variables relative to the Git url.

Host Variables Path Path of the Host Variables relative to the Git url.

Use Ansible Galaxy Install roles defined in `requirements.yml`

Enable Verbose Logging Enable to output verbose logging for Ansible task history

Use Morpheus Agent Command Bus Enable for Ansible Playbooks to be executed via Morpheus Agent Command Bus instead of SSH

4. Save Changes

Once you have completed this section and saved your changes you can set up a Cloud or Group to utilize this integration.

Ansible on Windows

When executing Ansible playbooks on Windows platforms, a few requirements must be met:

- `pywinrm` may need to be installed on the Morpheus Appliance via `pip install pywinrm`
- An Ansible Integration must be scoped to a Group or Cloud for Ansible to execute on Windows, as Morpheus assumes Ansible local when no group or cloud is scoped to Ansible. The playbooks do not need to be executed solely in the Group or Cloud, one just needs to be scoped to an Ansible Integration for Ansible Windows to run properly.

Scope Ansible Integration to a Cloud

1. Navigate to *Infrastructure > Clouds*
2. Edit the target Cloud
3. Expand the *Advanced Options* section
4. In the *Config Management* dropdown, select the Ansible Integration.
5. Save Changes

Once an Ansible integration is added to a Cloud, a new “ANSIBLE” tab will appear on the Cloud details page, populated with the Ansible integrations Playbook and Roles, as well as an editable Inventory list.

Scope Ansible Integration to a Group

1. Navigate to *Infrastructure > Groups*
2. Edit the target Group
3. Expand the *Advanced Options* section
4. In the *Config Management* dropdown, select the Ansible Integration.
5. Save Changes

Once an Ansible integration is added to a Group, a new “ANSIBLE” tab will appear on the Group details page, populated with the Ansible integrations Playbook and Roles, as well as an editable Inventory list.

Provisioning Options

When provisioning Instances into a Cloud or Group with a Ansible Integration added, an *Ansible* section will appear in the Config section of the provisioning wizard. By default, Ansible is enabled, but can be disabled by expanding the *Ansible* section and unchecking *Enable Ansible*.

Ansible Integration Provisioning options:

Enable Ansible Select to bootstrap

Ansible Group Ansible Inventory Group. Use existing group or enter a new group name to create a new group. Leaving this field blank will place instance in the “unassigned” inventory group.

Note: An instance can belong to multiple groups by separating group names with a comma

Playbook Playbook(s) to run. The .yml extension is optional.

Running Playbooks

Playbooks can also be run on all inventory groups, individual groups, or added as a task and ran with workflows.

To run Ansible on all or a single inventory group, in the Ansible tab of the Morpheus Group page, select the *Actions* dropdown and click *Run*.

In the *Run Ansible* modal, you can then select all or an individual group, and then all or a single Playbook, as well as add custom tags.

Playbook's can also be added as tasks to workflows in the Library > Automation section, and then selected in the Automation pane during provisioning of new instances, when creating app blueprints, or ran on existing instances using the *Actions > Run Workflow* on the Instance or Host pages.

Using variables

Morpheus variables can be used in playbooks.

Use Case:

Create a user as instance hostname during provisioning.

Below is the playbook. Add this playbook to a task and run it as a workflow on the instance.

```
---
- name: Add a user
  hosts: all
  gather_facts: false
  tasks:
    - name: Add User
      win_user:
        name: "{{ morpheus['instance']['hostname'] }}"
        password: "xxxxxxx"
        state: present
```

Note: `{{ morpheus['instance']['hostname'] }}` is the format of using Morpheus Variables

Create a user with a name which you enter during provisioning using a custom Instance type.

This instance type has a *Text* Input that provides a text box to enter a username. The `fieldName` of the Input in this c

```
---
- name: Add a user
  hosts: all
  gather_facts: false
  tasks:
    - name: Add User
      win_user:
        name: "{{ morpheus['customOptions']['username'] }}"
        password: "xxxxxxx"
        state: present
```

Note: `{{ morpheus['customOptions']['username'] }}` will be the format.

Using Secrets

Another great feature with using Ansible and Morpheus together is the built in support for utilizing some of the services that Morpheus exposes for automation. One of these great services is known as Cypher (please see documentation on *Cypher* for more details). Cypher allows one to store secret data in a highly encrypted way for future retrieval. Referencing keys stored in cypher in your playbooks is a matter of using a built-in lookup plugin for ansible.

```
- name: Add a user
  win_user:
    name: "myusername"
    password: "{{ lookup('cypher', 'secret=password/myusername') }}"
    state: present
```

By using the `{{ lookup('cypher', 'secret=password/myusername') }}` syntax. One can grab the value directly out of the key for use. This lookup plugin also supports a few other fancy shortcuts. In this above example the `password/` mountpoint is capable of autogenerating passwords if they have not previously been defined and storing them within cypher for reference later.

Another capability is accessing properties from within a key in cypher. The value of a key can also be a JSON object which can be referenced for properties within. For example:

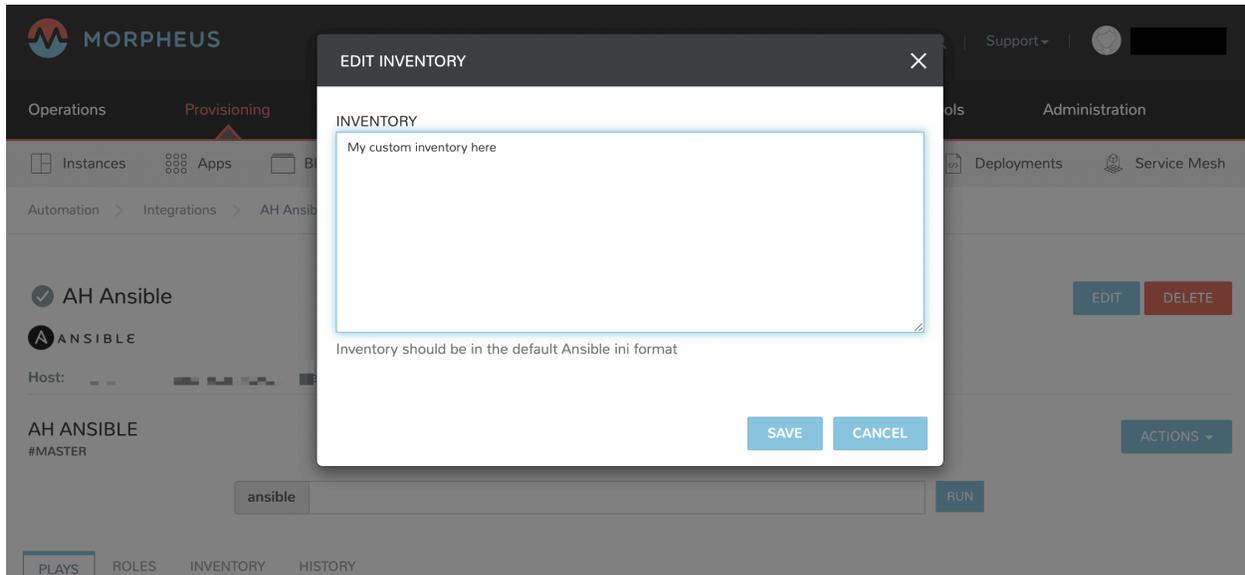
```
{{ lookup('cypher', 'secret=secret/myjsonobject:value') }}
```

This would grab the *value* property off the nested json data stored within the key.

Cypher is very powerful for storing these temporary or permanent secrets that one may need to orchestrate various tasks and workflows within Ansible.

Custom Inventory Entries

With Morpheus it is possible to add custom inventory entries that exist outside of morpheus host/server entry. This is global across cloud or group and is done on the integration details page of the Ansible integration. To add a custom inventory entry navigate to Library > Integrations > (Your specific Ansible integration). Click on the ACTIONS button, then click EDIT INVENTORY. Inventory should be in the default Ansible ini format.



Using Ansible over the Morpheus Agent Command Bus

In many environments, there may be security restrictions on utilizing SSH or WinRM to run playbooks from an Ansible server on the appliance to a target machine. This could be due to being a customer network (in the environment of an MSP), or various security restrictions put in place by tighter industries (i.e. Government, Medical, Finance).

Ansible can get one in trouble in a hurry. It is limited in scalability due to its fundamental design decisions that seem to bypass concepts core to all other configuration management frameworks (i.e. SaltStack, Chef, and Puppet). Because of its lack of an agent, the Ansible execution binary itself has to handle all the load and logic of executing playbooks on all the machines in the inventory of an Ansible project. This differs from other tools where the workload is distributed across the agents of each vm. Because of this (reaching out) approach, Ansible is very easy to get started with, but can be quite a bit slower as well as harder to scale up. However, Morpheus offers some solutions to help mitigate these issues and increase scalability while, at the same time improving security.

How does the Morpheus Agent Command Bus Work?

One of the great things about Morpheus is its Agent Optional approach. This means that this functionality can work without the Agent, however the agent is what adds the security benefits being represented here. When an instance is provisioned (or converted to managed) within Morpheus, an agent can be installed. This agent opens a secure websocket back to the Morpheus appliance (over port 443). This agent is responsible for sending back logs, guest statistics, and a command bus for automation. Since it is a WebSocket, bidirectional communication is possible over a STOMP communication bus.

When this functionality is enabled on an Ansible integration, a *connection_plugin* is registered with Ansible of type *morpheus* and *morpheus_win*. These direct bash or powershell commands, in their raw form, from Ansible to run over a Morpheus api. The Ansible binary sends commands to be executed as an https request over the API utilizing a one time execution lease token that is sent to the Ansible binary. File transfers can also be enacted by this API interface.

When Morpheus receives these commands, they are sent to the target instances agent to be executed. Once they have completed a response is sent back and updated on the *ExecutionRequest* within Morpheus. Ansible polls for the state and output on these requests and uses those as the response of the execution. This means Ansible needs zero knowledge of a machines target ip address, nor its credentials. These are all stored and safely encrypted within Morpheus.

It has also been pointed out that this execution bus is dramatically simpler than utilizing *pywinrm* when it comes to orchestrating Windows as the winrm configurations can be cumbersome to properly setup, especially in tightly secured Enterprise environments.

Using Ansible Galaxy

Morpheus can use a `requirements.yml` file to define Ansible roles to download prior to running your playbook. Place `requirements.yml` into the root of your Git repository and make sure *Use Ansible Galaxy* is checked in the integration. Roles will be installed in the root of the repository if a directory is not defined in *Roles Path*.

- Example requirements.yml:

```
- src: https://github.com/geerlingguy/ansible-role-java
  name: java
```

- Example playbook.yml:

```
- hosts: all
  gather_facts: true
  roles:
    - java
```

Troubleshooting Ansible

- When a workflow is executed manually, the Ansible run output is available in the Instance History tab. Select the **i** bubble next to the Ansible task to see the output. You can also see the run output in the ui logs in `/var/log/morpheus/morpheus-ui/current` which can be tailed by running `morpheus-ctl tail morpheus-ui`.
- Verify Ansible is installed on the Morpheus Appliance.

Ansible should be automatically installed but certain OS or network conditions can prevent the automated install. You can confirm installation by running `ansible --version` in the Morpheus appliance, or by viewing the Ansible integration details page (Administration > Integrations > Select Ansible Integration). We also see it in the Ansible tab of a Group or Cloud scoped to Ansible, just run `--version` as ansible is already included in the command.

If Ansible is not installed, follow these instructions to install, or use your preferred installation method:

Ubuntu:

```
sudo apt-get install software-properties-common
sudo apt-add-repository ppa:ansible/ansible
sudo apt-get update
sudo apt-get install ansible
```

CentOS:

```
sudo yum install epel-release
sudo yum install ansible
```

Then create the working Ansible directory for Morpheus:

```
mkdir -p /opt/morpheus/.ansible/tmp
chown -R morpheus-local. /opt/morpheus/.ansible
chmod -R 775 /opt/morpheus/.ansible
```

- Validate the git repo is authorizing and the paths are configured correctly.

The public and private ssh keys need to be added to the Morpheus appliance via “Infrastructure > Keys & Certs” and the public key needs to be added to the git repo via user settings. If both are set up right, you will see the playbooks and roles populate in the Ansible Integration details page.

- The Git Ref field on playbook tasks is to specify a different git branch than default. It can be left to use the default branch. If your playbooks are in a different branch you can add the brach name in the Git Ref field.
- When running a playbook that is in a workflow, the additional playbooks fields do not need to be populated, they are for running a different playbook than the one set in the Ansible task in the Workflow, or using a different Git Ref.
- If you are manually running Workflows with Ansible tasks on existing Instances through *Actions > Run Workflow* and not seeing results, set the Provision Phase on the Ansible task to Provision as there may be issues with executing tasks on other phases when executing manually.

Ansible Tower

Overview

Morpheus supports Ansible Tower for configuration management. Morpheus accomplishes this by integrating with an existing instance running Ansible Tower (AT) 3.3.0-1 and earlier. The username and password required for integration can be a user with admin access or a user with project admin access

Morpheus will import the current Inventory, Templates, Hosts, Groups and Projects. In the integration view it will add a Job tab which will have information of all the jobs executed from Morpheus.

Note: This integration will not import data of the jobs which are not executed from Morpheus.

Add Ansible Tower Integration

1. Navigate to Administration > Integrations and select + *New Integration*
2. Select Integration Type “Ansible Tower”
3. Populate the following fields:
 - Name: Name of the Ansible Tower Integration in Morpheus
 - Enabled: To disable the integration, uncheck this option
 - Ansible Tower URL: An HTTPS or HTTP Ansible Tower URL
 - Username: The user Morpheus would use to communicate with Ansible Tower
 - Password: Enter the password. Password is encrypted and saved in DB
 - API Version: This drop down has one option (v2) for now but may have others in future
4. Save Changes

Once you have completed this section and saved your changes you can set up a Cloud or Group to utilize this integration.

Scope Ansible Tower Integration to a Cloud

All instances provisioned in this cloud will have the Ansible Tower config option during provisioning. See below the Provisioning Options for more details about the options.

1. Navigate to *Infrastructure > Clouds*
2. Edit the target Cloud
3. Expand the *Advanced Options* section
4. In the *Config Management* dropdown, select the Ansible Tower Integration.
5. Save Changes

Scope Ansible Tower Integration to a Group

All instances provisioned in this Group will have the Ansible Tower config option during provisioning in any cloud part of the Group. See below the Provisioning Options for more details about the options.

1. Navigate to *Infrastructure > Groups*
2. Edit the target Group
3. Expand the *Advanced Options* section
4. In the *Config Management* dropdown, select the Ansible Tower Integration.
5. Save Changes

Provisioning Options

When provisioning Instances into a Cloud or Group with a Ansible Tower Integration added, an *Ansible Tower* section will appear in the Config section of the provisioning wizard. By default, Ansible Tower is enabled, but can be disabled by expanding the *Ansible Tower* section and unchecking *Enable Ansible Tower*.

Ansible Integration Provisioning options:

Enable Ansible Tower Select to bootstrap

Inventory A list of Inventory available in Ansible Tower will appear in the drop down. Select an existing inventory. The instance will be added to the inventory selected.

Ansible Group Enter the name of a new or an existing Group in the inventory selected above.

Template

Select an existing template or select the option 'Create New Template'. If 'Create New Template' is selected below fields w

Template Name Enter the template name

Project Select an existing project from the drop down options

Playbook Select a playbook from the dropdown to be associated with the template. Note: Morpheus doesn't store a local copy of the playbooks visible in Ansible Tower. SCM or local path for playbooks should be maintained in Ansible Tower.

Execute Mode

Select one of the options from the dropdown

Limit to instance This will execute the template on the instance provisioned.

Limit to Group This will execute the template on all hosts attached to the group entered in the ‘Ansible Group’ field.

Run for all This will execute the template on all hosts in the inventory

Skip execution This will skip the execution of the template on the instance provisioned.

Scoping Ansible Tower Jobs to Tenant-Default Inventories

Users in the Primary Tenant have an additional Inventory execution option when creating Ansible Tower Job-type Tasks. When making a selection in the Inventory field, “Use Tenant Default” may be selected rather than a specific Inventory. This is because Ansible Tower Jobs created in the Primary Tenant may be shared publicly to other Tenants through public Workflows or when associated with public Library items.

The screenshot shows a 'NEW TASK' form with the following fields and values:

- NAME: [Empty]
- CODE: [Empty]
- TYPE: Ansible Tower Job
- TOWER INTEGRATION: Lets Ansible Tower
- INVENTORY: Use Tenant Default (highlighted with a red box)
- GROUP: [Empty]
- JOB TEMPLATE: Ans User
- SCM OVERRIDE: [Empty]
- EXECUTE MODE: Limit to instance
- EXECUTE TARGET: Resource

Buttons: 'Execute Options' (with a right-pointing arrow), 'SAVE CHANGES'.

When this option is selected and the Task is run in a Subtenant, it will automatically be run against the default Inventory which is configured for the Subtenant. The next section includes steps for associating Tenants and default Inventories.

Important: An Ansible Tower Job configured to run against a Tenant-default Inventory will fail when run by a user whose Tenant does not have a default Inventory set.

Setting Default Inventories for Tenants

When creating or editing Ansible Tower integrations, navigate to the Inventory tab to view all Inventories synced from the selected integration. Click “Permissions” inside the “MORE” action menu at the end of a row for the selected Inventory. Within the PERMISSIONS modal, there is a single typeahead field where a Tenant can be selected. Once the Tenant is selected, click *SAVE CHANGES*. Now back on the Inventory list view, you’ll see the default Tenant which is associated with each Inventory.

Note: Tenants may only be associated with one Inventory, though an Inventory can have multiple Tenant associations. If a Tenant is selected to be associated with a new Inventory, its association with a previous Inventory will automatically be removed.

The screenshot shows the Morpheus Administration interface. The top navigation bar includes 'MORPHEUS', a search bar, and user information 'A Harter'. Below the navigation bar are tabs for 'Operations', 'Provisioning', 'Infrastructure', 'Backups', 'Logs', 'Monitoring', 'Tools', and 'Administration'. Under 'Administration', there are sub-tabs for 'Tenants', 'Plans & Pricing', 'Roles', 'Users', 'Integrations', 'Policies', 'Health', and 'Settings'. The 'Integrations' section is active, showing 'Labs Ansible Tower' with 'EDIT' and 'DELETE' buttons. Below this, there are tabs for 'Summary', 'Templates', 'Jobs', and 'Inventory'. The 'Inventory' tab is selected, displaying a table with the following data:

NAME	DESCRIPTION	TENANT DEFAULT	
Windows			MORE
Unix		Morpheus QA	MORE
OA		Auto Tenant 2	MORE
Demo Inventory			MORE
CTDemoInventory			MORE

Ansible Tower Configuration

When using an Ansible Tower task type or associating the Ansible Tower integration with a cloud/group, there are a few options that can be configured:

- Inventory
- Group
- Job Template
- Execute Mode

Prompt at Launch

Some options used to configure your deployments have the related option of **Prompt at Launch** in Ansible Tower, which should be enabled on the template to be chosen in the **Job Template** field. If **Prompt at Launch** is not enabled, the values configured on the template in Ansible Tower will be used instead. **Prompt at Launch** can be seen below on the **Inventory** and **Limit** fields:

CT - Hello World



- DETAILS
- PERMISSIONS
- NOTIFICATIONS
- COMPLETED JOBS
- SCHEDULES

* NAME CT - Hello World	DESCRIPTION Output Hello World	* JOB TYPE ? Run	<input type="checkbox"/> PROMPT ON LAUNCH
* INVENTORY ? CT-Demo-Inventory	* PROJECT ? CT-Demo-Project	* PLAYBOOK ? helloworld.yml	
CREDENTIAL ? ctaylor	FORKS ? DEFAULT	LIMIT ?	<input checked="" type="checkbox"/> PROMPT ON LAUNCH
* VERBOSITY ? 0 (Normal)	JOB TAGS ?	SKIP TAGS ?	<input type="checkbox"/> PROMPT ON LAUNCH
LABELS ?	INSTANCE GROUPS ?	SHOW CHANGES ? OFF	<input type="checkbox"/> PROMPT ON LAUNCH

- OPTIONS
- Enable Privilege Escalation ?
 - Allow Provisioning Callbacks ?
 - Enable Concurrent Jobs ?
 - Use Fact Cache ?

EXTRA VARIABLES ? YAML JSON PROMPT ON LAUNCH

```
1 ---
```

CANCEL SAVE

Group

The **Group** field is optional but a group can be entered into the field to associate the host to, in the target inventory. If the group is existing, then the instance will be associated as a host to that group. If the group does not exist, the group will be created and the instance will be associated as a host to that group.

KG-Synced-Inventory

DETAILS PERMISSIONS GROUPS **HOSTS** SOURCES COMPLETED JOBS

SEARCH KEY

HOSTS ▲	RELATED GROUPS
<input type="checkbox"/> <input type="button" value="ON"/> <input type="button" value="!"/> 10.32.20.179	<input type="button" value="newgroup"/> <input type="button" value="x"/>
<input type="checkbox"/> <input type="button" value="ON"/> <input type="button" value="!"/> testname1.localdomain	<input type="button" value="premade"/> <input type="button" value="x"/>

Inventory

When provisioning on a cloud with a configured Ansible Tower integration or using an Ansible Tower task type against an instance, the instance will be added as a host to the inventory chosen in the **Inventory** field. As mentioned, if specified, these instance will be associated with groups in the inventory as well. When using an inventory that syncs from a project, the instance will still be added as a host in the inventory, in addition to the sync'd inventory. This means that Ansible Tower will aggregate the manually added hosts from Morpheus with the sync'd project inventory. However, if the **Overwrite** option is enabled on the source for the project that contains the inventory, any hosts added by Morpheus **will be overwritten**. In some cases, a separate Morpheus inventory may be desired, if **Overwrite** is required on your sources.

morpheus-ansible-example as source ✕

DETAILS NOTIFICATIONS SCHEDULES

* NAME: morpheus-ansible-example as source DESCRIPTION: DESCRIPTION SOURCE: Sourced from a Project

SOURCE DETAILS

CREDENTIAL: CREDENTIAL PROJECT: KG-Project INVENTORY: inventory.ini

VERBOSITY: 1 (INFO) UPDATE OPTIONS: Overwrite Overwrite Variables Update on Launch Update on Project

ENVIRONMENT VARIABLES: [YAML] [JSON]

1 ---

CANCEL SAVE

OVERWRITE

If checked, any hosts and groups that were previously present on the external source but are now removed will be removed from the Tower inventory. Hosts and groups that were not managed by the inventory source will be promoted to the next manually created group or if there is no manually created group to promote them into, they will be left in the "all" default group for the inventory.

When not checked, local child hosts and groups not found on the external source will remain untouched by the inventory update process.

Passing extra_vars to Ansible Tower Job

When provisioning or when running Ansible Tower Jobs as Morpheus Tasks, you may pass the extra_vars stack to the Tower Job. First, ensure the Job Template has extra variables “Prompt on Launch” enabled as shown below:

CT - Hello World ✕

DETAILS PERMISSIONS NOTIFICATIONS COMPLETED JOBS SCHEDULES

* NAME: CT - Hello World DESCRIPTION: Output Hello World * JOB TYPE: Run PROMPT ON LAUNCH

* INVENTORY: CT-Demo-Inventory PROMPT ON LAUNCH * PROJECT: CT-Demo-Project * PLAYBOOK: helloworld.yml

CREDENTIAL: ctaylor PROMPT ON LAUNCH FORKS: DEFAULT LIMIT: PROMPT ON LAUNCH

* VERBOSITY: 0 (Normal) PROMPT ON LAUNCH JOB TAGS: PROMPT ON LAUNCH SKIP TAGS: PROMPT ON LAUNCH

LABELS: INSTANCE GROUPS: SHOW CHANGES: OFF PROMPT ON LAUNCH

OPTIONS: Enable Privilege Escalation Allow Provisioning Callbacks Enable Concurrent Jobs Use Fact Cache

EXTRA VARIABLES: [YAML] [JSON]

1 ---

CANCEL SAVE



The sample Playbook below is associated with the Tower Job Template.

```

---
- hosts: all
vars:
  Opensource_Team: "Customer"
tasks:
- name: Print Hello World
  debug:
    msg:
      - "Hello World {{ Opensource_Team }}. Here are Morpheus extra_vars: {{ morpheus }}"

```

After executing the Tower Job, we can see the variable stack surfaced into the results as defined in the Playbook:

```

4
5 TASK [Gathering Facts] ***** 04:32:59
6 ok: [ctspcen1001.localdomain]
7
8 TASK [Print Hello World] ***** 04:33:01
9 ok: [ctspcen1001.localdomain] => {
10   "msg": [
11     "Hello World Customer. Here are Morpheus extra_vars: {u'platfo
rm': u'linux', u'sshKey': None, u'maxCpu': None, u'sequence': 1, u'isD
ocker': False, u'ansibleTowerExecuteMode': u'executeHost', u'isEC2': F
alse, u'ansibleTowerJobTemplateId': u'66', u'nutanixUsr': None, u'evan
s': {u'HOST_INTERNAL': u'container357', u'INSTANCE_HOSTNAME': u'ctspce
n1001', u'IP_INTERNAL': u'10.30.20.105', u'IP_PUBLIC': u'10.30.20.10
5', u'HOST_EXTERNAL': u'ctspcen1001', u'PORT_22': 22}, u'groupName':

```

Use Case

You have Job template(s) in Ansible Tower to do post build config after the OS is deployed. The playbook with roles and tasks to do post build will add specific users and groups, install required packages, remove packages, disable services, change config for ntp, resolv, hosts etc. You want to add the instance to an existing Group/Inventory in Tower.

You can achieve this by adding the Ansible Tower Integration and then scope it to a Cloud or Group. While provisioning an instance, in the config stage you have the Ansible Tower section with option to select the post build job template, select the Inventory and provide an existing Group Name or if the Group doesn't exist Morpheus will create it and submit for provisioning.

Morpheus will provision the instance, once it is in the finalize state where the instance has an ip and has completed domain join if required, added user(s) or User Groups if specified then Morpheus will add the instance to the inventory and Group and run the Template which will do all the post build of the server.

The output of the post build template execution can be seen under Instance history.

Chef

Overview

Morpheus integrates with one or multiple Chef servers to be used for bootstrapping while provisioning or as tasks in workflows in the Automation section. These workflows can then be run during provisioning in the provisioning wizard Automation pane, or on an existing Instance by selecting Run Workflows from the Actions menu on the Instance detail page. Workflows can also be added to Instances in the blueprint and app sections.

Add Chef Integration

1. Navigate to Administration > Integrations and select + *New Integration*
2. Select Integration Type “Chef”
3. Populate the following fields:
 - Name: Name of the Chef Integration in Morpheus
 - Chef Endpoint: url of chef server api endpoint in <https://api.example.com> format. Do not add /organization/xxxx here, which is populated in the Chef Organization field
 - Chef Version: 12.3.0 by default, can be changed to use a different/more recent version of chef
 - Chef Organization: Chef Server Organization
 - Chef User: Chef Server User
 - User Private Key: The private key of the user with access to this chef server
 - Organization Validator: Validator key for the organization
4. Save Changes

The added Chef Integration is now available for use in Morpheus. The Chef Integration can be added to Clouds or Groups to auto-bootstrap nodes and specify Environment, Node ID, Runlist, Attributes and Tags when creating instances. The Chef integration can also be selected in the Chef Server dropdown when creating a Chef Bootstrap-type Task.

Note: Integrating Chef with Morpheus enables a user to bootstrap a node using Chef Bootstrap-type Tasks or Chef associations with Clouds or infrastructure Groups. It does not enable users to update run-lists on the Chef server for nodes which have already been bootstrapped.

Scope Chef Integration to a Cloud

1. Navigate to *Infrastructure > Clouds*
2. Edit the target Cloud
3. Expand the *Advanced Options* section
4. In the *Config Management* dropdown, select the Chef Integration.
5. Save Changes

Scope Chef Integration to a Group

1. Navigate to *Infrastructure > Groups*
2. Edit the target Group
3. Expand the *Advanced Options* section
4. In the *Config Management* dropdown, select the Chef Integration.
5. Save Changes

Provisioning Options

When provisioning Instances into a Cloud or Group with a Chef Integration added, a *Chef* section will appear in the Config section of the provisioning wizard. By default, Chef is enabled, but can be disabled by expanding the *Chef* section and unchecking *Enable Chef*.

Chef Integration Provisioning options:

Enable Chef Select to bootstrap

Chef Environment Populate Chef environment, or leave as *_default*

Chef Node ID Defaults to instance name, configurable.

Chef Runlist Add Runlist

CHEF ATTRIBUTES Add Chef Attributes

CHEF TAGS Add Chef tags

Puppet

Each Morpheus Puppet integration ties to a specific Puppet Master and makes it easy to install the Puppet Agent onto target Instances and Servers. Once integrated, we can trigger the agent installation by creating Puppet Agent Install Tasks through Morpheus automation or at provision time when spinning up instances in Clouds associated with a Puppet integration.

Add Puppet Integration

1. Navigate to Administration > Integrations
2. Click + *NEW INTEGRATION*
3. Select Integration type “Puppet”
4. Populate the following fields
 - Name: A friendly name for this Puppet integration in Morpheus
 - Enabled: When marked, this Puppet integration will be available to users at provision time and when creating Tasks
 - Puppet Master (Hostname): The resolvable DNS name to the Puppet Master, communicating on port 443 by default
 - Allow Immediate Execution: Yes or No (See additional details below)
5. Click *SAVE CHANGES*

NEW PUPPET INTEGRATION ×

NAME

ENABLED

PUPPET MASTER (HOSTNAME)

ALLOW IMMEDIATE EXECUTION? YES NO

PUPPET MASTER SSH USER

PUPPET MASTER SSH PASSWORD

SAVE CHANGES

Allow Immediate Execution

When adding or editing a Puppet integration, users have the option to “Allow Immediate Execution”. Using this feature will connect to the Puppet Master after the Puppet Agent is installed and force configuration application immediately. This is an optional convenience feature, if it’s not used, the Puppet Master will still apply configuration to the node on its own schedule. Using this feature requires supplying SSH credentials. If you do not wish to provide SSH credentials to Morpheus select “No” to opt out of this feature and they do not need to be provided.

Associating a Puppet Integration with a Cloud

By associating a Puppet Integration with a Cloud, users are able to automatically install the Puppet Agent, select a Puppet environment, and supply a Puppet Node Name at provision time. The integration association is made when adding or editing a Cloud integration in Morpheus.

1. Navigate to Infrastructure > Clouds
2. Identify the Cloud to associate with Puppet and edit it by clicking on the pencil icon at the right end of the row
3. In the EDIT CLOUD modal, expand the Advanced Options section
4. In the CONFIG MANAGEMENT dropdown, choose the Puppet integration
5. Save changes to the Cloud integration

Creating Puppet Agent Install Tasks

Puppet Agent Install Tasks automate the process of installing the Puppet Agent, selecting the Puppet environment, and supplying the Puppet Node Name. We can run this Task on-demand as needed for individual Instances or servers or add them to workflows to build a Puppet Agent installation step into larger automation suites.

1. Navigate to Library > Automation
2. Select the Tasks tab
3. Click + *ADD*
4. From the “Type” field, select Puppet Agent Install
5. Complete the fields as needed to target a specific Puppet Master and to identify Puppet environment and node names
6. Save changes

NEW TASK ×

NAME

CODE

TYPE

PUPPET MASTER

PUPPET NODE NAME

PUPPET ENVIRONMENT

Target Options

EXECUTE TARGET

▶ Execute Options

SAVE CHANGES

Installing Puppet Agent at Provision Time

With a Puppet integration associated with a Cloud (as described above), users can opt to install the Puppet Agent at provision time. When provisioning into an associated Cloud, a new “Puppet” section will appear on the CONFIGURE tab of the provisioning wizard. Here users can select a specific Puppet Master, select the Puppet environment, and select a Puppet node name. During provisioning, the Puppet Agent will be automatically installed and configured for the selected Master.

Salt

Overview

Morpheus integrates with an existing Salt Master for seamless deployment of Salt States to Minions provisioned from Morpheus .

Add Salt Integration

To get started browse to Admin > Integrations from within Morpheus .

Once there simply add a New Integration

The screenshot shows the Morpheus Admin console interface. At the top, there is a navigation bar with the Morpheus logo and the user name 'Adam Hicks'. Below the navigation bar, there are several menu items: Operations, Provisioning, Infrastructure, Backups, Logs, Monitoring, Services, and Admin (highlighted in red). Under the Admin menu, there are sub-menu items: Accounts, Service Plans, Roles, Users, Integrations (highlighted in blue), Provisioning, Monitoring, Backups, Logs, and Settings. The main content area is titled 'INTEGRATIONS' and features a '+ NEW INTEGRATION' button in a blue box. Below this, there is a table listing various integrations. Each row includes a logo, the integration name, the type, and edit/delete icons.

	NAME	TYPE		
	Labs Salt Master	saltMaster		
	morpheus-ansible	ansible		
	Labs Puppet Master	puppet		
	Labs Private	docker.registry		
	SNOW Approvals	serviceNow		
	Labs Chef	chef		
	Consul	consulRegistry		
	Labs Git A	git		
	My Private Docker Repo	docker.registry		
	Labs Salt Master 2	saltMaster		

And then scope the integration to your existing Salt Master by ip address. Make sure that the username entered is one with proper escalation privileges for running Salt, and point the Working Directory at the directory on your Master where your States live.

Note: Morpheus will allow you to run States from a git backend, but in v2.10 you will not see states from a git backend within Morpheus

INTEGRATION ✕

TYPE

NAME

SALT MASTER

SSH PORT

USERNAME

PASSWORD

KEY PAIR

WORKING DIRECTORY

SALT VERSION

SAVE CHANGES

Scope Salt Integration to Group Or Cloud

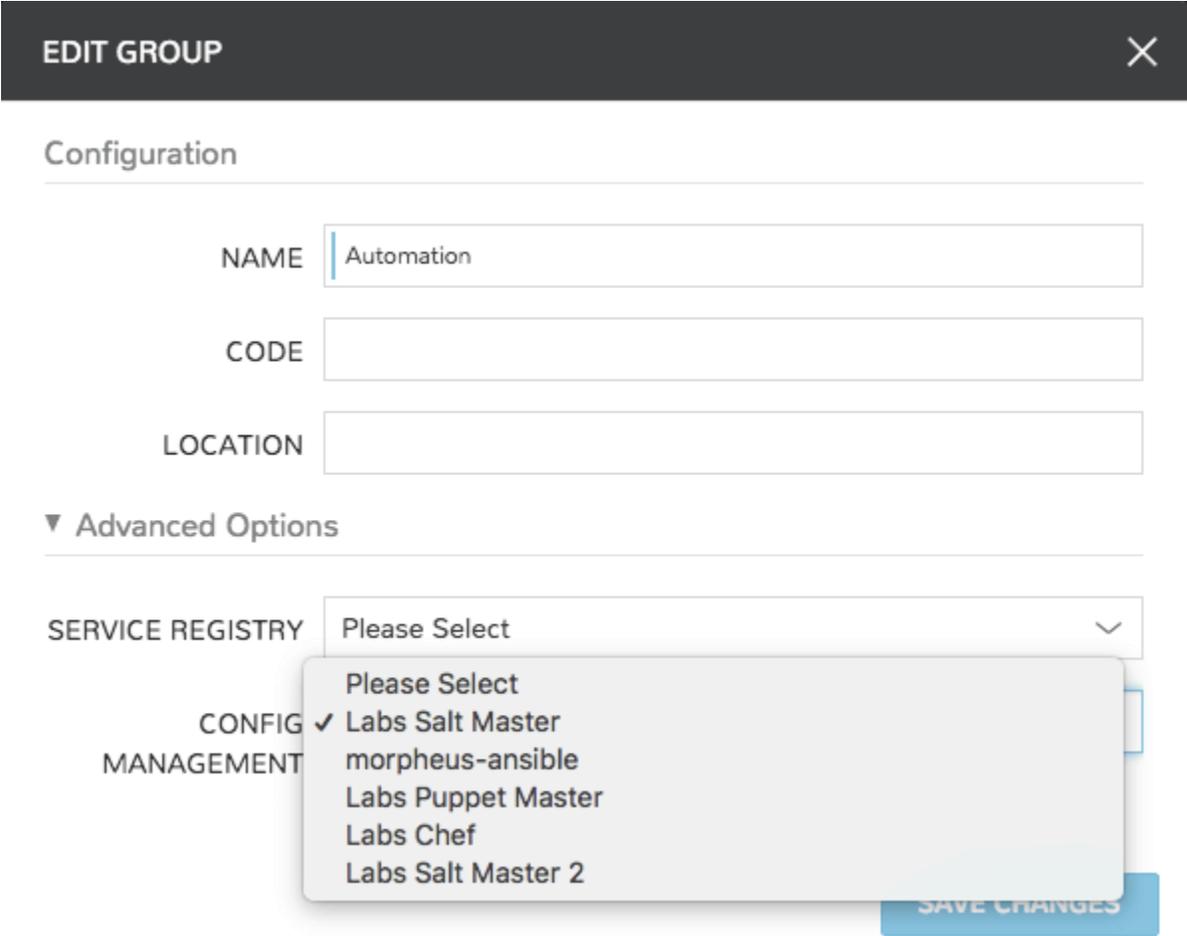
Configuration Management integrations like Saltstack apply to the Infrastructure Group abstraction in Morpheus . To ties yours in, browse to Infrastructure > Groups in Morpheus and select the group that you would like to tie to your Salt Master.

From here select *Edit*

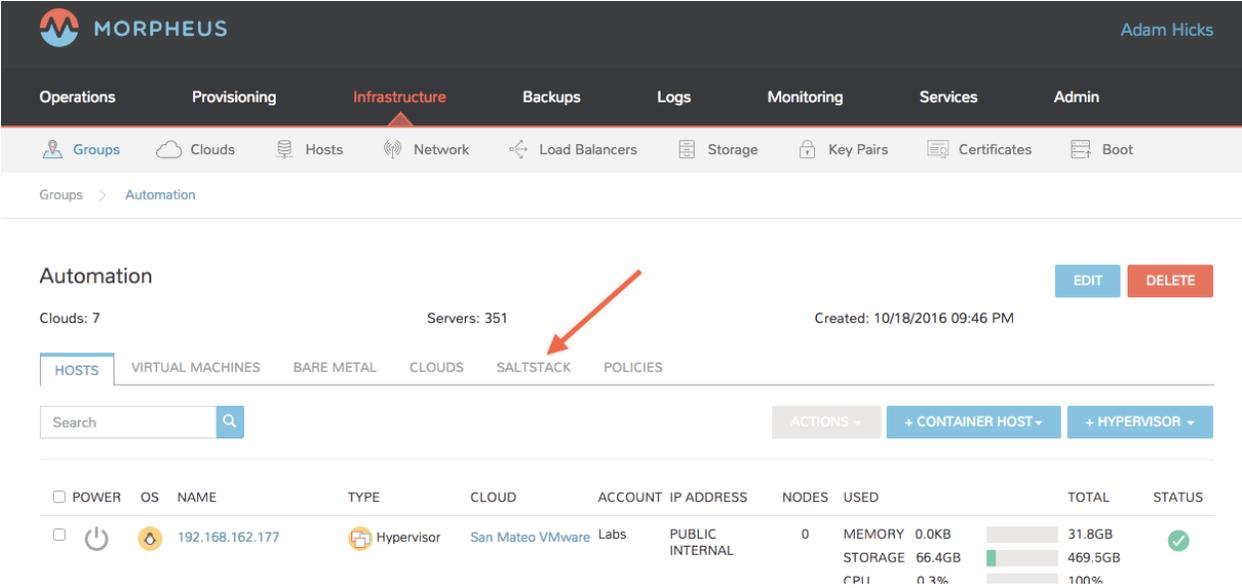
The screenshot shows the Morpheus web interface. The top navigation bar includes 'Operations', 'Provisioning', 'Infrastructure' (selected), 'Backups', 'Logs', 'Monitoring', 'Services', and 'Admin'. Below this is a secondary navigation bar with 'Groups', 'Clouds', 'Hosts', 'Network', 'Load Balancers', 'Storage', 'Key Pairs', 'Certificates', and 'Boot'. The main content area is titled 'Automation' and shows 'Clouds: 7' and 'Servers: 351'. There are tabs for 'HOSTS', 'VIRTUAL MACHINES', 'BARE METAL', 'CLOUDS', and 'POLICIES'. A search bar is present. Below the search bar are buttons for 'ACTIONS', '+ CONTAINER HOST', and '+ HYPERVISOR'. A table lists hosts with columns: POWER, OS, NAME, TYPE, CLOUD, ACCOUNT, IP ADDRESS, NODES, USED, TOTAL, and STATUS. The first host has IP 192.168.162.177 and 0 nodes. The second host has IP 192.168.163.130 and 4 nodes. An arrow points to the 'EDIT' button for the first host.

POWER	OS	NAME	TYPE	CLOUD	ACCOUNT	IP ADDRESS	NODES	USED	TOTAL	STATUS
<input type="checkbox"/>		192.168.162.177	Hypervisor	San Mateo VMware Labs	PUBLIC INTERNAL	0	MEMORY 0.0KB STORAGE 66.4GB CPU 0.3%	31.8GB 469.5GB 100%		
<input type="checkbox"/>		192.168.163.130	Hypervisor	San Mateo VMware Labs	PUBLIC INTERNAL	4	MEMORY 0.0KB STORAGE 41.9MB CPU 1.8%	127.8GB 5.1TB 100%		

And from the options toggle Advanced Options and select your Saltstack integration in the Config Management drop-down.



After a page refresh you should see your Saltstack tab in your group page



Clicking on it will reveal a page that includes:

- 1. An interface to run Salt Master commands

2. Parsed Top File

3. Available States

Automation [EDIT](#) [DELETE](#)

Clouds: 7 Servers: 351 Created: 10/18/2016 09:46 PM

[HOSTS](#) [VIRTUAL MACHINES](#) [BARE METAL](#) [CLOUDS](#) [SALTSTACK](#) [POLICIES](#)

LABS SALT MASTER [ACTIONS](#) ▾

@ 192.168.162.51

[RUN](#)

▼ Top File

```
base:
  '*':
    - git
    - cmatrix

'web*':
  - apache

'web*2':
  - webservers
```

▼ States

top.sls webservers.sls

The classic example of running

```
salt '*' test.ping
```

will return empty unless there are existing Minions with accepted keys on the Master. However, provisioning Minions via Morpheus is extremely easy.

Provisioning with Saltstack

To do so, provision as usual and Instances within the Group tied to the Saltstack Integration will now show additional options on the Configure pane

CREATE INSTANCE
✕

TYPE
GROUP
CONFIGURE
AUTOMATION
REVIEW

Configuration Options

VERSION	14.04	▼
INSTANCE CONFIGURATION	VMware VM	
PLAN	Memory: 512MB Storage: 10GB	
VOLUMES	root	10
	GB	SCSI 0:0
		Auto - Cluster
	+	
NETWORKS	Select Network	E1000
	+	
PLAN PRICE	\$0.00 / Month	
RESOURCE POOL	Resources	
PUBLIC KEY	Select	

► Create User (optional)

▼ Advanced Options

STORAGE CONTROLLERS	Type: IDE	Bus: 0	+
	Type: IDE	Bus: 1	
	Type: SCSI LSI Logic Parallel	Bus: 0	
DOMAIN NAME			
CUSTOMIZATION SPEC	Select		
HOSTNAME	webserver		
MINION ID			
STATE			

PREV
NEXT

Minion ID defaults to the hostname, and a State can be applied directly at provision time.

Note: Only States served from the Master's Working Directory can be applied at provision, not States from a git backend

Once your instance is provisioned and key negotiation has completed you will be able to access it and run commands via the integrated Salt command center in your Group.

Automation [EDIT](#) [DELETE](#)

Clouds: 7 Servers: 359 Created: 10/18/2016 09:46 PM

[HOSTS](#) [VIRTUAL MACHINES](#) [BARE METAL](#) [CLOUDS](#) [SALTSTACK](#) [POLICIES](#)

LABS SALT MASTER [ACTIONS -](#)

@ 192.168.162.51

salt RUN

RUN RESULTS

```
[WARNING ] The file_roots parameter is not properly formatted, using defaults
webserver:
  True
```

▼ Top File

```
base:
  **;
  .site
```

If you did not apply a state at provision time now you will be able to run State commands through Morpheus .

LABS SALT MASTER

@ 192.168.162.51

ACTIONS ▾

salt ** state.highstate

RUN

RUN RESULTS

```
[WARNING ] The file_roots parameter is not properly formatted, using defaults
webservice:
```

```
-----
ID: GIT software
Function: pkg.installed
Result: True
Comment: The following packages were installed/updated: python-git
         The following packages were already installed: git
Started: 18:52:07.812386
Duration: 23912.153 ms
Changes:
-----
```

```
python-async:
```

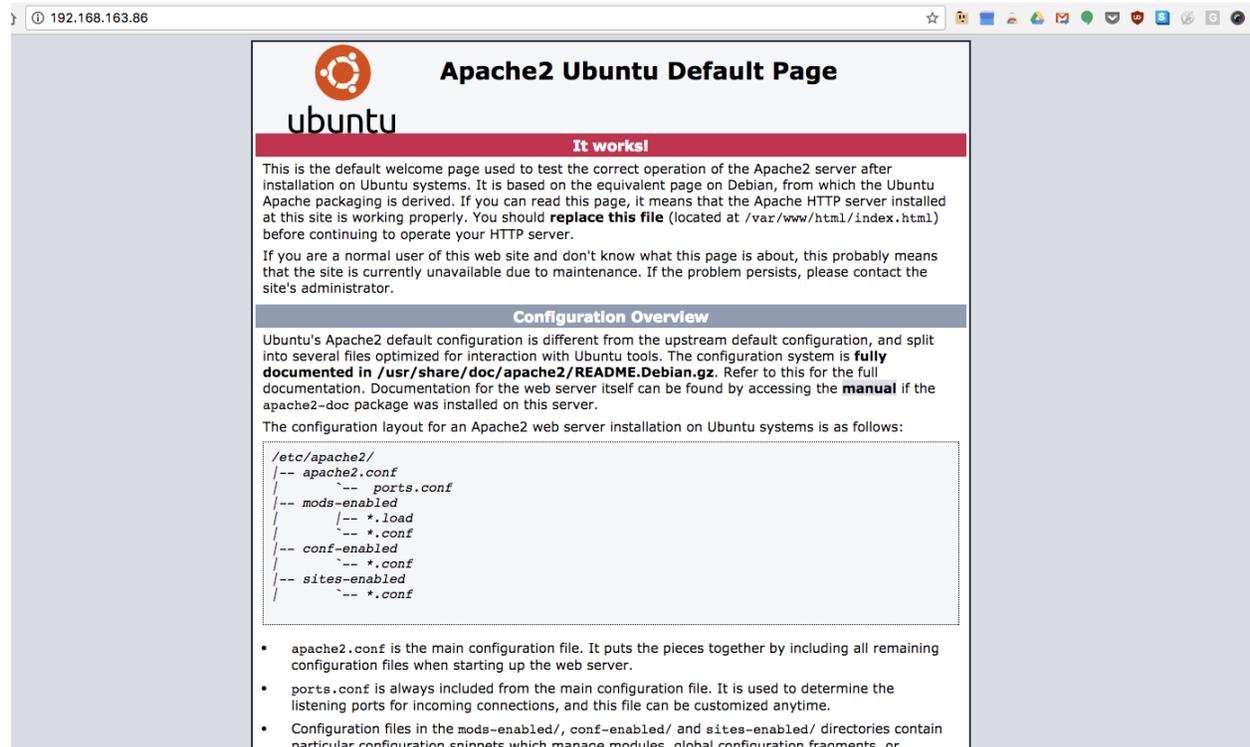
```
new:
    0.6.1-1
old:
```

```
python-git:
```

```
new:
    0.3.2~RC1-3
```

▼ Top File

In our example the Apache State from a git backend was applied successfully to our newly created vm.



192.168.163.86

Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|   |-- mods-enabled
|       |-- *.load
|       |-- *.conf
|   |-- conf-enabled
|       |-- *.conf
|   |-- sites-enabled
|       |-- *.conf
|
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or

Terraform

Requirements

Role Access

- In order to see the Terraform App Blueprint type option and create Terraform App Blueprints in Library > Blueprints > App Blueprints, the Morpheus user must have Role permissions for *Provisioning: Blueprints - Terraform* set to *Full*.
- In order to provision Terraform Apps in Provisioning > Apps, the Morpheus user must have Role permissions for *Provisioning: Blueprints - Terraform* set to *Provision* or *Full*.
- Existing Terraform Blueprints must be added before they can be provisioned from Provisioning > Apps.
- In order to provision Terraform Apps, the user must have Role permissions for *Provisioning: Apps* set to *Full*.

Github/Git Repo

- To use .tf files from a Git repo, a Git or Github integration needs to be configured in Administration > Integrations. If one is not configured, .tf or .tf.json files can be manually drafted in Morpheus and added to Terraform App Blueprints but they could not be sourced from version control repositories.

Supported App Provisioning Targets

- VMware
- Amazon AWS
- Microsoft Azure
- Google Cloud Platform (GCP)
- Oracle Cloud

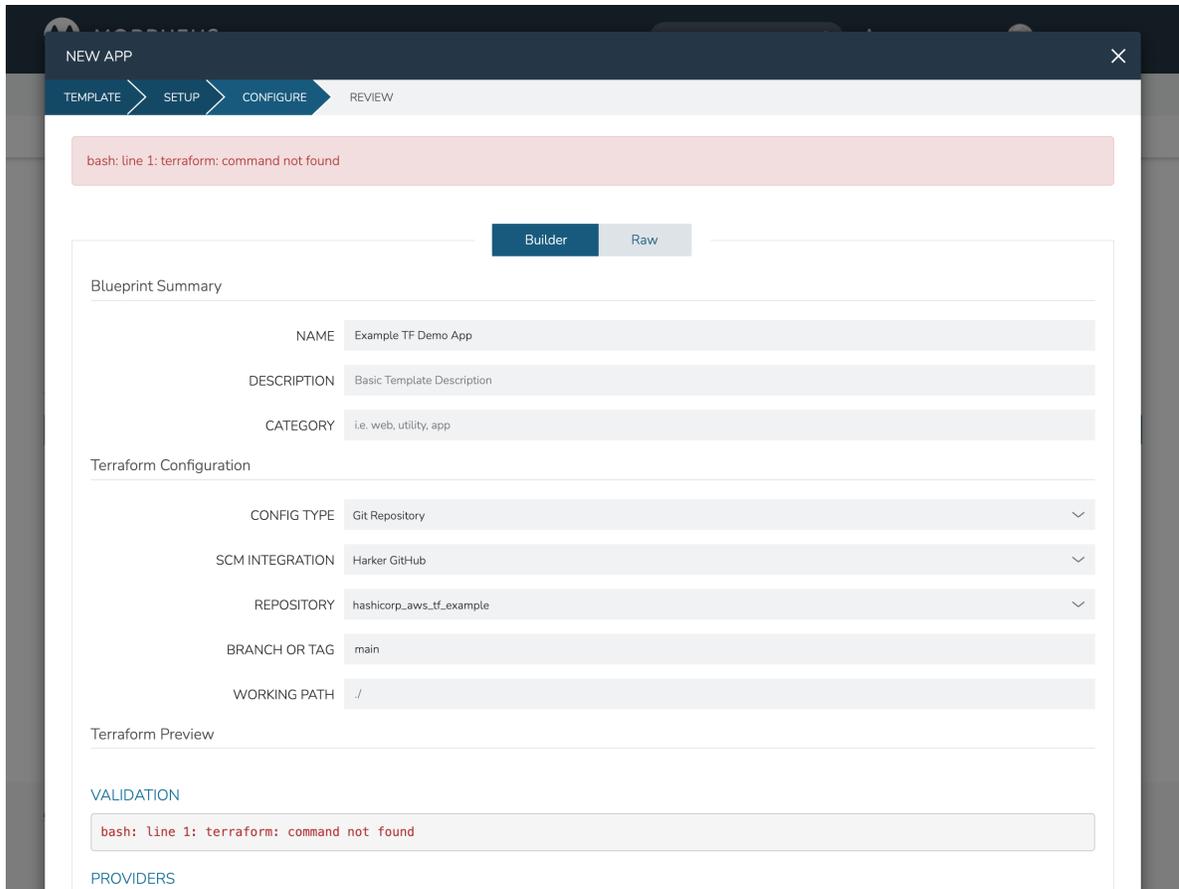
Note: Additional clouds are planned for later releases.

Terraform Installation

The first time you attempt to provision a Terraform App, you may come across an error indicating that Terraform is not installed:

```
bash: line 1: terraform: command not found
```

- **Command Not Found Error Screenshot**



This likely means you've not yet configured Terraform Settings within Morpheus global settings. Navigate to Administration > Settings > Provisioning and scroll down to the Terraform Settings section. By default, the Terraform Runtime field is set to "Manual". When set this way, Morpheus will attempt to use Terraform as installed on the appliance box and it may not be currently installed. To have Morpheus manage the Terraform installation process for you and manage Terraform versioning on a per-App basis, set the Terraform Runtime to "Auto". You should also set the Default Terraform Version field as well. When a version is set on a Terraform Spec Template or Terraform App Blueprint, that version will supersede the default version indicated in global settings.

- **Configured Terraform Runtime Screenshot**

The screenshot displays the Morpheus Administration interface. The top navigation bar includes the Morpheus logo, a search bar, a notification bell, a 'Support' dropdown, and a user profile for 'Alex Harker'. Below the navigation bar, the 'Administration' tab is active, showing a menu with options like Tenants, Plans & Pricing, Roles, Users, Integrations, Policies, Health, and Settings. The main content area is the 'Settings' page, which includes several sections: 'Enable Catalog' (checked), 'Enable Apps' (checked), 'Enable Instances' (checked), 'Provisioning Order' (set to 'Instances, Apps, Catalog'), 'PXE Boot Settings' (with a 'Default Root Password' field), 'App Blueprint Settings' (with 'Default Blueprint Type' set to 'Morpheus'), and 'Terraform Settings' (with 'Terraform Runtime' set to 'auto' and 'Default Terraform Version' set to '1.2.0'). A blue 'SAVE' button is located at the bottom of the settings section.

Important: Morpheus appliances which do not have access to the Internet will need to leave Terraform Runtime settings on “Manual” and ensure Terraform is installed appropriately on the appliance. Install Terraform in the `/usr/sbin/terraform` directory and make sure it’s owned by the `morpheus-local` user.

Creating Terraform App Blueprints

In order to provision Terraform apps, Terraform App Blueprints must be created first.

1. Navigate to Library > Blueprints > App Blueprints
2. Select + *ADD*
3. Name the Blueprint and select *Terraform* type.

Note: In order to see the Terraform Blueprint type option, the Morpheus user must have Role permissions for *Provisioning: Blueprints - Terraform* set to *Full*.

4. Select *NEXT*

5. Configure the following:

NAME Friendly name for the App Blueprint in Morpheus

DESCRIPTION Description for your App Blueprint shown in the Apps list (optional)

CATEGORY A category for your App (optional)

IMAGE Add reference icon for your App Blueprint to make it more identifiable at provision time (optional)

CONFIG TYPE (select Terraform Specs, Terraform (.tf), Terraform.json, or Git Repository)

- **Terraform (.tf)**

CONFIG Draft or paste in .tf content in the config text area. Variables will be presented as input fields during App provisioning, or auto-populated with matching values if contained in a selected TFVAR Secret file added to the Cypher service.

TFVAR SECRET Select an existing tfvar secret file stored in Morpheus Cypher service. This list is automatically filtered to show all Cypher entries which are currently stored at the “tfvar/*” Cypher mount point. Note that tfvars already set in any existing Terraform Cloud Profiles will already be available to your App and wouldn’t need to be set here.

VERSION Specify a version required by your Terraform App (optional). If specified, the given version will supersede the global Terraform version specified in Morpheus global settings (Administration > Settings > Provisioning). “Terraform Runtime” must also be set to “auto” in global settings for Morpheus to manage the Terraform version for you. When set to “manual”, Morpheus will use the Terraform version installed on the appliance box.

OPTIONS Specify any valid Terraform command line options Morpheus should append to its internal “terraform apply” call when the App is provisioned

- **Terraform (.tf.json)**

CONFIG Draft or paste in .tf.json content in the config text area. Variables will be presented as input fields during App provisioning, or auto-populated with matching values if contained in a selected TFVAR Secret file added to the Cypher service.

TFVAR SECRET Select an existing tfvar secret file stored in Morpheus Cypher service. This list is automatically filtered to show all Cypher entries which are currently stored at the “tfvar/*” Cypher mount point. Note that tfvars already set in any existing Terraform Cloud Profiles will already be available to your App and wouldn’t need to be set here.

VERSION Specify a version required by your Terraform App (optional). If specified, the given version will supersede the global Terraform version specified in Morpheus global settings (Administration > Settings > Provisioning). “Terraform Runtime” must also be set to “auto” in global settings for Morpheus to manage the Terraform version for you. When set to “manual”, Morpheus will use the Terraform version installed on the appliance box.

OPTIONS Specify any valid Terraform command line options Morpheus should append to its internal “terraform apply” call when the App is provisioned

- **Terraform Specs**

SPEC TEMPLATE Using the typeahead field, select all Terraform-type Spec Templates which make up your App. Variables will be presented as input fields during App provisioning, or auto-populated with matching values if contained in a selected TFVAR Secret file added to the Cypher service.

TFVAR SECRET Select an existing tfvar secret file stored in Morpheus Cypher service. This list is automatically filtered to show all Cypher entries which are currently stored at the “tfvar/*” Cypher mount point. Note that tfvars already set in any existing Terraform Cloud Profiles will already be available to your App and wouldn’t need to be set here.

VERSION Specify a version required by your Terraform App (optional). If specified, the given version will supersede the global Terraform version specified in Morpheus global settings (Administration > Settings > Provisioning). “Terraform Runtime” must also be set to “auto” in global settings for Morpheus to manage the Terraform version for you. When set to “manual”, Morpheus will use the Terraform version installed on the appliance box.

OPTIONS Specify any valid Terraform command line options Morpheus should append to its internal “terraform apply” call when the App is provisioned.

- **Git Repository**

SCM INTEGRATION Select a Github or Git integration that has been added in Administration > Integrations and which contains relevant .tf files. Integrations must be pre-existing prior to creating the App Blueprint.

Repository Select a repository which contains relevant .tf files from the Github or Git integration selected in the prior step.

BRANCH OR TAG Select the Git branch containing the desired version of .tf files for the App. “master” is chosen by default if no value is entered.

WORKING PATH Enter the repo path for the .tf file(s). ./ is default if no value is entered.

TFVAR SECRET Select an existing tfvar secret file stored in Morpheus Cypher service. This list is automatically filtered to show all Cypher entries which are currently stored at the “tfvar/*” Cypher mount point. Note that tfvars already set in any existing Terraform Cloud Profiles will already be available to your App and wouldn’t need to be set here.

VERSION Specify a version required by your Terraform App (optional). If specified, the given version will supersede the global Terraform version specified in Morpheus global settings (Administration > Settings > Provisioning). “Terraform Runtime” must also be set to “auto” in global settings for Morpheus to manage the Terraform version for you. When set to “manual”, Morpheus will use the Terraform version installed on the appliance box.

OPTIONS Specify any valid Terraform command line options Morpheus should append to its internal “terraform apply” call when the App is provisioned.

6. Select *COMPLETE*

Morpheus will scan the blueprint to check for validity and will surface any errors which need correcting before the App Blueprint can be saved. Your Terraform App is ready to be provisioned from Provisioning > Apps.

Cloud Profiles

Access to **Profiles** tab is determined by the following role permissions:

Role: Feature Access: Admin: Profiles

- None: Cannot access Profiles tab or create/view/edit/delete profiles
- Read: Can access Profiles tab, can view profiles, cannot create/edit/delete profiles
- Full: Can access Profiles tab, can create/view/edit/delete profiles

Terraform Profiles

- Terraform Profiles allow creation of Cloud-associated tfvars secrets, allowing tf apps and specs to be provisioned across multiple clouds that required different tfvars.
- Target Cloud Terraform Profiles are automatically mapped to tf apps/specs during provisioning, no manual scoping is required.
- Terraform Profiles are encrypted in Cypher and creating a profile creates a Cypher entry with key `tfvars/profile/cloud/$cloudCode/variables``
- Terraform Profiles can be edited after creation
- Terraform Profiles are limited to one per Cloud, once one is created for the Cloud the option to create a Terraform Profile is no longer present. Edit the existing Terraform Profile to make changes at that point

Important: Since Morpheus mounts Terraform Profiles in Cypher using a mount point which contains the Cloud code value, any Clouds which have the same code will share a Terraform Profile. Create or edit Clouds to have a unique code value if they should have a unique Terraform Profile. It's also important to understand that Morpheus does not require Clouds have a code at creation time. When Clouds are created without a code, Morpheus applies a generic non-unique code based on the Cloud type ("amazon" for AWS Clouds, as an example). This sets up a potential situation where all Clouds of the same type have the same generic Cloud code and thus share a Terraform Profile. To avoid this situation, enter a Cloud code value at creation time or edit existing Clouds to have a unique code.

Create a Terraform Profile

1. Navigate to Infrastructure > Clouds and select a Cloud
2. Select the "Profiles" tab
3. Select + *ADD PROFILE*
4. Select Terraform Profile Type
5. Enter tfvars in the *Terraform Profile Variables* field
 - example Terraform Profile Variables

```
access_key="****accessKey****"
secret_key="*****secretKey*****"
region="us-west-1"
```

6. Select *SAVE CHANGES*

Now, when provisioning a Terraform Instance or App to the Cloud the profile was created in, the tfvars in the profile become available to Terraform. It is not necessary to manually tie this tfvars files to your App Blueprint, these tfvars will automatically be available to Terraform whenever you provision an App to this cloud.

Provisioning Terraform Apps

Note: Terraform App Blueprints must be added to Library > Blueprints > App Blueprints before they can be provisioned. At least one Terraform App Blueprint must exist before Terraform Apps can be provisioned from Provisioning > Apps.

Note: In order to provision Terraform Apps in Provisioning > Apps, the Morpheus user must have Role permissions for *Provisioning: Blueprints - Terraform* set to *Provision* or *Full*.

1. Navigate to Provisioning > Apps
2. Select + *ADD*
3. Choose an existing Terraform App Blueprint
4. Select *NEXT*
5. Enter a *NAME* for the App and select the Group, Default Cloud and Environment (optional)
6. Select *NEXT*
7. Configure the following sections:

- **App Settings**

BACKEND TYPE Internal is the default selection and the most secure choice. This sends the Terraform state back to the appliance via HTTP loopback to be stored in the appliance database. Users may also select Local which will write Terraform state to the local filesystem first before being stored in the database and removed from the local filesystem. If unsure, use the default Internal value. **Backend Type selection is currently only available for Terraform Apps. Selecting Backend Type for Terraform Instance Types is a planned future feature addition.**

VALIDATE PLAN Marked by default. When unmarked, Morpheus will not perform its typical validation to ensure the Terraform spec is valid. Typically users would only unmark this box if onboarding an existing App with Initial State (see Advanced Options configuration below).

RUN APPLY Marked by default. When unmarked, Morpheus will not apply the plan at completion of the App wizard. Typically users would only unmark this box if onboarding an existing App with Initial State (see Advanced Options configuration below).

- **Terraform Variables**

Populate any required or optional variables here. Variables whose values are stored in an associated tfvars file sourced from Morpheus Cypher are masked from the user. Variables stored in any Terraform Cloud Profile associated with the selected Cloud also are automatically masked. Other variables will be presented in the Terraform Variables section for user entry and any configured default values on the Terraform spec will be pre-loaded.

- **Advanced Options**

REFRESH MODE Set an interval at which Morpheus should automatically check the App for drift from the Terraform spec. If set to Manual, Morpheus will never perform automatic checks and the user must do so when desired. The results of drift checks are reported on the App detail page and users may apply the plan at any time to bring the App back into alignment with the Terraform spec.

INITIAL STATE Paste in existing Terraform state to onboard an existing Terraform App for Morpheus management. Though the field is small, it will accept large, multiline Terraform state. When creating

an App from existing state, users may want to skip plan validation or may not want to apply right away. Opt out of these functions by unmarking the corresponding box in the App Settings section.

Note: Terraform state import is a new feature. At this time, some state file components may break the import process. This feature is being iterated on rapidly to increase coverage to as many application types and conditions as possible.

- **Terraform Preview**

Review the Terraform App components here, including any providers invoked, variables surfaced from the App spec, resources to be created, and .tf files utilized. There is no user input to be entered into this section.

8. Select *NEXT*

9. Morpheus will now validate the App (unless the user has opted out of this check) and surface any errors which would cause provisioning issues. If all is well, click *COMPLETE*

Tip: Review the App in the Terraform Preview section. If any config data needs to be edited, select the *RAW* tab, edit the config, and then select the *BUILDER* tab once again. The config changes from the RAW edit will be updated in the preview section for further review. Permanent edits can be made by editing the App Blueprint, pushing .tf changes to your code repository, or Terraform Spec Templates (depending on how the .tf files are sourced for your App Blueprint).

The Terraform App will begin to provision.

Once provisioning is completed, note the State tab in the App details page (Provisioning > Apps > select the App). This tab contains subsections related to state management which is discussed in greater detail in the next section.

Terraform App State Management

State management is handled from the State Tab of the Terraform App detail page (Provisioning > Apps > selected App). With the tab selected, the Terraform command field will be present regardless of the selected subsection. Use this field to send Terraform commands to your apps just like using Terraform from the command line. Press return on the keyboard or click on the “play” button to the right of the text field to execute the command.

Tip: “terraform” is automatically entered for each command as printed along the left edge of the text field. Thus, you don’t need to enter “terraform” with each command sent. Entering “state” or “plan” is equal to entering “terraform state” or “terraform plan” from the command line.

When Terraform commands are executed against the application, Morpheus provides progress bars and command output in the UI. Command output is shown underneath the Terraform command field. Users can dismiss individual output windows by clicking the “x” button in the upper-right of each window. All command output can be dismissed by clicking the blue “x” button to the right of the command field itself.

Within the ACTIONS reside four selections: Refresh State, Apply State, Edit Inputs, and Edit STATE. Selecting Refresh State is equivalent to using the “terraform plan” command from the command line. This will read the existing state of any existing objects which are part of the App and compare their current configuration against the prior state. Any differences will be noted in the output. If differences are discovered, the App is considered to be in a “drift” state. This drift status is shown in the UI when the user is viewing the “State” subsection (which is described in greater detail in the next section). The output of the Refresh State command, including detailed information about changes Terraform would make to App objects in order to realign them with the App spec are shown in the UI.

The Apply State selection brings up a modal which allows the user to view the App spec once again. This includes being able to view and edit Terraform variables if needed. After making any needed edits, click *NEXT* and Morpheus will

validate the App once again, just like it did at provision time. On the next tab of the wizard, Morpheus will show the user and planned changes that would be executed if the user completes the modal. An output will be shown as if “terraform plan” were run from the command line. Make note of any App objects which would be created, altered, or destroyed if the actions are accepted as Morpheus would immediately take them if desired. When ready, click *COMPLETE*. This will execute all planned changes as if the user had run “terraform apply -auto-approve” from a terminal session.

Edit Inputs allows for editing of Input values without going through the process of applying state. Edit State displays the state in a large text area for direct editing.

State Subsection

The State Subsection shows the current drift state of the App. This includes when Morpheus has last checked for drift and whether the App is currently in a “Drift” or “No Drift” state. If the App is currently in a Drift state, users can select Refresh State from the ACTIONS menu to identify which objects and attributes have deviated from the App configuration.

Summary **State** Resources Security Groups Environment Logs History Monitoring Wiki

terraform [] [] [] ACTIONS ▾

State Specs Plan Input Output

MY TEST APP - MY TEST APP

Last Check: 09/06/2022 08:00 PM Status: OK Drift Status: No Drift

STATE DATA

```

1 {
2   "format_version": "1.0",
3   "terraform_version": "1.2.8",
4   "values": {
5     "root_module": {
6       "resources": [
7         {
8           "address": "aws_instance.app_server",
9           "mode": "managed",
10          "type": "aws_instance",
11          "name": "app_server",
12          "provider_name": "registry.terraform.io/hashicorp/aws",
13          "schema_version": 1,
14          "values": {

```

Specs Subsection

The Specs Subsection will show the user all Morpheus Spec Templates (Library > Templates > Spec Templates) which make up the App. Users may even edit Spec Template config directly from this view by clicking the Edit (pencil) icon to the right of each Spec Template listed.

Tip: Editing a Spec Template here will detach it from the source object, essentially making it a brand new object that exists only here. All future updates to that Spec Template would have to be made here going forward. In most cases, it’s advisable to edit the Spec Template directly at the source. For example, if this Spec Template were sourced from an integrated version control repository (ex. Github), it’s likely the best option to make a new commit into your repository and then let Terraform handle the process of bringing your App in line with the new specifications.

EDIT RESOURCE SPEC



Editing this resource spec will detach it from the underlying source template

NAME My Test App

TYPE

TEMPLATE

```
terraform:
  required_providers:
    aws:
      source: hashicorp/aws
      version: ~> 4.16
  required_version: '>= 1.2.0'
  variable:
    access_key:
      type: !!com.bertramlabs.plugins.hcl4j.RuntimeSymbols.StringPrimitiveType
      length: null
      name: string
      parent: !!com.bertramlabs.plugins.hcl4j.symbols.HCLAttribute
      length: null
      name: type
      parent: !!com.bertramlabs.plugins.hcl4j.symbols.HCLBlock
```

SAVE CHANGES

Plan Subsection

This section displays the output of the most recent “terraform plan” run against your App. This will either indicate that your infrastructure (App) matches the configuration or it will indicate that a drift of some sort has taken place.

Input Subsection

This section lists all Terraform inputs, such as variables, which are relevant to the App. Variable values are shown unless they are flagged as sensitive in your configuration. All variables sourced from a Morpheus Cypher tfvars mount will automatically be masked.

Output Subsection

This section lists all configured Terraform output.

Terraform Instance Type Example

Terraform Spec can also be used within the Morpheus Instance Type construct in addition to App Blueprints and Apps. Expand the section below to see a complete end-to-end example of a Terraform Instance Type from drafting new Spec Templates through to provisioning.

- **Terraform Instance Type Example**

Terraform is a common tool that allows IT administrators to map out infrastructure as code in configuration files and supports all of the popular providers used in the modern datacenter. Once configured, Terraform will plan, deploy, and manage the infrastructure as needed. Configuration files can be brought under version control so teams can easily make changes to environments. Infrastructure can also be monitored for drift and corrective action can easily be taken.

Morpheus allows users to on-board or even draft Terraform spec directly. With the configuration on-board, we can begin to piece together infrastructure constructs into the Morpheus Library as Layouts and Instance Types. With the Library items staged, users can deploy new infrastructure directly into the selected providers. Once deployed, infrastructure can be monitored for drift from within Morpheus UI. When needed, we can plan and take corrective action easily from the detail page of a Morpheus Terraform Instance.

In this section, we've discussed a high-level overview of Terraform and working with Terraform in the Morpheus context. In the next section, we'll actually onboard Terraform spec, create Library items, and deploy real infrastructure to AWS with Morpheus.

In this example, we're going to deploy a VPC and three subnets to AWS using Terraform and Morpheus. I've created Spec Templates to onboard `.tf` configuration files which handle the AWS provider (with assume role for account flexibility), VPC creation, subnet creation, and variable injection.

We'll onboard the Terraform configuration files as modular Spec Templates, create new Instance Types with custom Layouts for the Morpheus Library, and set up Inputs to inject variable values at provision time. Once deployed, we'll take a look at the new infrastructure in Morpheus and go over the management capabilities for the new environment.

Terraform configuration is stored as a Spec Template in Morpheus. You can store your configuration as one monolithic file for each Instance Type you intend to create or you can create individual Spec Templates for modular pieces which can be reused across multiple Instance Types. When added to the Layout later, we'll be able to include as many Spec Templates as we wish which enables us to reuse smaller modular pieces if desired.

Spec Templates are added in the Morpheus Library (Library > Templates > Spec Templates tab). We can pull in the template from some type of repository, such as through a Github integration, or write new spec directly into the New Spec Template modal. In most cases, the spec will be pre-existing and pulled in from a version-controlled repository but here I have my Terraform spec entered locally. Click **+ ADD**, complete the fields as I've done in the example below and click **SAVE CHANGES**.

In the **VERSION** field at the bottom of the TF Spec Template modal, enter a Terraform version number to force that version to be used. This version is only honored if the Terraform Runtime setting (Administration > Settings > Provisioning) is set to "auto". When "manual" is selected as the Terraform Runtime setting, Morpheus will simply use the version installed on the appliance box.

Tip: When declaring variables, keep in mind that Morpheus expects users to follow Terraform best practices. For example, when a variable type is not defined, it defaults to string. See [Terraform Documentation](#) for additional resources on variable declaration.

EDIT SPEC TEMPLATE
✕

NAME

CATEGORY

TYPE

SOURCE

CONTENT

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = ">= 3.35.0"
    }
  }
}
```

provider "aws" {

VERSION

SAVE CHANGES

– AWS Subnet by Count

```
# This spec template creates AWS subnets based on the count requested utilizing
↳ the vpc cidr provided in var.vpc_cidr variable

locals {
  bitCount = sum([tonumber(local.subnet_options.cidrMask), -tonumber(split("/",
↳ var.vpc_cidr)[1])])
}

resource "aws_subnet" "main" {
  count = tonumber(var.subnetCount)
  vpc_id    = aws_vpc.main.id
  cidr_block = cidrsubnet(var.vpc_cidr, local.bitCount, count.index)

  tags = merge(
    local.default_tags,
    {
      Name = "${var.vpc_name}-subnet-0${count.index}"
    }
  )
}

output "aws_subnet" {
```

(continues on next page)

(continued from previous page)

```

value = aws_subnet.main
sensitive = true
}

```

– AWS Terraform Default Vars

```

variable "access_key" {
  type    = string
}

variable "secret_key" {
  type    = string
}

variable "subnetCount" {
  type = number
  default = "<%=customOptions.subnetCount%>"
}

variable "sensitive_thing" {
  type = string
  default = "this_var_is_sensitive"
  sensitive = true
}

```

– AWS Provider Role Assume

```

terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = ">= 3.35.0"
    }
  }
}

provider "aws" {
  region      = local.vpc_options.region
  access_key  = var.access_key
  secret_key  = var.secret_key

  assume_role {
    # The role ARN within Account B to AssumeRole into.
    role_arn = "arn:aws:iam::${local.vpc_options.aws_account}:role/
    ↪OrganizationAccountAccessRole"
  }
}

```

– AWS Terraform Locals

```

locals {
  # Common tags to be assigned to all resources
  default_tags = {

```

(continues on next page)

```
Owner    = "<%=username%>"
Group    = "<%=groupName%>"
Management_Tool = "Terraform"
Management_Platform = "Morpheus"
}

subnet_options = {
  cidrMask = "<%=customOptions.cidrMask%>"
  subnetCount = "<%=customOptions.subnetCount%>"
}
vpc_options = {
  region = "<%=customOptions.awsRegion%>"
  aws_account = "<%=customOptions.awsAccount%>"
}
}
```

– AWS VPC

```
variable "vpc_cidr" {
  type        = string
  description = "CIDR for the the VPC"
  default     = "172.16.0.0/24"
}

variable "vpc_name" {
  type        = string
  description = "Name for the VPC"
  default     = "durka"
}

resource "aws_vpc" "main" {
  cidr_block = var.vpc_cidr

  tags = merge(
    local.default_tags,
    {
      Name = var.vpc_name
    }
  )
}

output "aws_vpc" {
  value = aws_vpc.main
  sensitive = true
}
```

Note: In the AWS Terraform Locals example Spec Template above, pre-provision variables are used. Note the use of [pre-provision](#) variables to store the value for Owner and Group, among other things. See the variables section of Morpheus documentation (linked in the prior sentence) for a listing of other possible pre-provision variables and a complete map of variables which can be resolved after provisioning has completed.

In order to create the Layout later in the guide, I need to create four Inputs so the user can make certain selections at provision time. I wrote my Terraform Configuration with this flexibility in mind so that the same Instance Type can be reused in different scenarios. In this particular case, I'm populating the Inputs with manual Option Lists but they can also be populated through REST calls or calls to the Morpheus API when needed.

Option Lists are created in the Library (Library) under the Option Lists tab. These are lists of items which will be used to create dropdown selections at provision time. Click + *ADD*, complete the fields as I've done in the example below and click *SAVE CHANGES*. I've created one each for the AWS account selection, region selection, and CIDR mask input.

EDIT OPTION LIST ×

NAME

DESCRIPTION

TYPE

VISIBILITY

If the account assigned is not the master tenant, visibility will automatically change to private.

DATASET Create an initial JSON or CSV dataset to be used as the collection for this option list. It should be a list containing objects with properties 'name', and 'value'.

```
{{"name": "Acct1", "value": "111111111"}, {"name": "Acct2", "value": "222222222"}}
```

REAL TIME

Inputs are also created in the Library under the Inputs tab. In this case, I'm creating four Inputs. Three of them will display as dropdown selections and will be tied to one of the Option Lists we just made. The other will be a simple text input where the user can indicate the total number of subnets that should be created. Click + *ADD*, complete the fields as I've done in the example below and click *SAVE CHANGES*.

NEW OPTION TYPE
✕

NAME

DESCRIPTION

FIELD NAME
This is the input fieldName property that the value gets assigned to.

EXPORT AS TAG

DEPENDENT FIELD
A fieldName that will trigger reloading this option list

VISIBILITY FIELD
A fieldName that will trigger visibility of this option

DISPLAY VALUE ON DETAILS

TYPE

LABEL
This is the input label that shows typically to the left of a custom option.

At this point we're ready to create a new Instance Type. We'll give the Instance Type a name, which users will use to identify the Instance Type from the list in the provisioning wizard. We don't need to set much else in this case, most of the pieces we've created in previous steps will be associated with the Layout that we create next. The Layout will also be tied to the Instance Type we're creating now. Instance Types are also created in the Library (Library) under the Instance Types tab. Click + *ADD*, complete the fields as I've done in the example below and click *SAVE CHANGES*.

NEW INSTANCE TYPE
✕

NAME

CODE
Useful shortcode for provisioning naming schemes and export reference.

DESCRIPTION

CATEGORY

ICON

Suggested Dimensions: 150 x 51

The Layout will bring together everything we've made to this point, the Spec Templates, Inputs and the Instance Type. We can add a new one from the Instance Type detail page (Library > Blueprints > Instance Types > Selected Instance Type) by clicking + *ADD LAYOUT*. We can also create one from the Layouts section (Library > Blueprints > Layouts) by clicking + *ADD*.

First, change the TECHNOLOGY value to Terraform and the fields will change to allow proper configuration. Next, provide a name for your Layout. If you're creating the Layout through the Layout tab rather than from the Instance Type detail page, you'll need to identify the Instance Type the Layout goes with. Using the typeahead fields at bottom of the modal window, add our four Inputs and our five Spec Templates to the Layout. Finally, point the layout to a TFVAR SECRET from Morpheus Cypher if needed. You can see a screenshot of my Layout

configuration below

Option Types

AWS Account <small>(select)</small>	☰ ×
AWS Region <small>(select)</small>	☰ ×
Subnet Count <small>(text)</small>	☰ ×
CIDR Mask <small>(select)</small>	☰ ×

Spec Templates

AWS Subnet by Count	⊗
AWS Terraform Default Vars	⊗
AWS Provider Role Assume	⊗
AWS Terraform Locals	⊗
AWS VPC	⊗

SAVE CHANGES

Now, we're ready to provision new infrastructure into AWS using Morpheus and Terraform. Just like any other Instance Type, we begin from the Instances list page (Provisioning > Instances) and click + *ADD*. Select the Instance Type we've just created and move on to the **GROUP** tab of the wizard. Here you'll give the new instance a name and select your Group and Cloud. Once finished, you'll move on to the **CONFIGURE** tab where we'll see the Inputs we created and associated with the Layout. Once finished with this tab, step through the rest of the wizard to complete the process. You can see the options I've selected for this configuration in the image below.

CREATE INSTANCE
×

TYPE >
GROUP >
CONFIGURE
AUTOMATION >
REVIEW

Configuration Options

LAYOUT ▼
AWS Cloud Resources

PLAN ▼
Terraform
Cores: 1

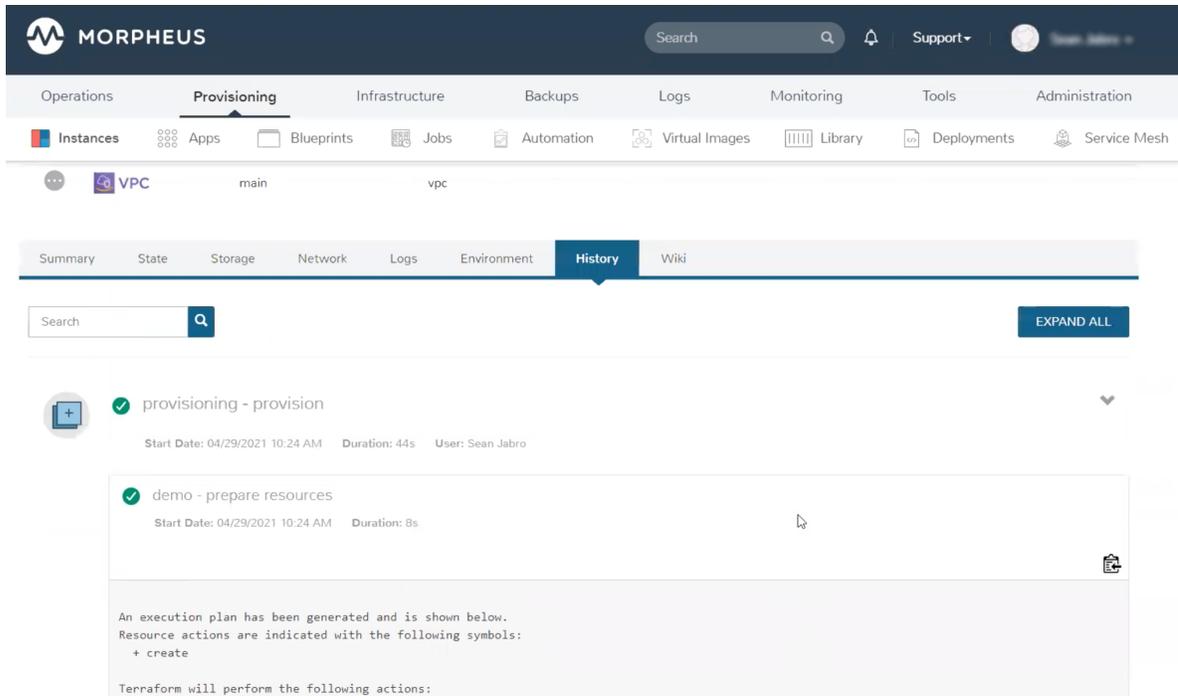
SELECT ACCOUNT ▼
AWS TW1

SUBNET CIDR ▼
/28
Bitmask for the subnet CIDR

SELECT REGION ▼
us-east-1

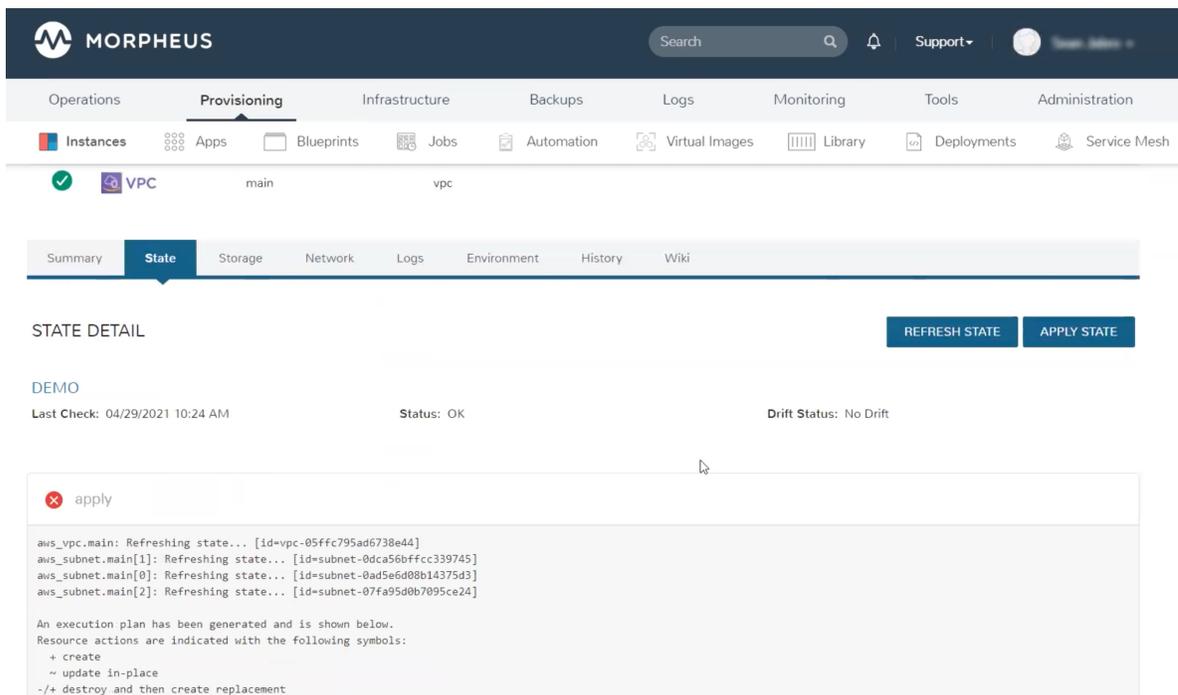
SUBNET COUNT ▼
| I

After completing the wizard, from the History tab of the Instance detail page users can review the Terraform plan being executed and see the output while the resources are still being provisioned.



Once the provisioning process is complete, head to the State tab. Here we can see and link through to the associated Spec Templates. If needed, you can also edit the configuration spec by clicking on the pencil icon at the end of the row for any listed Spec Template.

By clicking *APPLY STATE*, the user can once again see the Input selections which were presented during the initial provisioning and make changes when needed. After making changes and clicking *NEXT*, Morpheus will show the plan output no different than if you'd run `terraform plan`. On clicking *COMPLETE*, the plan will be executed as if you'd run `terraform apply`. Back on the State tab you will see the output from the Apply process as well as an indicator of the success or failure of the operation.



Morpheus will also regularly check for drift from the Terraform configuration. On the State tab near the top is a “Drift Status” indicator. This will either show Drift or No Drift depending on the situation. Morpheus will automatically check for drift every few minutes but you can perform a manual check at any time by clicking *REFRESH STATE*. Drift can be corrected when needed by reapplying state (*APPLY STATE* button).

vRealize Orchestrator

The vRealize Orchestrator (vRO) Integration provided for Morpheus enables users to easily trigger existing workflows that may already exist in vRealize Orchestrator. Not only can the user trigger these workflows, but they can also be chained easily into non-vRO workflows and process both output and input parameters of a workflow.

Adding the Integration

Setting up the vRO integration involves some steps which vary depending on the authentication model being used.

When using OAUTH, the Client ID must be gathered first. This can be found by browsing a file on the actual VRA server using SSH. On the vRA server, run the following command: `grep -i cafe_cli= /etc/vcac/solution-users.properties | sed -e 's/cafe_cli=/'`

Secondly, you will need the username, password, and host API URL. Typically, the API URL is run on port 8283. A sample API URL may look like the following example: `https://vrahost.com:8283/`

Be sure to fill in the tenant token as the domain or tenant ID, for example: `vsphere.local`, with a username of `administrator@vsphere.local`.

Note: At times, this can vary depending on how authentication and role assignments for the user have been set up for vRO.

vRA auth uses vRA identity Bearer tokens for API consumption. The only real difference in field requirements when using this authentication mode is that the *Client ID* is no longer needed.

Using vRealize Orchestrator

One of the first things Morpheus does when it is tied into a vRO integration is sync all available workflows by category. These workflows become available when creating a new Morpheus task in Library > Automation. Morpheus allows a user to map these vRO workflows into the task engine. The task engine allows users to design workflows that chain tasks in order or operate at different phases of a provisioning request. For more information on tasks, please read the Automation documentation.

Creating a task for VRO is simple.

First, go to Library > Automation and create a new task. Choose a task type of *vRealize Orchestrator Workflow*. A dropdown will appear allowing one to first select the active vRO Integration you would like to use. Once that is selected, a list of workflows becomes available.

Note: The next part is where things can get a bit tricky. The parameter body (expected in JSON) format can be a bit difficult to track down. One way is to use the Network Chrome inspector when kicking off a sample workflow from the vRO HTML5 client and grabbing the parameter JSON. Another is to query the API yourself and look at the samples from historical run history.

An example payload for the *SSH / Run SSH Command* Workflow would look like this:

```
{
  "parameters": [
    {
      "name": "hostNameOrIP",
      "type": "string",
      "value": {
        "string": {
          "value": "x.x.x.x"
        }
      }
    },
    {
      "name": "port",
      "type": "number",
      "value": {
        "number": {
          "value": 22
        }
      }
    },
    {
      "name": "cmd",
      "type": "string",
      "value": {
        "string": {
          "value": "echo \"Hello <%=instance.name%>\""
        }
      }
    },
    {
      "name": "encoding",
      "type": "string",
      "value": {
        "string": {
          "value": ""
        }
      }
    },
    {
      "name": "username",
      "type": "string",
      "value": {
        "string": {
          "value": "myuser"
        }
      }
    },
    {
      "name": "passwordAuthentication",
      "type": "boolean",
      "value": {
        "boolean": {
          "value": true
        }
      }
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```

    }
  },
  {
    "name": "password",
    "type": "string",
    "value": {
      "string": {
        "value": "password"
      }
    }
  },
  {
    "name": "path",
    "type": "string",
    "value": {
      "string": {
        "value": "\\var\\lib\\vco\\app-server\\conf\\vco_key"
      }
    }
  },
  {
    "name": "passphrase",
    "type": "string",
    "value": {
      "string": {
        "value": ""
      }
    }
  }
]
}

```

Note that all Morpheus variables can be injected into the parameter body. In the above example we inject the instance name into the sample command with `<%=instance.name%>`.

Adding this task to a workflow allows the result parameters to be referenced in subsequent tasks called throughout the workflow. For example, a local script task type could reference the output text of the above ssh command by injecting the following results map: `echo "results.vro: <%=results.vro.find{it.name == 'outputText'}?.value?.string?.value%>"`

There are very powerful options available for chaining results and injecting variables relevant to the instance being provisioned or even custom inputs from an operational workflow. Please reference the rest of the Automation documentation for examples.

5.13.16 Backups

Commvault

Morpheus integrates with Commvault for selection as a Cloud backup target. Compatible Clouds include VMware and OpenStack. Morpheus integrates with your existing Commvault appliance, which can then be set as the preferred backup solution for any existing Clouds. From there, easily schedule backup routines during Instance provisioning and restore Instances when needed. This section discusses the process for integrating Commvault with Morpheus, sharing a Commvault integration with multiple Tenants, setting backup during Instance provisioning, and restoring Instances from backup.

Features

- Share one backup provider across multiple Tenants
- Apply Commvault integration as the default backup target for compatible clouds
- Select Commvault integrations as backup provider at Instance provision time
- Automate backups with Morpheus Backup Jobs
- When needed, easily restore Instance backups

Adding Commvault Integration

1. Navigate to *Backups > Services*
2. Select + *ADD*
3. Select Commvault
4. Fill in the following:

Name Name of the Integration in Morpheus

Enabled Enable the Commvault integration

Host IP or Hostname of the Commvault server.

Port Port number configured to access the Commvault server

Tip: Enter the port to connect with the Commvault web server, by default this is port 81. For more, see [Commvault port requirement documentation](#).

Username Admin Username for Commvault

Password Password for Username provided (encrypted in Morpheus).

Visibility

Sets Multi-Tenant Visibility

Private Only Available to the Tenant the Integration is added by

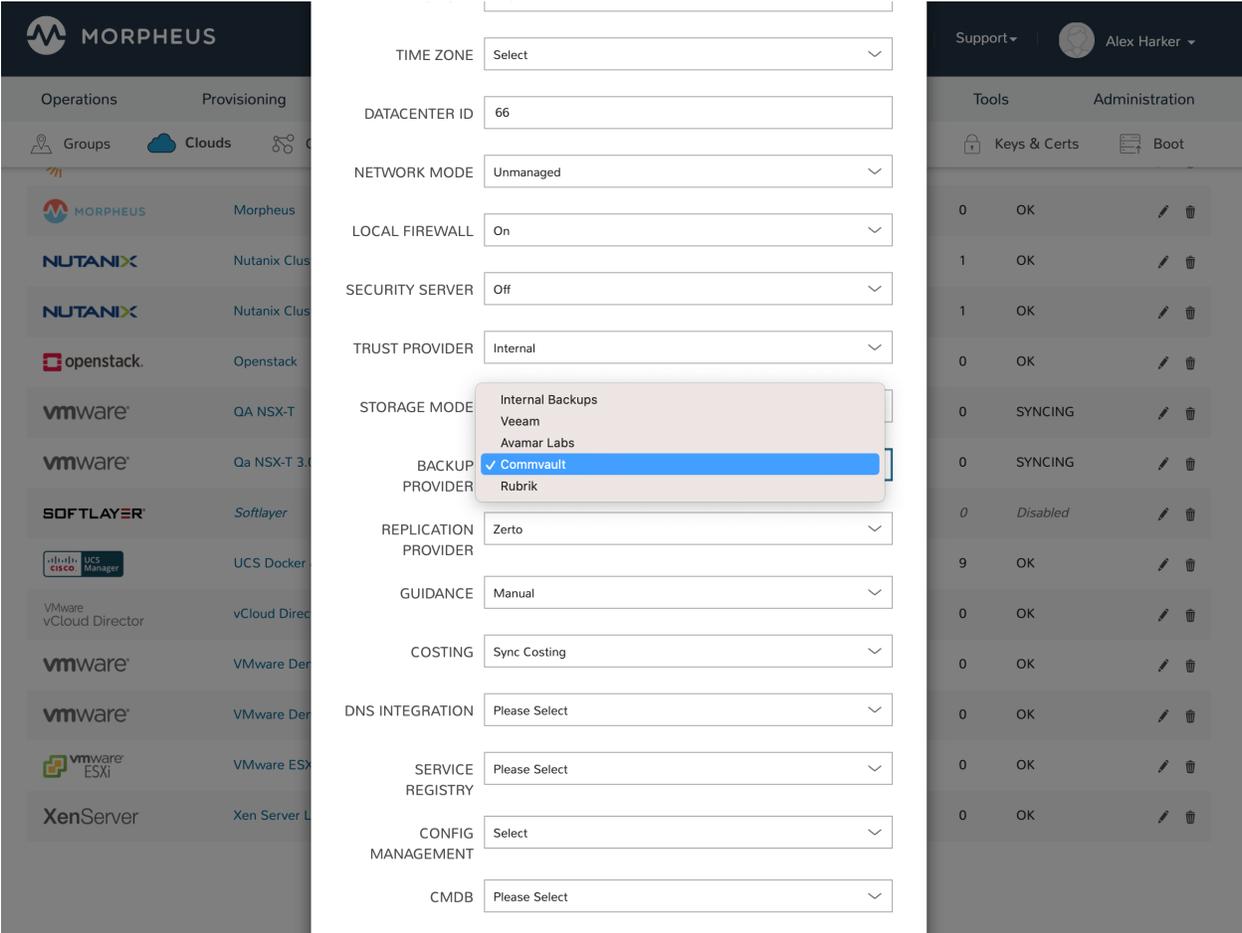
Public Available to Sub-Tenants (master tenant option only)

5. *SAVE*

Set Commvault as Cloud Backup Target

Once the initial integration is made, set this integration as the backup provider for as many supported Clouds as needed. Commvault integrations are supported as backup target for VMware and OpenStack Clouds at this time.

- 1. Navigate to Infrastructure > Clouds
- 2. Select an existing VMware or OpenStack Cloud
- 3. Click *EDIT*
- 4. Expand the Advanced Options section
- 5. Under “Backup Provider”, select the relevant Commvault integration
- 6. Click *SAVE CHANGES*



Configure Backup at Provision Time

When provisioning an Instance into a Cloud where Commvault is set as the backup provider, Commvault will be selectable as the “Backup Type” on the Automation tab of the provisioning wizard. Expand the Backups section on this tab and enter the following:

- **Backup Type:** Select the desired Commvault backup type
- **Backup Server:** Select the desired server synced from the Commvault backup provider associated with the Cloud
- **Backup Set:** Select a configured backup set synced from the Commvault backup provider associated with the Cloud
- **Storage Policy:** Gold, Silver, or Bronze. Select the applicable SLA or retention policy for the workload being provisioned. The meanings of these retention tiers are configurable in Commvault
- **Backup Name:** A name for the backup in Morpheus, this field is pre-populated with the Instance name but can be overwritten
- **Backup Job Type:** Clone an existing backup job (Backups > Jobs) or add this backup to an existing job. A job contains a retention count and backup frequency schedule and can have as many Instances backing up under it as needed
- **Backup Job:** Select the job which will be cloned or have a backup added to it depending on your selection in the prior field
- **Job Name:** A name for the new cloned job (if you are cloning and not creating a new Backup Job)

The screenshot displays the Morpheus Automation configuration interface. The top navigation bar includes tabs for TYPE, GROUP, CONFIGURE, AUTOMATION (selected), and REVIEW. The left sidebar shows the 'Instances' section with a search bar and a list of instances including ARDMCE, DMHAAP Demo, DMHADB Production, and DMHAES Production. The main content area is titled 'Automation' and contains several expandable sections: Deployment, Load Balancer, Scale, Backups (expanded), Replication, and Lifecycle. The 'Backups' section is active, showing the following configuration fields:

- WORKFLOW: Select
- BACKUP TYPE: Commvault VMware Backup
- BACKUP SERVER: 10.30.21.180
- BACKUP SET: defaultBackupSet
- STORAGE POLICY: Gold
- BACKUP NAME: AHDMAPA1010
- BACKUP JOB TYPE: New job
- JOB NAME: AHDMAPA1010

At the bottom right of the configuration panel, there are 'PREVIOUS' and 'NEXT' buttons. The right sidebar shows a user profile for Alex Harker and a '35% MEMORY' gauge.

Viewing Backups

After provisioning, users can review backup details from the Instance detail page (Provisioning > Instances > Selected Instance > Backups tab). Additionally, backups can be configured here if this was not done during provision time by clicking *ADD BACKUP*. Users can also run one-off backups from this page by opening the **ACTIONS** menu and clicking Backup. Backups will still continue to run based on the schedule configured in their job but additional runs can be made on-demand this way.

Within the Backups section (Backups > Backups) users can also view all currently-configured backups and whether or not recent backup runs have succeeded.

The screenshot shows the Morpheus web interface with the 'Backups' tab selected. Below the navigation bar, there is a 'BACKUP LIST' section with search and filter controls. A '+ ADD' button is visible in the top right of the list area. The list contains four backup entries:

Job	Duration	Type	Tenant	Progress	Success	Total Size	Backups
AHDMCEN1002 Job: AHDMCEN1002 Latest: N/A Next: N/A	-	Rubrik VMware Backup	Morpheus	0%	0KB	0	0
Demo Bucket Backup Job: Bucket Backups Latest: 11/14/2019 02:25 PM Next: N/A	00:24:16	Object Storage Backup	Morpheus	100%	1.84GB	2	2
demo-ha-app1 Job: N/A Latest: 04/04/2020 08:00 PM Next: N/A	00:01:04	Nutanix VM Snapshot	Morpheus	100%	0KB	1	1
demo-ha-db1 Job: demo-ha-db1 Latest: 08/07/2021 08:00 PM	00:01:06	Nutanix VM Snapshot	Morpheus	100%	0KB	1	1

Restore an Instance from Commvault

For Instances with current backups, the Backup Results section will be populated on the Instance detail page (Provisioning > Instances > Selected Instance > Backup tab). If the Instance needs restored, simply click Actions (within the Backup Results area, not the main actions menu for the Instance itself) and then click Restore. The status icon at the top of the Instance detail page will turn green once this process is finished and the Instance will be fully restored from your selected backup.

Veeam

Veeam is a backup and replication platform designed to work with popular on-prem cloud providers, including VMware, Microsoft Hyper-V, and vCloud Director. Morpheus integrates with your existing Veeam appliance which can then be set as the preferred backup solution for any existing Clouds. From there, easily schedule backup routines during Instance provisioning and restore Instances when needed. This section discusses the process for integrating Veeam with Morpheus, sharing a Veeam integration with multiple Tenants, setting backup during Instance provisioning, and restoring Instances from Veeam backup.

Features

- Share one backup provider across multiple Tenants
- Apply Veeam integration as the default backup target for compatible clouds
- Select Veeam integrations as backup provider at Instance provision time
- Automate backups with Morpheus Backup Jobs
- When needed, easily restore Instance backups

Adding Veeam Integration

1. Navigate to *Backups > Integrations*
2. Select + *ADD*
3. Select Veeam
4. Fill in the following:

Name Friendly name for the integration in Morpheus

Enabled When marked, this integration will be active and available for use

API URL IP or Hostname of the Veeam Enterprise Manager, must be HTTPS for VEEAM 10

Port Port number configured to access the Veeam server, must be 9398 for VEEAM 10

Username Username for an admin service account in Veeam

Password Password for Username provided (encrypted in Morpheus)

Visibility

Sets Multi-Tenant Visibility

Private Only available to the Tenant that added the integration

Public Available to Subtenants (primary tenant option only)

5. Click *SAVE*

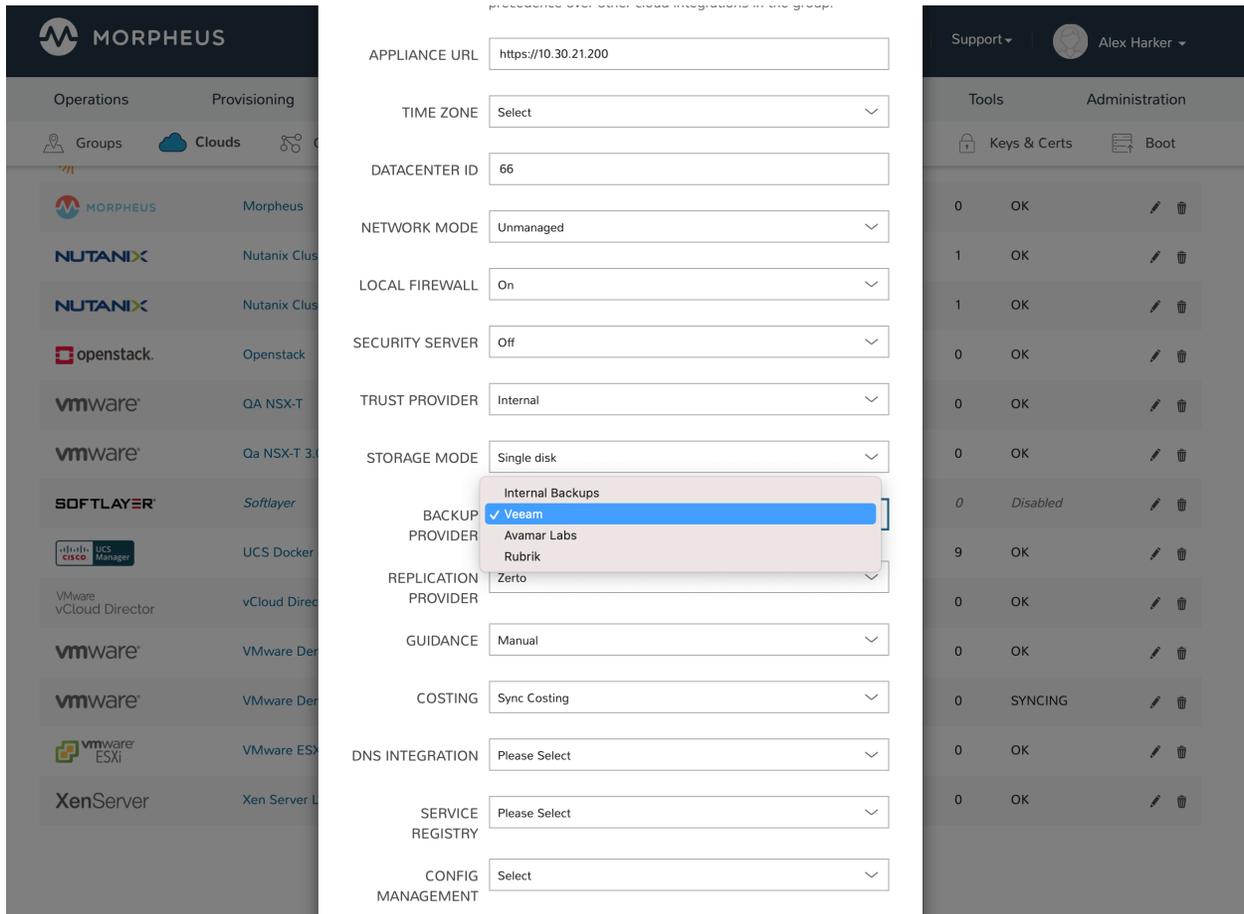
Note: Veeam Backup Enterprise Manager must be installed in order to successfully integrate Morpheus with Veeam.

Important: Once Veeam service has been integrated with Morpheus, Veeam server(s) will be available to select as the backup provider for VMware, Hyper-V, and vCloud Director cloud integrations (Infrastructure > Clouds > Edit a compatible Cloud). To enable Veeam backups, select the appropriate Veeam server as the “backup provider” for your cloud integrations as needed. Failure to do so will result in blank Backup Repositories and Backup Job Templates options when configuring Veeam Backups during provisioning.

Set Veeam as Cloud Backup Target

Once the initial integration is made, set this integration as the backup provider for as many supported Clouds as needed. Veeam integrations are supported as backup target for VMware, Hyper-V, and vCloud Director Clouds at this time.

1. Navigate to Infrastructure > Clouds
2. Select an existing VMware, Hyper-V, or vCD Cloud
3. Click *EDIT*
4. Expand the Advanced Options section
5. Under “Backup Provider”, select the relevant Veeam integration
6. Click *SAVE CHANGES*



Configure Backup at Provision Time

When provisioning an Instance into a Cloud where Veeam is set as the backup provider, Veeam will be selectable as the “Backup Type” on the Automation tab of the provisioning wizard. Expand the Backups section on this tab and enter the following:

- **Backup Type:** Select the desired Veeam backup type
- **Repository:** Select a repository synced from the Veeam backup provider associated with the Cloud
- **Managed Server:** Select a managed server associated with the backup server selected in the previous step
- **Backup Name:** A name for the backup in Morpheus, this field is pre-populated with the Instance name but can be overwritten
- **Backup Job Type:** Clone an existing backup job (Backups > Jobs) or add this backup to an existing job. A job contains a retention count and backup frequency schedule and can have as many Instances backing up under it as needed
- **Backup Job:** Select the job which will be cloned or have a backup added to it depending on your selection in the prior field
- **Job Name:** A name for the new cloned job (if you are cloning and not creating a new Backup Job)

The screenshot shows the 'CREATE INSTANCE' wizard in the Morpheus interface, specifically the 'AUTOMATION' tab. The 'Backups' section is expanded, showing the following configuration:

- WORKFLOW:** Select
- Scale:** (Collapsed)
- Backups:**
 - BACKUP TYPE:** Veeam VMware VM Backup
 - REPOSITORY:** Default Backup Repository (Backup Server: 10.30.22.140)
 - MANAGED SERVER:** 10.30.21.180 (Backup Server: 10.30.22.140)
 - BACKUP NAME:** AHDMUBU1021
 - BACKUP JOB TYPE:** Add to existing job
 - BACKUP JOB:** aa-test
- Replication:** (Collapsed)
- Lifecycle:** (Collapsed)

At the bottom of the wizard, there are 'PREVIOUS' and 'NEXT' buttons. The background shows the Morpheus dashboard with a '37% MEMORY' gauge and various system metrics.

Viewing Backups

After provisioning, users can review backup details from the Instance detail page (Provisioning > Instances > Selected Instance > Backups tab). Additionally, backups can be configured here if this was not done during provision time by clicking *ADD BACKUP*. Users can also run one-off backups from this page by opening the *ACTIONS* menu and clicking Backup. Backups will still continue to run based on the schedule configured in their job but additional runs can be made on-demand this way.

Within the Backups section (Backups > Backups) users can also view all currently-configured backups and whether or not recent backup runs have succeeded.

The screenshot shows the Morpheus interface for the Backups section. At the top, there is a navigation bar with the Morpheus logo, a search bar, and user information (Alex Harker). Below this is a secondary navigation bar with tabs for Operations, Provisioning, Infrastructure, Backups (selected), Logs, Monitoring, Tools, and Administration. Under the Backups tab, there are sub-tabs for Summary, Jobs, Backups, History, and Integrations. The main content area is titled 'BACKUP LIST' and includes a search bar, filters for 'LAST STATUS' and 'TYPE', and a '+ ADD' button. The backup list contains four entries:

Job Name	Duration	Type	Tenant	Progress	Success	Total Size	Backups
AHDMCEN1002 Job: AHDMCEN1002 Latest: Next: N/A	-	Rubrik VMWare Backup	Morpheus	0%	0KB	0	0
Demo Bucket Backup Job: Bucket Backups Latest: 11/14/2019 02:25 PM Next: N/A	00:24:16	Object Storage Backup	Morpheus	100%	1.84GB	2	2
demo-ha-app1 Job: N/A Latest: 04/04/2020 08:00 PM Next: N/A	00:01:04	Nutanix VM Snapshot	Morpheus	100%	0KB	1	1
demo-ha-db1 Job: demo-ha-db1 Latest: 08/07/2021 08:00 PM	00:01:06	Nutanix VM Snapshot	Morpheus	100%	0KB	1	1

Restore an Instance from Veeam

For Instances with current backups, the Backup Results section will be populated on the Instance detail page (Provisioning > Instances > Selected Instance > Backup tab). If the Instance needs restored, simply click Actions (within the Backup Results area, not the main actions menu for the Instance itself) and then click Restore. The status icon at the top of the Instance detail page will turn green once this process is finished and the Instance will be fully restored from your selected backup.

Rubrik

The embedded Morpheus Rubrik Backup integration allow syncing, creation and management of Rubrik Backups for vCenter clouds.

Features

- Backup sync & association
- Sla Domain sync & selection
- Backup creation, deletion & restore
- Restore Backups over existing vm's
- Restore Backup as new

Adding Rubrik Integration

Note: The Rubrik backup service is currently only supported on the VMware cloud type.

1. Navigate to *Backups > Services*
2. Select + *ADD*
3. Select Rubrik
4. Fill in the following:

Name Name of the Integration in Morpheus

Enabled Enable the Integration

Host IP or Hostname of the Rubrik api server

API Token The API token to authenticate with the Rubrik API server

Visibility

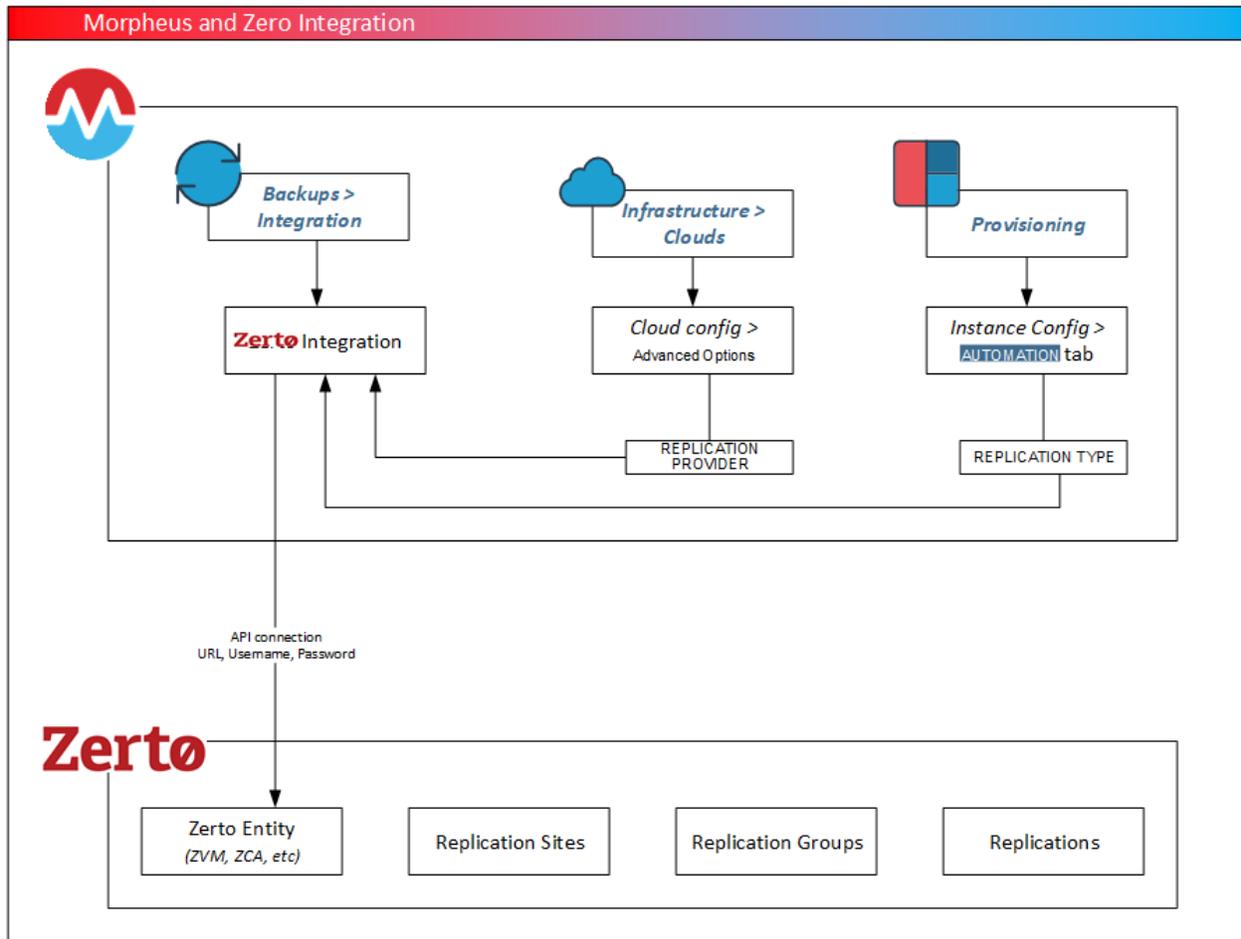
Sets Multi-Tenant Visibility

Private Only Available to the Tenant the Integration is added by

Public Available to Sub-Tenants (master tenant option only)

5. *SAVE*

Zerto



Adding Zerto Integration

1. Navigate to *Backups > Integrations*
2. Select + *ADD*
3. Select Zerto
4. Fill in the following:

Name Name of the Integration in Morpheus

Enabled Enable the Integration

API URL

API URL for Zerto Virtual Manager Example `API URL: https://zvm_IP:9669`

Username Admin Username for Zerto

Password Password for Username provided (encrypted in Morpheus).

Visibility

Sets Multi-Tenant Visibility

Private Only Available to the Tenant the Integration is added by

Public Available to Sub-Tenants (master tenant option only)

5. *SAVE*

Avamar

IMPORTANT: Avamar API must be installed on Avamar server (not installed by default)

Adding Avamar Integration

1. Navigate to *Backups > Services*
2. Select + *ADD*
3. Select Avamar
4. Fill in the following:

Name Name of the Integration in Morpheus

Enabled Enable the Integration

Host IP or Hostname of the Avamar api server.

Port Port number configured to access the Avamar server

Username Admin Username for Avamar

Password Password for Username provided (encrypted in Morpheus).

Tenant Avamar Tenant/Domain to scope Integration to

Hypervisor Avamar Hypervisor to scope Integration to

Visibility

Sets Multi-Tenant Visibility

Private Only Available to the Tenant the Integration is added by

Public Available to Sub-Tenants (master tenant option only)

5. *SAVE*

5.13.17 Clouds

Alibaba

Features

- Brownfield discovery
- Instance cloning
- Security group creation and management
- Docker and Kubernetes provisioning and management
- Usage tracking

- Two-way tag sync

Supported Regions

- ap-northeast-1 (1)
- ap-south-1 (1)
- ap-southeast-1 (1)
- ap-southeast-2 (2)
- ap-southeast-3 (3)
- ap-southeast-5 (5)
- cn-beijing (2)
- cn-chengdu (1)
- cn-guangzhou (3)
- cn-hangzhou (1)
- cn-heyuan (2)
- cn-hongkong ()
- cn-huhehaote (5)
- cn-qingdao (1)
- cn-shanghai (2)
- cn-shenzhen (1)
- cn-wulanchabu (6)
- cn-zhangjiakou (3)
- eu-central-1 (1)
- eu-west-1 ()
- me-east-1 (1)
- us-east-1 (1)

Integrate an Alibaba Cloud with Morpheus

To add a new Cloud, navigate to Infrastructure > Clouds and click + *ADD*. Select “Alibaba Cloud” and click *NEXT*. Once the “ADD CLOUD” modal appears, configure the following:

- **NAME:** A friendly name for the Cloud in Morpheus
- **CODE:** A Cloud code used to reference this Cloud in Morpheus API
- **LOCATION:** An optional field for tracking location data related to this Cloud
- **VISIBILITY:** Public Clouds are available to all Tenants, Private Clouds are available to one selected Tenant
- **TENANT:** If the Cloud visibility is set to “Private”, this field determines which Tenant the Cloud is exposed to
- **ENABLED:** When marked, the Cloud is available as a provisioning target

- **AUTOMATICALLY POWER ON VMS:** When marked, Morpheus is the source of truth for the expected power state of Instances. Morpheus tools should be used to control power state and Morpheus will override any unexpected power states (such as if an instance were powered on or off from the Alibaba web console)
- **CREDENTIALS:** Select Local Credentials to enter authentication credentials on this modal, Existing Credentials to choose a pre-saved credential set, or New Credentials to enter authentication credentials on this modal and save them (in Infrastructure > Trust) for other uses later
- **ACCESS KEY:** (When Local Credentials or New Credentials are selected) A valid Access Key for an Alibaba Cloud account
- **SECRET KEY:** (When Local Credentials or New Credentials are selected) A valid Secret Key for an Alibaba Cloud account
- **INVENTORY:** When marked, Morpheus will automatically onboard existing instances in the Alibaba Cloud account as unmanaged servers
- **REGION:** Select the Alibaba Cloud region to associate with the Cloud (if this list is empty, check your Access and Secret Key credentials)
- **VPC:** Select the Alibaba Cloud VPC to associate with the Cloud (if this list is empty, check your Access and Secret Key credentials)

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM's if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT INSTALL MODE

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud Init / Unattend (when available): (DEFAULT) Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

Following Integration

After the integration has been created, Morpheus will sync existing workloads (if you've opted to inventory), security groups, tags, and more. Synced workloads can be viewed from Infrastructure > Compute. If Plans aren't immediately available within a few minutes after the integration is created, navigate to the Cloud detail page (Infrastructure > Clouds > Your Integrated Alibaba Cloud), click *REFRESH* and click "Daily". Shortly thereafter, the Plans should be synced and selectable at provision time. Without manually syncing the Plans, you may be unable to provision to this Cloud until it undertakes its next daily sync overnight as Plan selection is required.

You're now able to provision new Instances and Apps to the Alibaba Cloud. Morpheus includes a default catalog that includes Alibaba images which can be provisioned out of the box. Additionally, you can begin to create your own custom library of Alibaba workloads by adding Virtual Images and building out Instance Types.

AWS

Overview

AWS is the Amazon public cloud, offering a full range of services and features across the globe in various datacenters. AWS provides businesses with a flexible, highly scalable, and low-cost way to deliver a variety of services using open standard technologies as well as proprietary solutions. This section of documentation will help you get Morpheus and AWS connected to utilize the features below:

Features

- Instance, Service, Infrastructure Provisioning & Synchronization
- EKS Cluster Creation & Synchronization
- Morpheus Kubernetes, Docker & KVM Cluster Creation
- ELB Classic Load Balancer Creation & Synchronization
- ELB Application Load Balancer (ALB) Creation & Synchronization
- Security Group Creation & Synchronization
- Security Group Rule Creation & Synchronization
- Network Synchronization
- VPC Creation & Synchronization
- CloudFormation Provisioning & Resource Synchronization
- Terraform Provisioning & Resource Synchronization
- Pricing & Costing Synchronization
- MetaData Tag Creation & Synchronization
- S3 Bucket Creation & Synchronization
- Route53 Automation & Synchronization
- IAM Profile Synchronization and Assignment
- RDS Support
- Backups / Snapshots
- Migrations
- Auto Scaling
- Remote Console (SSH & RDP)
- Lifecycle Management and Resize
- Restore from Snapshots
- Elastic IP Assignment
- Network Pools
- Enhanced Invoice Costing

Requirements

AWS IAM Security Credentials Access Key Secret Key Sufficient User Privileges (see *MinimumIAMPolicies* section for more info)

Security Group Configuration for Agent Install, Script Execution, and Remote Console Access

- Typical Inbound ports open from Morpheus Appliance: 22, 5985, 3389 (22 & 3389 required for Console. 22 & 5985 required for agent-less comms)
- Typical Outbound to Morpheus Appliance: 443 (Required for Agent install & comms)

Note: These are required for Morpheus agent install, communication, and remote console access for windows and linux. Other configurations, such as docker instances, will need the appropriate ports opened as well. Cloud-init Agent Install mode does not require incoming access for port 22.

Network(s) IP assignment required for Agent install, Script Execution, and Console if the Morpheus Appliance is not able to communicate with AWS instances private ip's.

Adding an AWS Cloud

1. Navigate to *Infrastructure > Clouds*
2. Select + *Create Cloud*
3. Select AWS
4. Enter the following:

Cloud Configuration

NAME Name of the Cloud in Morpheus

CODE Unique code used for api/cli, automation and policies.

LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT If Visibility is set to Private, select the Tenant the Cloud resources will assigned to.

ENABLED When disabled, automatic Cloud sync is paused and the Cloud will not be selectable for provisioning.

AUTOMATICALLY POWER ON VMS When enabled, Morpheus will maintain the expected power state of managed VMs. Morpheus will power on any managed VMs in the Cloud that have been shut down for unknown reasons (not powered off by Morpheus) to ensure availability of services.

Note: When “AUTOMATICALLY POWER ON VMS” is enabled, the power state of managed VMs should be maintained in Morpheus. This setting is not applicable to discovered/unmanaged resources.

Details

REGION Select AWS Region for the Cloud. AWS Clouds may also be scoped to all regions.

ACCESS KEY Access Key ID from AWS IAM User Security Credentials.

SECRET KEY Secret Access Key associated with the Access Key ID.

USE HOST IAM CREDENTIALS Check to use use Host IAM Credentials

ROLE ARN Supports security token service (STS) to AssumeRole by entering an AWS Role ARN

EXTERNAL ID When required to AssumeRole, included the needed External ID

INVENTORY

Basic Morpheus will sync information on all EC2 Instances in the selected VPC the IAM user has access to, including Name, IP Addresses, Platform Type, Power Status, and overall resources sizing for Storage, CPU and RAM, every 5 minutes. Inventoried EC2 Instances will appear as Unmanaged VM's.

Full In addition to the information synced from Basic Inventory level, Morpheus will gather Resource Utilization metrics for Memory, Storage and CPU utilization per VM.

Off Existing EC2 Instances will not be inventoried

Note: Cloud Watch must be configured in AWS for Morpheus to collect Memory and Storage utilization metrics on inventoried EC2 instances.

USE VPC Specify if the target account is using EC2-VPC or EC2-Classic Platform. In almost all cases, VPC should be selected, and then select the target VPC from the synced available VPC's list, or *All VPC's*.

5. The AWS cloud is ready to be added to a group and saved. Additional configuration options available:

IMAGE TRANSFER STORE S3 bucket for Image transfers, required for migrations into AWS.

EBS ENCRYPTION Enable or disable encryption of EBS Volumes

COSTING KEY For Gov Cloud pricing only, key for standard managing cost account

COSTING SECRET For Gov Cloud pricing only, secret for standard managing cost account

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM's if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT INSTALL MODE

- **SSH / WINRM:** Morpheus will use SSH or WINRM for Agent install.
- **Cloud Init / Unattend (when available):** (DEFAULT) Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

KMS KEY ID For specify an AWS KMS key for encrypting EBS Volumes when volume encryption is enabled

Provisioning and Keypairs

During provisioning, users do not need to create or select a keypair as you would do when provisioning directly from the AWS console. If needed, Morpheus will create an AWS keypair without input from the user. In the AWS console and in the Morpheus Keypairs Section (Infrastructure > Trust > Key Pairs) you will see these keypairs listed. Morpheus will also inventory other keypairs which are created in your integrated AWS clouds and these will be listed out in the Morpheus Keypairs section.

These created keypairs are not accessible to the user as you do not see them at creation time and they are encrypted as stored in the Morpheus database. To have access to the Instance after provisioning, the user should have his or her user created during provisioning. This is done by creating a Linux user in Morpheus [User Settings](#) and associating a keypair with the user. If you don't currently have a keypair to use, Morpheus can generate one for you in the Keypairs section.

Linux Settings

Username	<input type="text" value="al"/>
Password	<input type="password" value="••••••••"/>
Confirm	<input type="text"/>
SSH Key	<input type="text" value="morpheus"/>

With your user set in Morpheus User Settings, ensure that your user is being created during Instance provisioning. On the CONFIGURE tab of the Instance provisioning wizard, mark the “CREATE YOUR USER” box. This is normally checked by default if a user is configured in Morpheus User Settings but you can confirm that here.

CREATE INSTANCE ✕

TYPE > GROUP > CONFIGURE > AUTOMATION > REVIEW

Configuration Options

VERSION 22.04 ▼

LAYOUT Amazon VM ▼

PLAN T2 Nano - 1 Core, 0.5GB Memory ▼
Cores: 1 Memory: 512 MB Price: \$5.2456 / Month ⓘ

RESOURCE POOL vpc-0 ▼

VOLUMES root | 10 GB gp2 ▼ +

NETWORKS Select Network ▼ +

SECURITY GROUPS Select ▼ +

PUBLIC IP Subnet Default ▼

▼ User Config

CREATE YOUR USER

USER GROUP Select ▼

► Advanced Options

PREVIOUS NEXT

It's worth noting that adding your user to the provisioned workload does not replace the keypair used by Morpheus during provisioning. Instead it adds the public key for your user to the authorized keys file on the workload which allows you to access it. You can confirm this by catting out the authorized key file on any of your Morpheus-provisioned workloads (`~/.ssh/authorized_keys`). You will at least see two keys, one for the Morpheus service user and one for your own user.

Note: In general it should not cause problems if Morpheus-generated keypairs are deleted from the AWS console or from the Morpheus keypairs section. You would not lose console access to the VM. If needed, however, Morpheus will create more keypairs and there is currently no functionality to automatically remove these keypairs. They can be safely ignored, generally speaking there is no need to manually delete them.

Enhanced Invoice Costing Configuration

AWS cloud integrations in Morpheus will sync highly-granular costing data through the use of AWS Costing & Utilization Reports (CUR). If desired, users can turn on costing in the Morpheus Cloud configuration without linking a CUR to use AWS Cost Explorer instead. Morpheus version 4.2.3 also simplified the way CUR reports can be selected or created in order to sync costing data. The section below discusses setting up enhanced costing through CUR reports both in 4.2.3 and versions prior. For additional details on setting up costing with AWS GovCloud, see the next section.

Note: Even with a costing report configured in the Cloud integration as described below, the COSTING value must also be set to “Costing and Reservations” in order for enhanced invoice data to be brought into Morpheus. Confirm this setting by editing the Amazon Cloud integration, and checking the COSTING value in the Advanced Options panel before continuing.

v4.2.3 and Above

In Morpheus 4.2.3+, edit the Amazon cloud integration or create a new Amazon Cloud to get started. On the Create/Edit Cloud modal, open the advanced options section. The relevant fields for configuring invoice costing are shown below:

COSTING REPORT	<input type="text" value="New Report"/>
COSTING REPORT NAME	<input type="text" value="test_report"/>
COSTING FOLDER	<input type="text" value="reports"/>
COSTING BUCKET	<input type="text" value="New Report Bucket"/>
COSTING BUCKET NAME	<input type="text" value="new_bucket"/>
COSTING BUCKET REGION	<input type="text" value="US West (N. California)"/>
COSTING KEY	<input type="text"/>
COSTING SECRET	<input type="text"/>
LINKED ACCOUNT ID	<input type="text"/>

In the example case above, a new report and a new S3 bucket are created but Morpheus will also sync in buckets and

reports that meet the required parameters if they already exist. For reports to be synced they must meet the requirements listed below:

- Hourly time granularity
- Include resource IDs
- GZIP compression
- CSV format

If you don't currently have a report meeting those criteria, you can create one by selecting "New Report" from the REPORT NAME dropdown menu. A new S3 bucket can be created in similar fashion if needed. You may also want to review the section below on configuration for Morpheus 4.2.2 and below to note policies that will be applied to your selected bucket and Cost Explorer permissions required for the AWS cloud user associated with the Morpheus Cloud integration.

In the end, the following fields must be filled in order to complete the process:

- **COSTING BUCKET:** The S3 bucket name
- **COSTING REGION:** The region the bucket was created in
- **COSTING FOLDER:** This is the report path prefix if you configured one earlier
- **COSTING REPORT NAME:** The name given to your CUR report
- **COSTING KEY:** If the IAM user for this AWS cloud integration does not have access to the S3 bucket with the CUR data, enter the AWS Key ID for an IAM user with access
- **COSTING SECRET:** If the IAM user for this AWS cloud integration does not have access to the S3 bucket with the CUR data, enter the AWS Secret Key for the IAM account whose Key ID you entered in the previous field
- **LINKED ACCOUNT ID:** If the IAM user for this AWS cloud integration does not have access to the S3 bucket with the CUR data, enter the AWS account number that the IAM user from the above step resides in

Note: If the AWS cloud account is a GovCloud account, enter the COSTING KEY, COSTING SECRET, and LINKED ACCOUNT ID for the master commercial account your GovCloud account is associated with.

Using the Morpheus interface to create a CUR, and optionally a new S3 bucket, the IAM user requires permissions in AWS to be successful. By default, only the root user will have access to billing (including CUR), but billing can be enabled for IAM users following Amazon's documentation for [Activating IAM Access](#). Once IAM access is activated, policies need to be assigned to the user being configured on the cloud in Morpheus. If the user is an administrator of the AWS account, no additional policies are needed. Below are some example policies needed for different scenarios:

Creating a new CUR and new S3 bucket **Click to Expand/Hide**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutBucketPolicy",
        "s3:CreateBucket",
        "s3:ListBucket",
        "s3:GetBucketPolicy",
        "cur:DescribeReportDefinitions",
        "cur:PutReportDefinition"
      ]
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```

    ],
    "Resource": "*"
  }
]
}

```

Creating a new CUR and using an existing S3 bucket [Click to Expand/Hide](#)

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketPolicy",
        "cur:DescribeReportDefinitions",
        "cur:PutReportDefinition"
      ],
      "Resource": "*"
    }
  ]
}

```

Using an existing CUR [Click to Expand/Hide](#)

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cur:DescribeReportDefinitions"
      ],
      "Resource": "*"
    }
  ]
}

```

Important: The user configured on the cloud will need access to the objects in the S3 bucket configured on the CUR. If creating the CUR via the Morpheus UI, this should be done automatically. If an existing CUR report or existing S3 bucket was selected, ensure the user has permissions to access the objects in the configured S3 bucket. Alternatively, the **COSTING KEY** and **COSTING SECRET** can be used to configure a different user that has access to the S3 bucket.

v4.2.2 and Below

Begin by logging into the [AWS Billing Console](#), then click *Create report*.

The screenshot shows the AWS console interface for 'AWS Cost and Usage Reports'. On the left is a navigation sidebar with categories like Home, Cost Management, Billing, and Preferences. The main content area has a header 'AWS Cost and Usage Reports' and a sub-header explaining that these reports provide detailed data for cost analysis. A prominent blue 'Create report' button is visible. Below this, there's a section titled 'Other Reports' with an 'Enable Reports' button. Three report options are listed, each with an icon and a brief description: a bar chart for 'Analyze your cost and usage using AWS Cost Explorer', a line graph for 'Monitor your Reserved Instance (RI) utilization using AWS Cost Explorer', and a document icon for 'AWS Usage Report'.

Include a name for your report and mark the box to “Include resource IDs”. Morpheus uses these resource IDs to map costs to various resources. Click *Next*.

aws Services Resource Groups

AWS Cost and Usage Reports > Create report

Step 1
Report content

Step 2
Delivery options

Step 3
Review

Report content

Report name - required

Morpheus

Report includes

- Account identifiers
- Invoice and Bill Information
- Usage Amount and Unit
- Rates and Cost
- Product Attributes (e.g., instance type, operating system, and region)
- Pricing Attributes (e.g., offer types, and lease lengths)
- Reservation identifiers and related details (for reserved instances only)

Additional report details

Include resource IDs ⓘ

Data refresh settings ⓘ

Automatically refresh your Cost & Usage Report when charges are detected for previous months with closed bills.

Cancel Next

On the following page, begin by identifying an S3 bucket to house reports. Click *Configure* near the top of the page and select an existing bucket or create a new one.

✕
Step 1 of 2: Configure S3 Bucket

In order to receive AWS Cost & Usage Reports, you must have an Amazon S3 bucket created and configured with the appropriate access permissions. You can add an existing bucket or create a new one.

Select existing bucket

S3 bucket name

Choose an existing S3 Bucket ▼

Create a bucket

S3 bucket name

Enter a Bucket name

Region

US East (N. Virginia) ▼

OR

If you create a bucket for AWS Cost and Usage reports, default settings and permissions are applied. After the bucket is created, you can update the settings and permissions on the Amazon S3 console.

Some Regions are disabled by default. To create a bucket in one of these Regions, you must enable the Region first. [Learn more](#)

Cancel
Next

After identifying the bucket, you must mark the box to accept the default policy being applied to the bucket. Click *Save*.

✕
Step 2 of 2: Verify policy

The following default policy will be applied to your bucket:

```
{
  "Version": "2008-10-17",
  "Id": "XXXXXXXXXXXXXXXXXXXX",
  "Statement": [
    {
      "Sid": "XXXXXXXXXXXXXXXXXXXX",
      "Effect": "Allow",
      "Principal": {
        "Service": "billingreports.amazonaws.com"
      }
    }
  ],
}
```

I have confirmed that this policy is correct

Note: Saving the report in the existing S3 bucket will overwrite the current policy.

Cancel
Previous
Save

The default policy applied to the bucket is below:

```

{
  "Version": "2008-10-17",
  "Id": "SomeID",
  "Statement": [
    {
      "Sid": "SomeStmtID",
      "Effect": "Allow",
      "Principal": {
        "Service": "billingreports.amazonaws.com"
      },
      "Action": [
        "s3:GetBucketAcl",
        "s3:GetBucketPolicy"
      ],
      "Resource": "arn:aws:s3:::bucket-name"
    },
    {
      "Sid": "SomeStmtID",
      "Effect": "Allow",
      "Principal": {
        "Service": "billingreports.amazonaws.com"
      },
      "Action": [
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::bucket-name/*"
    }
  ]
}

```

After choosing a bucket, accepting the default policy, and saving the change, you're brought back to the report delivery page. By default, CUR reports are saved to a folder at the path `my-report-name/date-folder`. If this bucket already contains CUR reports, you may want to specify a prefix path in the "Report path prefix" field. Outside of this field, use the default values as shown in the screenshot below, then click *Next*.

AWS
Services ▾
Resource Groups ▾
★

[AWS Cost and Usage Reports](#) > Create report

Step 1
Report content

Step 2
Delivery options

Step 3
Review

Delivery options

S3 bucket - required

Configure

Verify

✔

Valid Bucket

Report path prefix

?

Time granularity

Hourly

Daily

The time granularity on which report data are measured and displayed.

Report versioning

Create new report version

Overwrite existing report

Enable report data integration for

Amazon Athena

Amazon Redshift

Amazon QuickSight

Compression type

GZIP
▾

File format

text/csv

Cancel

Previous

Next

On the following page, make your final review and click *Review and Complete*. Following this, you will see your newly configured report in the list of CUR report(s).

In addition, the AWS cloud user associated with the integration in Morpheus needs IAM policy permission to access Cost Explorer. Attach a policy like the one below to this cloud user:

```
{
  "Version": "2012-10-17",
  "Id": "SomeID",
  "Statement": [
    {
      "Sid": "SomeStmntID",
      "Effect": "Allow",
      "Action": [
        "ce:DescribeReportDefinitions",
        "ce:DescribeCostCategoryDefinition",
        "ce:ListCostCategoryDefinitions"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```
]
}
```

Note: If the Cost Explorer permissions are granted at the master account level, the user will see all costs for each member account; if granted at the member account, only the costs for that member account are available.

With the AWS console configuration steps complete, we can move back into Morpheus. Keep in mind it is only necessary to set up one AWS cloud for Costing since we process all records in the CUR report.

Once back in Morpheus, add or edit the relevant AWS cloud integration (Infrastructure > Clouds > + *ADD OR* click the pencil icon in the row for the chosen AWS integration). Expand the Advanced Options drawer and complete the following fields:

- **COSTING BUCKET:** The S3 bucket name
- **COSTING REGION:** The region the bucket was created in
- **COSTING FOLDER:** This is the report path prefix if you configured one earlier
- **COSTING REPORT NAME:** The name given to your CUR report
- **COSTING KEY:** If the IAM user for this AWS cloud integration does not have access to the S3 bucket with the CUR data, enter the AWS Key ID for an IAM user with access
- **COSTING SECRET:** If the IAM user for this AWS cloud integration does not have access to the S3 bucket with the CUR data, enter the AWS Secret Key for the IAM account whose Key ID you entered in the previous field
- **LINKED ACCOUNT ID:** If the IAM user for this AWS cloud integration does not have access to the S3 bucket with the CUR data, enter the AWS account number that the IAM user from the above step resides in

Note: If the AWS cloud account is a GovCloud account, enter the COSTING KEY, COSTING SECRET, and LINKED ACCOUNT ID for the master commercial account your GovCloud account is associated with.

COSTING BUCKET	<input type="text"/>
COSTING REGION	<input type="text" value="us-east-2"/>
COSTING FOLDER	<input type="text"/>
COSTING REPORT NAME	<input type="text" value="Morpheus"/>
COSTING KEY	<input type="text"/>
COSTING SECRET	<input type="text"/>
LINKED ACCOUNT ID	<input type="text"/>

Save changes to your cloud integration.

Important: It may take as long as one hour for Morpheus to process the next CUR report.

Global (Costing Aggregator Only) (v5.5.1+)

Costing can be created for specific clouds individually, following the steps previously mentioned. If the same account is added multiple times, using different regions, the same CUR file is available to be selected (if the configured user has access). However, this can become a tedious process in needing to configure the CUR on each cloud added to Morpheus. However, Morpheus has a region of *Global (Costing Aggregator Only)*, which can be chosen at the time of adding a cloud. This region is not designed for deploying workloads, it is here primarily for syncing costs. This means that the AWS account added as a cloud in Morpheus as a Global region can sync the cost for all the other regions of the same account added as clouds into Morpheus.

When using AWS Organizations, if the AWS account added as a global region is the management account (formerly known as master account) and consolidated billing is enabled, costs for **all** accounts can be sync'd using the Global region. This means when any AWS and/or regions in the organization are added as clouds in Morpheus, the appropriate costs are applied to them automatically. It does require that **Costing** is enabled on the cloud to see the costs but a Costing Report does not need to be chosen. This enables the use of one cloud added as Global to sync all costs and apply to all AWS clouds added in Morpheus.

Additionally, some costs in AWS are not region specific, for example the *Global* and *No Region* regions. These costs do not apply to the regions of the clouds added in Morpheus. With the Global region added as a cloud in Morpheus, you would be able to see these costs that are applicable to your accounts.

Costing and AWS GovCloud

AWS GovCloud delivers Amazon public cloud infrastructure and features in a way that complies with U.S. government standards for security. GovCloud accounts are applied for and must be associated with a pre-existing standard AWS account and the usage and billing data for the GovCloud account is rolled up into that of the standard AWS account. For that reason, Amazon recommends creating a new standard account solely to house the GovCloud account if usage and billing must be tracked separately.

Since GovCloud accounts do not have access to billing data directly, Morpheus must be able to access it through the associated standard account. You could do this by creating the Morpheus cloud integration through the standard account itself or by integrating the GovCloud account and supplying an Access Key and Secret Key for the standard account when configuring costing. When needed, add the additional credentials for the standard commercial account as described below:

1. Add a new AWS Cloud or edit an existing one
2. Expand the Advanced Options section
3. Complete the following fields in addition to other required fields needed to set up costing as described in the previous section:
 - **COSTING KEY:** The AWS Key ID for an IAM user with sufficient access who is associated with the standard commercial account
 - **COSTING SECRET:** The AWS Secret Key for an IAM user with sufficient access who is associated with the standard commercial account
 - **LINKED ACCOUNT ID:** The AWS account number for the standard commercial account in which the IAM user referenced in the prior bullets resides
4. Save the changes to the AWS Cloud integration

When credentials are configured correctly, you will be able to select an existing Costing and Usage Report (CUR) from the appropriate S3 bucket if it already exists. If not, you can create one directly from the add/edit AWS Cloud modal in Morpheus.

▼ Advanced Options

IMAGE TRANSFER STORE	AWS VM Import
EBS ENCRYPTION	Off
COSTING BUCKET	bq-...-...
COSTING FOLDER	
COSTING REPORT NAME	
COSTING KEY	AKIA...
COSTING SECRET
LINKED ACCOUNT ID	...

AWS Reserved Instances and Savings Plans

Amazon AWS public cloud offers Reserved Instances (RI) and Savings Plans, which allow organizations with consistent use patterns to reduce cloud spend significantly. Morpheus analyzes AWS cloud usage and spend, which allows it to make intelligent recommendations that can lead to significant savings. This data can be reviewed in the Reservation Recommendations and Savings Plan Recommendations tables on any AWS Cloud detail page (Infrastructure > Clouds > Selected Amazon Cloud).

Savings Plans potentially offer greater than 70% savings in exchange for a commitment to consistent usage levels for a 1- or 3-year term. Morpheus provides Savings Plan guidance based on learned analytics; allowing you to analyze Savings Plans based on different term commitments and upfront costs to choose the best savings plan.

SAVINGS PLAN RECOMMENDATIONS

EC2 Instance 3-Year No upfront 30 Days

ITEM		CURRENT SPEND		RECOMMENDATION				TERM COST			
SERVICE	TYPE	AVG / HOUR	TOTAL / MONTH	COMMITMENT	EST. ON-DEMAND	TOTAL / MONTH	SAVINGS	SAVINGS %	ROI (MONTHS)	UPFRONT	TOTAL
EC2	T2	\$2.65	\$1,936.24	\$1.25	\$0.25	\$1,095.28	\$840.96	43.4%	15.64	\$0.00	\$32,928.84
EC2	M4	\$0.35	\$255.87	\$0.17	\$0.02	\$134.20	\$121.67	47.6%	17.12	\$0.00	\$4,362.48
EC2	M4	\$1.15	\$839.54	\$0.75	\$0.25	\$728.72	\$110.81	13.2%	4.75	\$0.00	\$19,710.00
TOTAL			\$3,031.65			\$1,958.21	\$1,073.44			\$0.00	\$57,001.32

Reserved Instances (RI) provide a discounted hourly rate and optional capacity reservation for EC2 instances. AWS billing automatically applies your RI-discounted rate when the attributes of EC2 instance usage match attributes of an active RI. Morpheus provides RI guidance based on learned analytics.

RESERVATION RECOMMENDATIONS

Amazon Elastic Compute Cloud - Compute 1-Year No upfront

ITEM		CURRENT USAGE		RECOMMENDATION				TERM COST			
SERVICE	TYPE	AVG	TOTAL / MONTH	RESERVED	TOTAL / MONTH	SAVINGS	SAVINGS %	ROI (MONTHS)	UPFRONT	TOTAL	
EC2	T2.NANO	16.9	\$71.55	17	\$44.68	\$26.87	38%	4.51	\$0.00	\$536.11	
EC2	M4.LARGE	3	\$256.23	3	\$186.37	\$69.86	27%	3.27	\$0.00	\$2,236.43	
EC2	T2.NANO	38797	\$1,954.22	346	\$1,262.90	\$468.72	24%	2.88	\$0.00	\$15,154.80	
EC2	T2.NANO	1.99	\$9.87	2	\$6.86	\$3.01	30%	3.66	\$0.00	\$82.34	
TOTAL			\$2,291.87		\$1,500.81	\$568.47			\$0.00	\$18,009.68	

To retrieve Reserved Instances and Savings Plans data, the user configured on the cloud will require access to **Cost Explorer**. Below is an example IAM policy with Cost Explorer access:

Cost Explorer Policy [Click to Expand/Hide](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ce:*"
      ]
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```
    ],  
    "Resource": "*"    
  }  
]  
}
```

AWS IAM Permissions

Below are the AWS IAM Permissions required for Morpheus services.

autoscaling

```
"autoscaling:AttachInstances",  
"autoscaling:AttachLoadBalancerTargetGroups",  
"autoscaling:CreateAutoScalingGroup",  
"autoscaling>DeleteAutoScalingGroup",  
"autoscaling>DeleteLaunchConfiguration",  
"autoscaling>DeletePolicy",  
"autoscaling:DescribeAutoScalingGroups",  
"autoscaling:DescribeLaunchConfigurations",  
"autoscaling:DescribeLoadBalancers",  
"autoscaling:DescribePolicies",  
"autoscaling:DetachInstances",  
"autoscaling:PutScalingPolicy",  
"autoscaling:UpdateAutoScalingGroup",
```

cloudformation

```
"cloudformation:CreateStack",  
"cloudformation>DeleteStack",  
"cloudformation:DescribeStackEvents",  
"cloudformation:DescribeStackResources",  
"cloudformation:DescribeStacks",  
"cloudformation:GetTemplate",  
"cloudformation:UpdateStack",  
"cloudformation:ValidateTemplate",
```

cloudwatch

```
"cloudwatch>DeleteAlarms",  
"cloudwatch:DescribeAlarms",  
"cloudwatch:GetMetricStatistics",  
"cloudwatch:PutMetricAlarm",
```

costexplorer

```
"ce:*",
```

Cost and Usage Reports

```
"cur:DescribeReportDefinitions",  
"cur:PutReportDefinition",
```

ec2

```
"ec2:AllocateAddress",  
"ec2:AssignPrivateIpAddresses",  
"ec2:AssociateAddress",  
"ec2:AttachInternetGateway",  
"ec2:AttachNetworkInterface",  
"ec2:AttachVolume",  
"ec2:AuthorizeSecurityGroupEgress",  
"ec2:AuthorizeSecurityGroupIngress",  
"ec2:CancelExportTask",  
"ec2:CancelImportTask",  
"ec2:CopyImage",  
"ec2:CopySnapshot",  
"ec2:CreateEgressOnlyInternetGateway",  
"ec2:CreateImage",  
"ec2:CreateInstanceExportTask",  
"ec2:CreateInternetGateway",  
"ec2:CreateKeyPair",  
"ec2:CreateNatGateway",  
"ec2:CreateNetworkAcl",  
"ec2:CreateNetworkAclEntry",  
"ec2:CreateNetworkInterface",  
"ec2:CreateRoute",  
"ec2:CreateSecurityGroup",  
"ec2:CreateSnapshot",  
"ec2:CreateSubnet",  
"ec2:CreateTags",  
"ec2:CreateVolume",  
"ec2:CreateVpc",  
"ec2>DeleteEgressOnlyInternetGateway",  
"ec2>DeleteInternetGateway",  
"ec2>DeleteKeyPair",  
"ec2>DeleteNatGateway",  
"ec2>DeleteNetworkAcl",  
"ec2>DeleteNetworkAclEntry",  
"ec2>DeleteNetworkInterface",  
"ec2>DeleteRoute",  
"ec2>DeleteSecurityGroup",  
"ec2>DeleteSnapshot",
```

(continues on next page)

(continued from previous page)

```
"ec2:DeleteSubnet",
"ec2:DeleteTags",
"ec2:DeleteVolume",
"ec2:DeleteVpc",
"ec2:DeregisterImage",
"ec2:DescribeAccountAttributes",
"ec2:DescribeAddresses",
"ec2:DescribeAvailabilityZones",
"ec2:DescribeClassicLinkInstances",
"ec2:DescribeClientVpnConnections",
"ec2:DescribeClientVpnEndpoints",
"ec2:DescribeConversionTasks",
"ec2:DescribeEgressOnlyInternetGateways",
"ec2:DescribeExportTasks",
"ec2:DescribeImageAttribute",
"ec2:DescribeImages",
"ec2:DescribeImportImageTasks",
"ec2:DescribeImportSnapshotTasks",
"ec2:DescribeInstances",
"ec2:DescribeInstanceStatus",
"ec2:DescribeInstanceTypes",
"ec2:DescribeInternetGateways",
"ec2:DescribeKeyPairs",
"ec2:DescribeNatGateways",
"ec2:DescribeNetworkAcls",
"ec2:DescribeNetworkInterfaceAttribute",
"ec2:DescribeNetworkInterfaces",
"ec2:DescribeRegions",
"ec2:DescribeRouteTables",
"ec2:DescribeSecurityGroupReferences",
"ec2:DescribeSecurityGroups",
"ec2:DescribeSnapshotAttribute",
"ec2:DescribeSnapshots",
"ec2:DescribeStaleSecurityGroups",
"ec2:DescribeSubnets",
"ec2:DescribeTags",
"ec2:DescribeTransitGateways",
"ec2:DescribeTransitGatewayVpcAttachments",
"ec2:DescribeVolumeAttribute",
"ec2:DescribeVolumes",
"ec2:DescribeVolumeStatus",
"ec2:DescribeVpcAttribute",
"ec2:DescribeVpcClassicLink",
"ec2:DescribeVpcClassicLinkDnsSupport",
"ec2:DescribeVpcEndpoints",
"ec2:DescribeVpcEndpointServices",
"ec2:DescribeVpcPeeringConnections",
"ec2:DescribeVpcs",
"ec2:DescribeVpnGateways",
"ec2:DetachInternetGateway",
"ec2:DetachNetworkInterface",
"ec2:DetachVolume",
```

(continues on next page)

(continued from previous page)

```
"ec2:DisassociateAddress",
"ec2:GetPasswordData",
"ec2:ImportImage",
"ec2:ImportInstance",
"ec2:ImportKeyPair",
"ec2:ImportSnapshot",
"ec2:ImportVolume",
"ec2:ModifyImageAttribute",
"ec2:ModifyInstanceAttribute",
"ec2:ModifyNetworkInterfaceAttribute",
"ec2:ModifySnapshotAttribute",
"ec2:ModifySubnetAttribute",
"ec2:ModifyVolumeAttribute",
"ec2:RebootInstances",
"ec2:RegisterImage",
"ec2:ReleaseAddress",
"ec2:ReplaceNetworkAclAssociation",
"ec2:ReplaceNetworkAclEntry",
"ec2:ResetImageAttribute",
"ec2:ResetInstanceAttribute",
"ec2:ResetNetworkInterfaceAttribute",
"ec2:ResetSnapshotAttribute",
"ec2:RevokeSecurityGroupEgress",
"ec2:RevokeSecurityGroupIngress",
"ec2:RunInstances",
"ec2:StartInstances",
"ec2:StopInstances",
"ec2:TerminateInstances",
"ec2:UnassignPrivateIpAddresses",
"ec2:UpdateSecurityGroupRuleDescriptionsEgress",
```

eks

```
"eks:*",
```

elasticloadbalancing

```
"elasticloadbalancing:AddTags",
"elasticloadbalancing:ApplySecurityGroupsToLoadBalancer",
"elasticloadbalancing:AttachLoadBalancerToSubnets",
"elasticloadbalancing:CreateListener",
"elasticloadbalancing:CreateLoadBalancer",
"elasticloadbalancing:CreateRule",
"elasticloadbalancing:CreateTargetGroup",
"elasticloadbalancing>DeleteListener",
"elasticloadbalancing>DeleteLoadBalancer",
"elasticloadbalancing>DeleteRule",
"elasticloadbalancing>DeleteTargetGroup",
```

(continues on next page)

(continued from previous page)

```
"elasticloadbalancing:DescribeLoadBalancers",  
"elasticloadbalancing:DescribeRules",  
"elasticloadbalancing:DescribeTargetGroups",  
"elasticloadbalancing:ModifyListener",  
"elasticloadbalancing:ModifyTargetGroupAttributes",  
"elasticloadbalancing:RegisterTargets",  
"elasticloadbalancing:SetSecurityGroups",  
"elasticloadbalancing:SetSubnets",
```

elasticsearch

```
"es:DescribeElasticsearchDomains",  
"es:ListDomainNames",
```

iam

```
"iam:ListGroups",  
"iam:ListInstanceProfiles",  
"iam:ListRoles",
```

kms

```
"kms:Decrypt",  
"kms:GenerateDataKey",
```

rds

```
"rds:AddRoleToDBCluster",  
"rds:AddTagsToResource",  
"rds:ApplyPendingMaintenanceAction",  
"rds:AuthorizeDBSecurityGroupIngress",  
"rds:CopyDBClusterSnapshot",  
"rds:CopyDBParameterGroup",  
"rds:CopyDBSnapshot",  
"rds>CreateDBCluster",  
"rds>CreateDBClusterSnapshot",  
"rds>CreateDBInstance",  
"rds>CreateDBInstanceReadReplica",  
"rds>CreateDBSecurityGroup",  
"rds>CreateDBSnapshot",  
"rds>DeleteDBCluster",  
"rds>DeleteDBInstance",  
"rds>DeleteDBSecurityGroup",  
"rds>DeleteDBSnapshot",  
"rds:DescribeAccountAttributes",
```

(continues on next page)

```
"rds:DescribeCertificates",
"rds:DescribeDBClusterParameterGroups",
"rds:DescribeDBClusterParameters",
"rds:DescribeDBClusters",
"rds:DescribeDBClusterSnapshotAttributes",
"rds:DescribeDBClusterSnapshots",
"rds:DescribeDBEngineVersions",
"rds:DescribeDBInstances",
"rds:DescribeDBLogFiles",
"rds:DescribeDBParameterGroups",
"rds:DescribeDBParameters",
"rds:DescribeDBSecurityGroups",
"rds:DescribeDBSnapshotAttributes",
"rds:DescribeDBSnapshots",
"rds:DescribeDBSubnetGroups",
"rds:DescribeEngineDefaultClusterParameters",
"rds:DescribeEngineDefaultParameters",
"rds:DescribeEventCategories",
"rds:DescribeEvents",
"rds:DescribeOptionGroupOptions",
"rds:DescribeOptionGroups",
"rds:DescribeOrderableDBInstanceOptions",
"rds:ListTagsForResource",
"rds:ModifyDBCluster",
"rds:ModifyDBClusterParameterGroup",
"rds:ModifyDBClusterSnapshotAttribute",
"rds:ModifyDBInstance",
"rds:ModifyDBParameterGroup",
"rds:ModifyDBSnapshotAttribute",
"rds:PromoteReadReplica",
"rds:RebootDBInstance",
"rds:RemoveTagsFromResource",
"rds:RestoreDBClusterFromSnapshot",
"rds:RestoreDBClusterToPointInTime",
"rds:RestoreDBInstanceFromDBSnapshot",
"rds:RestoreDBInstanceToPointInTime",
"rds:RevokeDBSecurityGroupIngress",
"rds:StartDBInstance",
"rds:StopDBInstance",
```

route53

```
"route53:ChangeResourceRecordSets",
"route53:GetHostedZone",
"route53:ListHostedZones",
"route53:ListResourceRecordSets",
```

s3

```
"s3:AbortMultipartUpload",  
"s3:CreateBucket",  
"s3>DeleteBucket",  
"s3>DeleteObject",  
"s3>DeleteObjectVersion",  
"s3:GetBucketLocation",  
"s3:GetBucketPolicy",  
"s3:GetObject",  
"s3:GetObjectVersion",  
"s3:ListAllMyBuckets",  
"s3:ListBucket",  
"s3:ListBucketMultipartUploads",  
"s3:ListBucketVersions",  
"s3:ListMultipartUploadParts",  
"s3:PutBucketPolicy",  
"s3:PutObject",
```

Systems Manager

```
"ssm:GetParameters",
```

Azure (Public)**Overview**

Morpheus offers a complete Integration with Microsoft Azure including the following:

- Virtual Machine Sync, Create, Delete, Manage, RBAC, Tenant Permissions, Policies
- Resource Group Sync, Create, Delete, RBAC, Tenant Permissions
- Network Sync, Create, Delete, RBAC, Tenant Permissions
- Subnet Sync, Create, Delete, RBAC, Tenant Permissions
- Security Group Sync, Create, Delete, Tenant Permissions
- Security Group Rule Sync, Create, Delete, Tenant Permissions
- ARM Blueprints, Spec Templates, Deployment Logs Sync, Git/GitHub Integration
- MSSQL Service Sync, Create, Delete, Manage, RBAC, Tenant Permissions
- AKS Sync, Sync, Create, Delete, Manage, RBAC, Tenant Permissions
- Backup Create, Delete, Manage, RBAC, Policies
- Storage Sync, Create, Delete, Manage, Browse, RBAC, Tenant Permissions, Policies
- Marketplace Sync
- Private Image Sync & Upload
- Azure Marketplace Custom Library Item Support

- Remote Console (SSH & RDP)
- Lifecycle Management
- Availability Set Support
- Scale Set Sync, Create, Assign, Manage, Delete
- Azure Load Balancer Create, Assign, Manage, Delete, RBAC, Tenant Permissions
- Docker (VM) Cluster Sync, Create, Delete, Manage, RBAC, Tenant Permissions
- Kubernetes (VM) Cluster Sync, Create, Delete, Manage, RBAC, Tenant Permissions
- Service Plan Sync, Tenant Permissions, RBAC
- Pricing Sync RBAC, Tenant Permissions, Markup
- Costing Sync, Reporting, Invoicing
- Reservations Sync, Guidance Recommendations
- Azure Stack Support
- Tag Bi-Directional Sync, Creation, Deletion Policy Enforcement
- Cost Estimator
- Azure US Gov Support
- Azure China Support
- Azure Germany Support
- CSP Account Support

Requirements

Morpheus Azure Integration requires Owner or Contributor access to subscription via App Registration. Adding an Azure Cloud or Clouds to Morpheus will require the following:

- Azure Subscription ID
- Directory (tenant) ID
- Application (client) ID
- Application (client) Secret
- Application (client) must be Owner or Contributor of Subscription

CSP Accounts require the additional following input:

- CSP Directory (tenant) ID
- CSP Application (client) ID
- CSP Application (client) SECRET (Web App Key)

The Morpheus appliance requires outbound HTTPS (443) access to the Azure endpoints. Depending on the type of cloud you choose when adding Azure, ensure the proper endpoints are allowed:

Global Azure Cloud

- <https://management.core.windows.net> (ServiceManagementUrl)
- <https://management.azure.com> (ResourceManagerUrl)

- <https://login.microsoftonline.com> (ActiveDirectoryAuthority)

US Gov Cloud

- <https://management.core.usgovcloudapi.net> (ServiceManagementUrl)
- <https://management.usgovcloudapi.net> (ResourceManagerUrl)
- <https://login.microsoftonline.us> (ActiveDirectoryAuthority)

Germany Cloud

- <https://management.core.cloudapi.de> (ServiceManagementUrl)
- <https://management.microsoftazure.de> (ResourceManagerUrl)
- <https://login.microsoftonline.de> (ActiveDirectoryAuthority)

China Cloud

- <https://management.core.chinacloudapi.cn> (ServiceManagementUrl)
- <https://management.chinacloudapi.cn> (ResourceManagerUrl)
- <https://login.chinacloudapi.cn> (ActiveDirectoryAuthority)

Credentials & Permissions

Morpheus authenticates with Azure via an App Registration with an Owner or Contributor Role on a Subscription. Use the steps below to create and collect the required credentials and assign the required permissions to integrate Azure with Morpheus.

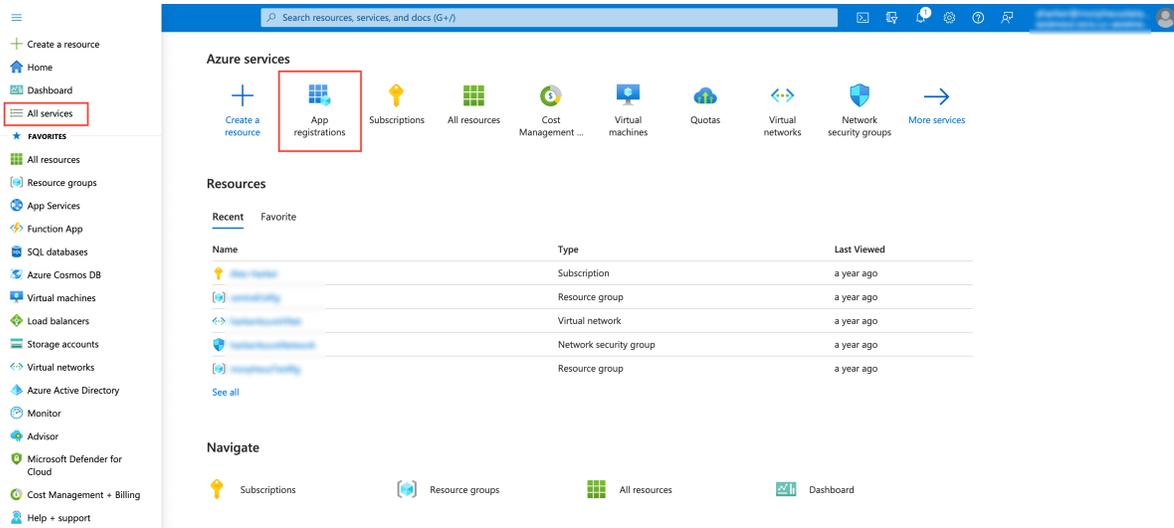
Warning: Using an App Registration (service principal) that has selective resource permissions and is not an Owner or Contributor of the Subscription is not supported and will cause failures/issues. Please confirm the App Registration you use to integrate Azure with Morpheus has Owner or Contributor permissions on the specified Subscription before contacting support.

Create an App Registration

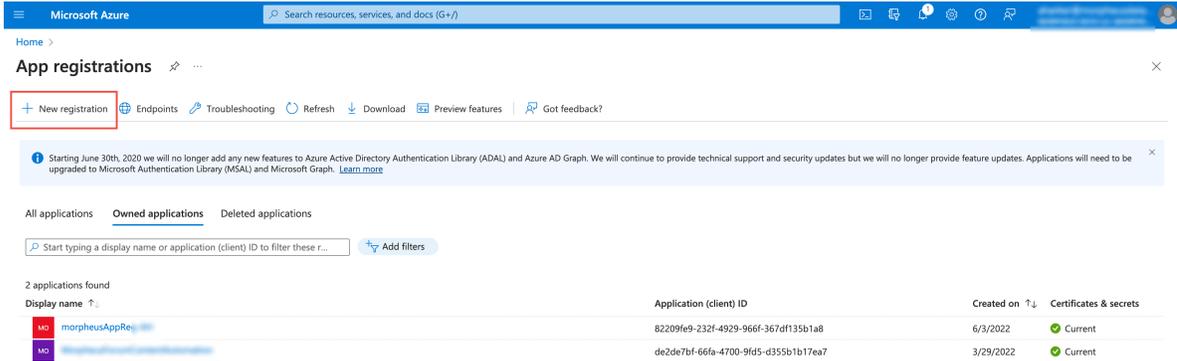
If you do not have an existing Azure Active Directory App Registration, or you wish to use a new one for Morpheus, you will need to create one.

Tip: If you can't find sections of the Azure portal discussed in this integration guide, such as App Registrations or Subscriptions, you may be able to find them under the "Favorites" section within the main hamburger menu, at the "All Resources" page from within the main hamburger menu, or by using the global search bar.

1. Log into the Azure web console
2. Select "App Registrations", you may have to go to the "All Services" page to find it



3. Click “+ New registration”



- Next, give app a name, specify which accounts may access this API, specify Web for the Redirect URI type and enter any url for the Sign-on URL:
- Click Register and your new App Registration will be created.

Microsoft Azure

Home > App registrations > Register an application

* Name
The user-facing display name for this application (this can be changed later).
morpheusAppReg

Supported account types
Who can use this application or access this API?
 Accounts in this organizational directory only (Morpheus Data LLC only - Single tenant)
 Accounts in any organizational directory (Any Azure AD directory - Multitenant)
 Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
 Personal Microsoft accounts only
[Help me choose...](#)

Redirect URI (optional)
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.
 Web | https://morpheusAppReg

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from [Enterprise applications](#).

By proceeding, you agree to the [Microsoft Platform Policies](#)

Register

Now that we have (or already had) our App Registration, we will gather the credentials required for the Morpheus Azure integration.

Copy Directory (tenant) and Application (client) IDs

The App Registration Directory (tenant) and Application (client) ID are required for the Morpheus Azure integration. Both can be found in the overview section of the App Registration.

1. Go to the Overview section of your App Registration
2. Copy the Directory (tenant) ID
3. Store/Paste for use as the Tenant ID when Adding your Azure cloud in Morpheus
4. Copy the Application (client) ID
5. Store/Paste for use as the Client ID when Adding your Azure cloud in Morpheus

Microsoft Azure

Home > App registrations > morpheusAppReg-AH

Overview

Essentials

Display name	: morpheusAppReg	Client credentials	: 0_certificate_2_secret
Application (client) ID	: 822091...	Redirect URIs	: 1_web_0_spa_0_public_client
Object ID	: a4d061...	Application ID URI	: Add an Application ID URI
Directory (tenant) ID	: b16e21...	Managed application in...	: morpheusAppReg-AH

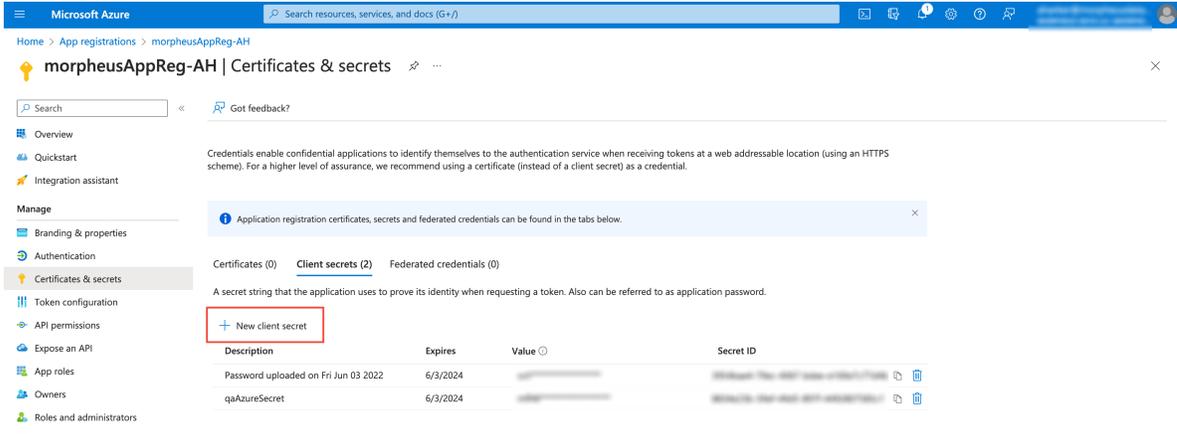
Supported account types: My organization only

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. [Learn more](#)

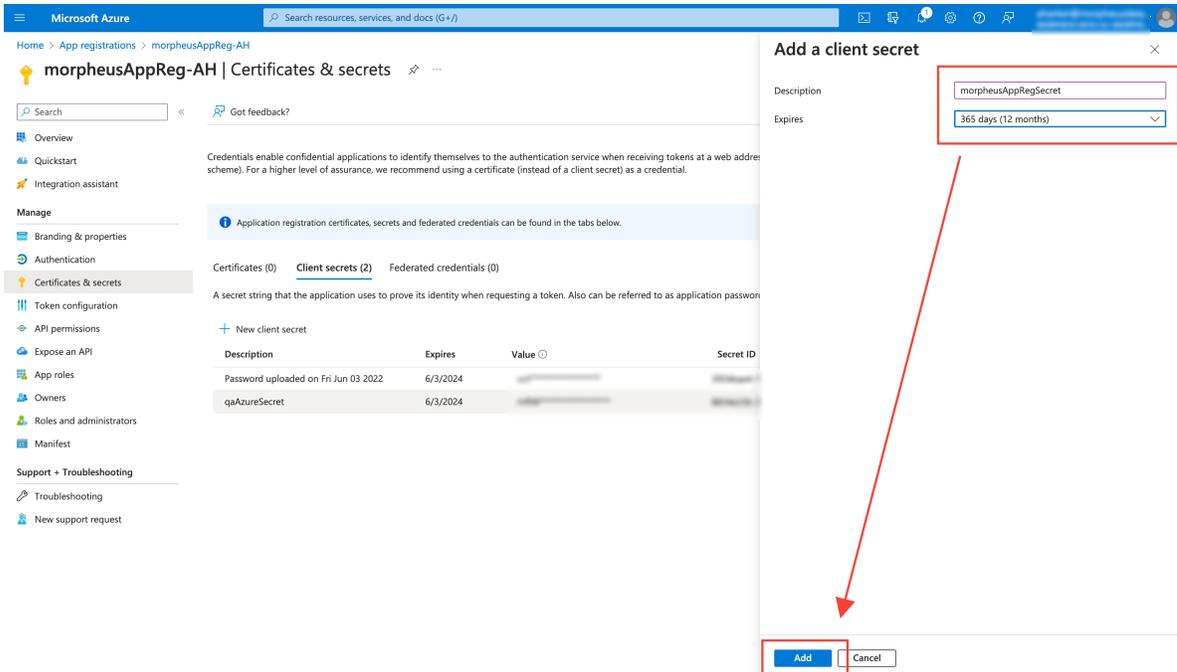
Generate a Client Secret

While still in your App Registration:

1. Select Certificates & secrets in the Manage Section
2. Select + New client secret



3. The “Add a client secret” modal will come up
4. Add a description to help identify the secret in the future
5. Select a duration
6. Select *Add*



7. Copy the newly generated Client Secret Value. It is important to copy the Client Secret Value now as it will not be displayed/available

Important: Copy the key value before continuing as it will not be displayed/available again.

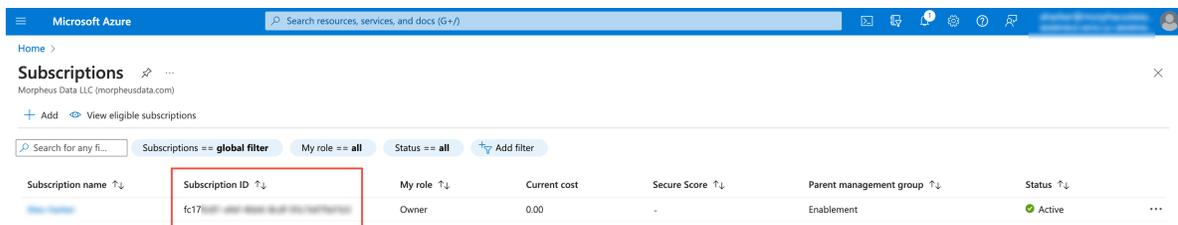
8. Store/Paste for use as the Client Secret when Adding your Azure cloud in Morpheus

You now have 3 of the 4 credentials required for Morpheus Azure cloud integration. The last credential required is the Azure Subscription ID.

Subscription ID

To get the Azure Subscription ID:

1. Navigate to the main Subscriptions section. One way is to search for “Subscriptions” and select Subscriptions in the search results
2. In the main “Subscriptions” section, copy the Subscription ID

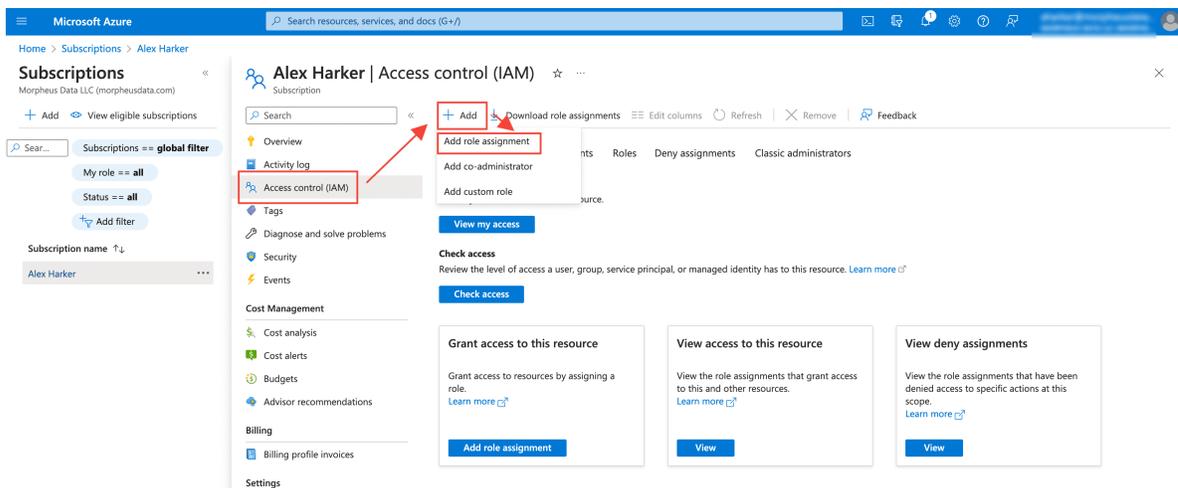


3. Store/Paste for use as the Subscription ID when Adding your Azure cloud in Morpheus

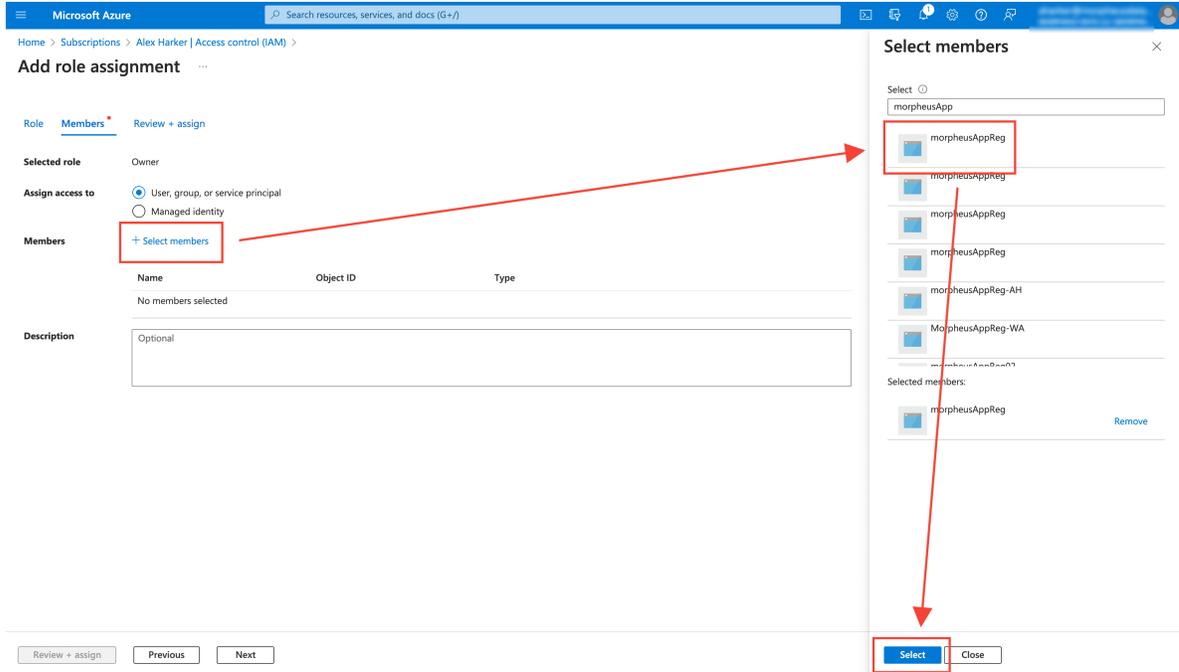
Make App Registration owner or contributor of Subscription

The App Registration created/used needs to be an owner of the Azure Subscription used for the Morpheus cloud integration. If lesser permissions are given or permissions are assigned at individual resource levels, Morpheus will not be able to properly inventory/sync, create and/or remove resources.

1. In the main Subscriptions section, click on the name of the Subscription
2. With the Subscription detail open, select “Access Control (IAM)”
3. Click “+ ADD” and then click “Add role assignment” from the pop-out menu



4. Click on the tab for “Privileged administrator roles”
5. Select Owner or Contributor and click “Next”
6. Add Members to the Role Assignment by clicking “+ Select members”
7. Select the App Registration from the search results and click Select



8. Click “Review + Assign”

Add an Azure Cloud Integration

To add a new Azure Cloud integration into Morpheus using the credentials created/collected from the previous section, perform the following:

1. In Morpheus, navigate to Infrastructure > Clouds and select + ADD

The screenshot shows the Morpheus web interface. The top navigation bar includes the Morpheus logo, a search bar, and a 'Support' link. Below this is a secondary navigation bar with tabs for 'Operations', 'Provisioning', 'Infrastructure' (highlighted with a red box), 'Backups', 'Logs', 'Monitoring', 'Tools', and 'Administration'. Underneath are icons for 'Groups', 'Clouds' (highlighted with a red box), 'Clusters', 'Hosts', 'Network', 'Load Balancers', 'Storage', 'Keys & Certs', and 'Boot'. A red arrow points from the 'Clouds' icon to a '+ ADD' button in the 'CLOUDS' section. The 'CLOUDS' section has a search bar, a dropdown for 'All', and a 'Select Type' dropdown. Below this is a table with columns: TYPE, NAME, LOCATION, TENANT, HOSTS, VMS, BARE METAL, and STATUS. The table content is blurred. At the bottom of the page, there is a footer with the Morpheus logo and copyright information: '© 2020 MORPHEUS DATA, LLC. ALL RIGHTS RESERVED — TERMS & CONDITIONS | PRIVACY POLICY'.

2. Select “AZURE (PUBLIC)” from the Cloud Types list and click *NEXT*

CREATE CLOUD ✕

CLOUD
CONFIGURE
GROUP
REVIEW

	Alibaba Cloud	ALIBABA CLOUD	Alibaba is a high performance , highly scalable global public cloud with large Chinese customers.
	amazon web services	AMAZON	Amazon cloud
	Microsoft Azure	AZURE (PUBLIC)	Azure
	DELL EMC	DELL	Dell Server zone - manually managed servers
	MISSING LOGO	DIGITAL OCEAN PLUGIN	Digital Ocean Plugin Description
	DigitalOcean	DIGITALOCEAN	Digital Ocean
	GOOGLE CLOUD		

PREVIOUS
NEXT

3. Populate the Following

Cloud Configuration

NAME Name of the Cloud in Morpheus

CODE Unique code used for api/cli, automation and policies.

LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT If Visibility is set to Private, select the Tenant the Cloud resources will assigned to.

ENABLED When disabled, automatic Cloud sync is paused and the Cloud will not be selectable for provisioning.

AUTOMATICALLY POWER ON VMS When enabled, Morpheus will maintain the expected power state of managed VMs. Morpheus will power on any managed VMs in the Cloud that have been shut down for unknown reasons (not powered off by Morpheus) to ensure availability of services.

Note: When “AUTOMATICALLY POWER ON VMS” is enabled, the power state of managed VMs should be maintained in Morpheus. This setting is not applicable to discovered/unmanaged resources.

Details

CLOUD TYPE

- Global (Azure Cloud)
- US Gov (Azure US Government)
- German (Azure German Cloud)
- China (Azure China Cloud)

SUBSCRIPTION ID The target Azure Subscription ID obtained from the previous section

TENANT ID The Directory (tenant) ID obtained from the previous section

CREDENTIALS If available, select a pre-stored Client ID and Client Secret set. Alternatively, select “Local Credentials” to enter the Client ID and Client Secret without storing them for later access. “client id and secret” under the “New Credentials” heading can also be selected so your Client ID and Client Secret will be stored for later use in either the Morpheus credential store or in an integrated third party secret store

CLIENT ID The Application (client) ID obtained from the previous section

CLIENT SECRET The Application (client) Secret obtained from the previous section

LOCATION Once valid credentials are populated above and Morpheus is able to successfully authenticate with Azure, the available locations/regions will populate.

REGION Scope the Cloud to either “All” regions or a selected Azure region

RESOURCE GROUP

- Select “All” to scope the Cloud to all available Resource Groups in the specified location/region.
- Select a single Resource Group to limit Morpheus resource creation, selection and discovery to just this Resource Group.

INVENTORY EXISTING INSTANCES Check to enable discovery/inventory of existing VM’s in the scoped Region and Resource Group(s)

ACCOUNT TYPE Standard, EA or CSP

Note: For CSP Accounts, also enter CSP TENANT ID, CSP CLIENT ID and CSP CLIENT SECRET in the Advanced Options section. In order to enable cost sync for CSP accounts, the “CSP CUSTOMER” checkbox must be marked and “COSTING” should be set to “Costing” rather than “Costing and Reservations”.

For the CSP Client Secret, enter the Web App Key rather than the Native App Key. This should be accessed from the Microsoft Partner Center rather than the Azure web console. If this is not, Plans may sync but Price Sets and Prices would not.

CREATE CLOUD ✕

CLOUD > **CONFIGURE** > GROUP > REVIEW

NAME

CODE

LOCATION

Details

CLOUD TYPE

SUBSCRIPTION ID

TENANT ID

CLIENT ID

CLIENT SECRET

LOCATION

RESOURCE GROUP

INVENTORY EXISTING INSTANCES

INVENTORY LEVEL

ACCOUNT TYPE

▶ Advanced Options

▶ Provisioning Command

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM's if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT INSTALL MODE

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud Init / Unattend (when available): (DEFAULT) Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus

will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

AZURE COSTING MODE Standard, CSP, or Azure Plan

Example configurations but choose what is applicable to the tenant:

Table 23: Example Azure Costing Configurations

Account Type	Azure Costing Mode	Notes
Standard (Pay as you go)	Standard	
EA (Enterprise Agreement)	Standard	
CSP (Cloud Solution Provider)	CSP	CSP Tenant, ID, Client ID, and Client Secret required
CSP (Cloud Solution Provider)	Azure Plan (Microsoft Customer Agreement)	CSP Tenant, ID, Client ID, and Client Secret required on the primary cloud

4. Once done configuring the Cloud, select *NEXT*. NOTE all specified values except the Subscription ID can be changes after the Cloud is created.
5. Next select an existing Group to add the Azure Cloud to, or create a new Group, then select *NEXT*

CREATE CLOUD ✕

CLOUD > CONFIGURE > GROUP > REVIEW

USE EXISTING CREATE NEW

Configuration

NAME

CODE

LOCATION

▶ Advanced Options

6. Review the configuration and then select *COMPLETE*

CREATE CLOUD ✕

CLOUD > CONFIGURE > GROUP > REVIEW

Name: newAzureCloud
Location: West Central US
Domain: localdomain
Group: Azure
Visibility: private
Scale Priority: 1
Type: Azure (Public)

[PREVIOUS](#) [COMPLETE](#)

Your new Azure Cloud integration will be created and begin to sync.

Note: The initial sync of an Azure Cloud can take some time due to Marketplace data sync.

TYPE	NAME	LOCATION	TENANT	HOSTS	VMS	BARE METAL	STATUS
Microsoft Azure	newAzureCloud	West Central US	master	0	0	0	OK

Azure Stack

Overview

Azure Stack is Microsoft's Azure Cloud for on-premises environments. Azure Stack contains the core Azure services, allowing organizations to take advantage of Azure's offerings with the security, compliance, and financial benefits of hosting it in their own data-centers.

- Virtual Machine Provisioning
- Backups / Snapshots
- Resource Group Sync & Selection
- Network Sync & Selection
- Security Group Sync & Selection
- Storage Account Sync & Selection
- Marketplace Search and Provisioning
- Remote Console
- Periodic Synchronization
- Lifecycle Management and Resize
- Availability Set Support
- Azure Load Balancers
- Azure Storage
- Docker Host Provisioning & Management
- Service Plan Sync
- Pricing Sync with markup options
- Cost Estimator

Combine these features with public Azure and Morpheus can provide a single pane of glass and self service portal for managing instances scattered across both Azure offerings.

Requirements

Azure Stack Accessibility

By default, the Azure Stack management url's are not accessible from an external network. Port mappings and DNS must be configured for communication between the Morpheus Appliance and Azure Stack.

Important: In order to communicate with Azure Stack, Morpheus must be able to reach the internal Azure Stack network. The Azure Stack Portal needs to be exposed to the Morpheus Appliances' network with corresponding entries added to DNS.

One option to expose the Internal Azure Stack network to the Morpheus Appliances network is to use the `Expose-AzureStackPortal.ps1` powershell script from <https://gallery.technet.microsoft.com/scriptcenter/Expose-the-Azure-Stack-7ef68b19>. An Azure Stack Port Mapping Tool is also available.

Below is a sample output from the script for reference:

```
[Admin Portal] Created port mappings on 10.30.23.120 to 192.168.102.8
[Admin Portal] Ports: 13011 30015 13001 13010 13021 13020 443 13003 12646 12647 12648.
↪12649 12650 12495 13026 12499
[Admin Portal] DNS: 10.30.23.120 - adminportal.local.azurestack.external adminmanagement.
↪local.azurestack.external

[Tenant Portal] Created port mappings on 10.30.23.121 to 192.168.102.10
[Tenant Portal] Ports: 13011 30015 13001 13010 13021 13020 443 13003 12646 12647 12648.
↪12649 12650 12495 13026 12499
[Tenant Portal] DNS: 10.30.23.121 - portal.local.azurestack.external management.local.
↪azurestack.external

[Blob Storage] Created port mappings on 10.30.23.122 to 192.168.102.4
[Blob Storage] Ports: 80 443
[Blob Storage] DNS: 10.30.23.122 *.blob.local.azurestack.external

VERBOSE: DNS delegation/forwarding is optional, change the DNS records on MAS-DC01.
↪manually (dnsmgmt.msc from Host).
[DNS Delegation] Created port mappings on 10.30.23.120 to 192.168.200.224
[DNS Delegation] Ports: 53 (TCP/UDP)
[DNS Delegation] DNS: local.azurestack.external NS 10.30.23.120
[DNS Delegation] Change records on MAS-DC01 manually `if` you plan to use DNS forwarding.
[DNS Delegation] Change records back to the original internal IPs before running this.
↪script again.

VERBOSE: App Service detected and external IPs specified, creating mappings.
[App Service API] Created port mappings on 10.30.23.123 to 192.168.102.17
[App Service API] Ports: 443
[App Service API] DNS: 10.30.23.123 api.appservice.local.azurestack.external
[App Service Apps] Created port mappings on 10.30.23.124 to 192.168.102.15
[App Service Apps] Ports: 80 443 21 990
[App Service Apps] DNS: 10.30.23.124 *.appservice.local.azurestack.external
```

Azure Stack Resources

The following resources need to be created and configured inside Azure Stack for successful provisioning:

- Resource Group(s)
- Virtual Network(s)
- Storage Account(s)
- Network Security Group(s)
 - Inbound ports open from Morpheus Appliance: 22, 5985, 3389
 - Outbound ports open to Morpheus Appliance: 80, 443

Note: Proper Network and Network Security Group configuration is required for Morpheus agent install, communication, and remote console access. Other configurations, such as docker instances, will need the appropriate ports opened as well.

Required Credentials & Permissions

Credentials to integrate Morpheus with Azure Stack are located in both the public Azure Portal and the Private Azure Stack Portal. The Azure Active Directory Application used must be an owner of the Azure Stack subscription.

Azure Portal:

- Azure Active Directory Application Credentials
- Directory ID
- Management URL
- Identity Resource URL
- Application ID
- Key Value

Azure Stack Portal:

- Azure Stack Subscription ID
- Active Directory App from Azure portal added as owner of the Azure Stack Subscription in Azure Stack.

Adding an Azure Stack Cloud

Configure

1. In the Morpheus UI, navigate to **Infrastructure** > **Clouds** and Select + *CREATE CLOUD*
2. Select *AZURE STACK (PRIVATE)* from the Clouds list and select *NEXT*
3. In the Configure section, enter:

Cloud Configuration

NAME Name of the Cloud in Morpheus

CODE Unique code used for api/cli, automation and policies.

LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT If Visibility is set to Private, select the Tenant the Cloud resources will assigned to.

ENABLED When disabled, automatic Cloud sync is paused and the Cloud will not be selectable for provisioning.

AUTOMATICALLY POWER ON VMS When enabled, Morpheus will maintain the expected power state of managed VMs. Morpheus will power on any managed VMs in the Cloud that have been shut down for unknown reasons (not powered off by Morpheus) to ensure availability of services.

Note: When “AUTOMATICALLY POWER ON VMS” is enabled, the power state of managed VMs should be maintained in Morpheus. This setting is not applicable to discovered/unmanaged resources.

Details

IDENTITY URL <https://login.microsoftonline.com>

MANAGEMENT URL* Azure AD Azure Stack Administrator app or Microsoft Azure Stack Administrator app url. Example: <https://adminmanagement.local.azurestack.external/>

IDENTITY RESOURCE URL Azure AD Azure Stack Administrator App ID URI Example: <https://adminmanagement.xxxxxxx.onmicrosoft.com/4a80e607-4259-4ac6-83e2-2fabeaf2eh83>

BASE DOMAIN This should match the base domain in your Management url. Example: local.azurestack.external

SUBSCRIPTION ID Subscription ID from Azure Stack portal (this is different from the Subscription ID in you Azure portal used when configuring Azure Stack)

TENANT ID This is the Directory ID from the Azure AD directory

CLIENT ID Application ID of Azure AD app with Azure Stack permissions granted, and has been added as an owner of the Azure Stack subscription (in the Azure Stack portal).

CLIENT SECRET Key Value of Application ID used above

Note: Once all credentials are entered and validated, the Location and Resource Group fields will populate.

Location Select an Azure Stack region for the cloud to scope to. This typically will be “local”.

Resource Group Select All or a single Resource Group to scope the cloud to. Selecting a single Resource Group will only sync resources in that Resource Group and disable Resource Group selection during provisioning. All will sync all resources and allow specifying the Resource Group during provisioning.

Inventory Existing Instances If enabled, existing Virtual Machines will be inventoried and appear as unmanaged Virtual Machines in Morpheus.

4. The Azure Stack cloud is ready to be added to a group and saved. Additional configuration options available:

Note: All fields and options can be edited after the Cloud is created.

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM's if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT INSTALL MODE

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud Init / Unattend (when available): (DEFAULT) Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

5. Once all options are configured, select NEXT to add the cloud to a Group.

GROUP A Group must be specified or created for the new Cloud to be added to. Clouds can be added to additional Groups or removed from Groups after being created.

USE EXISTING Add the new Cloud to an existing Group in Morpheus .

CREATE NEW Creates a new Group in Morpheus and adds the Cloud to the Group.

6. Confirm all settings are correct and select **COMPLETE**. The Azure Stack Cloud will be added, and Morpheus will perform the initial cloud sync of:
 - Virtual Machines (if Inventory Existing Instances is enabled)
 - Networks
 - Virtual Images/Blueprints
 - Network Security Groups
 - Storage Accounts
 - Marketplace Catalog
 - Availability Sets

Tip: Synced Networks can be configured or deactivated from the Networks section in this Clouds detail page, or in the *Infrastructure > Networks* section.

Cloud Foundry

Configuration

Adding PCF Cloud From *Infrastructure > Clouds*

1. Navigate to *Infrastructure > Clouds*
2. Select + *ADD*
3. Select **CLOUD FOUNDRY** from the Clouds list
4. Select *NEXT*
5. Populate the following:

Cloud Configuration

NAME Name of the Cloud in Morpheus

CODE Unique code used for api/cli, automation and policies.

LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT If Visibility is set to Private, select the Tenant the Cloud resources will assigned to.

ENABLED When disabled, automatic Cloud sync is paused and the Cloud will not be selectable for provisioning.

AUTOMATICALLY POWER ON VMS When enabled, Morpheus will maintain the expected power state of managed VMs. Morpheus will power on any managed VMs in the Cloud that have been shut down for unknown reasons (not powered off by Morpheus) to ensure availability of services.

Note: When “AUTOMATICALLY POWER ON VMS” is enabled, the power state of managed VMs should be maintained in Morpheus. This setting is not applicable to discovered/unmanaged resources.

Details

API URL Cloud Foundry API Url

CLIENT ID Typically cf

CLIENT SECRET Typically blank

USERNAME Enter Username. If using an API Key, enter `apikey` for username, and the API Key as the password.

PASSWORD Enter Password. If using an API Key, the API Key as the password.

ORGANIZATION Select Organization. Dropdown populates upon successful authorization.

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM's if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherrwell

AGENT INSTALL MODE

- **SSH / WINRM:** Morpheus will use SSH or WINRM for Agent install.
- **Cloud Init / Unattend (when available): (DEFAULT)** Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

6. Select *NEXT*
7. Select an existing or create a new Group to add the Cloud to. The Cloud can be added to additional Groups in a Groups *CLOUDS* tab.
8. Select *NEXT*
9. Review and then Select *COMPLETE*

Adding PCF Cloud From *Infrastructure > Groups*

1. Navigate to *Infrastructure > Groups*
2. Select a Group
3. Select the *CLOUDS* tab
4. Scroll down to *CLOUD FOUNDRY* and select + *ADD*
5. Populate the following:

Name Name of the Cloud in Morpheus

Location Description field for adding notes on the cloud, such as location.

Visibility For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT Select a Tenant if Visibility is set to Private to assign to Cloud to that Tenant. Multiple Tenants can be added by editing the cloud after creation.

API URL Cloud Foundry API Url. Example `https://api.cf.morpheusdata.com`

CLIENT ID Typically `cf`

CLIENT SECRET Typically blank

USERNAME Enter Username. If using an API Key, enter `apikey` for username, and the API Key as the password.

PASSWORD Enter Password. If using an API Key, the API Key as the password.

ORGANIZATION Select Organization. Dropdown populates upon successful authorization.

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM's if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT INSTALL MODE

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud Init / Unattend (when available): (DEFAULT) Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

6. Select *NEXT*

7. Review and then Select *COMPLETE*

Adding Spaces

Cloud Foundry Spaces are referred to as Resource Pools in Morpheus. You can add a new Space by:

1. Navigating to the Cloud and selecting the Resources tab.
2. Then, click :guilabel: '+ Add Resource'.
3. Give the Resource a Name
4. Expand the Managers, Developers, and Auditors section to add specific Cloud Foundry users to the roles. When adding a user to these sections, use their Cloud Foundry email addresses.

Provisioning

Morpheus automatically seeds MySQL, Redis and RabbitMQ PCF Instance Types, as well as a generic Cloud Foundry Instance Type that will create a shell app used in conjunction with deployments. PCF Marketplace items can also be added to the Provisioning Library in the Cloud detail view Marketplace tab. The Marketplace item will be added to the selected Instance Type and available when selecting the Cloud Foundry Cloud during Instance or App Template creation.

Deployments

The Cloud Foundry App Instance Type is used in conjunction with deployments. Users do not have to pick deployment when creating a Cloud Foundry App Instance Type, but then Instance will only be a shell of a Cloud Foundry Application.

A deployment in Morpheus can either point to a git hub repository or contain the actual manifest.yml and associated artifacts required for a Cloud Foundry deployment. During the deployment, Morpheus will gather up the files required. Therefore, if the deployment points to a git hub repository, Morpheus will fetch the files from git hub. Once the files are obtained, Morpheus will deploy the artifacts in a similar fashion to the Cloud Foundry cli. This includes parsing the manifest to obtain the parameters to create or update the Cloud Foundry application. Morpheus will ignore certain fields such as memory and disk size because they are dictated by the selected plan. Other fields are utilized such as routes. After parsing the manifest.yml file (including overwriting certain fields), Morpheus is ready to update or create the App in Cloud Foundry.

After the App is configured, the artifacts references in the Morpheus deployment are uploaded to Cloud Foundry for the App. Note that when paths are referenced in the manifest.yml file, the paths continue to be relative to the manifest. So, a jar file under build/libs would need to be found under the build/libs directory.

If Cloud Foundry services are specified in the manifest, they must already exist within Cloud Foundry. Morpheus App templates can be utilized to wire up Cloud Foundry services created by Morpheus. In this case, Morpheus will add all of the included service names defined in the App template to the manifest.yml services section. Therefore, multiple services can be used and wired up by Morpheus.”

Example

To better understand how Morpheus parses the manifest.yml file, lets take a closer look at the Cloud Foundry ‘spring-music’ project. The project can be found here (<https://github.com/cloudfoundry-samples/spring-music>).

The project contains the required manifest.yml file as well as the source code and build.gradle file to define how the project is to be built. After downloading the project to your local machine, build the project to generate the jar.

Now, let’s take a look at the manifest.yml file:

```
---
applications:
- name: spring-music
  memory: 1G
  random-route: true
  path: build/libs/spring-music.jar
```

Using the Cloud Foundry docs (<https://docs.cloudfoundry.org/devguide/deploy-apps/manifest.html>), we can gain a better understanding of how this file is utilized by Cloud Foundry.

- The -name parameter defines the name that will be given to the application in Cloud Foundry. Morpheus will overwrite this value with the name given to the Instance being created in Morpheus.

- The `-memory` parameter (as well as the `disk_quota` parameter if specified) will be overwritten by Morpheus based on the plan specified for the Instance.
- The `-path` parameter defines, where relative to the `manifest.yml` file, your Cloud Foundry application can be found.
- The `-random-route` parameter, as well as all other parameters described in the Cloud Foundry documentation will simply be passed through to Cloud Foundry.

Adding Marketplace Items

1. Navigate to **Infrastructure** > **Clouds** and select your Cloud Foundry Cloud
2. Select the **MARKETPLACE** tab
3. Select **+ ADD MARKETPLACE ITEM**
4. Select the Morpheus Instance Type to add the Marketplace Item to.
5. Enter version
6. Search for and select Marketplace Item
7. Select **SAVE CHANGES**

A Node Type and layout will be created in the Library section and the layout will be automatically added to the Instance Type selected when adding the Marketplace Item.

Provisioning Instances

Seeded and Marketplace Items

Morpheus automatically seeds MySQL, Redis and RabbitMQ PCF Instance Types, and PCF Marketplace items can also be easily added to the Provisioning Library in the Cloud detail view Marketplace tab. The Marketplace item will be added to the selected Instance Type and available when selecting the Cloud Foundry Cloud during Instance or App Template creation.

1. Navigate to **Provisioning** > **Instances** and select an Instance Type with a Cloud Foundry layout (MySQL, Redis and RabbitMQ plus Marketplace additions)
2. Select **NEXT**
3. Select a Group and PCF Cloud
4. Add an Instance Name
5. Optionally select an Environment Tag and/or add a custom Tag
6. Select **NEXT**
7. Select Version and Instance Configuration for a Cloud Foundry layout, ex: *Cloud Foundry MySQL*
8. Select a Plan and available options for the Plan, or use the custom Plan
9. Select a Space to add the Instance to
10. Optionally configure advanced options
11. Select **NEXT**
12. Optionally configure Automation options
13. Select **NEXT**

14. Select *COMPLETE*

Note: Compute, Memory, and CPU stats will be pulled, and a Cloud Foundry monitoring health check will be automatically configured for the instance.

Cloud Foundry App Instance Type

Important: Add Deployments in Provisioning > Code > Deployments to be used when provisioning a Cloud Foundry App Instance Type.

Note: Minimal options are outlined below.

1. Navigate to Provisioning > Instances and select the *Cloud Foundry App* Instance Type
2. Select *NEXT*
3. Select a Group and PCF Cloud
4. Add an Instance Name
5. Optionally select an Environment Tag and/or add a custom Tag
6. Select *NEXT*
7. Select a Plan and available options for the Plan, or use the custom Plan
8. Select a Space to add the Instance to
9. Select *NEXT*
10. In the Deployments section, select a Deployment and Version to be deployed. These can be git repos or files added in Provisioning > Code > Deployments

Important: If services are specified in a git repo manifest, Morpheus assumes they already exist in the PCF cloud with matching names.

11. Select *NEXT*
12. Select *COMPLETE*

This will quickly create the Cloud Foundry Application, and then the deployment will follow which may take longer depending on the app configuration. The location will be updated with the route once it is configured.

Note: Compute, Memory, and CPU stats will be pulled, and a Cloud Foundry monitoring health check will be automatically configured for the instance.

Digital Ocean

Add a Digital Ocean Cloud

DigitalOcean Cloud Integration Detail fields:

Cloud Configuration

NAME Name of the Cloud in Morpheus

CODE Unique code used for api/cli, automation and policies.

LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT If Visibility is set to Private, select the Tenant the Cloud resources will assigned to.

ENABLED When disabled, automatic Cloud sync is paused and the Cloud will not be selectable for provisioning.

AUTOMATICALLY POWER ON VMS When enabled, Morpheus will maintain the expected power state of managed VMs. Morpheus will power on any managed VMs in the Cloud that have been shut down for unknown reasons (not powered off by Morpheus) to ensure availability of services.

Note: When “AUTOMATICALLY POWER ON VMS” is enabled, the power state of managed VMs should be maintained in Morpheus. This setting is not applicable to discovered/unmanaged resources.

Details

Username DigitalOcean Username

API Key Personal access tokens/Key from the DigitalOcean *API > Tokens/Keys* section.

Data Center Select DigitalOcean DataCenter Region

The Cloud can now be added to a Group or configured with additional Advanced options.

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM's if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT INSTALL MODE

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud Init / Unattend (when available): (DEFAULT) Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

ESXi

The ESXi Cloud type enables managing and provisioning to ESXi hosts, even without the ESXi API enabled.

Important: The VMware ESXi integration is for adding a single ESXi / vSphere Hypervisor host. If you have vCenter please use the VMWare vCenter cloud type for full vSphere integration features.

To get started with VMware ESXi, simply add a VMware ESXi Cloud in either the **Infrastructure > Clouds** or **Infrastructure > Groups** section.

1. Select + Create Cloud Button
2. Select ESXi from the Add Cloud modal
3. Select NEXT
4. Provide the following information.

Cloud Configuration

NAME Name of the Cloud in Morpheus

CODE Unique code used for api/cli, automation and policies.

LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT If Visibility is set to Private, select the Tenant the Cloud resources will assigned to.

ENABLED When disabled, automatic Cloud sync is paused and the Cloud will not be selectable for provisioning.

AUTOMATICALLY POWER ON VMS When enabled, Morpheus will maintain the expected power state of managed VMs. Morpheus will power on any managed VMs in the Cloud that have been shut down for unknown reasons (not powered off by Morpheus) to ensure availability of services.

Note: When “AUTOMATICALLY POWER ON VMS” is enabled, the power state of managed VMs should be maintained in Morpheus. This setting is not applicable to discovered/unmanaged resources.

Details

- ESXi Host name or IP address
- Username (This is normally root)
- Password

Note: If you receive the message “Error! Invalid cloud config” Please ensure you have ssh enabled on the ESXi host.

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM's if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT INSTALL MODE

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud Init / Unattend (when available): (DEFAULT) Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

Important: ESXi provisioning require a vmx file, which is not included in an OVF/OVA export from vCenter. A proper vmx file must be included when adding a vmdk/ovf/ova image to Virtual Images in Morpheus for successful provisioning.

Google Cloud Platform (GCP)

Integration Features

- Provisioning Virtual Machines
- Network tagging
- Private and Local Images
- Google VM Snapshots
- Brownfield Inventory
- Costing
- Right-sizing
- Shared Network Support

Requirements for Integration with Morpheus

To integrate Morpheus with Google Cloud Platform, you will need the following. APIs are enabled in “APIs & Services” and must be enabled for all projects or the selected project (depending on your GCP Cloud integration settings). The next section contains more detailed steps for enabling API in the GCP web console.

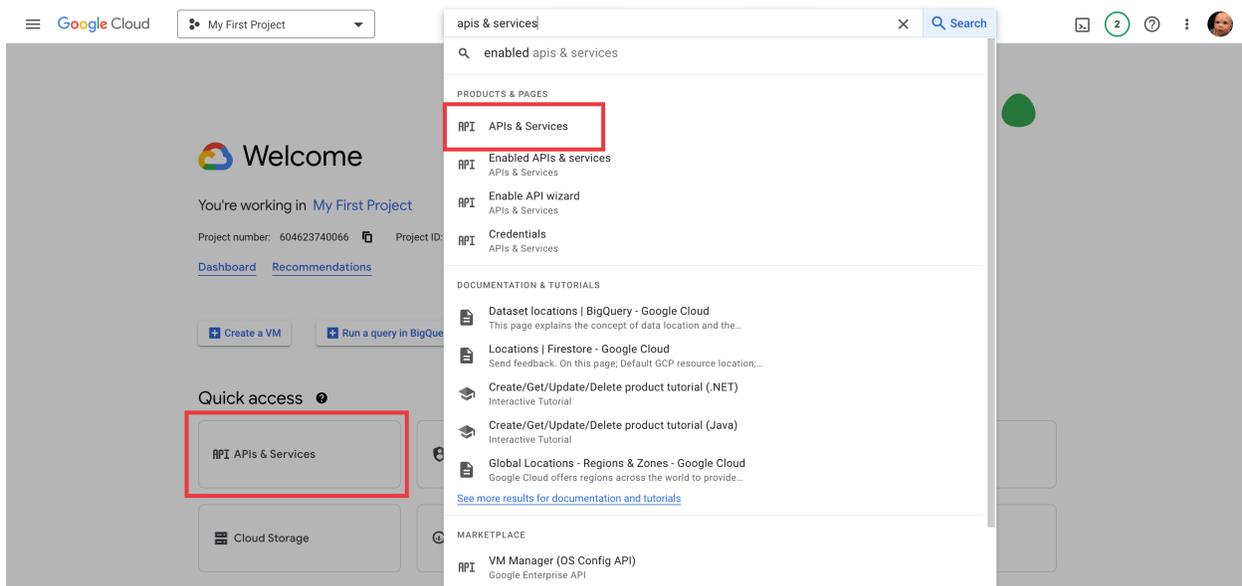
- The Compute Engine API enabled
- The Cloud Resource Manager API enabled
- The Cloud Billing API
- The Identity and Access Management (IAM) API enabled
- The BigQuery API enabled
- The BigQuery Data Transfer API enabled
- The Kubernetes Engine API enabled (required to provision GKE clusters)
- Credentials for an IAM service account with Owner or Compute Admin role permissions
- The private key and client email for the service account

This integration guide goes through the process of configuring your account and obtaining the information necessary to integrate with Morpheus. Continue to the next section for a detailed look at enabling the APIs mentioned above.

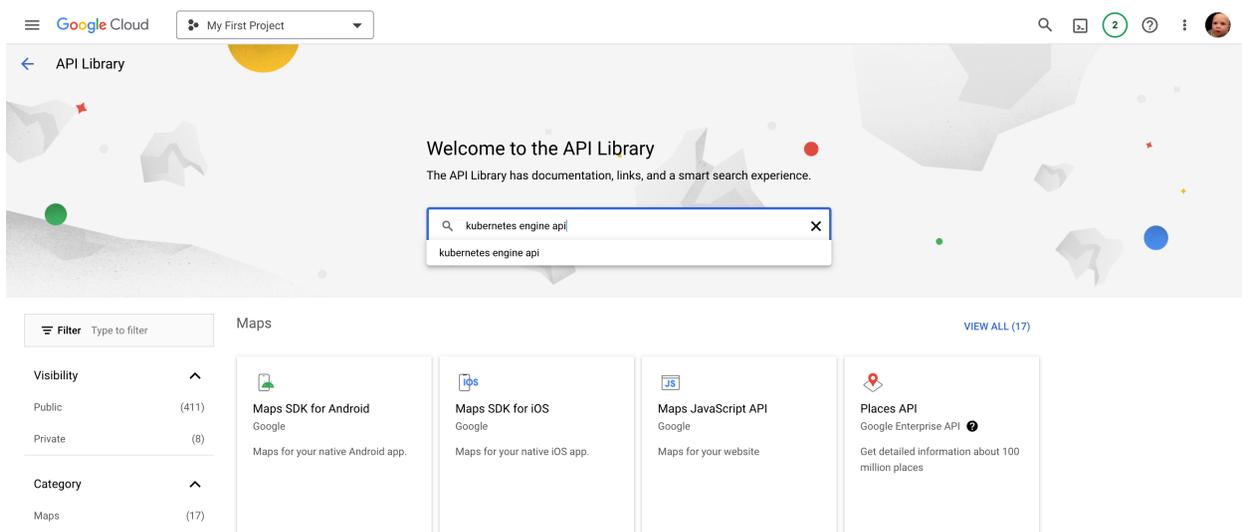
Enabling the Required APIs

In order to take full advantage of the Morpheus integration with Google Cloud, a number of APIs must be enabled within the GCP web console. It's recommended that you enable all of the APIs listed in the preceding section regardless of the Morpheus feature set you intend to use. This will ensure you do not run into problems down the road stemming from a lack of access which may take time to diagnose.

Log into the Google Cloud web console and navigate to the APIs and Services page. You may find this in the Quick access area of the welcome page or you can search for it as shown in the screenshot below.

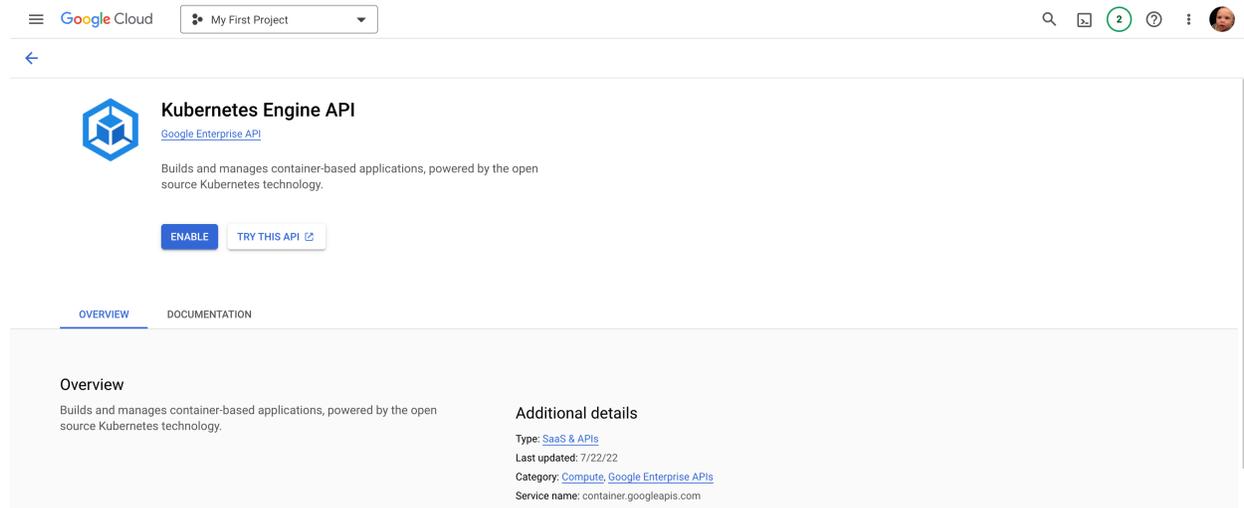


From the APIs and Services page, a list of enabled APIs and some details about your usage are shown. To enable new APIs, click + *ENABLE APIS & SERVICES* near the top of the window. Now on the API library page, search for the API you wish to enable. Here I've searched for the Kubernetes Engine API.



From the search results, click on the API you wish to enable to view its detail page. Click *ENABLE*. Once successfully

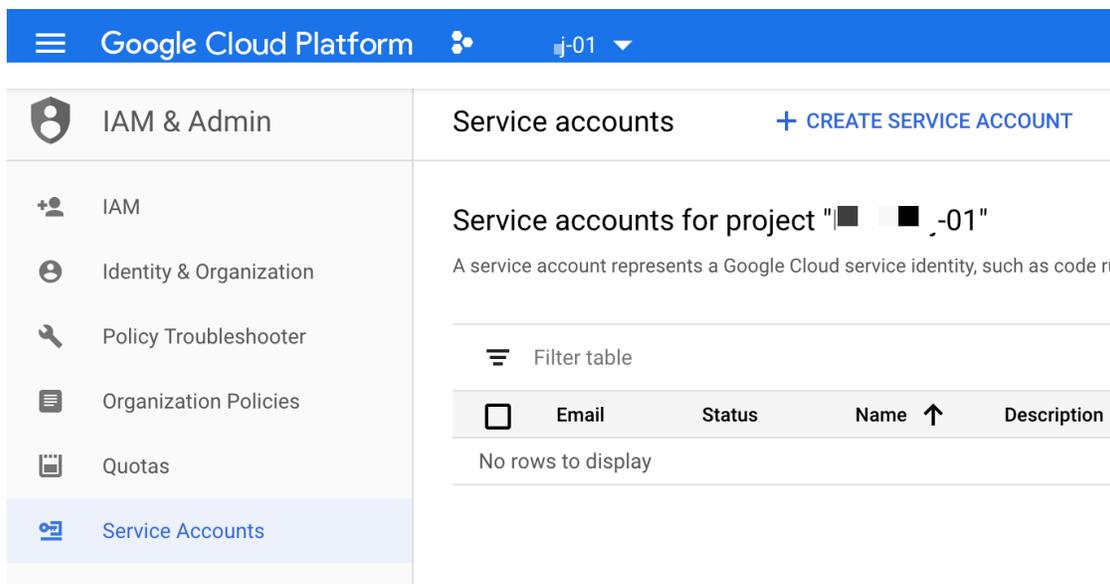
enabled, the button will change to a *MANAGE* button. It may take a few moments for the API to be fully enabled. You may also be prompted to enable the Cloud Billing API or create an association with a Billing Account when enabling APIs. Go ahead and do so if prompted.



Repeat this process until all required APIs (listed in the previous section) are enabled.

Creating a Service Account

1. From anywhere in the GCP web console, search for “service accounts” in the global search bar at the top of the window
2. Click on the Service Accounts page (within the IAM & Admin stack)
3. A list of existing service accounts within the selected Project is shown (if any)
4. To create a new one, click + *CREATE SERVICE ACCOUNT*



3. Enter at least a name for your new service and click *CREATE AND CONTINUE*

The screenshot shows the 'Create service account' wizard in the Google Cloud Platform console. The left sidebar contains the 'IAM & Admin' menu with 'Service Accounts' selected. The main content area shows the 'Service account details' step, which is the first of two steps. The 'Service account name' field contains 'svcaccount'. The 'Service account ID' field also contains 'svcaccount'. The 'Service account description' field is empty. At the bottom, there are 'CREATE' and 'CANCEL' buttons.

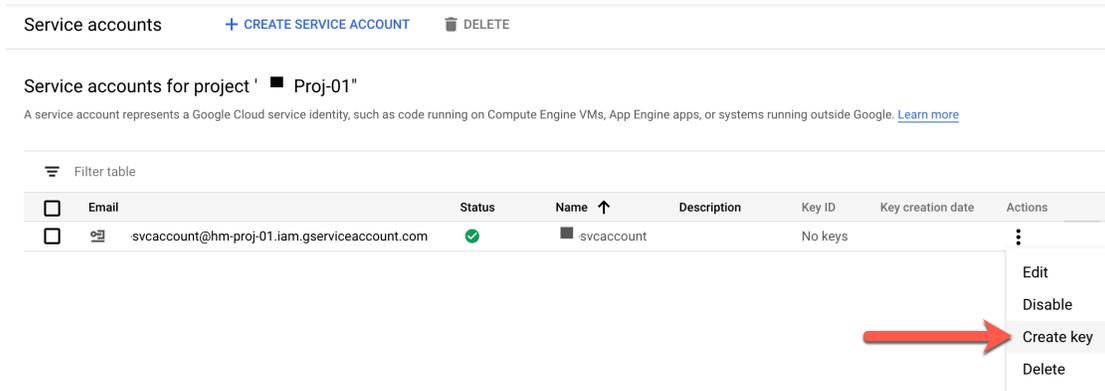
4. After creating the service account, you'll be prompted to set a role for the account. In order to fully integrate with Morpheus, you must use an account in the Owner role or the Compute Admin role
5. Click CONTINUE

The screenshot shows the 'Create service account' wizard in the Google Cloud Platform console, now at the 'Service account permissions (optional)' step. The 'Service account details' step is marked as completed with a checkmark. The 'Service account permissions (optional)' section shows that the service account is granted access to the project 'Proj-01'. The 'Role' is set to 'Owner', which provides 'Full access to all resources.' There is an 'Add condition' link and a trash icon. At the bottom, there are 'CONTINUE' and 'CANCEL' buttons.

6. Following creation of the service account, you'll be taken back to the list of existing service accounts

Generating Keys and Integrating with Morpheus

1. From the list of service accounts, click the ellipsis button (...) to the right of a selected account
2. Click “Manage Keys”



3. On the Keys page, click “Add Key” and then “Create New Key”
4. Select JSON format and click CREATE
5. A JSON-formatted document will be downloaded, this document contains the Project ID, private key, and client email values needed to complete the integration process in the next step

Add a GCP Cloud

Note: The JSON-formatted document downloaded when creating a key for your service account contains all of the required values for completing the integration. Consult the above section on generating keys if needed.

1. Navigate to Infrastructure > Clouds
2. Select + *CREATE CLOUD*, select Google Cloud, and then click *NEXT*.
3. Enter the following into the Create Cloud modal:

Cloud Configuration

NAME Name of the Cloud in Morpheus

CODE Unique code used for api/cli, automation and policies.

LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT If Visibility is set to Private, select the Tenant the Cloud resources will assigned to.

ENABLED When disabled, automatic Cloud sync is paused and the Cloud will not be selectable for provisioning.

AUTOMATICALLY POWER ON VMS When enabled, Morpheus will maintain the expected power state of managed VMs. Morpheus will power on any managed VMs in the Cloud that have been shut down for unknown reasons (not powered off by Morpheus) to ensure availability of services.

Note: When “AUTOMATICALLY POWER ON VMS” is enabled, the power state of managed VMs should be maintained in Morpheus. This setting is not applicable to discovered/unmanaged resources.

Details

PRIVATE KEY The service account private key. Paste in the entire value between (but not including) the quotation marks in your downloaded JSON document, formatted like the following example: -----BEGIN PRIVATE KEY-----*(your_key)*-----END PRIVATE KEY-----

CLIENT EMAIL The service account client email, ex: *morpheus@morpheus.iam.gserviceaccount.com*

PROJECT ID Projects will auto-populate upon successful entry of the private key and client email. You can opt to scope the GCP integration to a single Project or select “All” to instead select the Project from the Resource Pool dropdown at provision time

REGION Regions will auto-populate upon successful entry of the private key and client email. Select the appropriate region for this Cloud, if applicable. You can also opt to scope the GCP integration to all regions to allow users to select from any region at provision time

INVENTORY EXISTING INSTANCES If checked, existing GCP resources will be inventoried and appear as unmanaged virtual machines in Morpheus.

If advanced options are not needed, click *NEXT* to advance to the Group selection page. Otherwise, continue on with this guide and review advanced or provisioning options.

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM's if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If

your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT INSTALL MODE

- **SSH / WINRM:** Morpheus will use SSH or WINRM for Agent install.
- **Cloud Init / Unattend (when available): (DEFAULT)** Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

4. After reviewing all options, click *NEXT* to advance to the Group selection page. Following Group selection, click *COMPLETE* to finish the integration process. If you've opted to inventory existing Instances, they will be viewable in Morpheus shortly. At this point, you are ready to provision new resources in Google Cloud Platform as needed!

Important: If you experience difficulties adding a GCP Cloud, review the above guide and ensure you've met all requirements for completing the integration. For example, if the Compute Engine API is not enabled, Morpheus will not accept credentials entered on the Create Cloud modal. If you repeatedly run into problems completing the integration process, review the above guide in its entirety and double check that each step is completed and your account meets all configuration requirements.

Create a GCP Project

On initial integration, Morpheus will sync Projects and allow you to scope the integration to a specific Project or to scope the integration to all Projects. As time goes on, additional Projects are continually synced and can be managed from within the Resources tab on the Cloud detail page (Infrastructure > Clouds > Selected GCP Cloud). Within the Resources tab, users can edit some Project settings as well as delete Projects if needed.

To create a new GCP Project:

1. Click **+ ADD RESOURCE POOL**
2. Enter a name value for the new Project
3. Mark the “DEFAULT” box if you’d prefer newly provisioned Instances default to the new Project
4. Enter a Project ID and ensure it meets the listed validation requirements
5. Set a Parent value if the new Project should exist underneath a parent organization
6. Finally, select a billing account
7. Click **SAVE CHANGES**

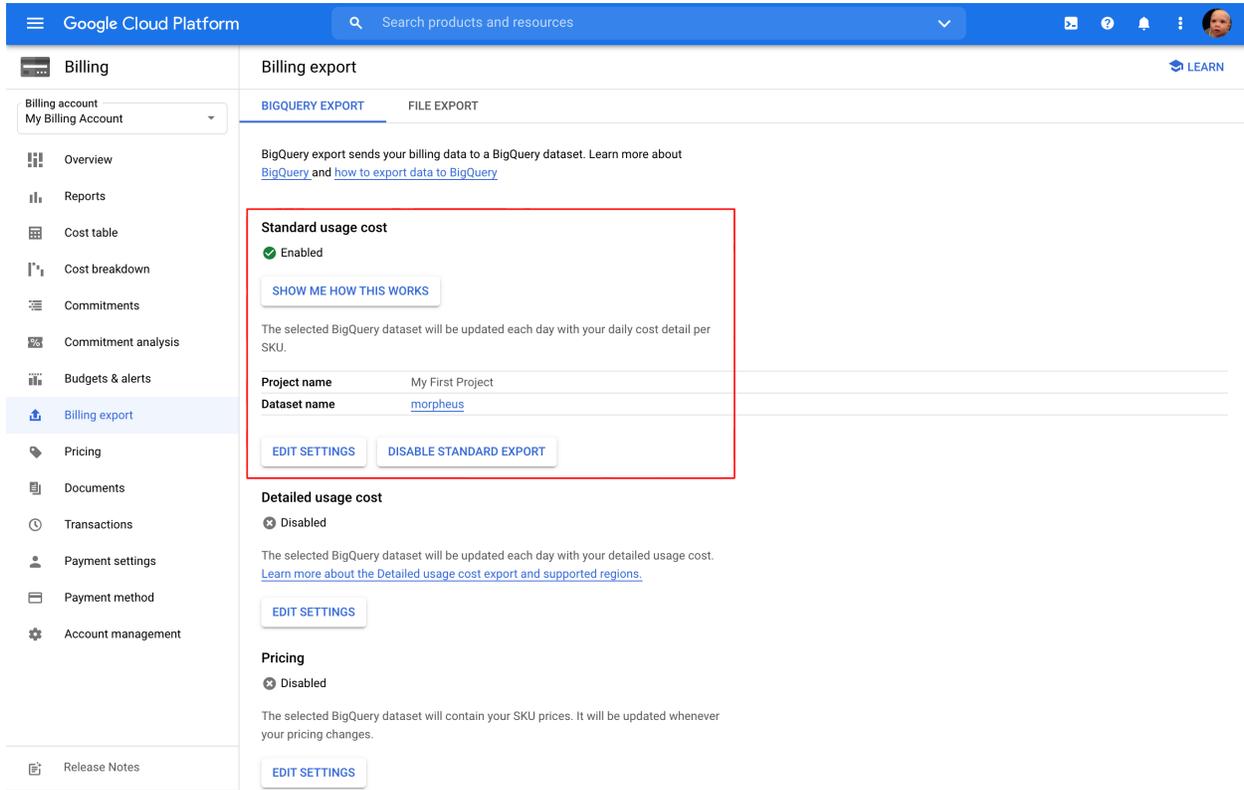
After a few minutes, the new Project will be ready on the GCP side and Morpheus will be ready to provision new resources into it.

Enabling Live Costing for GCP

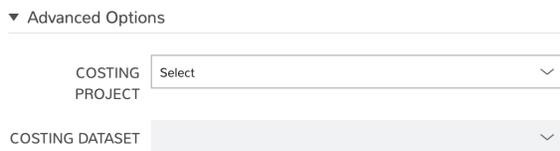
GCP costing is done at the Billing Account level. Each Billing Account can be linked to one or more GCP Projects. All projects which are linked to the Billing Account will have their costing data available to Morpheus but if the GCP Cloud has been scoped to only one Project, Morpheus will ingest costing data only for that Project. Users can view the Billing Account linked to a particular project by clicking on the hamburger menu (main menu button in the far upper-left of the console window) and selecting billing. A pop-up window will give users the option to navigate to the Billing Account which is linked to the currently-selected Project.

The screenshot displays the Google Cloud Platform Billing console. The top navigation bar includes the Google Cloud Platform logo, a search bar for products and resources, and user profile information. The left sidebar contains a menu with options like Billing, Reports, Cost table, and Billing export (highlighted with a red box). The main content area shows the 'Billing account overview' for 'My Billing Account'. It includes a 'Current month' summary for September 1-3, 2021, with 'Month-to-date total cost' at \$0.00 and 'End-of-month total cost (forecasted)' as 'Not enough historical data to project cost'. Below this is a 'Cost trend' section for the period from September 1, 2020, to September 30, 2021, showing an 'Average monthly total cost' of \$0.00. On the right, there are sections for 'Billing account' details, 'Organization' (No organization), and 'Billing health checks' (0 critical, 1 warning, 1 success).

Within the Billing Account, Standard Usage Cost must be enabled for Morpheus to access costing data. From the page for the appropriate Billing Account, click on Billing Export and then click “Edit Settings” under the “Standard usage cost heading”. Specify a project and create a dataset or specify an existing one. In doing this, you’re specifying a location for the dataset *which will be for the entire billing account and not just for the Project the dataset resides in.*



With configuration in the GCP console completed, we can now enable cost onboarding from the Morpheus side. Add or edit an existing GCP Cloud (Infrastructure > Clouds). Within the Advanced Options section, note the COSTING PROJECT and COSTING DATASET fields. When selecting a Project, associated datasets (if any) will automatically be loaded into the dropdown in the next field for selection. Additionally, the COSTING field should be set to “Sync Costing” rather than “Off”. Recall from the previous paragraph that this is merely pointing to the Project that houses the appropriate dataset. If your GCP Cloud in Morpheus is configured for all Projects, all costing data will be consumed for the Projects linked to the associated Billing Account (assuming those Projects have billing enabled). If the GCP Cloud in Morpheus is scoped to just one Project, only billing data for that Project will be onboarded. For this reason, the selected Costing Project can be (but is not necessarily) the Project to which the Morpheus Cloud is scoped.



Windows Images

Morpheus can add custom metadata that will be injected into the unattend conf by GCP during provisioning. This is required for customizations including setting the Windows Administrator password during provisioning. GCP Windows Images must be syspreped using the `GCESysprep` command prior to image creation, and must have `platform/os` set on the Virtual Image record in Morpheus after image sync for successful customization and Agent Installation.

GCP Windows Requirements

- GCP Windows Images must be syspreped using the `GCESysprep` command prior to Image creation in GCP. Refer to Google's "creating-windows-os-image" doc.
- Once the Image is synced into Morpheus, the Platform (Windows, Windows 2016 etc) must be set on the Morpheus Virtual Image record, otherwise linux is assumed and the metadata will not be generated correctly.
- The Global Windows "Administrator" password must be set in Morpheus under `/admin/provisioning/settings > Windows Settings > Administrator Password`, or Administrator and password defined on the Morpheus Virtual Image record.
- Be aware the unattend configuration during startup after sysprep delays causes a reboot and a prolonged finalization process during provisioning, and console/rdp may not be available during this time as windows is configuring.

Note: Some Google provided Windows Images have slow startups that cause the Morpheus Agent service to not start within the default 30 second service startup timeframe, including after initial reboot after sysprep/unattend configuration. This can be adjusted by running `New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\" -Name "ServicesPipeTimeout" -PropertyType DWORD -Value 180000` in powershell on the Windows Image.

Important: Failure to use a GCP Windows Image that has not been syspreped using `GCESysprep` will cause Agent Installation, Automation, and Console issues as Morpheus will not be able to set user credentials and authenticate.

Huawei Cloud

Features

- Virtual machine provisioning
- Backups
- Brownfield VM management and migration
- Hypervisor remote console
- Cloud sync
- Lifecycle management and resizing
- Network security group creation
- Network security group management
- Router and network creation
- Load balancer services

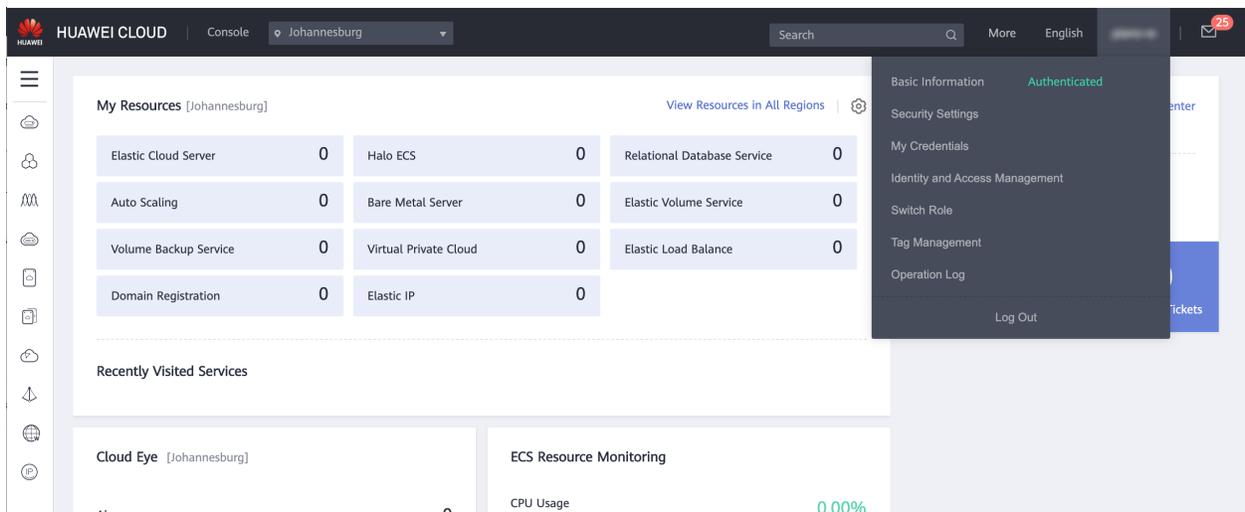
- Docker host management and configuration
- Floating IP assignment
- Huawei OBS buckets (create, manage, delete, and discovery)
- Huawei SFS (create, manage, and delete)

Integrate Huawei Cloud with Morpheus

To integrate Huawei Cloud with Morpheus, we'll gather the following pieces of information:

- Account Name
- Identity (IAM) API URL
- Project
- Username
- Password

Begin by logging into your [Huawei Cloud console](#). If you're not currently logged in, you will be prompted to do so. Once on the console page, hover over your username in the upper-right corner of the application window and select "My Credentials".



From the credentials page, we can gather the Account Name and the Project Name, record them for later when we provide the integration information to Morpheus.

The screenshot shows the 'API Credentials' page in the Huawei Cloud console. The 'Account Name' field is highlighted in red. Below, a table lists various projects with their IDs, names, and regions. The 'af-south-1' project is highlighted in red.

Project ID	Project Name	Region
	cn-north-1	CN North-Beijing1
	cn-north-4	CN North-Beijing4
	cn-east-2	CN East-Shanghai2
	cn-south-1	CN South-Guangzhou
	cn-southwest-2	CN Southwest-Guiyang1
	ap-southeast-1	AP-Hong Kong
	ap-southeast-2	AP-Bangkok
	ap-southeast-3	AP-Singapore
	af-south-1	AF-Johannesburg

To gather the API endpoint URL, take a look at the complete list of [endpoints](#). If a specific endpoint exists for your region, use it. In any other case use the endpoint for all regions. It will be formatted like this: <https://iam.myhuaweicloud.com/v3>.

Identity and Access Management (IAM)

Region Name	Region	Endpoint	Protocol Type
ALL	ALL	iam.myhuaweicloud.com	HTTPS
CN East-Shanghai1	cn-east-3	iam.cn-east-3.myhuaweicloud.com	HTTPS
CN East-Shanghai2	cn-east-2	iam.cn-east-2.myhuaweicloud.com	HTTPS
CN North-Beijing1	cn-north-1	iam.cn-north-1.myhuaweicloud.com	HTTPS
CN North-Beijing2	cn-north-2	iam.cn-north-2.myhuaweicloud.com	HTTPS
CN North-Beijing4	cn-north-4	iam.cn-north-4.myhuaweicloud.com	HTTPS
CN South-Guangzhou	cn-south-1	iam.cn-south-1.myhuaweicloud.com	HTTPS
CN South-Shenzhen	cn-south-2	iam.cn-south-2.myhuaweicloud.com	HTTPS
CN Southwest-Guiyan g1	cn-southwest-2	iam.cn-southwest-2.myhuaweicloud.com	HTTPS

With this information gathered, and presuming you know the credentials for the service account you wish to use, we can move back into Morpheus-UI.

Important: Integrating Morpheus with Huawei Cloud requires a service account that has programmatic access.

Navigate to Infrastructure > Clouds and click + *ADD*. Scroll to Huawei Cloud and click *NEXT*. The information we've gathered will be plugged into the CREATE CLOUD modal. The DOMAIN ID field will accept the Account Name field we gathered. Your completed CREATE CLOUD modal will look similar to the one pictured below:

Details

IDENTITY API URL	<input type="text" value="https://iam.myhuaweicloud.com/v3"/>
DOMAIN ID	<input type="text" value="my_account_name"/> <small>This pertains to the OpenStack V3 API and should be ignored when using V2. This is the Domain ID (Not to be confused with Domain Name). The Domain ID can be found via the CLI by typing <code>openstack domain list</code>.</small>
PROJECT	<input type="text" value="af-south-1"/>
REGION	<input type="text"/>
USERNAME	<input type="text" value="my_username"/>
PASSWORD	<input type="password" value="....."/>
IMAGE FORMAT	<input type="text" value="QCOW2"/>

After clicking *NEXT*, add this new Cloud to a Group or create a new Group. On finalizing the wizard, Huawei Cloud will be integrated into Morpheus and ready for provisioning. If you opted to inventory existing workloads, those will be onboarded shortly.

Add/Edit Huawei Cloud Modal Fields

Cloud Configuration

NAME Name of the Cloud in Morpheus

CODE Unique code used for api/cli, automation and policies.

LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT If Visibility is set to Private, select the Tenant the Cloud resources will assigned to.

ENABLED When disabled, automatic Cloud sync is paused and the Cloud will not be selectable for provisioning.

AUTOMATICALLY POWER ON VMS When enabled, Morpheus will maintain the expected power state of managed VMs. Morpheus will power on any managed VMs in the Cloud that have been shut down for unknown reasons (not powered off by Morpheus) to ensure availability of services.

Note: When “AUTOMATICALLY POWER ON VMS” is enabled, the power state of managed VMs should be maintained in Morpheus. This setting is not applicable to discovered/unmanaged resources.

Details

IDENTITY API URL The v3 identity endpoint. See the integration steps above for more detail

DOMAIN ID The DOMAIN ID field takes the “Account Name” as shown on the [Basic Information page](#) of the account. See the integration steps above for more detail

PROJECT The target project name. See the integration steps above for more detail

USERNAME The service account username. See the integration steps above for more detail

PASSWORD The integration service account password. See the integration steps above for more detail

IMAGE FORMAT Select QCOW2, RAW or VMDK image type

Image Store Set an OBS bucket as a permanent store location for Morpheus virtual images. Users are limited to uploading images of 2GB or less in size if an OBS bucket is not specified here

Inventory Existing Instances Select for Morpheus to discover and sync existing VMs

Enable Hypervisor Console Hypervisor console support for openstack currently only supports novnc. Be sure the novnc proxy is configured properly in your openstack environment.

Tip: When using the RAW image format, you can bypass the image conversion service within the cloud leading to quicker performance. Other image formats are converted to RAW format and back when performing various actions. Using the RAW format from the start will bypass these conversion steps.

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM’s if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT INSTALL MODE

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud Init / Unattend (when available): (DEFAULT) Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

Huawei Scalable File Service (SFS)

The Morpheus integration with Huawei Cloud includes the capability to work with Huawei Scalable File Service (SFS). SFS is shared file storage hosted on Huawei Cloud. By integrating Morpheus with Huawei Cloud you can discover, create, manage, and delete SFS servers, as well as view and work with the file shares and files contained therein.

SFS Server Discovery and Management

On integrating Huawei Cloud with Morpheus, SFS servers and file shares are discovered automatically after a short time. The server(s) can be viewed in Infrastructure > Storage > Servers. By viewing the server detail page and clicking *EDIT*, the storage server can be scoped as needed. Administrators can choose to scope to other Huawei Cloud integrations (if more than one relevant integration currently exists), select from synced availability zones, and scope the storage server to specific Tenants if desired.

EDIT STORAGE SERVER ✕

NAME

DESCRIPTION

ENABLED

TYPE

CLOUD

AVAILABILITY ZONE

Permissions

VISIBILITY

TENANTS

SAVE CHANGES

Additionally, Huawei SFS servers can be created from the storage server list page (Infrastructure > Storage > Servers) directly in Morpheus. Click + *ADD* to begin and set the storage server type value to “Huawei SFS”. Just like with existing synced SFS servers, those created from Morpheus can be scoped as needed.

ADD STORAGE SERVER ✕

NAME

DESCRIPTION

ENABLED

TYPE

CLOUD

AVAILABILITY ZONE

Permissions

VISIBILITY

TENANTS

SAVE CHANGES

SFS File Share Discovery and Management

Discovered file shares will appear among other file shares synced with Morpheus in Infrastructure > Storage > File Shares. Depending on the number of cloud integrations in your Morpheus appliance and the number of cloud integrations available to your user account, this list may be quite large. Using the search bar on this page, we can narrow down the list to file shares displayed to those whose names match the search terms.

The screenshot shows the Morpheus web interface. At the top, there is a navigation bar with the Morpheus logo, a search bar, and user information (Alex Harker). Below this is a secondary navigation bar with tabs for Operations, Provisioning, Infrastructure (selected), Backups, Logs, Monitoring, Tools, and Administration. Under Infrastructure, there are sub-tabs for Groups, Clouds, Clusters, Hosts, Network, Load Balancers, Storage (selected), Keys & Certs, and Boot. The main content area is titled 'STORAGE' and has sub-tabs for Buckets, File Shares (selected), Volumes, Data Stores, and Servers. Below this is a 'FILE SHARES' section with a search bar containing 'qa' and a '+ ADD' button. A table lists several file shares with columns for NAME, PROVIDER TYPE, SHARE PATH, SOURCE, BACKUP, and DEPLOYMENTS. The table contains six rows, with the fifth row highlighted in blue and labeled 'qa-huawei-sfs'.

NAME	PROVIDER TYPE	SHARE PATH	SOURCE	BACKUP	DEPLOYMENTS
...	No	No
...	No	No
...	No	No
qa-huawei-sfs	Huawei SFS Share	...	Labs Huawei SFS	No	No
...	No	No

We can drill into individual file shares by clicking on their hyperlinked name in the list of all integrated file shares. From the file share detail page, a list of files will appear on the files tab. Begin the process of adding a new file by clicking + *ADD*. The Access tab on the file shares detail page allows users to view and manage ACL rules.

Note: A “Failed to load files from storage provider” is present when the Morpheus appliance doesn’t have access to the file share.

New Huawei SFS file shares can be created directly in Morpheus. From the file shares list page, get started by clicking + *ADD*. Select the type “Huawei SFS Share”. Set the storage service field to a pre-existing Huawei SFS server. A friendly name for the file share in Morpheus and selecting from synced availability zones are required fields.

NEW FILE SHARE ✕

NAME

STORAGE SERVICE

AVAILABILITY ZONE

SHARE SIZE GB

ACTIVE

DEFAULT BACKUP TARGET

DEFAULT DEPLOYMENT ARCHIVE TARGET

DEFAULT VIRTUAL IMAGE STORE

Retention

RETENTION POLICY

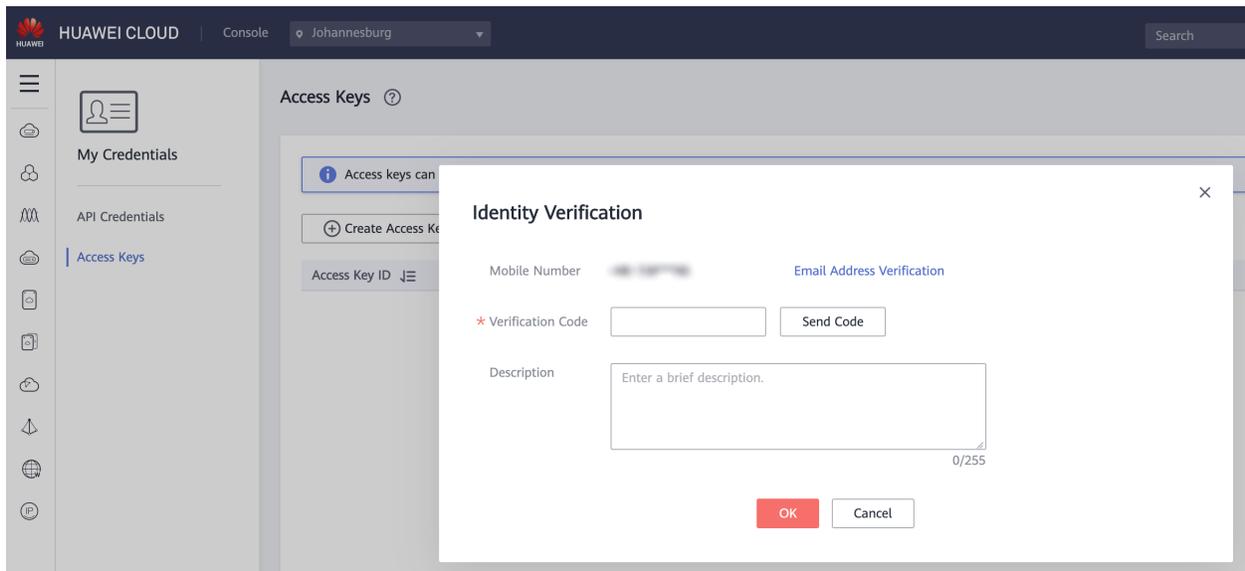
SAVE CHANGES

Huawei Object Storage Service (OBS)

The Morpheus integration with Huawei Cloud also supports Object Storage Service (OBS). Morpheus will automatically onboard existing OBS servers and buckets shortly after completing the cloud integration. Before you can add a new OBS server from Morpheus, you must know or generate a key and secret value from the Huawei console and must provide a Huawei OBS API endpoint.

Generate a Key and Secret

From the Huawei web console, log into the account used to integrate Huawei Cloud with Morpheus. Hover over your account name in the upper-right corner of the application window and click “My Credentials”. Select “Access Keys” from the left-hand sidebar. To create a new key, click + *Create Access Key*. Complete the two-factor authentication steps in the box that appears.



Once the key is generated, download or record the key and store it in a safe location. The key will not be viewable or available for download again after this point.

Create OBS Server in Morpheus

With the key and secret value in hand from the previous section, navigate to Infrastructure > Storage > Servers. Click + *ADD*. On changing the server type to Huawei OBS, you will see the fields for the access key and the secret key. OBS API endpoints can be found in [Huawei endpoint documentation](#). Include those three values in the Create Server modal along with a friendly name for use in Morpheus UI. Just like with SFS objects, we can choose to scope the server to all or specific Tenants at this time.

ADD STORAGE SERVER ✕

NAME

DESCRIPTION

ENABLED

TYPE

ACCESS KEY

SECRET KEY

ENDPOINT

Permissions

VISIBILITY

TENANTS

SAVE CHANGES

Create Huawei OBS Bucket

With an OBS server onboarded or created in Morpheus, you're able to create and manage Huawei OBS buckets as needed. To create a new bucket, navigate to Infrastructure > Storage > Buckets. Click + *ADD* and select "Huawei OBS Bucket". The following fields are required when creating a Huawei OBS bucket:

- **NAME:** A friendly name for use in Morpheus UI
- **STORAGE SERVICE:** Choose the OBS server to associate the new bucket with
- **BUCKET NAME:** The name of the bucket in Huawei Cloud, this must be unique
- **STORAGE CLASS:** If needed, view the [discussion of storage classes](#) in Huawei support documentation
- **BUCKET ACL:** Public Read, Public Read/Write, or Private
- **BUCKET POLICY:** Public Read, Public Read/Write, or Private
- **STORAGE QUOTA:** Set to 0 for no quota

Once finished, click *SAVE CHANGES*

NEW BUCKET ✕

NAME

STORAGE SERVICE

BUCKET NAME

STORAGE CLASS

BUCKET ACL

BUCKET POLICY

STORAGE QUOTA GB

CREATE BUCKET

ACTIVE

DEFAULT BACKUP TARGET

DEFAULT DEPLOYMENT ARCHIVE TARGET

DEFAULT VIRTUAL IMAGE STORE

Retention

RETENTION POLICY

SAVE CHANGES

Network and Router Creation

Once a Huawei Cloud is integrated into Morpheus, new network creation options become available. When adding a new network (Infrastructure > Networks > Networks Tab), a new type labeled “Huawei Private Network” is available when clicking *+ADD*. When the user creates this network construct in Morpheus, a layer two subnet is created but it’s not connected to a Virtual Private Cloud (VPC). This is by design as an Internet-routable network is not always desired. Continue on with this section after creating the network to also create a VPC (router).

Create a network

1. Navigate to Infrastructure > Networks
2. Click on the Networks tab
3. Click *+ADD*
4. Select Huawei Private Network
5. Complete the modal based on requirements for the new network
6. Click *SAVE CHANGES*

Create a router

1. Navigate to Infrastructure > Networks
2. Click on the Routers tab
3. Click *+ADD*
4. Select Huawei Router
5. Complete the modal based on requirements for the new router
6. Click *SAVE CHANGES*

When creating a router, it’s helpful to note that the External Network is the floating IP network that has been assigned to the Huawei project. This network will grant your Instances their routes out to the Internet. The Internal Subnet can be a layer two subnet that you may have created in the previous step. In addition, multiple subnets can be added to the router (VPC) and the IP address on the subnet would be the router’s internal IP address.

Hyper-V

Hyper-V is the virtualized server computing environment introduced by Microsoft. Hyper-V is consumed by Morpheus as a private cloud offering and is a common hypervisor technology in data centers. Morpheus provides an avenue to aggregate Hyper-V resources together to allow efficient and seamless deployment of applications as a virtual machine (VM) or Docker host in the world of Hyper-V.

Features

- Virtual Machine Provisioning
- Discovery of Existing Instances
- Containers
- Backups / Snapshots
- Resources Groups
- Migrations
- Auto Scaling
- Load Balancing
- Remote Console
- Periodic Synchronization
- Veeam Integration
- Lifecycle Management and Resize
- Unique Kerberos Authentication

Getting Started

To get started, a few prerequisites must first be met. The Hyper-V host must be installed with its firewall enabled and it can either be joined to a domain or standalone. The Hyper-V host must also have the external network of Hyper-V configured and it can share this network with the management operating system. A user account that is part of the local administrators group on the Hyper-V host is also required. This document covers Hyper-V 2008 and Hyper-V 2012.

Understanding WinRM

Morpheus uses WinRM to communicate to the Hyper-V host for deployment of the Morpheus Agent. The Morpheus Agent allows for the host dashboard to be populated with information in the form of graphs that cover CPU, Network, Storage, and memory consumption. Furthermore, this Agent provides logging and monitoring capabilities.

If Windows Remote Management (WinRM) is not installed and configured, WinRM scripts do not run and the WinRM command line tool cannot perform data operations or allow for the Morpheus Agent to be installed. WinRM uses HTTP port 5985 or HTTPS port 5986 for communications.

To better understand all of the default settings of WinRM please refer to the [following page](#) in Microsoft documentation.

Native Authentication

To configure WinRM with default settings (WINRM_NATIVE)

Type the following command at a command prompt:

```
$ winrm quickconfig
```

If you are not running under the local computer Administrator account, you must either select Run as Administrator from the Start menu or use the *Runas* command at a command prompt.

When the tool displays Make these changes [y/n]?, type y.

If configuration is successful, the following output is displayed:

```
$ WinRM has been updated for remote management.
$ WinRM service type changed to delayed auto start.
$ WinRM service started.
$ Created a WinRM listener on HTTP://* to accept WS-Man requests to any IP on this_
↪machine.
```

Keep the default settings for client and server components of WinRM, or customize them. By default Kerberos is enabled and if domain authentication is not being used we want to disable that. Issue the below commands to setup basic authentication:

```
$ winrm set winrm/config/service/Auth '@{Basic="true"}'
$ winrm set winrm/config/service '@{AllowUnencrypted="true"}'
$ winrm set winrm/config/service/Auth '@{Kerberos="false"}'
```

Domain Authentication

To configure WinRM with Domain Authentication (WINRM_INTERNAL)

Type the following command at a command prompt

```
$ winrm quickconfig
```

If you are not running under the local computer Administrator account, you must either select Run as Administrator from the Start menu or use the runas command at a command prompt.

When the tool displays Make these changes [y/n]?, type y.

If configuration is successful, the following output is displayed:

```
$ WinRM has been updated for remote management.
$ WinRM service type changed to delayed auto start.
$ WinRM service started.
$ Created a WinRM listener on HTTP://* to accept WS-Man requests to any IP on this_
↪machine.
```

Keep the default settings for client and server components of WinRM, or customize them. Issue the below commands to setup domain authentication:

```
$ winrm set winrm/config/service/Auth @{Basic="true"}
$ winrm set winrm/config/service @{AllowUnencrypted="false"}
$ winrm set winrm/config/service/Auth @{Kerberos="true"}
```

Kerberos authentication will also need to be configured on the Morpheus appliance to support Windows domain accounts to access the remote host with WINRM_INTERNAL connection type.

On the Morpheus appliance the krb5-user package must be installed.

For Ubuntu the command is as follows:

```
$ sudo apt-get install krb5-user
```

For Centos the command is as follows:

```
$ sudo yum install krb5-workstation pam_krb5 -y
```

Create a file in /etc called krb5.conf and replace the domain name with the name of the domain to be used. In this case we used Morpheus .com as the domain.

```
[libdefaults]
    default_realm = |morpheus| .COM
    dns_lookup_kdc = true
    verify_ap_req_nofail = false
    default_tgs_enctypes = rc4-hmac
    default_tkt_enctypes = rc4-hmac
[realms]
    |morpheus| .COM = {
        kdc = win-ad.|morpheus| .COM:88
        admin_server = win-ad.|morpheus| .COM:749
    }
[domain_realm]
    .|morpheus| .COM = |morpheus| .COM
    |morpheus| .COM = |morpheus| .COM
[login]
    krb4_convert = true
    krb4_get_tickets = false
```

After creation of the krb5.conf a keytab file is also required. See below for instructions on how to create a keytab file. <http://www.itadmintools.com/2011/07/creating-kerberos-keytab-files.html>

Adding Hyper-V as a Private Cloud

The Hyper-V host is prepared for Morpheus to communicate with it via WinRM so the Hyper-V private cloud is ready to be configured. Create a group and then create a Morpheus cloud for Hyper-V. Populated the information as show in Figure 1: specific for the environment being configured.

CREATE CLOUD ✕

CLOUD > **CONFIGURE** > GROUP > REVIEW

NAME

LOCATION

DNS DOMAIN

VISIBILITY ▼

SCALE PRIORITY

Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

Details

HYPER-V HOST

WORKING PATH

VM PATH

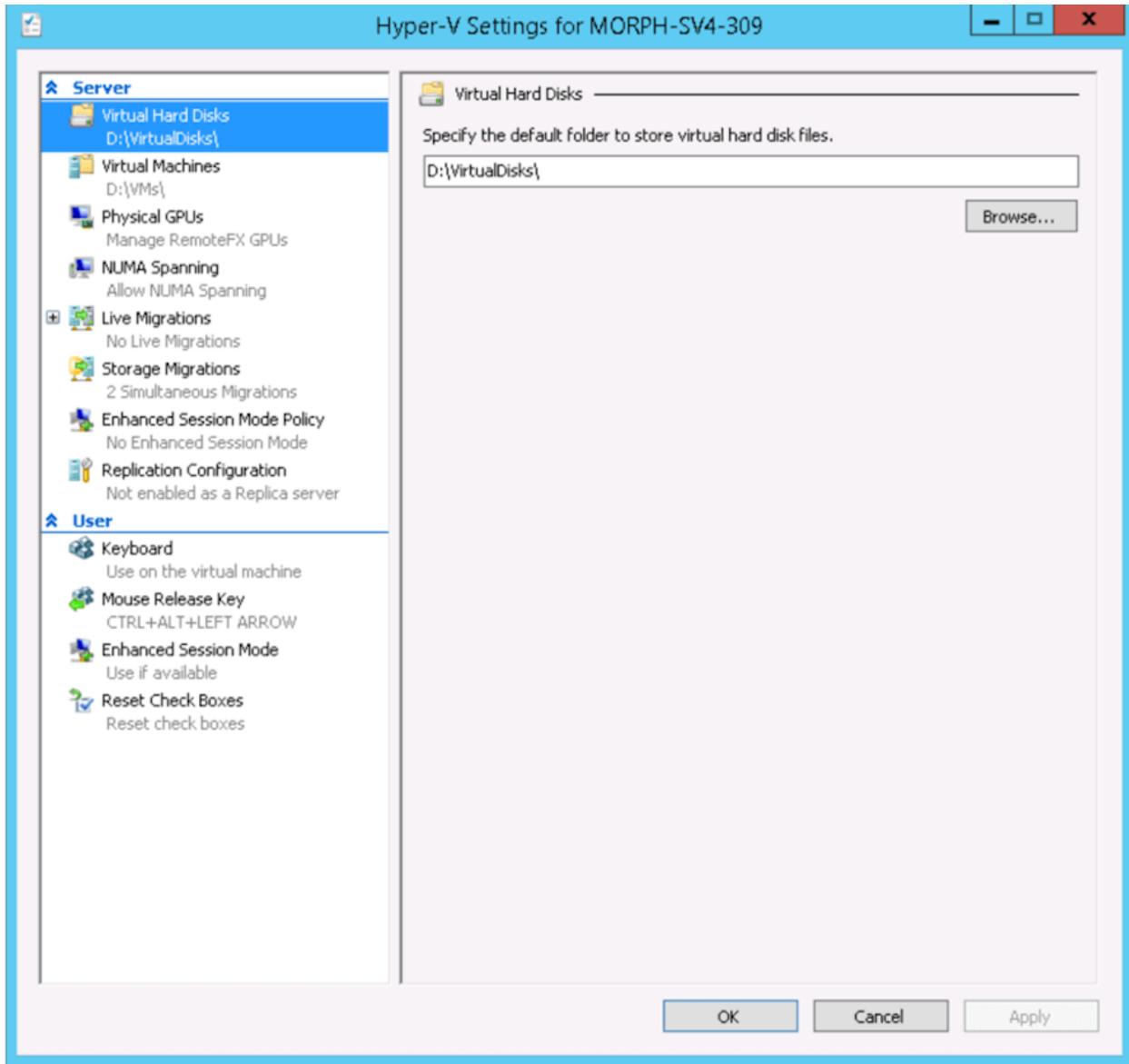
DISK PATH

USERNAME

PASSWORD

PREV **NEXT**

Note: The working path, vm path, and disk path should be created on the Hyper-V host by the Hyper-V administrator. If these paths are not created they will need to be setup and the Hyper-V settings will need to adjusted to reference them.



Service Plans

A default set of Service Plans are created in Morpheus for the VMware provisioning engine. These Service Plans can be considered akin to AWS Flavors or Openstack Flavors. They provide a means to set predefined tiers on memory, storage, cores, and cpu. Price tables can also be applied to these so estimated cost per virtual machine can be tracked as well as pricing for customers. By default, these options are fixed sizes but can be configured for dynamic sizing. A service plan can be configured to allow a custom user entry for memory, storage, or cpu. To configure this, simply edit an existing Service Plan tied to Hyper-V or create a new one. These are managed from the Admin > Plans & Pricing section.

TYPE NAME	CLOUDS	MEMORY	STORAGE	PRICE SETS		
 Memory: 3.5GB, 1 vCPU	Google Labs	3.5GB	0MB	0	/	🗑️
 Memory: 7.2GB, 2 vCPU	Google Labs	7.2GB	0MB	0	/	🗑️
 Memory: 15GB, 4 vCPU	Google Labs	15.0GB	0MB	0	/	🗑️
 Memory: 30GB, 8 vCPU	Google Labs	30.0GB	0MB	0	/	🗑️
 Memory: 60GB, 16 vCPU	Google Labs	60.0GB	0MB	0	/	🗑️
 Memory: 120GB, 32 vCPU	Google Labs	120.0GB	0MB	0	/	🗑️
 Memory: 13GB, 2 vCPU	Google Labs	13.0GB	0MB	0	/	🗑️
 Memory: 26GB, 4 vCPU	Google Labs	26.0GB	0MB	0	/	🗑️
 Memory: 52GB, 8 vCPU	Google Labs	52.0GB	0MB	0	/	🗑️
 Memory: 104GB, 16 vCPU	Google Labs	104.0GB	0MB	0	/	🗑️
 Memory: 208GB, 32 vCPU	Google Labs	208.0GB	0MB	0	/	🗑️
 Memory: 1.7GB, 2 vCPU	Google Labs	1.7GB	0MB	0	/	🗑️
 Memory: 3.5GB, 4 vCPU	Google Labs	3.5GB	0MB	0	/	🗑️
 Memory: 7.2GB, 8 vCPU	Google Labs	7.2GB	0MB	0	/	🗑️
 Memory: 14GB, 16 vCPU	Google Labs	14.0GB	0MB	0	/	🗑️
 Memory: 28GB, 32 vCPU	Google Labs	28.0GB	0MB	0	/	🗑️
 Memory: 512MB Storage: 10GB	San Mateo Hyper-V	512.0MB	10.0GB	0	/	🗑️
 Memory: 1GB Storage: 10GB	San Mateo Hyper-V	1.0GB	10.0GB	0	/	🗑️
 Memory: 2GB Storage: 20GB	San Mateo Hyper-V	2.0GB	20.0GB	0	/	🗑️
 Memory: 4GB Storage: 40GB	San Mateo Hyper-V	4.0GB	40.0GB	0	/	🗑️
 Memory: 8GB Storage: 80GB	San Mateo Hyper-V	8.0GB	80.0GB	0	/	🗑️
 Memory: 16GB Storage: 160GB	San Mateo Hyper-V	16.0GB	160.0GB	0	/	🗑️
 Memory: 24GB Storage: 240GB	San Mateo Hyper-V	24.0GB	240.0GB	0	/	🗑️
 Memory: 32GB Storage: 320GB	San Mateo Hyper-V	32.0GB	320.0GB	0	/	🗑️
 HV_4_240	San Mateo Hyper-V	3.9GB	234.3GB	0	/	🗑️

Docker

So far this document has covered how to add the Hyper-V cloud integration and has enabled users the ability to provision virtual machine-based instances via the Add Instance catalog under the Provisioning menu. Another great feature provided by Morpheus out of the box is the ability to use Docker containers and even support multiple containers per Docker host. To do this a Docker Host must first be provisioned into Hyper-V (multiple are needed when dealing with horizontal scaling scenarios).

To provision a Docker Host simply navigate to the Clusters tab of the Cloud detail page or Infrastructure > Clusters section. From there click + *ADD CLUSTER* to add a Hyper-V Docker Host. A cluster is created when adding Docker hosts, even when only one host is needed.

Morpheus views a Docker host just like any other hypervisor with the caveat being that it is used for running containerized images instead of virtualized ones. Once a Docker Host is successfully provisioned a green checkmark will appear to the right of the host marking it as available for use. In the event of a failure click into the relevant host that failed and an error explaining the failure will be displayed in red at the top.

Some common error scenarios include network connectivity. For a Docker Host to function properly, it must be able to resolve the Morpheus appliance url which can be configured in Admin | Settings. If it is unable to resolve and negotiate with the appliance than the agent installation will fail and provisioning instructions will not be able to be issued to the host.

IBM Cloud

IBM Cloud offers a wide set of cloud computing services including compute, storage, database, networking, and more. It is deployed in data centers across the world and its services can scale to meet the needs of small development teams as well as large-scale enterprises. Morpheus integrates with IBM Cloud to offer virtual machine provisioning, brownfield discovery, and lifecycle management.

Integrating IBM Cloud with Morpheus

Connecting an IBM Cloud account with Morpheus is a simple process requiring only an API key. You can create an API key from the IBM Cloud web console if you don't currently have access to one. Integrating with Morpheus requires a user API key. If needed, create one by clicking "Manage" in the upper menu bar of the IBM Cloud web console and clicking "Access (IAM)". Select "Users" from the left navigation menu and click on the desired user. Create a new API Key from within the API Keys section and store the API Key in a secure place to retrieve in the next step. It's recommended that you integrate Morpheus using an IBM Cloud admin account to ensure you won't run into any blockers while working in Morpheus. Using Morpheus role-based access you can limit which IBM Cloud constructs and capabilities are accessible to your users.

With the API key created, head back to Morpheus and continue with the following steps:

1. Navigate to Infrastructure > Clouds
2. Click + *ADD*
3. Complete the following required fields:
 - **NAME:** A friendly name for the cloud integration in Morpheus
 - **USERNAME:** Enter "apikey" (Do not enter your IBM Cloud username or anything other than "apikey")
 - **API Key:** The API key value
 - **DATACENTER:** Select the IBM Cloud data center
 - **OBJECT STORE:** If desired and if any are found, select an object store

1. Click *NEXT*
2. Select a Group for this Cloud or create a new one if desired
3. Complete the wizard to save the new integration

EDIT CLOUD
✕

NAME

CODE

LOCATION

VISIBILITY ▾

TENANT ▾

ENABLED

AUTOMATICALLY POWER ON VMS

Details

USERNAME

API KEY

DATACENTER ▾

OBJECT STORE ▾

Once the wizard has completed, the new IBM Cloud integration will appear alongside other Clouds currently available in Morpheus. If you've selected to automatically onboard existing resources, those will appear shortly.

IBM Cloud integrations also support a number of advanced options which were not discussed as part of the initial integration set up discussed above. For more information on the advanced options, consult the section below.

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM's if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT INSTALL MODE

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud Init / Unattend (when available): (DEFAULT) Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

KVM

Overview

Morpheus KVM is a powerful, cheaper alternative to virtualization compared with other hypervisor offerings. It is also very capable of setting up complex shared storage and multiple networks across many hosts. This guide goes over the process for onboarding brownfield KVM clusters and for provisioning new KVM clusters directly from Morpheus. When created or onboarded, KVM clusters are associated with the chosen Cloud and can then be selected as provisioning targets using existing Instance types and automation routines. In this example, baremetal KVM hosts are added to a Morpheus-type Cloud but similar combinations can be made with other Cloud types.

Requirements

At this time, Morpheus primarily supports CentOS 7 and Ubuntu-based KVM clusters. When creating KVM clusters directly in Morpheus these are the two base OS options. It's possible to onboard KVM clusters built on other Linux distributions, such as SUSE Linux but the user would need to ensure the correct packages were installed. The required packages are listed below.

- kvm
- libvirt
- virt-manager
- virt-install
- qemu-kvm-rhev
- genisoimage

Additionally, Morpheus will attempt to add a new network switch called 'morpheus' and storage pool when onboarding a brownfield KVM cluster.

When creating new clusters from Morpheus, users simply provide a basic Ubuntu or CentOS box. Morpheus takes care of installing the packages listed above as well as Morpheus Agent, if desired. The same can be said for adding a new host to an existing KVM cluster. Users need only provide access to an Ubuntu or CentOS box and Morpheus will install the required packages along with making sensible default configurations. Users can also add existing KVM hosts to a cluster. After providing SSH access into the host, if Morpheus detects that virsh is installed, it will treat it as a brownfield KVM host. Brownfield KVM hosts must have:

- libvirt and virsh installed
- A pool called morpheus-images defined as an image cache and ideally separate from the main datastore
- A pool called morpheus-cloud-init defined which stores small disk images for bootup (this pool can be small)

Note: Morpheus creates (or uses in the case of brownfield hosts) a morpheus-images pool which is separate from the main datastore. This is a host-local image cache which facilitates faster clone operations. The cache will automatically purge images once the allocation reaches 80% to avoid filling completely. Once it is 80% full, the oldest accessed volumes in the cache will be deleted first until the cache is under 50% full once again.

Creating the Cloud

Morpheus doesn't include a KVM-specific Cloud type. Instead, other Cloud types (either pre-existing or newly created) are used and KVM clusters are associated with the Cloud when they are onboarded or created by Morpheus. For example, a generic Morpheus-type Cloud could be created to associate with baremetal KVM clusters. Similarly, brownfield VMware KVM hosts could be onboarded into an existing VMware vCenter Cloud. Other combinations are possible as well. In the example in this section, a Morpheus Cloud will be created and KVM hosts will be associated with it to become Morpheus provisioning targets.

A Morpheus-type cloud is a generic Cloud construct that isn't designed to hook into a specific public or private cloud (such as VMware or Amazon AWS). Before onboarding an existing KVM host or creating one via Morpheus UI tools, create the Cloud:

1. Navigate to Infrastructure > Clouds
2. Click + *ADD*
3. Select the Morpheus Cloud type and click *NEXT*
4. On the Configure tab, provide:
 - **NAME:** A name for the new Cloud
 - **VISIBILITY:** Clouds with Public visibility will be accessible in other Tenants (if your appliance is configured for multitenancy)
 - **Automatically Power On VMs:** Mark this box for Morpheus to handle the power state of VMs
 - **Inventory Existing Instances:** If marked, Morpheus will automatically onboard VMs from any KVM hosts associated with this Cloud
5. On the Group tab, create a new Group or associate this Cloud with an existing Group. Click *NEXT*
6. After reviewing your configuration, click *COMPLETE*

CREATE CLOUD ×

CLOUD > CONFIGURE > GROUP > REVIEW

NAME

CODE

LOCATION

VISIBILITY

TENANT

ENABLED

AUTOMATICALLY POWER ON VMS

INVENTORY EXISTING INSTANCES

[▶ Advanced Options](#)

[▶ Provisioning Command](#)

Onboard an Existing KVM Cluster

With the Cloud created, we can add existing KVM hosts from the Cloud detail page (Infrastructure > Clouds > Selected Cloud). From the Hosts tab, open the ADD HYPERVISOR menu and select “Brownfield Libvirt KVM Host”.

On the first page of the Create Host modal, enter a name for the onboarded KVM host in Morpheus. Click *NEXT*. On the following page, enter the IP address for your host in addition to an SSH username and password for the host box. It’s recommended that you also copy the revealed SSH public key into your authorized_hosts file. Click *NEXT*.

- **SSH PORT:** The port for the SSH connection, typically 22
- **SSH USERNAME:** The username for an administrator user on the host box(es)
- **SSH PASSWORD:** The password for the account in the previous field
- **SSH KEY:** Select a stored SSH keypair from the dropdown menu
- **LVM ENABLED:** Mark if the destination box has LVM enabled
- **DATA DEVICE:** If “LVM ENABLED” is marked, this field is available. The indicated logical volume will be added to the logical volume group if it doesn’t already exist
- **SOFTWARE RAID?:** Mark to enable software RAID on the host box
- **NETWORK INTERFACE:** Select the interface to use for the Open vSwitch Bridge

6. Click *NEXT* to finish configuration, then complete the modal after final review

After adding the new cluster, you will see your newly created hosts listed on the Host Tab of the KVM Cloud detail page (Infrastructure > Clouds > Selected KVM Cloud).

Provisioning to KVM

With the Cloud and hosts available, users can now provision to the KVM host using custom Instance Types and automation routines built in the Morpheus library. To provision a new Instance, navigate to Provisioning > Instances and click + *ADD*. Select the Instance Type to provision, and click *NEXT*. Choose a Group that the KVM Cloud lives in and select the Cloud. Provide a name for the new Instance if a naming policy doesn’t already give it a name under current configurations. Click *NEXT* to advance to the Configuration tab. The fields here will differ based on the Instance Type and Layout used but in the example case, selections have been made for:

- **Layout:** Single KVM VM
- **Resource Pool:** The selected KVM cluster
- **Volumes:** Configure the needed volumes and the associated datastore for each
- **Networks:** The KVM network the VM(s) should belong to
- **Host:** The selected host the VM(s) should be provisioned onto

Complete the remaining steps to the provisioning wizard and the new KVM Instance will be created.

The screenshot shows the 'CREATE INSTANCE' wizard in the 'CONFIGURE' step. The configuration options are as follows:

- VERSION:** 20.04
- LAYOUT:** Single KVM VM
- PLAN:** 512MB Memory, 10GB Storage (Memory: 512 MB, Price: \$3.50 / Month)
- RESOURCE POOL:** QA KVM Cluster
- VOLUMES:** root (10 GB, morpheus)
- NETWORKS:** morpheus (DHCP)
- HOST:** qa-kvm-host1

At the bottom, there are 'PREVIOUS' and 'NEXT' buttons.

Adding VLANs to Morpheus KVM Hosts (CentOS)

Getting Started

This guide will go over how to configure VLANs on a Morpheus KVM Host. To get started, the first step is to go ahead and add the KVM host to morpheus and allow morpheus to configure it just like any other kvm host. When provisioning a manual kvm host be sure to enter the proper network interface name for the management network (not the trunk port). For example eno2 could be a management network while eno1 could be the trunk port network that the VLAN's are going to be on as in this example.

Setting up a VLAN Interface

Before a VLAN can be used by KVM, an interface definition must first be configured for said vlan. In CentOS this is done by defining a network script in `/etc/sysconfig/network-scripts`.

Note: It is highly recommended that `NM_CONTROLLED` is set to `NO` or NetworkManager is disabled entirely as it tends to get in the way.

If our trunk network is called eno1 we need to make a new script for each VLAN ID we would like to bridge onto. In our example we are going to look at **VLAN 211**. To do this we need to make a new script called `ifcfg-eno1.211` (note the VLAN Id is a suffix to the script name after a period as this is conventional and required).

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
NAME=eno1.211
DEVICE=eno1.211
ONBOOT=yes
NM_CONTROLLED=no
VLAN=yes DEVICETYPE=ovs
OVS_BRIDGE=br211
```

There are a few important things to note about this script. Firstly there is a flag called `VLAN=yes` that enables the kernel tagging of the VLAN. Secondly we have defined an `OVS_BRIDGE` name. Morpheus utilizes openvswitch for its networking which is a very powerful tool used even by Openstack's Neutron. It supports not just VLANs but VxLAN interfacing.

The `OVS_BRIDGE` name means we also need to define a bridge port script called `br211` by making a script called `ifcfg-br211`:

```
DEVICE=br211
ONBOOT=yes
DEVICETYPE=ovs
TYPE=OVSBridge
NM_CONTROLLED=no
BOOTPROTO=none
HOTPLUG=no
```

These configurations will enable persistence on these interfaces so that a reboot of the host will retain connectivity to the bridges. Next up, the interfaces need to be brought online. This can be done by restarting all network services but

if a typo is made networking could be stuck disabled and access over SSH could be broken. To do this by interface simply run:

```
ifup eno1.211
ifup br211
ovs-vsctl
add-br br211
```

Defining a LibVirt Network

Now that the bridge interface is defined properly for OVS, it must be defined in LibVirt so that Morpheus will detect the network and KVM can use it properly. By convention, these resource configurations are stored in `/var/morpheus/kvm/config`.

An XML definition must be created to properly define the network. In this case the network is named `public 185.3.48.0.xml`:

```
<network>
<name>public 185.3.48.0</name>
<forward mode="bridge"/>
<bridge name="br211"/>
<virtualport type="openvswitch"/>
</network>
```

This configuration defines the network name that will be synced into morpheus for selection as well as the type of interface being used (in this case a bridge to the `br211` interface over `openvswitch`).

Now that this xml specification is defined it must be registered with libvirt via the `virsh` commands:

```
virsh net-define "public 185.3.48.0.xml"
virsh net-autostart "public 185.3.48.0"
virsh net-start "public 185.3.48.0"
```

Once this is completed, simply refresh the cloud in morpheus and wait for the network to sync into the networks list. Once the network is synced make sure the appropriate settings are applied to it within Morpheus. This includes setting the CIDR, Gateway, Nameservers and if using IP Address Management, the IPAM Pool.

Canonical MAAS

MAAS from Canonical is an open-source server orchestration tool. It's designed to allow administrators to build a datacenter from on-premises, bare-metal servers where large networks of individual units can be discovered, deployed, and reconfigured.

Integrating MAAS and Morpheus

Integrating MAAS with Morpheus is a simple process requiring the MAAS API URL and API Key. We'll start by gathering what we need from the MAAS UI, then move back into Morpheus to store the required details.

We can gather the API key by clicking on the username in the upper-right corner of the MAAS UI window. From this preferences page, click on "API keys" as shown in the screenshot:

MAAS Machines Devices Controllers KVM Images DNS AZs Subnets Settings **Morpheus** Log out

My preferences

Details
API keys
SSH keys
SSL keys

Username *
Morpheus
Required: 150 characters or fewer. Letters, digits and @/./+/_ only.

Full name (optional)

Email address *
morpheus@morpheusdata.com

[Change password...](#)

Save

MAAS name: morph MAAS
MAAS version: 2.7.3 (8290-g.ebe2b9884)

[View release notes](#) · [View documentation](#) · [Legal information](#) · [Give feedback](#)

© 2020 Canonical Ltd. Ubuntu and Canonical are registered trademarks of Canonical Ltd.

CANONICAL

From the API keys page, select the displayed key and copy it. Alternatively, you can click the copy button in the UI to add the full key to your clipboard. Store this key in an accessible location for a later step.

MAAS Machines Devices Controllers KVM Images DNS AZs Subnets Settings Morpheus Log out

My preferences

Details
API keys
SSH keys
SSL keys

Generate MAAS API key

NAME	KEY	ACTIONS
MAAS consumer	...	

In addition to the API key, we need the MAAS API URL. As entered in Morpheus, the value will be like the following example: `http://<maas-hostname-or-ip>:5240`. We'll enter the API URL into Morpheus during the next step.

In Morpheus, navigate to the list of integrated Clouds and start a new MAAS Cloud integration:

1. Infrastructure > Clouds
2. Click + *ADD*
3. Select "MAAS"
4. Click *NEXT*

On the "CREATE CLOUD" modal, you must at least give a friendly name for the Cloud in Morpheus, MAAS API URL and API KEY. An example is shown below:

CREATE CLOUD

CLOUD > CONFIGURE > GROUP > REVIEW

NAME

CODE

LOCATION

Details

MAAS API URL

API KEY

RESOURCE POOL

▶ Advanced Options

▶ Provisioning Command

PREVIOUS NEXT

Tip: You'll know the credentials are entered correctly when your list of MAAS resource pools is synced in as you can see in the example screenshot.

Click *NEXT* and add this new Cloud to an existing Group or create a new Group for it. Once you advanced past the end of the wizard, the Cloud is added and Morpheus begins to inventory (if you've opted to inventory when adding the Cloud).

The screenshot displays the Morpheus MAAS dashboard. At the top, there is a navigation bar with the Morpheus logo, a search bar, and a 'Support' dropdown. Below this is a secondary navigation bar with tabs for Operations, Provisioning, Infrastructure (selected), Backups, Logs, Monitoring, Tools, and Administration. Under the Infrastructure tab, there are sub-tabs for Groups, Clouds, Clusters, Hosts, Network, Load Balancers, Storage, Keys & Certs, and Boot. The main content area shows the 'Morpheus MAAS' status with 'EDIT', 'REFRESH', and 'DELETE' buttons. It indicates the last sync was on 10/05/2020 at 11:11 AM with a duration of 8 seconds. Below this are five performance metrics: GBPO (COST THIS MONTH), COST HISTORY (a line graph showing 0.0 from May to Oct), MAX CPU (0%), MEMORY (0%), and STORAGE (0%). A summary section shows zero counts for Container Hosts, Hypervisors, Bare Metal, Virtual Machines, and Discovered. At the bottom, a 'Summary' tab is active, displaying configuration details for MAAS: Appliance URL (default), Domain, Storage Mode (standard), Local Firewall (off), Time Zone (default), Datacenter ID, Guidance (off), Security Server (off), Scale Priority (1), Agent Install Mode (cloudinit), and Network Mode (standard).

Mac Stadium

Overview

MacStadium is a provider of enterprise-class hosting solutions for Apple Mac infrastructure. It can be used to deploy a hosted private cloud for large-scale CI/CD or even a single Mac mini to test an iOS app. It allows virtualized Mac build machines

Features

- Virtual Machine Provisioning
- Backups / Snapshots
- Resource Groups
- Datastores and DRS Clusters
- Distributed Switches
- Datacenter / Cluster scoping
- Brownfield VM management and migration
- VMware to VMware migrations
- VMDK/OVF image conversion support

- Hypervisor Remote Console
- Periodic Synchronization
- Veeam Backup Integration
- Lifecycle Management and Resize

On top of all these features, Morpheus also adds additional features to VMware that do not exist out of the box to make it easier to manage in multitenant environments as well as hybrid cloud environments:

- Cloud-Init Support
- VHD to VMDK Image Conversion
- QCOW2 to VMDK Image Conversion
- Multitenancy resource allocation
- Virtual Image management (Blueprints)
- Auto-scaling and recovery

Getting Started

To get started with VMware, simply start by adding a Cloud in the **Infrastructure > Clouds** section.

To start adding a VMware cloud there will be some things you will need:

vCenter API Url Typically this is the url to the vCenter web client with a /sdk in the path

Username/Password A set of credentials with high level access to VMware (ensure the account has Datacenter level access)

Once these fields are entered, some selections will start pre-populating. A cloud integration is scoped to a specific data center, and can optionally be scoped down to a single cluster or even a single resource pool. If the drop downs do not populate, please verify the api url is resolvable, morpheus has access to vCenter on 443, and the provided credentials are correct and the user has sufficient permissions.

Another cool feature provided with the cloud integration is optional *Resource Pool* scoping. One can choose to allow the cloud to provision into All Resource Pools or a singular Resource Pool. When choosing *All*, these Resource Pools can be managed from a sub-account and visibility perspective via the Cloud Detail page (multi-tenancy).

The VMware cloud integration provides a few additional options including allowing users to make host selections or keeping that aspect hidden such that the best host is automatically chosen for the requested provision.

The *RPC Mode* feature can be configured to allow Morpheus to install its agent on the Guest operating system via either SSH/WinRM or VMware Tools Guest Process feature. The VMware tools Guest Execution API can be tricky so it is recommended to use SSH/WinRM if possible. However, if it is not possible for the Appliance to have outbound access to all networks in which VMs are being provisioned to the SSH/WinRM ports (22, 5985 respectively) then Guest Execution is the only option.

The *Use VNC* console option on the VMware cloud requires special configuration on each ESXI host but allowed hypervisor level remote console support. (See the Advanced Section for details)

When following this add cloud wizard an option will be presented to create a group or add to an existing group. These groups can be given provisioning permission via role based access control. It is normally recommended that groups are organized such that one cloud exists in one group unless the networks are setup such that internal routing is possible between the clouds. This is very useful for bursting, or hybrid cloud configurations.

Windows Provisioning Tips

By default when provisioning windows templates, Morpheus performs guest customizations which initiates a sysprep. This resets the Administrator user and password. Morpheus will set the Administrator password from Administration > Settings > Provisioning > Windows Settings > Password.

Users can also set the username on an image as Administrator and enter a different password if unique passwords are required per image.

Guest customizations are required when assigning static IP's manually or using IP pools. They can be disabled per virtual image advanced settings under Library > Virtual Images > Edit Image > Advanced > Uncheck "Force Guest Customization" if using DHCP. However the SID will not be changed from the source template. In addition, new VM's will not be able to join a domain that had already been joined by the source template or any other VM's with that SID.

Existing Instances

Morpheus provides several features regarding pulling in existing virtual machines and servers in an environment. Most cloud options contain a checkbox titled '*Inventory Existing Instances*'. When this option is selected, all VMs found within the specified scope of the cloud integration will be scanned periodically and Virtual Machines will be synced into Morpheus. Users may also choose to onboard only virtual machines that are running within specific Resource Pools. Once the vCenter Cloud is integrated, navigate to the detail page for the specific Cloud (select it from the list at Infrastructure > Clouds). From the Resources tab, locate the Pools section. Click ACTIONS > Edit next to a selected Resource Pool. If INVENTORY is checked, Morpheus will automatically onboard virtual machines from that Resource Pool.

By default these virtual machines are considered 'unmanaged' and do not appear in the Provisioning > Instances area but rather Infrastructure > Compute > Virtual Machines. However, a few features are provided with regards to unmanaged instances. They can be assigned to various accounts if using a multitenant master account, however it may be best suited to instead assign the 'Resource Pool' to an account and optionally move all servers with regards to that pool (more on this later).

A server can also be made into a managed server. During this process remote access is requested and an agent install is performed on the guest operating system. This allows for guest operations regarding log acquisition and stats. If the agent install fails, a server will still be marked as managed and an Instance will be created in *Provisioning*, however certain features will not function. This includes stats collection and logs.

Note: All Cloud data is resynchronized on a 5 minute interval. This includes Datastores, Resource Pools, Networks, Blueprints, and Virtual Machines.

Service Plans

A default set of Service Plans are created in Morpheus for the VMware provisioning engine. These Service Plans can be considered akin to AWS Flavors or Openstack Flavors. They provide a means to set predefined tiers on memory, storage, cores, and cpu. Price tables can also be applied to these so estimated cost per virtual machine can be tracked as well as pricing for customers. By default, these options are fixed sizes but can be configured for dynamic sizing. A service plan can be configured to allow a custom user entry for memory, storage, or cpu. To configure this, simply edit an existing Service Plan tied to VMware or create a new one. These all can be easily managed from the Admin > Plans & Pricing section.

Virtual Images / Blueprints

Morpheus will automatically take an inventory of all blueprints configured in vCenter and present them as options during provisioning. However, in order for Morpheus to properly provision these virtual machines and provide accurate stats and health of these virtual machines, an agent must be installed during virtual machine startup. This means remote access needs to be granted at the guest operating system level to Morpheus. To properly configure these virtual images, find the relevant images in Library > Virtual Images and edit the entry. On this form, a few options are presented. The first is a check box asking whether or not cloud-init is enabled. If cloud-init is enabled, simply provide the default OS username configured (for Ubuntu the username is *ubuntu* and for CentOS the username is *centos*). For those looking to add cloud-init to existing blueprints Morpheus requires no special configuration and can use the default *cloud.cfg* settings.

A global cloud-init username/password can also be configured per account as well as a keypair via the Admin->Provisioning settings section. The great benefit of utilizing cloud-init is default blueprints do not need common credential sets thereby increasing provisioning security.

Windows systems do not typically support cloud-init. So simply turn this checkbox off and provide the *Administrator* credentials. It should be noted that these credentials are encrypted in the database. If using WinRM for the RPC Mode instead of VMware tools, a Local or Domain Administrator account credential set can be provided instead.

Snapshots

Morpheus allows the ability to create a snapshot of a VM in VMware vCenter. From the instance detail page, simply select **Actions > Create Snapshot** to begin creation of a new Snapshot. Existing snapshots can be viewed in the **BACKUPS** tab on the instance detail page. Snapshots taken in vCenter will sync into Morpheus every five minutes. To revert to a previous snapshot, click on the revert icon located on the right side of the Snapshot. Snapshots can be deleted by clicking on the trash can icon.

Note: Access to Snapshots can be limited or removed entirely for specific user roles as needed. To edit a role's Snapshots permissions, go to Administration > Roles > (Your selected role) > Snapshots. Users can be given Full, Read-only, or No access.

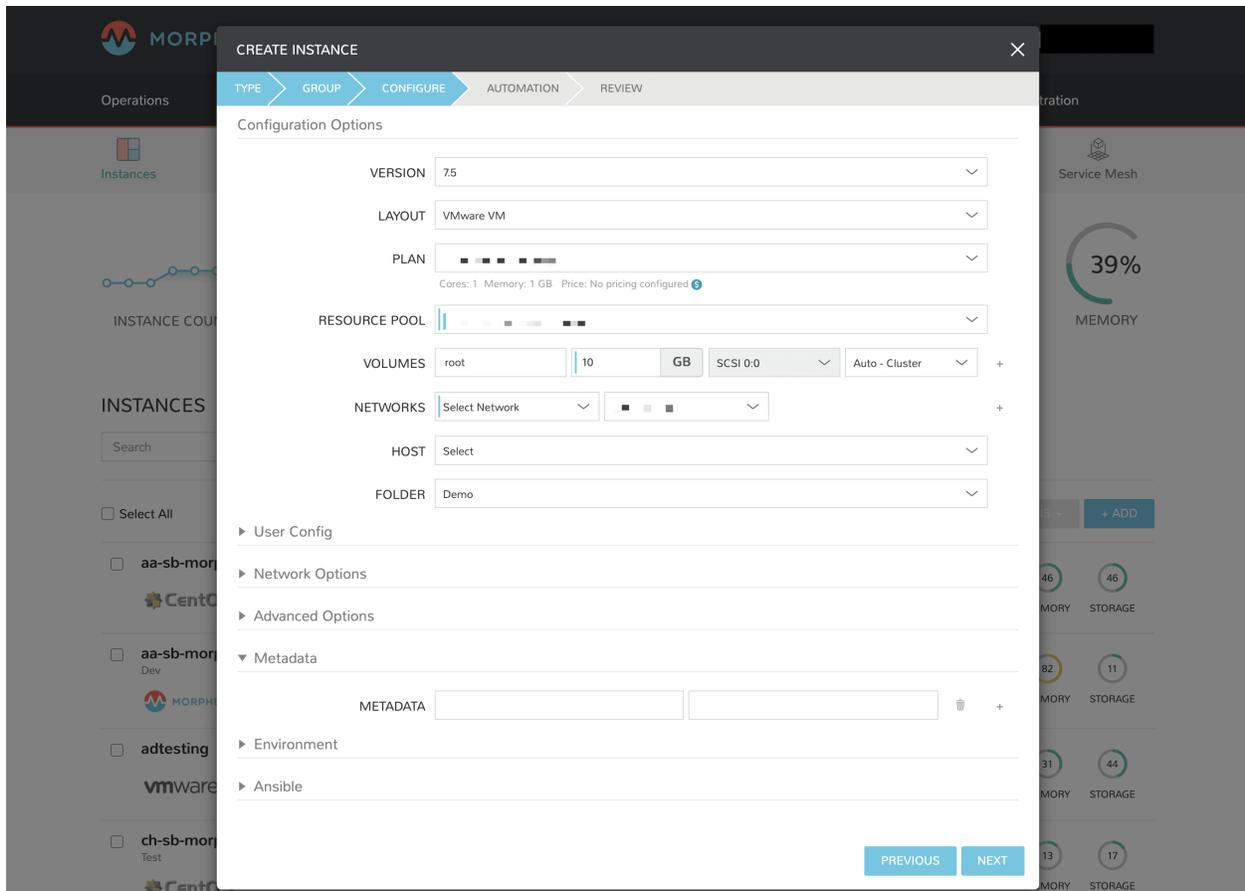
Tagging and Metadata

As of Morpheus version 4.1.0, tagging support is included for vCenter in addition to the other clouds that have already supported it in past versions. Tags will sync to vCenter from Morpheus and existing tags are also inventoried from vCenter into Morpheus.

Note: This feature requires a minimum API version of vCenter 6.5. The API version can be edited by navigating to 'Infrastructure > Clouds' and clicking the edit (pencil) button in the row for the relevant cloud. The field is labeled 'VERSION'.

Tags can be created on-demand when provisioning from the 'CONFIGURE' tab of the 'CREATE INSTANCE' wizard (Provisioning > Instances). Within the 'Metadata' drawer, you will see sets of fields to enter key/value pairs. On creation of the instance, this metadata will be synced into vCenter.

'Inputs' from your library can also be exported as metadata for use with vCenter. When adding or editing a new Input (Library > Options > Inputs), simply mark the box labeled 'EXPORT AS METADATA'. The 'FIELD NAME' becomes the tag category in VMWare.



Docker

So far this document has covered how to add the VMware cloud integration and has enabled users the ability to provision virtual machine based instances via the *Add Instance* catalog in *Provisioning*. Another great feature provided by Morpheus out of the box is the ability to use Docker containers and even support multiple containers per Docker host. To do this a Docker Host must first be provisioned into VMware (multiple are needed when dealing with horizontal scaling scenarios).

To provision a Docker Host simply navigate to the Clusters tab of the Cloud detail page or Infrastructure > Clusters section. From there, click **+ ADD CLUSTER** to add a VMware Docker Host. This host will show up in the Hosts tab next to other ESXi servers that were inventoried by the VMware cloud integration. Morpheus views a Docker host just like any other Hypervisor with the caveat being that it is used for running containerized images instead of virtualized ones. Once a Docker Host is successfully provisioned a green checkmark will appear to the right of the host marking it as available for use. In the event of a failure click into the relevant host that failed and an error explaining the failure will be displayed in red at the top.

Some common error scenarios include network connectivity. For a Docker Host to function properly, it must be able to resolve the Morpheus appliance url which can be configured in Administration > Settings. If it is unable to resolve and negotiate with the appliance than the agent installation will fail and provisioning instructions will not be able to be issued to the host.

Multitenancy

A very common scenario for Managed Service Providers is the need to provide access to VMware resources on a customer by customer basis. With VMware several administrative features have been added to ensure customer resources are properly scoped and isolated. For VMware it is possible to assign specific *Networks*, *Datastores*, and *Resource Pools* to customer accounts or even set the public visibility of certain resources, therefore allowing all sub accounts access to the resource.

Advanced

There are several advanced features provided within Morpheus that can leverage some cool aspects of VMware. One of these features is Remote Console support directly to the hypervisor. To enable this feature a few prerequisites must be met. First, the Morpheus appliance must have network access to the ESXi hosts within VCenter. Secondly, firewall settings need to be adjusted on each ESXi host. This can be done in VSphere under firewall configuration on the host. Simply check the *gdbserver* option, which will open up the necessary ports (starting at 5900 range).

Important: Hypervisor Console for vCenter 6.5 requires Morpheus v3.2.0+

Now that the ESXi hosts are ready to utilize remote console, simply edit the cloud in Morpheus via **Infrastructure > Clouds**. Check the option that says *Enable Hypervisor Console*. It is important to note that currently this functionality only works for newly provisioned vm's provisioned directly via Morpheus. This should change soon however.

It is also possible to import vm snapshots for backup or conversion purposes from VCenter and also an ESXi host. However, this does require that the ESXi host license has an enterprise level license as it will not allow the appliance to download a virtual image if it is not a paid VMware license.

Nutanix

Overview

Nutanix simplifies datacenter infrastructure by integrating server and storage resources allowing applications to run at scale. Morpheus provides an avenue to enhance the Nutanix resources to allow efficient and seamless deployment of applications as a virtual machine (VM) or as a container on a Docker host.

Features

- Virtual Machine Provisioning
- Containers
- Backups / Snapshots
- Resources Groups
- Migrations
- Auto Scaling
- Load Balancing
- Remote Console
- Periodic Synchronization
- Lifecycle Management and Resize

Morpheus can provide a single pane of glass and self-service portal for managing multiple Nutanix Clusters and allowing the seamless deployment of applications.

Note: Prism Central is currently supported and is maintained as a separate, plugin-based Cloud integration. See the [Prism Central plugin page](#) on the Morpheus plugin share site for additional details and access to the plugin.

Getting Started

To get started this a few prerequisites must first be met. The Nutanix cluster should be provisioned and available on the network. Morpheus will look login to the Nutanix cluster with the Nutanix admin credentials and is typically located at the <https://fqdn:9440> url.

Adding a Nutanix Cloud

The Nutanix cluster should be available and responding to the <https://fqdn:9440> url for authentication by Morpheus .

NAME Name of the Cloud in Morpheus

CODE Unique code used for api/cli, automation and policies.

LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT If Visibility is set to Private, select the Tenant the Cloud resources will assigned to.

ENABLED When disabled, automatic Cloud sync is paused and the Cloud will not be selectable for provisioning.

AUTOMATICALLY POWER ON VMS When enabled, Morpheus will maintain the expected powered on state of managed VM's and power on any managed VM's in the cloud that have been shut down for unknown reasons (not powered off by Morpheus) to ensure availability of services.

Note: When "AUTOMATICALLY POWER ON VMS" is enabled, the power state of managed VM's should be maintained in Morpheus. This setting is not applicable to discovered/unmanaged resources.

API URL URL of the Nutanix Prism API, example: <https://10.30.21.220:9440>

USERNAME Nutanix admin username

PASSWORD Nutanix admin password

Inventory Existing Instances If enabled, existing Virtual Machines will be inventoried and appear as unmanaged Virtual Machines in Morpheus .

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME_ZONE Configures the time zone on provisioned VM's if necessary.

DATACENTER_ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK_MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL_FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY_SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST_PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE_MODE Single Disk, LVM or Clustered

BACKUP_PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION_PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS_INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE_REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG_MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB_DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE_MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT_INSTALL_MODE

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud Init / Unattend (when available): (DEFAULT) Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API_PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL_AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

Service Plans

A default set of Service Plans are created in Morpheus for the VMware provisioning engine. These Service Plans can be considered akin to AWS Flavors or Openstack Flavors. They provide a means to set predefined tiers on memory, storage, cores, and cpu. Price tables can also be applied to these so estimated cost per virtual machine can be tracked as well as pricing for customers. By default, these options are fixed sizes but can be configured for dynamic sizing. A service plan can be configured to allow a custom user entry for memory, storage, or cpu. To configure this, simply edit an existing Service Plan tied to Nutanix or create a new one. These all can be easily managed from the Admin | Service Plans & Pricing section.

Docker

So far this document has covered how to add the Nutanix cloud integration and has enabled users the ability to provision virtual machine based instances via the Add Instance catalog in Provisioning. Another great feature provided by Morpheus out of the box is the ability to use Docker containers and even support multiple containers per Docker host. To do this a Docker Host must first be provisioned into Nutanix (multiple are needed when dealing with horizontal scaling scenarios).

To provision a Docker Host, simply navigate to the Cloud detail page or Infrastructure > Clusters section. From there click + *ADD CLUSTER* to add a Nutanix Docker Host. Morpheus views a Docker host just like any other Hypervisor with the caveat being that it is used for running containerized images instead of virtualized ones. Once a Docker Host is successfully provisioned a green checkmark will appear to the right of the host marking it as available for use. In the event of a failure click into the relevant host that failed and an error explaining the failure will be displayed in red at the top.

Some common error scenarios include network connectivity. For a Docker Host to function properly, it must be able to resolve the Morpheus appliance url which can be configured in Admin Settings. If it is unable to resolve and negotiate with the appliance than the agent installation will fail and provisioning instructions will not be able to be issued to the host.

Snapshots

Morpheus allows the ability to create a snapshot of a Nutanix instance. From the instance detail page, simply select **Actions > Create Snapshot** to begin creation of a new Snapshot. Existing snapshots can be viewed in the **BACKUPS** tab on the instance detail page. Snapshots taken outside Morpheus will be synced every five minutes. To revert to a previous snapshot, click on the revert icon located on the right side of the Snapshot. Snapshots can be deleted by clicking on the trash can icon.

Note: Access to Snapshots can be limited or removed entirely for specific user roles as needed. To edit a role's Snapshots permissions, go to Administration > Roles > (Your selected role) > Snapshots. Users can be given Full, Read-only, or No access.

Openstack

Overview

Openstack is becoming a widely used on-premise infrastructure orchestration platform. It has a wide array of contributors and enterprise sponsorships. There are several variations on Openstack as well. Morpheus supports integration with all the various platform offerings and ranges in support all the way back to Openstack Juno. The complete list of compatible versions is listed in our [Integration Compatibility table](#). It leverages the APIs and provides full functionality as a self service portal in front of Openstack.

Features

- Virtual Machine Provisioning
- Backups and Snapshots
- Security Group Management
- Disk Mode support Local/Image (via Ceph)
- Floating IP Assignment support
- Brownfield VM management and Migration
- Lifecycle Management and Resize
- Docker Host management and configuration
- Manila File Services (SFS)
- Object Storage (OBS)
- Network Lifecycle
- LBaaS/Octavia Load Balancing Services

On top of all these features, Morpheus also adds additional features to Openstack that do not exist out of the box to make it easier to manage in multitenant environments as well as hybrid cloud environments:

- Image to QCOW2 Image Conversion
- QCOW2 to RAW Image Conversion
- Multitenancy resource allocation
- Virtual Image management (Blueprints)
- Auto-scaling and recovery
- Instance Cloning
- Morpheus Kubernetes Cluster Deployment

Tip: To allow Morpheus to list Hypervisor Hosts, ensure the Openstack user used for the Cloud Integration has sufficient privileges for `os_compute_api:os-hypervisors` in `/etc/nova/policy.json` in Openstack.

Getting Started

OpenStack Clouds are very easy to integrate with Morpheus. First, go to the **Infrastructure > Clouds** section and click + *ADD*. Select OpenStack to begin the integration process, most branded flavors of OpenStack will work with this Cloud selection as well.

Warning: Support for OpenStack v2 Identity API has been removed in v5.3.3

Cloud Configuration

NAME Name of the Cloud in Morpheus

CODE Unique code used for api/cli, automation and policies.

LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT If Visibility is set to Private, select the Tenant the Cloud resources will assigned to.

ENABLED When disabled, automatic Cloud sync is paused and the Cloud will not be selectable for provisioning.

AUTOMATICALLY POWER ON VMS When enabled, Morpheus will maintain the expected power state of managed VMs. Morpheus will power on any managed VMs in the Cloud that have been shut down for unknown reasons (not powered off by Morpheus) to ensure availability of services.

Note: When “AUTOMATICALLY POWER ON VMS” is enabled, the power state of managed VMs should be maintained in Morpheus. This setting is not applicable to discovered/unmanaged resources.

Details

IDENTITY API URL v3 Identity endpoint.

DOMAIN ID For *Default* domains, Default can be used. For other domain the Domain ID must be entered, not the Domain Name.

PROJECT Enter the target project or leave this field blank to integrate all projects

USERNAME Service Username

PASSWORD Service user password

OS VERSION Select Openstack Version. Morpheus supports the latest versions of OpenStack, select the latest version available if your current version is not shown.

IMAGE FORMAT Select QCOW2, RAW or VMDK Image Type

LB TYPE Select LB Type for Openstack LB syncing and creation

Inventory Existing Instances Select for Morpheus to discover and sync existing VM's

Enable Hypervisor Console Hypervisor console support for openstack currently only supports novnc. Be sure the novnc proxy is configured properly in your openstack environment. When disabled Morpheus will use ssh and rdp for console connection (vm/host credentials required)

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM's if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT INSTALL MODE

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud Init / Unattend (when available): (DEFAULT) Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing sysprep images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

Note: v5.3.3 adds openstack project management which requires additional permissions in openstack:

```
identity:list_domain_roles
identity:list_role_assignments
identity:list_roles
identity:list_projects
identity:create_project
identity:update_project
identity:delete_project
identity:create_grant
identity:revoke_grant
```

Most of the information in the dialog can be acquired from the Openstack dashboard. under *Project > Access & Security > API Access*. The API URL that is needed is the one tied to *Identity*. The Domain and Project inputs typically correlate to the multitenant domain setup within Openstack (sometimes just left at default) as well as the project name given to instances. Morpheus allows multiple integrations to the same Openstack cluster to be scoped to various domains and projects as needed.

The remaining options help Morpheus determine which API capabilities exist in the selected Openstack environment. Hence the need for the Openstack version and image format. If a newer Openstack cluster is being used then exists in the dropdown, simply select the most recent version in the dropdown and this should function sufficiently until the new version is added.

Tip: Some Openstack environments do not support QCOW2 and force RAW image formats (like Metapod). This is due to some network overhead in Ceph created by using QCOW2. Morpheus keeps two copies of Openstack image templates for this exact purpose.

Saving this cloud integration should perform a verification step and close upon successful completion.

Existing Instances

Morpheus provides several features regarding pulling in existing virtual machines and servers in an environment. Most cloud options contain a checkbox titled *'Inventory Existing Instances'*. When this option is selected, all VMs found within the specified scope of the cloud integration will be scanned periodically and Virtual Machines will be synced into Morpheus.

By default these virtual machines are considered 'unmanaged' and do not appear in the Provisioning > Instances area but rather Infrastructure > Compute > Virtual Machines. However, a few features are provided with regards to unmanaged instances. They can be assigned to various accounts if using a multitenant master account, however it may be best suited to instead assign the 'Resource Pool' to an account and optionally move all servers with regards to that pool (more on this later).

A server can also be made into a managed server. During this process remote access is requested and an agent install is performed on the guest operating system. This allows for guest operations regarding log acquisition and stats. If the agent install fails, a server will still be marked as managed and an Instance will be created in *Provisioning*, however certain features will not function. This includes stats collection and logs.

Service Ports

Morpheus consumes the following OpenStack service ports by default as part of its cloud integration. If your OpenStack implementation has been configured to use alternate service ports, these can be overwritten in the Cloud configuration under the Service Endpoints section when adding or editing the Cloud integration.

▼ Service Endpoints

COMPUTE SERVICE	<input type="text" value="http://10.30.21.150:8774"/>
IMAGE SERVICE	<input type="text" value="http://10.30.21.150:9292"/>
STORAGE SERVICE	<input type="text" value="http://10.30.21.150:8776"/>
NETWORK SERVICE	<input type="text" value="http://10.30.21.150:9696"/>
LOAD BALANCER SERVICE	<input type="text"/>
OBJECT STORAGE SERVICE	<input type="text"/>
SHARED FILE SYSTEM SERVICE	<input type="text"/>

Default Service Ports

- Identity: 5000
- Compute: 8774
- Image: 9292
- Key Manager: 9311
- Network: 9696
- Volume API v3: 8776 v3
- Manila: 8786

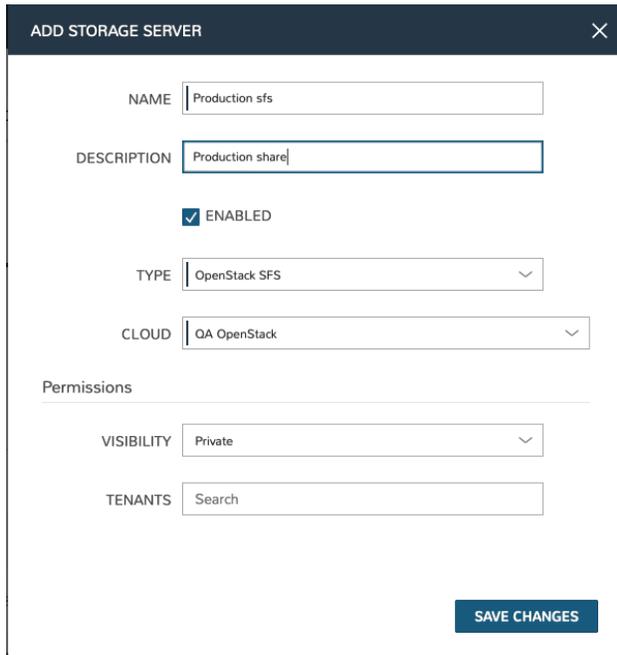
OpenStack Scalable File Service (SFS)

The Morpheus integration with Openstack Cloud includes the capability to work with Openstack Scalable File Service (SFS). SFS is shared file storage hosted on Openstack Cloud. By integrating Morpheus with Openstack you can discover, create, manage, and delete SFS servers, as well as view and work with the file shares and files contained therein.

SFS Server Discovery and Management

On integrating Openstack Cloud with Morpheus, SFS servers and file shares are discovered automatically after a short time. The server(s) can be viewed in Infrastructure > Storage > Servers. By viewing the server detail page and clicking *EDIT*, the storage server can be scoped as needed. Administrators can choose to scope to other Openstack Cloud integrations (if more than one relevant integration currently exists), select from synced availability zones, and scope the storage server to specific Tenants if desired.

Additionally, Openstack SFS servers can be created from the storage server list page (Infrastructure > Storage > Servers) directly in Morpheus. Click *+ADD* to begin and set the storage server type value to “Openstack SFS”. Just like with existing synced SFS servers, those created from Morpheus can be scoped as needed.



The screenshot shows a modal window titled "ADD STORAGE SERVER" with a close button (X) in the top right corner. The form contains the following fields and controls:

- NAME:** Text input field containing "Production sfs".
- DESCRIPTION:** Text input field containing "Production share".
- ENABLED:** A checked checkbox labeled "ENABLED".
- TYPE:** A dropdown menu with "OpenStack SFS" selected.
- CLOUD:** A dropdown menu with "QA OpenStack" selected.
- Permissions:** A section header.
- VISIBILITY:** A dropdown menu with "Private" selected.
- TENANTS:** Text input field containing "Search".
- SAVE CHANGES:** A blue button at the bottom right.

SFS File Share Discovery and Management

Discovered file shares will appear among other file shares synced with Morpheus in Infrastructure > Storage > File Shares. Depending on the number of cloud integrations in your Morpheus appliance and the number of cloud integrations available to your user account, this list may be quite large. Using the search bar on this page, we can narrow down the list to only file shares whose names match the search terms.

We can drill into individual file shares by clicking on the hyperlinked name in the list of all integrated file shares. From the file share detail page, a list of files will appear on the files tab. Begin the process of adding a new file by clicking *+ADD*. The Access tab on the file shares detail page allows users to view and manage ACL rules.

Note: “Failed to load files from storage provider” is present when the Morpheus appliance doesn’t have access to the file share.

New Openstack SFS file shares can also be created directly in Morpheus. From the file shares list page, get started by clicking *+ADD*. Select the type “Openstack SFS Share”. Set the storage service field to a pre-existing Openstack SFS server. Setting a friendly name for the file share in Morpheus and selecting from synced availability zones is required.

Network and Router Creation

Once an OpenStack Cloud is integrated into Morpheus, new network creation options become available. When adding a new network (Infrastructure > Networks > Networks Tab), a new type labeled “OpenStack Private Network” is available when clicking *+ADD*. When the user creates this network construct in Morpheus, a layer two subnet is created but it’s not connected to a Virtual Private Cloud (VPC). This is by design as an Internet-routable network is not always desired. Continue on with this section after creating the network to also create a VPC (router).

Create a network

1. Navigate to Infrastructure > Networks
2. Click on the Networks tab
3. Click *+ADD*
4. Select OpenStack Private Network
5. Complete the modal based on requirements for the new network
6. Click *SAVE CHANGES*

Create a router

1. Navigate to Infrastructure > Networks
2. Click on the Routers tab
3. Click *+ADD*
4. Select OpenStack Router
5. Complete the modal based on requirements for the new router
6. Click *SAVE CHANGES*

When creating a router, it’s helpful to note that the External Network is the floating IP network that has been assigned to the OpenStack project. This network will grant your Instances their routes out to the Internet. The Internal Subnet can be a layer two subnet that you may have created in the previous step. In addition, multiple subnets can be added to the router (VPC) and the IP address on the subnet would be the router’s internal IP address.

Advanced

There are a few advanced features when it comes to provisioning on top of Openstack. Most of these present themselves in the provisioning wizard. This includes OS Volume Type (Local or Volume) which dictates whether the main OS disk is copied and run off the hypervisor or remotely mounted as a volume via Glacier. Some Openstack setups only configure hypervisors with minimal local disks so volume type is needed.

Another option during provisioning is “Assign Floating IP”. This option does exactly what it says and is similar to the feature on the Openstack instance dashboard itself. It should be noted that this will attempt to acquire a floating IP from the project and, if out of capacity, will attempt to increase capacity to the project if the cloud credentials provided have sufficient administrative privileges to do so.

Docker

So far this document has covered how to add the Openstack cloud integration and has described how to provision virtual machine-based Instances via the *Add Instance* catalog in *Provisioning*. Another great feature provided by Morpheus out of the box is the ability to work with Docker containers and even support multiple containers per Docker host. To do this, a Docker host must first be provisioned into Openstack (multiple hosts are needed when dealing with horizontal scaling scenarios).

To provision a Docker Host, navigate to Infrastructure > Clusters and click + *ADD CLUSTER*. Complete the provisioning wizard including selecting the appropriate Group and Cloud. Alternatively, you can navigate to the Clusters tab for a specific Cloud (Infrastructure > Clouds > Specific Cloud detail page > Clusters tab) and begin the process of provisioning a Docker host to that Cloud from there. Once completed, this host will show up in the Hosts sections (Infrastructure > Hosts OR Infrastructure > Clouds > Specific Cloud detail page > Hosts tab). Morpheus views a Docker host just like any other Hypervisor with the caveat being that it is used for running containerized images instead of virtualized ones.

Once a Docker Host is successfully provisioned, a green checkmark will appear to the right of the host marking it as available for use. In the event of a failure, click into the relevant host that failed and an error explaining the failure will be displayed in red at the top.

Some common error scenarios include network connectivity. For a Docker Host to function properly, it must be able to resolve the Morpheus appliance URL which can be configured in Administration > Settings. If it is unable to resolve and negotiate with the appliance, the Morpheus Agent installation will fail and provisioning instructions will not be able to be issued to the host.

Oracle VM

Add an Oracle VM Cloud

Cloud Configuration

NAME Name of the Cloud in Morpheus

CODE Unique code used for api/cli, automation and policies.

LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT If Visibility is set to Private, select the Tenant the Cloud resources will assigned to.

ENABLED When disabled, automatic Cloud sync is paused and the Cloud will not be selectable for provisioning.

AUTOMATICALLY POWER ON VMS When enabled, Morpheus will maintain the expected power state of managed VMs. Morpheus will power on any managed VMs in the Cloud that have been shut down for unknown reasons (not powered off by Morpheus) to ensure availability of services.

Note: When “AUTOMATICALLY POWER ON VMS” is enabled, the power state of managed VMs should be maintained in Morpheus. This setting is not applicable to discovered/unmanaged resources.

Details

API URL Oracle VM API URL. ex: <https://10.20.30.40:7002/ovm/core/wsapi/rest>

USERNAME Oracle VM User

PASSWORD Oracle VM User Password

REPOSITORY Available repositories will auto-populate upon successful authentication with the above credentials. Select appropriate repository for this Cloud.

SERVER POOL Available server pools will auto-populate upon successful authentication with the above credentials. Select appropriate server pool for this Cloud.

Inventory Existing Instances If enabled, existing Virtual Machines will be inventoried and appear as unmanaged Virtual Machines in Morpheus .

The Cloud can now be added to a Group or configured with additional Advanced options.

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM's if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT INSTALL MODE

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud Init / Unattend (when available): (DEFAULT) Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

Oracle Cloud

Required Permissions

Integrating Oracle Public Cloud with Morpheus requires access to a service account with at least the permission set listed below. When creating an Oracle Cloud integration scoped to a specific compartment, the service account needs access only to the listed resource families within the chosen compartment. If the Cloud will be scoped to all compartments, the service account will need access to the listed resource families at the root compartment.

- **Oracle Cloud Policy Requirements**

Allow group <GROUP CONTAINING SERVICE USER> to manage cluster-family in compartment <CHOSEN COMPARTMENT OR ROOT COMPARTMENT>

Allow group <GROUP CONTAINING SERVICE USER> to manage compute-management-family in compartment <CHOSEN COMPARTMENT OR ROOT COMPARTMENT>

Allow group <GROUP CONTAINING SERVICE USER> to manage data-catalog-family in compartment <CHOSEN COMPARTMENT OR ROOT COMPARTMENT>

Allow group <GROUP CONTAINING SERVICE USER> to manage dns in compartment <CHOSEN COMPARTMENT OR ROOT COMPARTMENT>

Allow group <GROUP CONTAINING SERVICE USER> to manage file-family in compartment <CHOSEN COMPARTMENT OR ROOT COMPARTMENT>

Allow group <GROUP CONTAINING SERVICE USER> to manage instance-family in compartment <CHOSEN COMPARTMENT OR ROOT COMPARTMENT>

Allow group <GROUP CONTAINING SERVICE USER> to manage object-family in compartment <CHOSEN COMPARTMENT OR ROOT COMPARTMENT>

Allow group <GROUP CONTAINING SERVICE USER> to manage virtual-network-family in compartment <CHOSEN COMPARTMENT OR ROOT COMPARTMENT>

Allow group <GROUP CONTAINING SERVICE USER> to manage volume-family in compartment <CHOSEN COMPARTMENT OR ROOT COMPARTMENT>

Add Oracle Public Cloud

Important: A Keypair (both public and private key) must be added to Morpheus with the Public Key in ssh-rsa format. The Public Key in PEM format needs to be added to Oracle Cloud users keys in Oracle Cloud console for authentication.

Note: Information on uploading the Public Key and generating Tenancy's OCID and User's OCID can be found at <https://docs.cloud.oracle.com/iaas/Content/API/Concepts/apisigningkey.htm>

To get started, navigate to Infrastructure > Clouds. Click + *ADD* and select Oracle Public Cloud to begin a new one. Configure the following options for the new Cloud:

Cloud Configuration

NAME Name of the Cloud in Morpheus

CODE Unique code used for api/cli, automation and policies.

LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT If Visibility is set to Private, select the Tenant the Cloud resources will assigned to.

ENABLED When disabled, automatic Cloud sync is paused and the Cloud will not be selectable for provisioning.

AUTOMATICALLY POWER ON VMS When enabled, Morpheus will maintain the expected power state of managed VMs. Morpheus will power on any managed VMs in the Cloud that have been shut down for unknown reasons (not powered off by Morpheus) to ensure availability of services.

Note: When “AUTOMATICALLY POWER ON VMS” is enabled, the power state of managed VMs should be maintained in Morpheus. This setting is not applicable to discovered/unmanaged resources.

Details

TENANCY OCID The OCID string from *Tenancy Information* section in Oracle Cloud

USER OCID OCID String for the OPC API user

SELECT KEY PAIR Select a keypair added to Morpheus matching the public key added to specified OPC API user

REGION Select the OPC region (populates after successful account authentication)

COMPARTMENT Choose to scope the Cloud to all compartments or to one specific compartment (populates after successful account authentication)

INVENTORY Turn on for Morpheus to discover and sync existing VMs

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM's if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option

to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT INSTALL MODE

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud Init / Unattend (when available): (DEFAULT) Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

Enable Live Costing for Oracle Public Cloud

Morpheus version 4.2.1 and higher support live costing data from the Oracle Cloud metering API. In order to authenticate with this API, edit your existing Oracle Cloud account integration or begin the process of newly integrating an account that wasn't previously consumable in Morpheus (Infrastructure > Clouds > +ADD).

In the advanced options section of the add/edit cloud modal for Oracle Public Cloud, the **COSTING KEY** and **COSTING SECRET** fields must be completed to work with metering API data in Morpheus. Unlike the OCI API authentication used to initially integrate Oracle Cloud, the metering API uses token-based authentication. We must access a Client ID and Client Secret value from the Oracle Public Cloud console to complete these fields.

▼ Advanced Options

COSTING KEY

COSTING SECRET

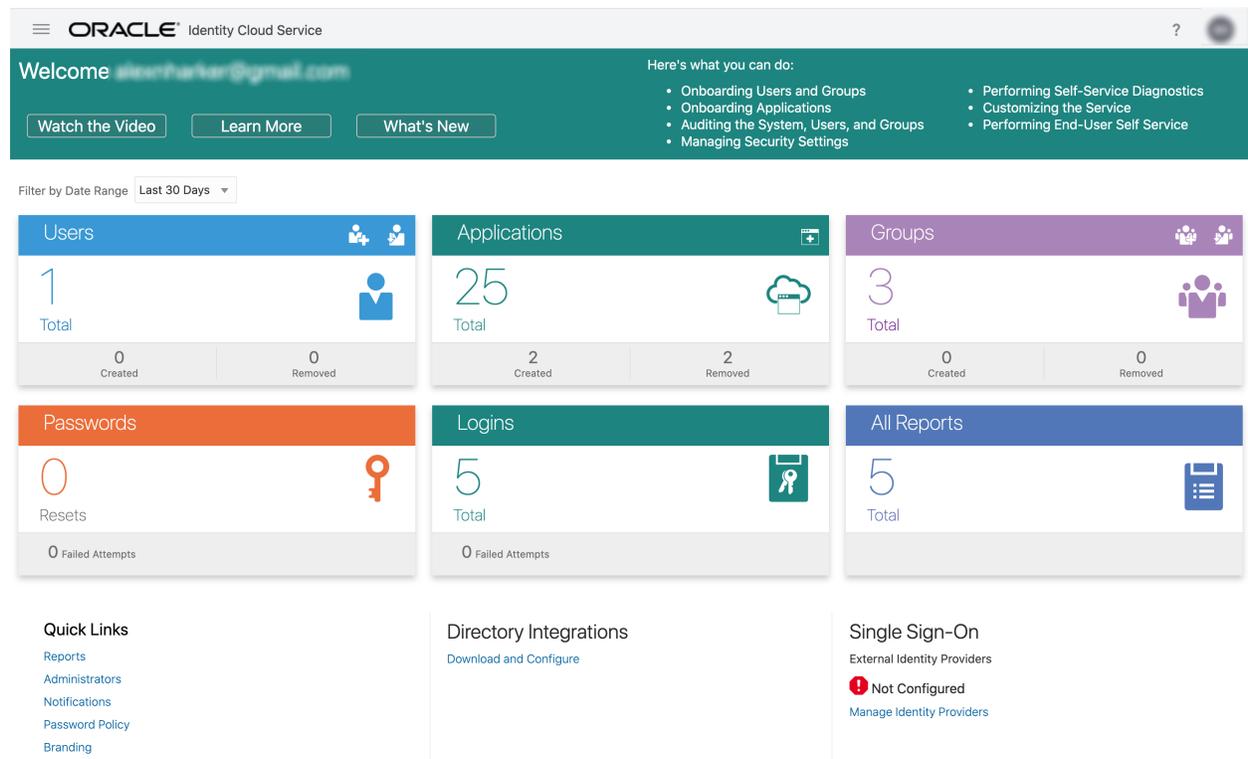
Navigate to Oracle cloud sign in page, the URL for which is similar to the following example:

`https://idcs-00a0xxxxxxxxxxxxx.identity.oraclecloud.com/ui/v1/signin`

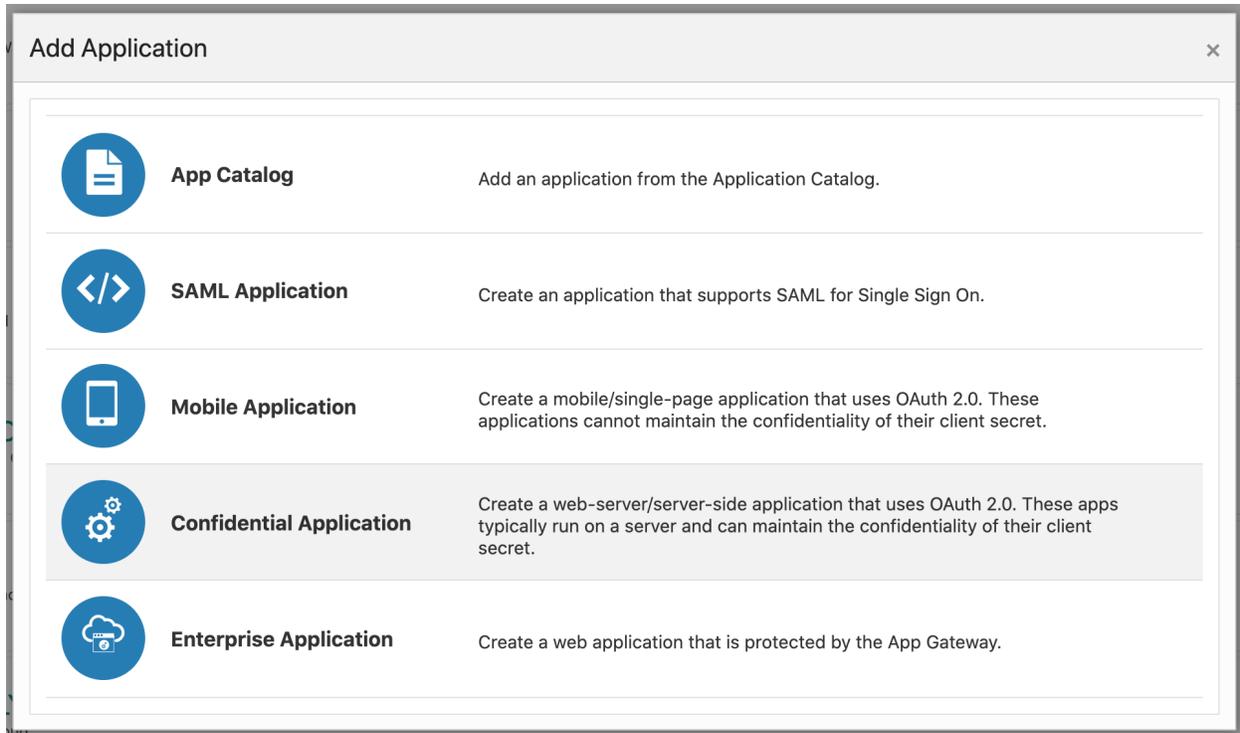
If you're not redirected to the admin console similar to the one pictured below, log out and replace 'signin' at the end of the URL with 'adminconsole' as in the following example:

`https:// idcs-00a0xxxxxxxxxxxxx.identity.oraclecloud.com/ui/v1/adminconsole`

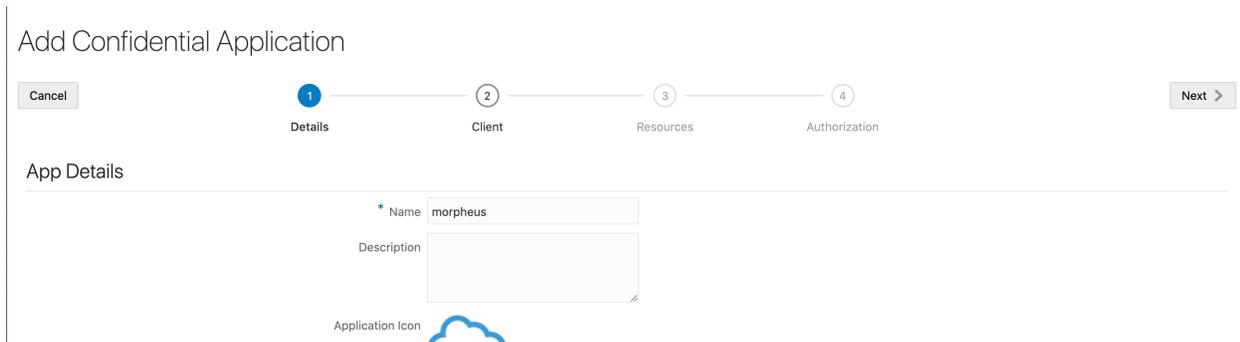
You'll immediately be redirected back to the same signin page but in doing that you should be taken to the admin console after authenticating your session once again.



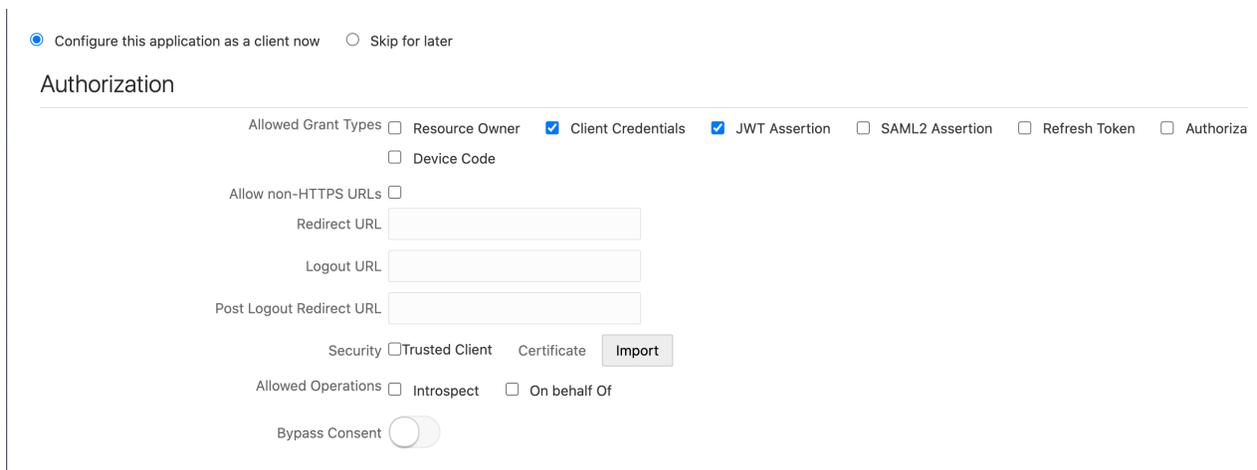
Create a new application and select the type "Confidential Application".



On the Details tab, enter a “Name” value and click “Next”.



On the Client tab, choose to “Configure this application as a client now” to reveal additional fields. Then, in the Authorization section, mark the boxes for “Client Credentials” and “JWT Assertion”.



In the Token Issuance Policy section, click the “+Add Scope” button. Click the right-facing arrow button in the row for “CloudPortalResourceApp”. Mark the box to give read access for metering and click “Add”.

Select Scope ×

< Back

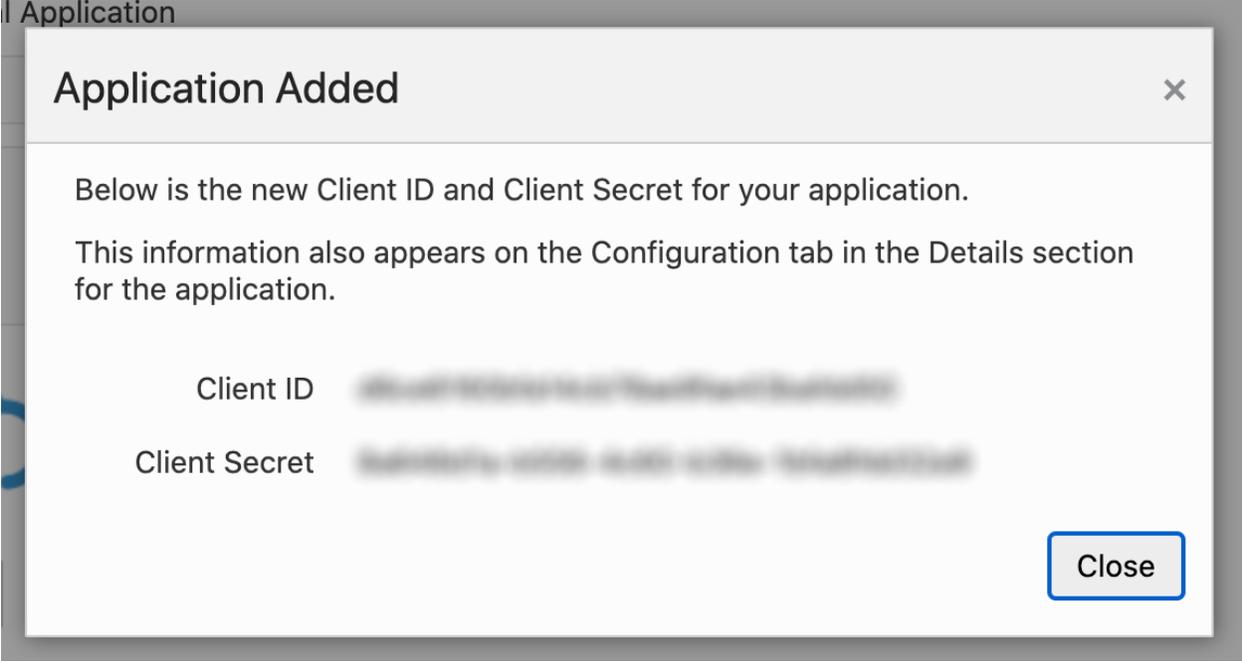
CloudPortalResourceApp

<input type="checkbox"/>	urn:opc:resource:consumer:cp:itas:myservices::read
<input type="checkbox"/>	urn:opc:resource:consumer:cp:itas:myservices::all
<input checked="" type="checkbox"/>	urn:opc:resource:consumer:cp:itas:metering::read
<input type="checkbox"/>	urn:opc:resource:consumer:cp:itas:metering::all

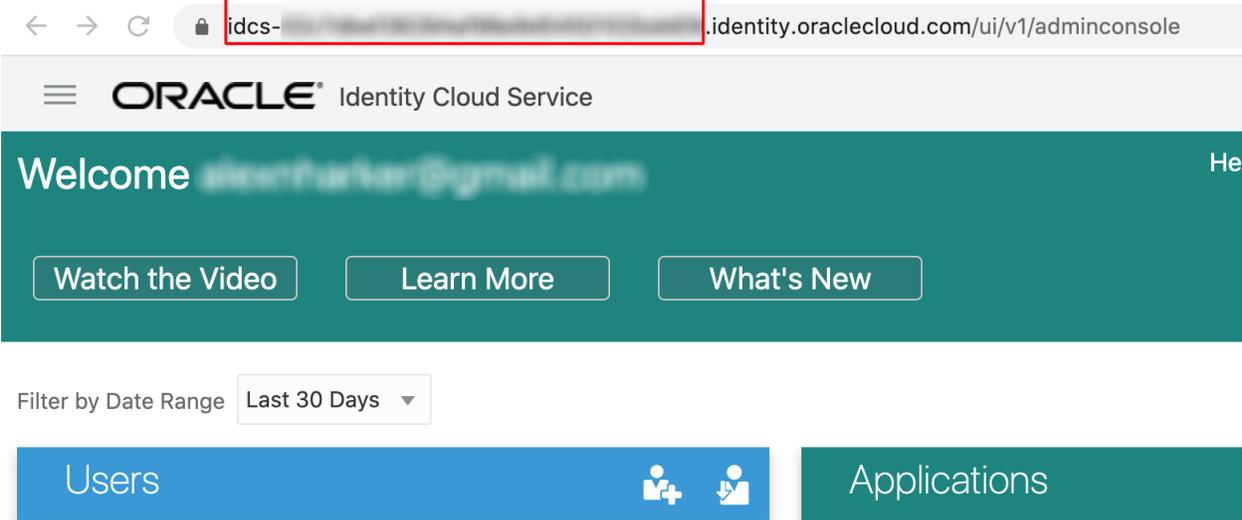
Add

Click “Next” until the “Finish” button is shown, then click “Finish”

The Client ID and Client Secret value will be shown at this point. If these values need to be referenced in the future, simply edit the application and go to the Configuration tab. The Client ID and Client Secret are shown in the General Information section.



Back in Morpheus, enter these values in the COSTING KEY and COSTING SECRET fields of the add/edit cloud modal for your Oracle Public Cloud integration. You also need to fill in the IDENTITY SERVICE value. This value can be found in the URL for your Oracle admin console as shown in the image below. It will be in a format `idcs-xxxxxx`.



Save changes to the Cloud.

Open Telekom Cloud

Open Telekom Cloud is an Openstack-based public cloud offering. Morpheus offers a robust integration into OTC and supports many of its features, including those listed in the next section.

Features

- Virtual machine provisioning
- Backups
- Brownfield VM management and migration
- Hypervisor remote console
- Cloud sync
- Lifecycle management and resizing
- Network security group creation
- Network security group management
- Router and network creation
- Load balancer services
- Docker host management and configuration
- Floating IP assignment
- OBS buckets (create, manage, delete, and discovery)

Add an Open Telekom Cloud

Navigate to Infrastructure > Clouds and click + *ADD*. Scroll to Open Telekom Cloud and click *NEXT*. Complete the ADD CLOUD modal, the remainder of this guide includes descriptions of the fields presented on this modal with advice on formatting needed values and where certain data can be located.

Cloud Configuration

NAME Name of the Cloud in Morpheus

CODE Unique code used for api/cli, automation and policies.

LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT If Visibility is set to Private, select the Tenant the Cloud resources will assigned to.

ENABLED When disabled, automatic Cloud sync is paused and the Cloud will not be selectable for provisioning.

AUTOMATICALLY POWER ON VMS When enabled, Morpheus will maintain the expected power state of managed VMs. Morpheus will power on any managed VMs in the Cloud that have been shut down for unknown reasons (not powered off by Morpheus) to ensure availability of services.

Note: When “AUTOMATICALLY POWER ON VMS” is enabled, the power state of managed VMs should be maintained in Morpheus. This setting is not applicable to discovered/unmanaged resources.

Details

IDENTITY API URL The v3 identity API URL, such as `https://iam.eu-de.otc.t-systems.com/v3`

DOMAIN ID Note that this is the Domain ID and not the Domain Name. The Domain ID can be found via the CLI by typing `openstack domain list`. For default domains, “Default” can be used

PROJECT OTC projects are groupings of resources and can include compute resources, storage or networking. Multiple projects may be nested under your account. Select the project to which Morpheus should onboard from (if desired) and provision

REGION OTC Region

USERNAME The username for the OTC service account that Morpheus will use. Ensure this account has sufficient cloud privileges to avoid interruption of work in Morpheus

PASSWORD The password for the above service account

IMAGE FORMAT Select QCOW, RAW or VMDK

IMAGE STORE Set an OBS bucket as a permanent store location for Morpheus virtual images. Users are limited to uploading images of 2GB or less in size if an OBS bucket is not specified here

INVENTORY EXISTING IMAGES When selected, Morpheus will automatically onboard existing cloud resources which can be converted to managed Instance if desired. View onboarded cloud resources in the Compute Section (Infrastructure > Compute)

ENABLE HYPERVISOR CONSOLE Hypervisor console support for Openstack currently only supports novnc. Be sure the novnc proxy is configured properly in your Openstack environment

Service Endpoints

If needed, update the following service endpoints. A complete listing of OTC API endpoints is [here](#).

- COMPUTE SERVICE
- IMAGE SERVICE
- STORAGE SERVICE
- NETWORK SERVICE
- LOAD BALANCER SERVICE
- OBJECT STORAGE SERVICE
- SHARED FILE SYSTEM SERVICE

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in `admin > settings`) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM’s if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT INSTALL MODE

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud Init / Unattend (when available): (DEFAULT) Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

Network and Router Creation

Once an Open Telekom Cloud is integrated into Morpheus, new network creation options become available. When adding a new network (Infrastructure > Networks > Networks Tab), a new type labeled “Open Telekom Private Network” is available when clicking *+ADD*. When the user creates this network construct in Morpheus, a layer two subnet is created but it’s not connected to a Virtual Private Cloud (VPC). This is by design as an Internet-routable network is not always desired. Continue on with this section after creating the network to also create a VPC (router).

Create a network

1. Navigate to Infrastructure > Networks
2. Click on the Networks tab
3. Click *+ADD*
4. Select Open Telekom Private Network
5. Complete the modal based on requirements for the new network
6. Click *SAVE CHANGES*

Create a router

1. Navigate to Infrastructure > Networks
2. Click on the Routers tab
3. Click *+ADD*
4. Select Open Telekom Router
5. Complete the modal based on requirements for the new router
6. Click *SAVE CHANGES*

When creating a router, it’s helpful to note that the External Network is the floating IP network that has been assigned to the OTC project. This network will grant your Instances their routes out to the Internet. The Internal Subnet can be a layer two subnet that you may have created in the previous step. In addition, multiple subnets can be added to the router (VPC) and the IP address on the subnet would be the router’s internal IP address.

SCVMM

Requirements

- Access to SCVMM host on port 5985 for Agent installation
- Access to Hyper-V host on port 2179 for hypervisor console access
- Morpheus Agent installation (installed on the target SCVMM host via port 5985 and WinRM)
- User with administrator privileges

Agent Requirement

SCVMM and Hyper-V integrations utilize the Morpheus Agent for communication with the Morpheus appliance, making the Morpheus Agent required. This also means SCVMM and Hyper-V Clouds can only point to one Morpheus Appliance at any given time. If another Morpheus Appliance adds an SCVMM or Hyper-V Cloud that is already managed by another Morpheus Appliance, the Morpheus Agent `appliance_url` will be updated to point to the new Morpheus appliance_url, and the previous Morpheus Appliance will no longer be able to communicate with the SCVMM Cloud or Hyper-V Cloud until the Agent configuration is updated to point to the previous appliance again. In Morpheus version 4.2.1 and higher, multiple Morpheus clouds can be created by integrating with the same SCVMM host. This allows users to create separate Clouds with are scoped to different SCVMM Cloud, Host and/or Cluster combinations.

Note: Morpheus only supports integration with standalone SCVMM installations and not high-availability cluster installation at this time.

Add a SCVMM Cloud

1. Navigate to **Infrastructure > Clouds**
2. Select + *CREATE CLOUD*, select SCVMM, and then click *Next*.
3. Enter the following into the Create Cloud modal:

Note: You will need to open port 5985 in order for Morpheus to communicate to SCVMM. You will also want to make sure the SCVMM Controller has WinRM enabled.

Cloud Configuration

NAME Name of the Cloud in Morpheus

CODE Unique code used for api/cli, automation and policies.

LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT If Visibility is set to Private, select the Tenant the Cloud resources will assigned to.

ENABLED When disabled, automatic Cloud sync is paused and the Cloud will not be selectable for provisioning.

AUTOMATICALLY POWER ON VMS When enabled, Morpheus will maintain the expected power state of managed VMs. Morpheus will power on any managed VMs in the Cloud that have been shut down for unknown reasons (not powered off by Morpheus) to ensure availability of services.

Note: When “AUTOMATICALLY POWER ON VMS” is enabled, the power state of managed VMs should be maintained in Morpheus. This setting is not applicable to discovered/unmanaged resources.

Details

SCVMM HOST IP address or URL of SCVMM host server

USERNAME SCVMM Username, for example: svc.scvmm

PASSWORD SCVMM user Password

CLOUD To scope the SCVMM Integration to a single Cloud, select it from the Cloud dropdown, which populates after establishing communication and authorization over port 5985 using the supplied username and password. To scope to all Clouds, leave the dropdown selection as `Select Cloud`

HOST GROUP To scope the SCVMM Integration to a single host group, select a host group from the dropdown list. To scope to all host groups, select `All Hosts`

CLUSTER To scope the SCVMM Integration to a single cluster, select a cluster from the dropdown list. To scope to all host groups, select `All`

LIBRARY SHARE Select a Library Share to be used with the cloud integration

SHARED CONTROLLER

When creating additional Morpheus clouds that point to an SCVMM host already integrated with this appliance, select the appropriate shared controller value from the dropdown.

Important: Only set `SHARED CONTROLLER` on additional Morpheus clouds and not on the Primary Morpheus SCVMM cloud. Failure to set the `SHARED CONTROLLER` on secondary Morpheus clouds pointed to the same SCVMM cluster will cause agent comm issues resulting in provisioning failures.

WORKING PATH Path for Morpheus to write to, for example `c:\cloud`

DISK PATH Path for Virtual Disks, for example `c:\virtualdisks`

HIDE HOST SELECTION FROM USERS Prevents host selection from appearing in provisioning wizards

INVENTORY EXISTING INSTANCES Enable for Morpheus to automatically discover existing VMs in the scoped resources

ENABLE HYPERVISOR CONSOLE Enable to use VNC Hypervisor Console for Morpheus console connection as opposed to the default SSH and RDP console connection methods. Requires resolution of all Hyper-V host names and access over port 443 from the Morpheus appliance to Hyper-V hosts.

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM's if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT INSTALL MODE

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud Init / Unattend (when available): (DEFAULT) Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

4. After clicking *NEXT*, the new Cloud can be added to a Group or configured with additional advanced options.

Softlayer

Add a Softlayer Cloud

Cloud Configuration

NAME Name of the Cloud in Morpheus

CODE Unique code used for api/cli, automation and policies.

LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT If Visibility is set to Private, select the Tenant the Cloud resources will assigned to.

ENABLED When disabled, automatic Cloud sync is paused and the Cloud will not be selectable for provisioning.

AUTOMATICALLY POWER ON VMS When enabled, Morpheus will maintain the expected power state of managed VMs. Morpheus will power on any managed VMs in the Cloud that have been shut down for unknown reasons (not powered off by Morpheus) to ensure availability of services.

Note: When “AUTOMATICALLY POWER ON VMS” is enabled, the power state of managed VMs should be maintained in Morpheus. This setting is not applicable to discovered/unmanaged resources.

Details

Username Softlayer Username

API Key Softlayer User API Key, accessible in the Softlayer Portal under `Account > Users > View API Key`

Datacenter Datacenters will auto-populate upon successful authentication with the above credentials. Select appropriate Datacenter for this Cloud.

Object Store Select the destination Object Store

Inventory Existing Instances If enabled, existing Softlayer Instances will be inventoried and appear as unmanaged Virtual Machines in Morpheus .

The Cloud can now be added to a Group or configured with additional Advanced options.

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM's if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT INSTALL MODE

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud Init / Unattend (when available): (DEFAULT) Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

UCS Manager

Overview

The Morpheus UCS Manager Integration enables UCS M B and C Chassis Inventory, VM and Container Host Bare Metal Provisioning, PXE boot with IPMI, Storage Profile, SAN Connection Profile, Server Pool, BIOS Profile, Boot Profile, Maintenance Profile, UUID Pool and Disk Group Profile sync.

Adding UCS Manager Cloud

1. Navigate to Infrastructure > Clouds
2. Select + *ADD*
3. Select **UCS MANAGER** from the Clouds list
4. Populate the following:

Cloud Configuration

NAME Name of the Cloud in Morpheus

CODE Unique code used for api/cli, automation and policies.

LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT If Visibility is set to Private, select the Tenant the Cloud resources will assigned to.

ENABLED When disabled, automatic Cloud sync is paused and the Cloud will not be selectable for provisioning.

AUTOMATICALLY POWER ON VMS When enabled, Morpheus will maintain the expected power state of managed VMs. Morpheus will power on any managed VMs in the Cloud that have been shut down for unknown reasons (not powered off by Morpheus) to ensure availability of services.

Note: When “AUTOMATICALLY POWER ON VMS” is enabled, the power state of managed VMs should be maintained in Morpheus. This setting is not applicable to discovered/unmanaged resources.

Details

UCS MANAGER IP or hostname of UCS Manager

USERNAME UCS Manager User

PASSWORD UCS Manager Password

ORGANIZATION

- EXISTING (select)
- **NEW (create)**

– **ORG NAME** Enter name for the new Organization

SERVER PREFIX String provisioned servers will be prefixed with

DATA DISK MODE

- LVM data disk
- Single Disk

DATA VOLUME Defaults to /dev/sdb * Check to enable SOFTWARE RAID

NET INTERFACE Defaults to eth0

5. Select *NEXT*
6. Select an existing or create a new Group to add the Cloud to. The Cloud can be added to additional Groups in a Groups *Clouds* tab.
7. Select *NEXT*
8. Review and then Select *COMPLETE*

UpCloud

Overview

UpCloud is a cloud hosting provider that offers both Linux and Windows virtual machines on their MAXIOPS infrastructure which is billed as I.A.A.S (infrastructure-as-a-service). They have datacenters based in the UK, USA, Germany, Netherlands, Singapore and Finland. Servers can be created a lightning fast 45 seconds with their faster than SSD technology.

Features

- Virtual Machine Provisioning
- Containers
- Backups / Snapshots
- Migrations
- Auto Scaling
- Load Balancing
- Remote Console
- Periodic Synchronization
- Lifecycle Management and Resize
- Inventory
- Cloudinit

Requirements

An UpCloud User with API, Server and Storage permissions is required.

To enable API access for a Main Account UpCloud User:

1. Login to UpCloud
2. Select **My Account > User Accounts**
3. Select *Change* on the target user
4. Check the box for *API connections: Allow API connections from*
5. Under **Access Permissions > Allow access to individual servers**, check the box for *User has control access to all servers*.
6. Under **Access Permissions > Allow control access to individual storages**, check the box for *User has control access to all storages*
7. Save

To Enable API, API, Server and Storage permissions for a SubAccount User:

When creating or editing a Sub Account UpCloud user:

1. Check the box for *API connections: Allow API connections from*
2. Under **Access Permissions > Allow access to individual servers**, check the box for *User has control access to all servers*.
3. Under **Access Permissions > Allow control access to individual storages**, check the box for *User has control access to all storages*
4. Save

Adding an UpCloud Cloud

Configure

1. Navigate to **Infrastructure > Clouds**
2. Select **+ Create Cloud Button**
3. Select UpCloud from the Add Cloud modal
4. Select *NEXT*
5. Enter the following:

Cloud Configuration

NAME Name of the Cloud in Morpheus

CODE Unique code used for api/cli, automation and policies.

LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT If Visibility is set to Private, select the Tenant the Cloud resources will assigned to.

ENABLED When disabled, automatic Cloud sync is paused and the Cloud will not be selectable for provisioning.

AUTOMATICALLY POWER ON VMS When enabled, Morpheus will maintain the expected power state of managed VMs. Morpheus will power on any managed VMs in the Cloud that have been shut down for unknown reasons (not powered off by Morpheus) to ensure availability of services.

Note: When “AUTOMATICALLY POWER ON VMS” is enabled, the power state of managed VMs should be maintained in Morpheus. This setting is not applicable to discovered/unmanaged resources.

Details

USERNAME UpCloud User Account Username

PASSWORD UpCloud User Account Password

ZONE Select UpCloud Datacenter to scope cloud to

INVENTORY

- *Off*: Existing UpCloud Servers will not be inventoried in Morpheus
- *Basic*: Existing Servers are Inventoried with Power state, Memory and Cores statistics synced.
- *Full*: Existing Servers are Inventoried with Power state, Memory and Cores statistics, plus IP Addresses, Storage Info, and Console VNC Information.

Note: Full Inventory level recommended. Basic Inventory level can reduce Cloud Sync times when inventorying Datacenters with large amounts of servers. Credentials need to be added by editing the Virtual Machine in order to connect.

The Cloud can now be added to a Group or configured with additional Advanced options.

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM's if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT INSTALL MODE

- **SSH / WINRM:** Morpheus will use SSH or WINRM for Agent install.
- **Cloud Init / Unattend (when available): (DEFAULT)** Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

Group

A Group must be specified or created for the new Cloud to be added to. Clouds can be added to additional Groups or removed from Groups after being created.

- *USE EXISTING*: Add the new Cloud to an existing Group in Morpheus .
- *CREATE NEW*: Creates a new Group in Morpheus and adds the Cloud to the Group.

Review

Confirm all settings are correct and select *COMPLETE*.

The UpCloud Cloud will be added, and Morpheus will perform the initial cloud sync of:

- UpCloud Servers will be added as Virtual Machines (if Inventory is enabled)
- UpCloud Templates (My Templates) will sync and be added to Library > Virtual Images.

Note: The Console tab will only appear for Inventoried Servers if Inventory Level is set to *Full*

Provisioning to UpCloud

Instances and Apps can be created using the private Images synced from UpCloud or from the Morpheus provided Image Catalog.

Provision a synced Image

Images synced from UpCloud can be provisioned by using:

- The *UPCLOUD* Instance Type and selecting the Image from the Image dropdown in the configure section when provisioning an Instance, App, or creating an App Blueprint.
- Creating custom Library Instance Types and selecting a synced Image when creating a Node Type for the custom Instance Type.

Important: Synced images should be configured prior to provisioning by editing the Image in the Library > Virtual Images section.

Provision a Morpheus provided UpCloud Image

Morpheus provides a number of pre-configured Images that are available in the default Morpheus Catalog when provisioning an Instance, App, or creating an App Blueprint. UpCloud Images are included in the following Instance Types in the default Morpheus catalog.

- ACTIVEMQ
- APACHE
- CASSANDRA
- DEBIAN

- ELASTICSEARCH
- GRAILS
- JAVA
- MONGO
- MYSQL
- NGINX
- PHP
- RABBITMQ
- REDIS
- OMCAT
- UBUNTU
- WINDOWS
- GRAILS

vCloud Director

Features

- Virtual Machine Provisioning
- Backups / Snapshots
- Datastores
- Brownfield VM management and migration
- Periodic Synchronization
- Lifecycle Management and Resize
- IP Pool Support
- Multi-NIC Interfaces
- Kubernetes and Docker
- Proxy Support
- Image Builder
- Monthly estimated pricing and usage tracker

Configuration

Add vCD Cloud From *Infrastructure* > *Clouds*

1. Navigate to *Infrastructure* > *Clouds*
2. Select + *ADD*
3. Select **VCLLOUD DIRECTOR** from the Clouds list
4. Select *NEXT*
5. Populate the following:

Cloud Configuration

NAME Name of the Cloud in Morpheus

CODE Unique code used for api/cli, automation and policies.

LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT If Visibility is set to Private, select the Tenant the Cloud resources will assigned to.

ENABLED When disabled, automatic Cloud sync is paused and the Cloud will not be selectable for provisioning.

AUTOMATICALLY POWER ON VMS When enabled, Morpheus will maintain the expected power state of managed VMs. Morpheus will power on any managed VMs in the Cloud that have been shut down for unknown reasons (not powered off by Morpheus) to ensure availability of services.

Note: When “AUTOMATICALLY POWER ON VMS” is enabled, the power state of managed VMs should be maintained in Morpheus. This setting is not applicable to discovered/unmanaged resources.

Details

API URL

vCloud Director API Url Example: `https://org.vcd.company.com`

USERNAME vCD Organization Administrator or System Administrator User

- User must have an Organizational Administrator or System Administrator Role
- Username must be in the format of <name>@<org>
- When using a user with the System Administrator role, give the username in the format of <username>@system. Additionally, ensure this user has permission set correctly, such as to view objects created by the organization administrator if needed. Otherwise, things like catalogs and vApps created by the Organization Administrator might not be visible to Morpheus
- In some cases, it may not be advisable to use the system administrator user. This is because some environments will have API access turned off for the system administrator for security reasons as the user would be able to remove key pieces of infrastructure. If your system administrator user does have API access and you understand the risks, you can authenticate Morpheus with this user

PASSWORD Password for the user indicated above

ORGANIZATION Select Organization. Dropdown populates upon successful authorization.

VDC Select VDC. Dropdown populates upon successful authorization.

API VERSION Example. 31.0 Note: Full version required. 31 would not be valid, must be xx.x (vCD API versions are strings)

CATALOG Optionally select a vCD catalog to store Morpheus artifacts or use the default “morpheus_auto” catalog

Inventory Existing Instances If enabled, existing Virtual Machines will be inventoried and appear as unmanaged Virtual Machines in Morpheus .

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM's if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT INSTALL MODE

- **SSH / WINRM:** Morpheus will use SSH or WINRM for Agent install.
- **Cloud Init / Unattend (when available):** (DEFAULT) Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

6. Select *NEXT*
7. Select an existing or create a new Group to add the Cloud to. The Cloud can be added to additional Groups in a Groups *Clouds* tab.
8. Select *NEXT*
9. Review and then Select *COMPLETE*

How to create vCloud Director templates for Morpheus

To create a Windows Template

Create a new machine in VMware vCenter and install a base version of your preferred Windows build.

1. Apply any service packs / updates to the operating system.
2. Set the Network location to Private the below PowerShell will set the location.

```
Get-NetConnectionProfile | Set-NetconnectionProfile -NetworkCategory private
```

3. Configure WinRM to allow remote management and open the firewall.
 - To do this, under local computer Administrator, open a command prompt and run `winrm quickconfig`
4. Install VMware tools
5. Install .Net at least 4.5
6. Enable remote PowerShell this can be done in PowerShell.

```
Enable-PSremoting
```

7. Shutdown the virtual machine and convert to a template.

Note: Do not run sysprep

To create a Linux Centos template

Create a new machine in VMware vCenter and install a base version of your preferred Linux distro build. If you are using cloud init as part of your image you will need to ensure your virtual machine has a cdrom.

1. Before installing the operating system setup a single `ext` or `xf`s partition without a swap disk (This is so that growpart can extend the disk. growpart currently does not support lvm)
2. Install the distro and apply any updates to the operating system and security updates
3. Install cloud-init using command `yum install cloud-init`
4. Install cloud-utils-growpart using command `yum install cloud-utils-growpart`
5. Install vmware tools
6. Install git by running `yum install git`
7. `epel-release`
8. `selinux` set to permissive (enforced can cause problems with cloud-init)

To create a Linux Ubuntu template

Create a new machine in VMware vCenter and install a base version of your preferred Linux distro build. If you are using cloud init as part of your image you will need to ensure your virtual machine has a cdrom.

1. Before installing the operating system setup a single `ext` partition without a swap disk (This is so that growpart can extend the disk. growpart currently does not support lvm)
2. Install the distro and apply any updates to the operating system and security updates
3. Ensure you have set a root password
4. Install cloud-init by running `sudo apt install cloud-init`
5. Install cloud-utils-growpart `sudo apt install cloud-utils`
6. Install desired hypervisor drivers (Virto, Open-VM Tools)
7. Install git by running `sudo apt install git`
8. As Debian 9 includes network manager ensure this is disabled. Change the below file

```
/etc/NetworkManager/NetworkManager.conf
```

to the following:

```
managed=false
```

We also recommend disabling network manager and setting the network adapter to eth0 rather than the automatically assigned name. See a more detailed guide on VMware image prep [here](#).

To import your template into vCloud director you will need to login as either an administrator or organisation administrator.

Once logged into vCloud director you will then need select **Manage Organizations** and then select your organization.

From within the organisation click on **Catalogues** > select an existing catalogue or create a new catalogue.

Note: Please note once you connect Morpheus to your vCD environment, it will create a catalogue called Auto Morpheus. This is a working catalogue and is ignored by Morpheus when searching for images, so any images in the catalogue will not be synced into Morpheus

Open the catalogue and select the import template from vCenter and then browse the data stores for your templates. Select your template and the type in a new name and description then check the copy template into vCloud director.

Once you click ok the import process will begin. When the import has completed the template will appear in Morpheus within **Library > Virtual Images**

If the image does not appear within the virtual images you may need to use the filters to filter the virtual images by the vmware (vmdk / ovf / ova) type.

You may also need to refresh the cloud. To do this go to **Infrastructure > Clouds** > select the vCloud Director cloud > select Refresh.

VMware vCenter

Overview

VMware is a very common cloud integration choice supported by Morpheus . They have provided a top notch virtualization solution and one might argue pioneered the virtualization space altogether. As such, many companies utilize this technology and all the features that come with it, so Morpheus covers a broad feature set in vCenter.

Features

- Virtual Machine Provisioning
- Backups / Snapshots
- Resource Groups
- Datastores and DRS Clusters
- Distributed Switches
- Datacenter / Cluster scoping
- Brownfield VM management and migration
- VMware to VMware migrations
- VMDK/OVF image conversion support
- Hypervisor Remote Console
- Periodic Synchronization

- Veeam Backup Integration
- Lifecycle Management and Resize
- Metadata tag sync

On top of all these features, Morpheus also adds additional features to VMware that do not exist out of the box to make it easier to manage in multitenant environments as well as hybrid cloud environments:

- Cloud-Init Support
- VHD to VMDK Image Conversion
- QCOW2 to VMDK Image Conversion
- Multitenancy resource allocation
- Virtual Image management (Blueprints)
- Auto-scaling and recovery

Getting Started

To get started with VMware, simply start by adding a Cloud in the **Infrastructure > Clouds** section.

To start adding a VMware cloud there will be some things you will need:

vCenter API Url Typically this is the url to the vCenter web client with a /sdk in the path

Username/Password A set of credentials with high level access to VMware (ensure the account has Datacenter level access)

Once these fields are entered, some selections will start pre-populating. A cloud integration is scoped to a specific data center, and can optionally be scoped down to a single cluster or even a single resource pool. If the drop downs do not populate, please verify the api url is resolvable, morpheus has access to vCenter on 443, and the provided credentials are correct and the user has sufficient permissions.

Another cool feature provided with the cloud integration is optional *Resource Pool* scoping. One can choose to allow the cloud to provision into All Resource Pools or a singular Resource Pool. When choosing *All*, these Resource Pools can be managed from a sub-account and visibility perspective via the Cloud Detail page (multi-tenancy).

The VMware cloud integration provides a few additional options including allowing users to make host selections or keeping that aspect hidden such that the best host is automatically chosen for the requested provision.

The *RPC Mode* feature can be configured to allow Morpheus to install its agent on the Guest operating system via either SSH/WinRM or VMware Tools Guest Process feature. The VMware tools Guest Execution API can be tricky so it is recommended to use SSH/WinRM if possible. However, if it is not possible for the Appliance to have outbound access to all networks in which VMs are being provisioned to the SSH/WinRM ports (22, 5985 respectively) then Guest Execution is the only option.

The *Use VNC* console option on the VMware cloud requires special configuration on each ESXI host but allowed hypervisor level remote console support. (See the Advanced Section for details)

When following this add cloud wizard an option will be presented to create a group or add to an existing group. These groups can be given provisioning permission via role based access control. It is normally recommended that groups are organized such that one cloud exists in one group unless the networks are setup such that internal routing is possible between the clouds. This is very useful for bursting, or hybrid cloud configurations.

Windows Provisioning Tips

By default when provisioning windows templates, Morpheus performs guest customizations which initiates a sysprep. This resets the Administrator user and password. Morpheus will set the Administrator password from Administration > Settings > Provisioning > Windows Settings > Password.

Users can also set the username on an image as Administrator and enter a different password if unique passwords are required per image.

Guest customizations are required when assigning static IP's manually or using IP pools. They can be disabled per virtual image advanced settings under Library > Virtual Images > Edit Image > Advanced > Uncheck "Force Guest Customization" if using DHCP. However the SID will not be changed from the source template. In addition, new VM's will not be able to join a domain that had already been joined by the source template or any other VM's with that SID.

Existing Instances

Morpheus provides several features regarding pulling in existing virtual machines and servers in an environment. Most cloud options contain a checkbox titled '*Inventory Existing Instances*'. When this option is selected, all VMs found within the specified scope of the cloud integration will be scanned periodically and Virtual Machines will be synced into Morpheus. Users may also choose to onboard only virtual machines that are running within specific Resource Pools. Once the vCenter Cloud is integrated, navigate to the detail page for the specific Cloud (select it from the list at Infrastructure > Clouds). From the Resources tab, locate the Pools section. Click ACTIONS > Edit next to a selected Resource Pool. If INVENTORY is checked, Morpheus will automatically onboard virtual machines from that Resource Pool.

By default these virtual machines are considered 'unmanaged' and do not appear in the Provisioning > Instances area but rather Infrastructure > Compute > Virtual Machines. However, a few features are provided with regards to unmanaged instances. They can be assigned to various accounts if using a multitenant master account, however it may be best suited to instead assign the 'Resource Pool' to an account and optionally move all servers with regards to that pool (more on this later).

A server can also be made into a managed server. During this process remote access is requested and an agent install is performed on the guest operating system. This allows for guest operations regarding log acquisition and stats. If the agent install fails, a server will still be marked as managed and an Instance will be created in *Provisioning*, however certain features will not function. This includes stats collection and logs.

Note: All Cloud data is resynchronized on a 5 minute interval. This includes Datastores, Resource Pools, Networks, Blueprints, and Virtual Machines.

Service Plans

A default set of Service Plans are created in Morpheus for the VMware provisioning engine. These Service Plans can be considered akin to AWS Flavors or Openstack Flavors. They provide a means to set predefined tiers on memory, storage, cores, and cpu. Price tables can also be applied to these so estimated cost per virtual machine can be tracked as well as pricing for customers. By default, these options are fixed sizes but can be configured for dynamic sizing. A service plan can be configured to allow a custom user entry for memory, storage, or cpu. To configure this, simply edit an existing Service Plan tied to VMware or create a new one. These all can be easily managed from the Admin > Plans & Pricing section.

Virtual Images / Blueprints

Morpheus will automatically take an inventory of all blueprints configured in vCenter and present them as options during provisioning. However, in order for Morpheus to properly provision these virtual machines and provide accurate stats and health of these virtual machines, an agent must be installed during virtual machine startup. This means remote access needs to be granted at the guest operating system level to Morpheus. To properly configure these virtual images, find the relevant images in Library > Virtual Images and edit the entry. On this form, a few options are presented. The first is a check box asking whether or not cloud-init is enabled. If cloud-init is enabled, simply provide the default OS username configured (for Ubuntu the username is *ubuntu* and for CentOS the username is *centos*). For those looking to add cloud-init to existing blueprints Morpheus requires no special configuration and can use the default *cloud.cfg* settings.

A global cloud-init username/password can also be configured per account as well as a keypair via the Admin->Provisioning settings section. The great benefit of utilizing cloud-init is default blueprints do not need common credential sets thereby increasing provisioning security.

Windows systems do not typically support cloud-init. So simply turn this checkbox off and provide the *Administrator* credentials. It should be noted that these credentials are encrypted in the database. If using WinRM for the RPC Mode instead of VMware tools, a Local or Domain Administrator account credential set can be provided instead.

Snapshots

Morpheus allows the ability to create a snapshot of a VM in VMware vCenter. From the instance detail page, simply select **Actions > Create Snapshot** to begin creation of a new Snapshot. Existing snapshots can be viewed in the **BACKUPS** tab on the instance detail page. Snapshots taken in vCenter will sync into Morpheus every five minutes. To revert to a previous snapshot, click on the revert icon located on the right side of the Snapshot. Snapshots can be deleted by clicking on the trash can icon.

Note: Access to Snapshots can be limited or removed entirely for specific user roles as needed. To edit a role's Snapshots permissions, go to Administration > Roles > (Your selected role) > Snapshots. Users can be given Full, Read-only, or No access.

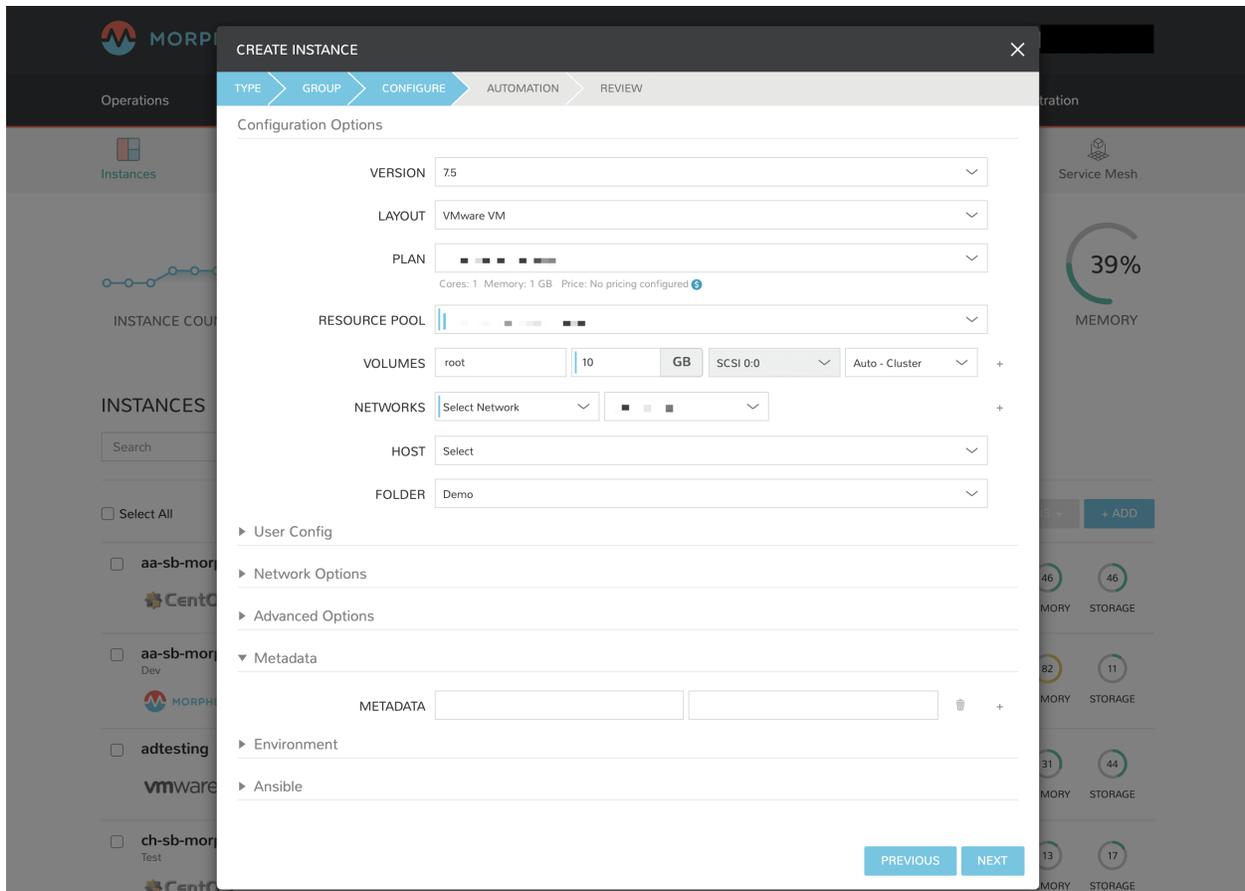
Tagging and Metadata

As of Morpheus version 4.1.0, tagging support is included for vCenter in addition to the other clouds that have already supported it in past versions. Tags will sync to vCenter from Morpheus and existing tags are also inventoried from vCenter into Morpheus.

Note: This feature requires a minimum API version of vCenter 6.5. The API version can be edited by navigating to 'Infrastructure > Clouds' and clicking the edit (pencil) button in the row for the relevant cloud. The field is labeled 'VERSION'.

Tags can be created on-demand when provisioning from the 'CONFIGURE' tab of the 'CREATE INSTANCE' wizard (Provisioning > Instances). Within the 'Metadata' drawer, you will see sets of fields to enter key/value pairs. On creation of the instance, this metadata will be synced into vCenter.

'Inputs' from your library can also be exported as metadata for use with vCenter. When adding or editing a new Input (Library > Options > Inputs), simply mark the box labeled 'EXPORT AS METADATA'. The 'FIELD NAME' becomes the tag category in VMWare.



Docker

So far this document has covered how to add the VMware cloud integration and has enabled users the ability to provision virtual machine based instances via the *Add Instance* catalog in *Provisioning*. Another great feature provided by Morpheus out of the box is the ability to use Docker containers and even support multiple containers per Docker host. To do this a Docker Host must first be provisioned into VMware (multiple are needed when dealing with horizontal scaling scenarios).

To provision a Docker Host simply navigate to the Clusters tab of the Cloud detail page or Infrastructure > Clusters section. From there, click **+ ADD CLUSTER** to add a VMware Docker Host. This host will show up in the Hosts tab next to other ESXi servers that were inventoried by the VMware cloud integration. Morpheus views a Docker host just like any other Hypervisor with the caveat being that it is used for running containerized images instead of virtualized ones. Once a Docker Host is successfully provisioned a green checkmark will appear to the right of the host marking it as available for use. In the event of a failure click into the relevant host that failed and an error explaining the failure will be displayed in red at the top.

Some common error scenarios include network connectivity. For a Docker Host to function properly, it must be able to resolve the Morpheus appliance url which can be configured in Administration > Settings. If it is unable to resolve and negotiate with the appliance than the agent installation will fail and provisioning instructions will not be able to be issued to the host.

Multitenancy

A very common scenario for Managed Service Providers is the need to provide access to VMware resources on a customer by customer basis. With VMware several administrative features have been added to ensure customer resources are properly scoped and isolated. For VMware it is possible to assign specific *Networks*, *Datastores*, and *Resource Pools* to customer accounts or even set the public visibility of certain resources, therefore allowing all sub accounts access to the resource.

Advanced

There are several advanced features provided within Morpheus that can leverage some cool aspects of VMware. One of these features is Remote Console support directly to the hypervisor. To enable this feature a few prerequisites must be met. First, the Morpheus appliance must have network access to the ESXi hosts within vCenter. Secondly, firewall settings need to be adjusted on each ESXi host. This can be done in vSphere under firewall configuration on the host. Simply check the *gdbserver* option, which will open up the necessary ports (starting at 5900 range).

Important: Hypervisor Console for vCenter 6.5 requires Morpheus v3.2.0+

Now that the ESXi hosts are ready to utilize remote console, simply edit the cloud in Morpheus via **Infrastructure > Clouds**. Check the option that says *Enable Hypervisor Console*. It is important to note that currently this functionality only works for newly provisioned vm's provisioned directly via Morpheus. This should change soon however.

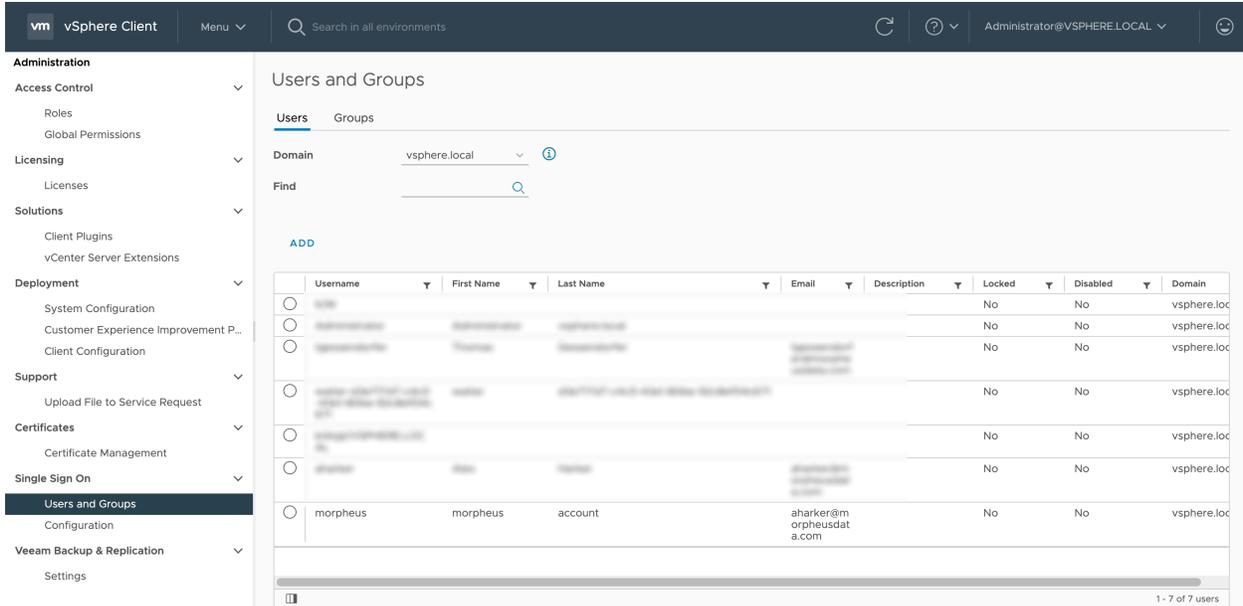
It is also possible to import vm snapshots for backup or conversion purposes from vCenter and also an ESXi host. However, this does require that the ESXi host license has an enterprise level license as it will not allow the appliance to download a virtual image if it is not a paid VMware license.

VMware Permissions

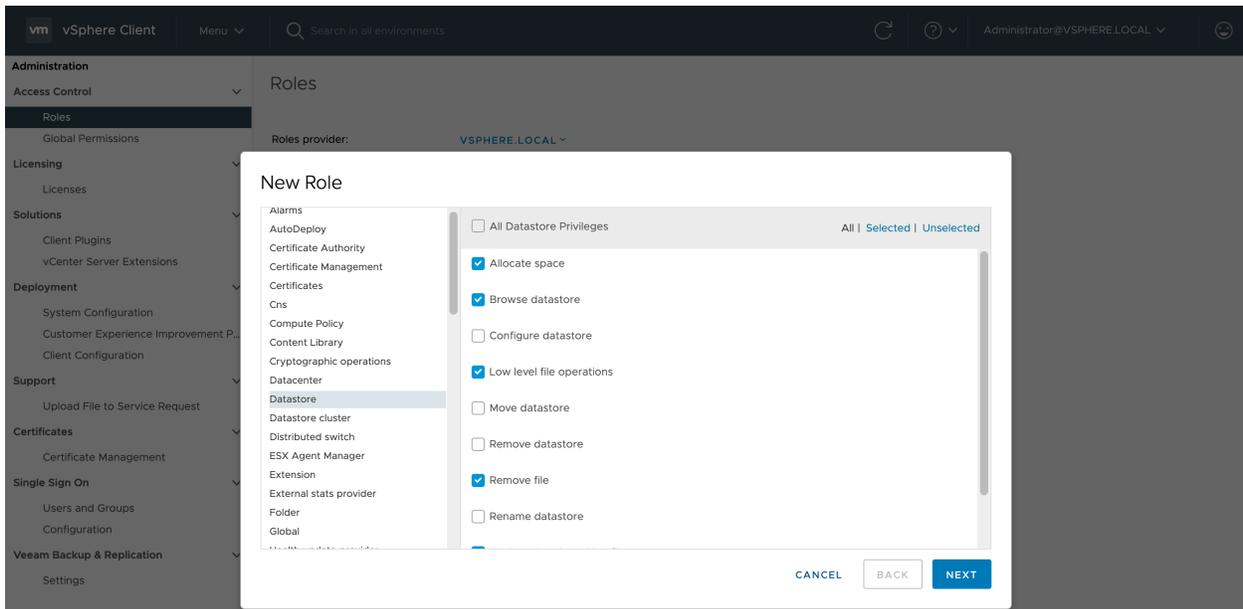
When integrating VMware vCenter with Morpheus, users must supply credentials for a vCenter account and Morpheus will only have access privileges equal to the integrated account. Many users will choose to use a vCenter administrator account so that Morpheus can freely do any function in vCenter without worrying about hitting access limits. Others, for security reasons, may want to restrict Morpheus only to the minimum permissions it needs to perform its functions. Follow the guide in this section to configure a user with minimal permissions and associate it with the appropriate usage levels before using it to create a Morpheus Cloud integration.

Create vCenter Users and Roles

For this example, I've added a new local user to be my Morpheus integration user (Menu > Administration > Users and Groups) but any existing user, whether locally-created or sourced from an identity integration (like Active Directory), works fine.



The next step is to create a Role (Menu > Administration > Roles). You can edit an existing Role to be sure it has the correct privileges, I've opted to create a new role and assign the correct privileges. Below the screenshot, take note of the complete set of required privileges. Once all privileges are set, name the Role (if it's a new one) and click Finish.



Privileges

Content Library

- All Content Library privileges

Datastore/Datastore Cluster

- Allocate Space
- Browse Datastore
- Low Level file Operations
- Remove File
- Update virtual machine files
- Update virtual machine metadata

Distributed Switch

- Port configuration operation
- Port setting operation

Global

- Log Event
- Manage custom attributes
- Set custom attribute

Network

- Assign Network
- Configure
- Remove

Resource

- Apply recommendation
- Assign vApp to resource pool
- Assign virtual machine to resource pool
- Migrate powered off virtual machine
- Migrate powered on virtual machine

Scheduled task

- Create tasks
- Modify task
- Remove task
- Run task

Tasks

- Create task
- Update task

Virtual Machine

- Configuration (all)
- Guest Operations (all)
- Interaction (all)
- Inventory (all)
- Provisioning (all)
- Service configuration (all)
- Snapshot management (all)
- vSphere Replication (all)

vApp

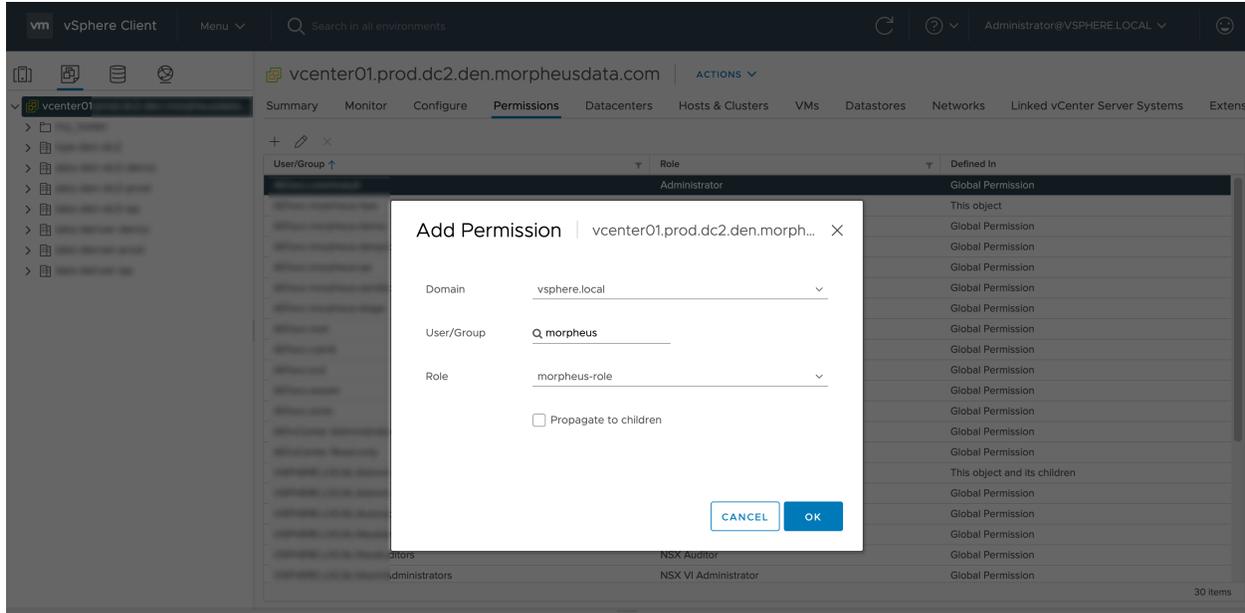
- Clone
- Export
- Import

vSphere Tagging

- Assign or Unassign vSphere Tag
- Create vSphere Tag
- Create vSphere Tag Category
- Delete vSphere Tag
- Delete vSphere Tag Category
- Edit vSphere Tag
- Edit vSphere Tag Category
- Modify UsedBy Field For Category
- Modify UsedBy Field For Tag
- `privilege.InventoryService.Tagging.CreateScope.label`
- `privilege.InventoryService.Tagging.DeleteScope.label`

With the User and Role created, add permissions to associate the User and Role to the appropriate usage constructs. Navigate to the usage construct you wish to work with, navigate to the permissions tab, click the plus (+) button. In the screenshot below, I'm adding the permission for the vCenter usage construct. The complete list of usages and whether or not to mark the propagation box is below the image.

Note: For organization and security purposes, permissions can also be added to folders. This allows Morpheus to see the folders and onboard any resources within them (if desired). Once the vCenter Cloud integration has been created in Morpheus, you can view folders from the Cloud Detail Page (Infrastructure > Clouds > Selected Cloud > Resources Tab). By editing the folder here (Actions > Edit), folders can be set as the "Default" and/or the "Image Target". When a folder is set as Default, this folder is pre-selected when provisioning new Instances into the Cloud. When a folder is set as the Image Target, Morpheus will look into this folder to onboard VMware images into Morpheus.



Usage

vCenter

- Non-Propagating

Datacenter

- Non-Propagating

Cluster

- Non-Propagating

Host

- Non-Propagating

Datastore/Datastore Cluster

- Propagating

After completing the above steps, all VMware Cloud functionality should be available in Morpheus without running into permissions errors.

Creating a Morpheus VMware Image

Morpheus comes out of the box with a default set of blueprints for use in many modern deployment scenarios. These consist mostly of base operating system images with a few additional adjustments. These adjustments typically include the addition of cloud-init (which is highly recommended to be used in most environments, but not mandatory). However, in many on-premise deployments there are custom image requirements as well as networking requirements. This guide will go over how to create a VMware Images for use within Morpheus.

Note: A Morpheus appliance may have many vCenter Clouds tied to any number of vCenter appliances. If the same images need to be available to multiple vCenter Clouds, you will need to download the OVF from one vCenter and upload it into the others. At that point you can make multiple Morpheus Node Types from the images and it

will be available to all needed vCenter Clouds. This is a vCenter limitation but one which may not be obvious when provisioning via Morpheus.

Creating a Windows Image

Supported Versions

2008R2, 2012, 2012R2, 2016, 2019, 2022

Image Preparation

Create a new machine in VMware vCenter and install a base version of your preferred Windows build. The smaller the VMDK drive, typically the faster you can clone and deploy. Utilizing Morpheus, provisioning and post deploy scripts can expand drives to desired sizing.

1. Ensure VMware Tools is installed on the operating system.
2. Apply any service packs / updates to the operating system.
3. Configure WinRM to allow remote management and open the firewall. This is optional if using VMware Tools RPC mode for agent install and Morpheus Agent for guest exec. To enable this, under local computer Administrator, open a command prompt and run

```
winrm quickconfig
```

4. Install .Net at least 4.5.2
5. Ensure Windows Firewall will allow WinRM connections.
6. Shutdown the virtual machine and convert to a template.

Note: WinRM is not required and is used as a fallback when using vmttools guest exec and customizations

Note: Morpheus will sysprep images based on the “Force Guest Customizations” flag under the Virtual Image’s settings when using DHCP. Ensure a sysprep has not been performed on the template if this flag is enabled or if using Static IPs/IP Pools when provisioning, which will always use Guest Customizations and trigger a sysprep.

Creating a CentOS/RHEL 7 Image

Create a new virtual machine in VMware vCenter and install a base version of your preferred Linux distro build. If you are using cloud init as part of your image you will need to ensure your virtual machine has a cdrom.

1. Before installing the operating system setup a single ext or xfs partition without a swap disk (This is so that growpart can extend the disk. growpart currently does not support lvm)
2. Install the distro and apply any updates to the operating system and security updates
3. Install cloud-init using command `yum install cloud-init`
4. Install cloud-utils-growpart using command `yum install cloud-utils-growpart`
5. Install open-vm-tools using command `yum install open-vm-tools`

6. Install git by running `yum install git`
7. Install epel-release repo using command `yum install epel-release`
8. selinux set to permissive (enforced can cause problems with cloud-init) `sudo vi /etc/selinux/config`

Cloud-Init

To get started with a base CentOS image we first install cloud-init. This is a relatively simple process using yum:

```
yum -y install epel-release
yum -y install git wget ntp curl cloud-init dracut-modules-growroot
rpm -qa kernel | sed 's/^kernel-//' | xargs -I {} dracut -f /boot/initramfs-{}.img {}
```

There are two parts to this yum installation. We are first ensuring some core dependencies are installed for automation as well as cloud-init. git for example is installed for use by ansible playbook automation down the line and is therefore optional if not using ansible. The dracut-modules-growroot is responsible for resizing the root partition upon first boot to match the virtual disk size that was potentially adjusted during provisioning.

A great benefit to using cloud-init is credentials don't have to be locked into the blueprint. It is advisable, within Morpheus, to configure the default cloud-init user that gets created when the vm boots automatically by cloud-init. This is located in Administration > Settings > Provisioning, within the Cloud-Init Settings section.

Network Interfaces

A slightly annoying change with CentOS 7 is that the network interfaces have changed naming convention. You may notice when running `ifconfig` that the primary network interface is set to something like `ens2344` or some other random number. This naming is dynamic typically by hardware id and we don't want this to fluctuate when provisioning the blueprint in various VMware environments. Fortunately, there is a way to turn this functionality off and restore the interface back to `eth0`.

Firstly we need to adjust our bootloader to disable interface naming like this.

```
sed -i -e 's/quiet/quiet net.ifnames=0 biosdevname=0/' /etc/default/grub
grub2-mkconfig -o /boot/grub2/grub.cfg
```

The above command adds a few arguments to the kernel args list (namely `net.ifnames=0` and `biosdevname=0`). It may be useful to view the `/etc/default/grub` file and ensure these settings were indeed applied.

The next step is to adjust the network-scripts in CentOS. we need to ensure we have a file called `/etc/sysconfig/network-scripts/ifcfg-eth0`

Below is a script that we run on our packer builds to prepare the machines network configuration files.

```
export iface_file=$(basename "$(find /etc/sysconfig/network-scripts/ -name 'ifcfg*' -not_
↪-name 'ifcfg-lo' | head -n 1)")
export iface_name=${iface_file:6}
echo $iface_file
echo $iface_name
sudo mv /etc/sysconfig/network-scripts/$iface_file /etc/sysconfig/network-scripts/ifcfg-
↪eth0
sudo sed -i -e "s/$iface_name/eth0/" /etc/sysconfig/network-scripts/ifcfg-eth0
sudo bash -c 'echo NM_CONTROLLED="no" >> /etc/sysconfig/network-scripts/ifcfg-eth0'
```

This script tries to ensure there is a new `ifcfg-eth0` config created to replace the old `ens` config file. Please do verify this config exists after running. If it does not you will have to be sure to build one on your own.

```
TYPE=Ethernet
DEVICE=eth0
NAME=eth0
ONBOOT=yes
NM_CONTROLLED="no"
BOOTPROTO="dhcp"
DEFROUTE=yes
```

Creating a CentOS/RHEL 8 Image

Create a new virtual machine in VMware vCenter and install a base version of your preferred Linux build. You must be running ESXi 6.7 Update 2 or later.

Prepare The New CentOS 8/RHEL8 Image

1. Install epel-release: `yum -y install epel-release` (This step is not necessary for RHEL)
2. Install git, wget, curl, cloud-init, cloud-utils-growpart, and open-vm-tools: `yum -y install git wget curl cloud-init cloud-utils-growpart open-vm-tools`
3. Update: `yum -y update`
4. Finally run: `rpm -qa kernel | sed 's/^kernel-//' | xargs -I {} dracut -f /boot/initramfs-{}.img {}`

SELinux Settings

If allowed by your internal IT policies, set SELinux to permissive to avoid potential issues with cloud-init down the road.

1. Edit the following: `vi /etc/selinux/config`
2. Make the following change: `setenforce 0`

Network Interfaces

Run the following to rename the network NIC. Values inside angle brackets should be filled in with the appropriate value for your environment (ex. <varname>):

1. `sed -i -e 's/quiet/quiet net.ifnames=0 biosdevname=0/' /etc/default/grub`
2. `grub2-mkconfig -o /boot/grub2/grub.cfg` (location may be different, could be located at /boot/efi/EFI/centos/grub.cfg)
3. `ifdown <orginal-nic>`
4. `mv /etc/sysconfig/network-scripts/<orginal-nic> /etc/sysconfig/network-scripts/ifcfg-eth0` (this changes name/device to eth0)
5. Edit `ifcfg-eth0` and change the NAME to `eth0`
6. `bash -c 'echo NM_CONTROLLED="no" >> /etc/sysconfig/network-scripts/ifcfg-eth0'`
7. `ip link set <orginal-nic> down`

8. `ip link set <original-nic> name eth0`
9. `ip link set eth0 up`
10. `ifup eth0`

Final VMWare Tasks

1. Detach any install media
2. Shutdown the VM
3. Convert the VM to template on the Morpheus side
4. Refresh the Morpheus Cloud to allow the new template to sync

Creating an Ubuntu 20.04 Image

Download the Ubuntu 20.04 ISO from Canonical, and upload the base image to vCenter. Then, create a new virtual machine in vCenter.

Note: Since we'll include cloud-init with our image, we will need to ensure the virtual machine has a cdrom. Select the Ubuntu 20.04 ISO we just downloaded from the CD/DVD drive dropdown menu when creating the new virtual machine.

Before installing the operating system, set up a single ext partition without a swap disk. Then, continue on installing Ubuntu making the following selections during the setup process:

- Update to the latest installer if a later version is available
- Use the entire disk and deselect the option to set up the disk as an LVM group
- Configure an account and set a password
- Opt to install OpenSSH Server
- Other optional packages aren't needed for this basic Ubuntu image

Complete the installation process and reboot the machine. Update the package list and apply any upgrades:

```
apt-get update
apt-get upgrade
```

Change the network interface to `eth0` by editing `/etc/default/grub`. The line `GRUB_CMDLINE_LINUX=""` should be edited to `GRUB_CMDLINE_LINUX="net.ifnames=0 biosdevname=0"`.

Update GRUB:

```
update-grub
```

Update the `70-persistent-net.rules` file:

```
cat << EOF > /etc/udev/rules.d/70-persistent-net.rules
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{dev_id}=="0x0", ATTR{type}=="1",
↳NAME="eth0"
EOF
```

Remove `subiquity-disable-cloudinit-networking.cfg` as cloud-init will skip network configuration if it's present:

```
rm -f /etc/cloud/cloud.cfg.d/subiquity-disable-cloudinit-networking.cfg
```

Update `99-pve.cfg`:

```
cat << EOF > /etc/cloud/cloud.cfg.d/99-pve.cfg
datasource_list: [ConfigDrive, NoCloud]
EOF
```

Remove Netplan files, they will not be regenerated if they exist:

```
rm -f /etc/netplan/00-installer-config.yaml
rm -f /etc/netplan/50-cloud-init.yaml
```

Run cloud-init clean:

```
cloud-init clean
```

Next, reboot the system and confirm the network interface is labeled `eth0` once the machine comes back up. Then, clear BASH history for root. The history entry has a copy in the memory and it will flush back to the file when you log out. You can avoid this with the following command:

```
cat /dev/null > ~/.bash_history && history -c && exit
```

Shutdown the system:

```
shutdown -h now
```

Convert the VM to a template in vCenter before moving back to Morpheus to onboard the image and use it to begin building your provisioning library.

Gotchas

SELinux can cause issues with cloud-init when in enforced mode. It may be advisable to set this to permissive unless it is mandatory within your organization to use an enforced SELinux configuration. If that is the case please see the documentation for the `cloud_init_t` security policies.

Network Manager will also prevent the required restart of the Network Service when assigning static IP's. Disable Network Manager when possible or Static IP assignment may not work until the Network Service is restarted manually.

A Note on Proxies

Proxy configurations are known to vary in some organizations and makes building a base blueprint a little more difficult. In order to fully configure proxies a few environment variables must be set in the `/etc/environment` file (This can be done automatically in a default user-data script for cloud-init as well in edit cloud).

```
http_proxy="http://myproxyaddress:8080"
https_proxy="http://myproxyaddress:8080"
ftp_proxy="http://myproxyaddress:8080"
no_proxy=127.0.0.1,localhost,applianceUrl
https_no_proxy=127.0.0.1,localhost,applianceUrl
```

Important: It is very important to properly set the `no_proxy` list (`applianceUrl`) should be replaced with the actual appliance url. In future releases, morpheus plans to automatically take care of this.

Note: If using cloud-init agent install mode these settings need to be set in the custom Cloud-Init User data section of “Edit Cloud” or “Edit Virtual Image”

Important: If using this virtual machine as a docker host, proxy settings must also be configured in the docker config. See Docker guides for instructions on how to properly set this. If necessary this can be wrapped in a task automation workflow for your own use.

VMware Fusion

Add a VMware Fusion Cloud

1. Navigate to **Infrastructure > Clouds**
2. Select + *CREATE CLOUD*, select VMware Fusion, and then click *Next*.
3. Enter the following into the Create Cloud modal:

Cloud Configuration

NAME Name of the Cloud in Morpheus

CODE Unique code used for api/cli, automation and policies.

LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT If Visibility is set to Private, select the Tenant the Cloud resources will assigned to.

ENABLED When disabled, automatic Cloud sync is paused and the Cloud will not be selectable for provisioning.

AUTOMATICALLY POWER ON VMS When enabled, Morpheus will maintain the expected power state of managed VMs. Morpheus will power on any managed VMs in the Cloud that have been shut down for unknown reasons (not powered off by Morpheus) to ensure availability of services.

Note: When “AUTOMATICALLY POWER ON VMS” is enabled, the power state of managed VMs should be maintained in Morpheus. This setting is not applicable to discovered/unmanaged resources.

Details

VMWARE FUSION HOST IP or URL of VMware Fusion Host

WORKING PATH Existing folder Morpheus will write to on Host

USERNAME Host Username

PASSWORD Host Password

BRIDGE NAME Will auto-populate upon successful authentication with the Fusion Host (E.X. ‘EN0: ETHERNET’)

4. The Cloud can now be added to a Group or configured with additional Advanced options.

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM’s if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider’s costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT INSTALL MODE

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.

- **Cloud Init / Unattend** (when available): (DEFAULT) Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

Xen Server

Add a Xen Server Cloud

1. Navigate to **Infrastructure > Clouds**
2. Select **+ CREATE CLOUD**, select Xen, and then click *Next*.
3. Enter the following into the Create Cloud modal:

Cloud Configuration

NAME Name of the Cloud in Morpheus

CODE Unique code used for api/cli, automation and policies.

LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY For setting cloud permissions in a multi-tenant environment. Not applicable in single tenant environments.

TENANT If Visibility is set to Private, select the Tenant the Cloud resources will assigned to.

ENABLED When disabled, automatic Cloud sync is paused and the Cloud will not be selectable for provisioning.

AUTOMATICALLY POWER ON VMS When enabled, Morpheus will maintain the expected power state of managed VMs. Morpheus will power on any managed VMs in the Cloud that have been shut down for unknown reasons (not powered off by Morpheus) to ensure availability of services.

Note: When “AUTOMATICALLY POWER ON VMS” is enabled, the power state of managed VMs should be maintained in Morpheus. This setting is not applicable to discovered/unmanaged resources.

Details

API URL IP or URL of Xen Host. ex: *xenserver.domain.com*

CUSTOM PORT Port for non standard xen server clouds

USERNAME Xen Host Username

PASSWORD Xen Host Password

Inventory Existing Instances If enabled, existing Virtual Machines will be inventoried and appear as unmanaged Virtual Machines in Morpheus .

4. The Cloud can now be added to a Group or configured with additional Advanced options.

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.

SCALE PRIORITY Only affects Docker Provisioning. Specifies the priority with which an instance will scale into the cloud. A lower priority number means this cloud integration will take scale precedence over other cloud integrations in the group.

APPLIANCE URL Alternate Appliance url for scenarios when the default Appliance URL (configured in *admin > settings*) is not reachable or resolvable for Instances provisioned in this cloud. The Appliance URL is used for Agent install and reporting.

TIME ZONE Configures the time zone on provisioned VM's if necessary.

DATACENTER ID Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE Unmanaged or select a Network Integration (NSX, ACI etc)

LOCAL FIREWALL On or Off. Enable to managed Host and VM firewall/IP Table rules (linux only)

SECURITY SERVER Security Server setting is for Security Service Integrations such as ACI

TRUST PROVIDER Select Internal (Morpheus) or an existing Trust Provider Integration

STORAGE MODE Single Disk, LVM or Clustered

BACKUP PROVIDER Select a backup provider. Depending on the Cloud type and any currently-configured backup plugins you may select Internal Backups (Morpheus) or another configured backup solution

REPLICATION PROVIDER Sets the default Replication Provider for the Cloud. Select an existing Replication Provider Integration

GUIDANCE Enable Guidance recommendations on cloud resources.

COSTING Enable for Morpheus to sync Costing data from the Cloud provider, when available. For on-prem Clouds, enabling costing activates a costing service designed to mirror the live costing experience of public clouds, including invoicing with line items and real-time cost data (Operations > Costing > Invoices). If your organization utilizes reserved instances and you want to pull in related pricing data, some Cloud integrations include the option to select *Costing and Reservations*. If this is not relevant, select *Costing* to save money on additional calls to the Cloud provider's costing API.

DNS INTEGRATION Records for instances provisioned in this cloud will be added to selected DNS integration.

SERVICE REGISTRY Services for instances provisioned in this cloud will be added to selected Service Registry integration.

CONFIG MANAGEMENT Select a Chef, Salt, Ansible or Puppet integration to be used with this Cloud.

CMDB Select CMDB Integration to automatically update selected CMDB.

CMDB DISCOVERY When checked, any automatically discovered (unmanaged) servers onboarded into Morpheus from this Cloud will also have CMDB records created for them.

CHANGE MANAGEMENT Select an existing Change Management Integration to set on the Cloud. ex: Cherwell

AGENT INSTALL MODE

- **SSH / WINRM:** Morpheus will use SSH or WINRM for Agent install.
- **Cloud Init / Unattend (when available): (DEFAULT)** Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fall back on SSH or WINRM if cloud-init is not installed on the provisioned image. Morpheus will also add Agent installation to Windows unattend.xml data when performing Guest Customizations or utilizing syspreped images.

API PROXY Set a proxy for outbound communication from the Morpheus Appliance to the Cloud endpoints. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

INSTALL AGENT Enable to have Agent Installation on by default for all provisioning into this Cloud. Disable for Agent Installation to be off by default for all provisioning into this Cloud.

Provisioning Options

PROXY Set a proxy for inbound communication from Instances to the Morpheus Appliance. Proxies can be added in the *Infrastructure > Networks > Proxies* tab.

Bypass Proxy for Appliance URL Enable to bypass proxy settings (if added) for Morpheus Agent communication to the Appliance URL.

NO PROXY Include a list of IP addresses or name servers to exclude from proxy traversal

USER DATA (LINUX) Add cloud-init user data. Morpheus 4.1.0 and earlier assumes bash syntax. Morpheus 4.1.1 and later supports all User Data formats. Refer to <https://cloudinit.readthedocs.io/en/latest/topics/format.html> for more information.

5.13.18 Containers

Docker Registry

Overview

Without any additional configuration Morpheus can provision images from Docker's public hub at <https://hub.docker.com/> using their public api at <https://index.docker.io/v1/>

However, many organizations maintain private Docker registries for security measures. Additional public and private Docker registries can be added to Morpheus.

Adding a Docker Registry Integration

1. Navigate to Administration > Integrations
2. Click "New Integration"
3. Select the *Docker Repository* Type
4. Add the following:

Name Name for the Registry in Morpheus

Repository url Docker Registry url or IP address

Username Username if private registry

Password Password if private registry

5. Save Changes

Note: You must either have signed certificates for your registry or configure your docker host(s) to accept insecure registries

Provisioning an Instance from Docker Registry

Docker images from the Integrated Registry can be provisioned using the generic *Docker* Instance Type, or by adding images to Node Types for custom Library Instance Types.

5.13.19 Deployment

Git

Git Repository integrations allow integration of public or private Git repositories in Github, GitLab or other services. Use your existing code and any future code to build automation Tasks and Workflows, onboard Spec Templates for Terraform and Kubernetes to build App configurations, and more. Each time code from your repositories is invoked, Morpheus will pull the current live version (depending on configuration) so any recent changes to the code are used.

Creating an Integration

New Git integrations are created either in the global integrations section (Administration > Integrations) or in the code integrations section (Provisioning > Code > Integrations). You will need an access code for authentication with private repositories, public repositories can be integrated simply with a URL.

1. Navigate to the global integrations section (Administration > Integrations) or the code integrations section (Provisioning > Code > Integrations)
2. Click *+ADD*
3. Click Git repository
4. Configure the following:
 - **NAME:** A friendly name for the Git repository integration in Morpheus
 - **ENABLED:** When marked, code in this repository will be available for creating automation or other Morpheus constructs
 - **DEFAULT BRANCH:** The default repository branch from which code should be sourced
 - **USERNAME:** For private repositories, enter an account name (such as a Github username)
 - **PASSWORD:** For Github and GitLab, password authentication is no longer supported **but access tokens should go in this field.**
 - **ACCESS TOKEN:** Currently an unused field, access tokens should go in the Password field
 - **KEY PAIR:** (Optional) Select a stored SSH keypair for Github SSH authentication
 - **ENABLE GIT REPOSITORY CACHING:** When unmarked, Morpheus retrieves code fresh from the repository each time it's invoked. When marked, Morpheus will use a cached version of the code if it's less than five minutes old. In general, this should be left unmarked unless you are experiencing performance issues related to very large amounts of code being invoked many times during a deployment

5. Click *SAVE CHANGES*

Github

Integrate your Github account with Morpheus and have access to all your public and private repositories within the platform. Use your existing code and any future code to build automation Tasks and Workflows, onboard Spec Templates for Terraform and Kubernetes to build App configurations, and more. Integrated Github repositories also populate the Code list page (Provisioning > Code) where repositories can be browsed and individual files can be viewed. Repositories can be configured as [import](#) and [export](#) targets from Code detail pages as well. Each time code from your repositories is invoked in Tasks, Workflows, Spec Templates, and more, Morpheus will pull the current live version (depending on configuration) so any recent changes to the code are used.

Generating a Personal Access Token (Classic)

As of August 13, 2021, due to policy changes on the Github platform, Morpheus requires an access token to authenticate with Github. Password authentication is no longer possible. Morpheus accepts both classic personal access tokens and fine-grained tokens (described in the next section). To create a new access token:

1. Log in to Github
2. Click on your account profile avatar in the upper-right portion of the window
3. Click Settings
4. From the left nav, click on Developer Settings
5. Once again from the left nav, click Personal access tokens
6. Click “Tokens (classic)”
7. Click Generate new token
8. Click “Generate new token (classic)”
9. Give the new token at least access to everything in the “repo” scope
10. Click Generate token
11. Copy the access token and save it for the next step when the integration is created

Tip: We recommend using a long-lived token, Github even includes the option to have no expiration on your tokens (for classic personal access tokens only, not fine-grained tokens). Should the token expire, any Morpheus automation, App Blueprints, and Terraform or Kubernetes Spec Templates which are based on code contained in your repositories will no longer work. This could lead to failed provisioning and time spent troubleshooting to isolate the issue until the Github integration is refreshed with a new access token.

Generating a Fine-Grained Token

As of August 13, 2021, due to policy changes on the Github platform, Morpheus requires an access token to authenticate with Github. Password authentication is no longer possible. Morpheus accepts both classic personal access tokens and fine-grained tokens. To create a new fine-grained access token:

1. Log in to Github
2. Click on your account profile avatar in the upper-right portion of the window
3. Click Settings
4. From the left nav, click on Developer Settings

5. Once again from the left nav, click Personal access tokens
6. Click “Fine-grained tokens”
7. Click Generate new token
8. Give the token access to all repositories unless you will only need to work with specific repositories within Morpheus
9. Click Generate token
10. Copy the access token and save it for the next step when the integration is created

Tip: Fine-grained tokens can be created with an expiration date of up to one year. Should the token expire, any Morpheus automation, App Blueprints, and Terraform or Kubernetes Spec Templates which are based on code contained in your repositories will no longer work. This could lead to failed provisioning and time spent troubleshooting to isolate the issue until the Github integration is refreshed with a new access token.

Creating an Integration

New Github integrations are created either in the global integrations section (Administration > Integrations) or in the code integrations section (Provisioning > Code > Integrations). You will need a Github access token to complete this step, see the prior section for instructions on obtaining a Github access code.

1. Navigate to the global integrations section (Administration > Integrations) or the code integrations section (Provisioning > Code > Integrations)
2. Click *+ADD*
3. Click Github
4. Configure the following:
 - **NAME:** A friendly name for the Github integration in Morpheus
 - **ENABLED:** When marked, this Github integration is active and any code repositories are available for building Morpheus automation and other constructs within the platform
 - **USERNAME:** The username for your Github account
 - **PASSWORD:** Copy your access token here rather than in the “ACCESS TOKEN” field. See the “Important” box below for additional details
 - **ACCESS TOKEN:** Do not use this field. See the “Important” box below for additional details
 - **KEY PAIR:** (Optional) Select a stored SSH keypair for Github SSH authentication
 - **ENABLE GIT REPOSITORY CACHING:** When unmarked, Morpheus retrieves code fresh from the repository each time it’s invoked. When marked, Morpheus will use a cached version of the code if it’s less than five minutes old. In general, this should be left unmarked unless you are experiencing performance issues related to very large amounts of code being invoked many times during a deployment
5. Click *SAVE CHANGES*

Important: Your access token should be pasted into the PASSWORD field and not the ACCESS TOKEN field. If the ACCESS TOKEN field is used, the repository will create successfully and many features will work. However, in Code detail pages (Provisioning > Code > Selected code detail page), you will not be able to browse files in private Github

repositories unless the access token is pasted into the PASSWORD field. New Github integrations should be created by pasting the access token in the PASSWORD field and the ACCESS TOKEN field should be ignored.

Note: In certain cases, it can take several seconds for the integration process to complete and the ADD INTEGRATION modal to be dismissed.

Viewing an Integrated Github Account

When authentication is successful, click into the new Github integration from the list of available integrations. The Organizations tab will list each organization the Github account is associated with. The Repositories tab lists all public and private repositories which are associated with the account. We can also click in to each repository to view its files and folders, as well as create specific types of automation or Spec Templates from the files directly in this view.

The screenshot shows the Morpheus Administration interface. At the top, there is a navigation bar with the Morpheus logo, a search bar, and user information for 'Alex Harker'. Below the navigation bar is a menu with categories: Operations, Provisioning, Library, Infrastructure, Backups, Monitoring, Tools, and Administration. Under Administration, there are sub-menus: Tenants, Plans & Pricing, Roles, Users, Integrations, Policies, Health, and Settings. The 'Integrations' sub-menu is selected, showing 'Integrations > Harker GitHub'. The main content area displays the 'Harker GitHub' integration with a green checkmark, the GitHub logo, and buttons for 'EDIT' and 'DELETE'. Below this, it shows 'Host: Last Update: 10/03/2023 10:21 AM'. There are two tabs: 'Organizations' and 'Repositories', with 'Repositories' being the active tab. Under the 'Repositories' tab, there is a search bar and a 'Select' dropdown. Below the search bar is a table of repositories.

NAME	ORG	LOCATION	PRIVATE	FORKED	DEFAULT BRANCH
[blurred]	[blurred]	[blurred]	No	No	master
[blurred]	[blurred]	[blurred]	No	No	master
[blurred]	[blurred]	[blurred]	No	No	master
[blurred]	[blurred]	[blurred]	No	No	master
[blurred]	[blurred]	[blurred]	No	No	master
[blurred]	[blurred]	[blurred]	No	No	master
[blurred]	[blurred]	[blurred]	No	No	master
[blurred]	[blurred]	[blurred]	No	No	main
[blurred]	[blurred]	[blurred]	No	No	master
[blurred]	[blurred]	[blurred]	No	No	master
[blurred]	[blurred]	[blurred]	No	No	main
[blurred]	[blurred]	[blurred]	Yes	No	master

5.13.20 DNS

AWS Route53

Overview

Morpheus integrates directly with Amazon Route 53 to automatically create DNS entries for Instances provisioned to a configured Cloud or Group. Morpheus also syncs in Route 53 Domains for easy selection while provisioning, or setting as the default Domain on a Cloud or Network.

Add Route 53 Integration

Route 53 can be added in the *Administration* or *Infrastructure* sections:

1. In Administration > Integrations, select + *New Integration*
2. In Infrastructure > Networks > Services, select *Add Service*
3. Provide the following:

TYPE Route 53

NAME Name for the Integration in Morpheus

REGION AWS Region for the Integration

ACCESS KEY AWS User IAM Access Key

SECRET KEY AWS User IAM Secret Key

4. Once saved the Integration will be added and visible in both Administration > Integrations and Infrastructure > Networks > Services

Note: All fields can be edited after saving.

Domains

Once the integration is added, Route 53 Domains will sync and listed under Infrastructure > Networks > Domains.

Note: Default Domains can be set on Networks and Clouds, and can be selected when provisioning. Additional configuration options are available by editing a domain in *Networks > Domains*

Configuring Route 53 with Clouds and Groups

DNS Integrations are available in the *DNS Integration* dropdown in Cloud and Group settings.

Morpheus will register Instances with the DNS provider when provisioned into a Cloud or Group with a DNS Integration added.

Add DNS Integration to a Cloud

1. In **Infrastructure** > **Clouds** edit the target Cloud.
2. Expand the *Advanced Options* section.
3. In the *DNS Integration* dropdown, select an available DNS Integration.
4. Save Changes

Add DNS Integration to a Group

1. In **Infrastructure** > **Groups** select the target Group.
2. Select the *Edit* button for the Group
3. Expand the *Advanced Options* section.
4. In the *DNS Integration* dropdown, select an available DNS Integration.
5. Save Changes

Note: Instances provisioned into a Cloud or Group with a DNS Integration added will be registered as instance-name.domain with the DNS Provider during provisioning, and de-registered at teardown.

Microsoft DNS

Overview

Morpheus integrates directly with Microsoft DNS to automatically create DNS entries for Instances provisioned to a configured Cloud or Group. Morpheus also syncs in Microsoft DNS Domains for easy selection while provisioning, or setting as the default Domain on a Cloud or Network.

Prepare DNS Server(s)

Note: This section will assume a the DNS server is in an Active Directory environment and joined to the domain. The process may be different for other configurations.

The easiest method to prepare DNS server(s) is to use a service account that is added to the `DnsAdmins` and `Remote Management Users` groups, either in Active Directory (if DNS is on domain controllers) or the local groups of a member server. The `DnsAdmins` group will provide permissions for the service account to make DNS changes, such as creating/deleting A and PTR records. The `Remote Management Users` group will allow Morpheus to connect to the server(s) via WinRM.

Additionally, ensure firewall rules have been updated if needed to allow WinRM through. In some cases, the default WinRM rules allow `Private` and `Domain` networks but not `Public`. Enable `Public` if the network Morpheus is connected is considered `Public`, or disable the firewall if permitted. If a jump box is required (discussed below), then ensure the firewall is configured to allow the jump box to connect to the DNS server instead.

Finally, `winrm quickconfig` may need to be run to enable WinRM, if the server is an older operating system.

Minimum Permissions

Some organizations may require that users cannot be added to the `DNSAdmin` group, mentioned previously. If this is a requirement, the following process/permissions would be required to ensure Morpheus can connect successfully. This process may be required on each DNS server, depending on the environment. Note if Morpheus adds additional functionality at a later time, these permissions may need to be updated to support those features.

- Run `dnsmgmt.msc`
- Right-click the DNS server object and choose `Properties`
- Add the service account to the user list and ensure the following permissions are applied:
 - Read
 - Create all child objects
 - Delete all child objects
- Run `wmimgmt.msc`
- Right-click `WMI Control (Local)` and choose `Properties`
- Click the `Security` tab
- Set the following permissions for each of the below nodes:
 - `CIMV2`
 - `MicrosoftDNS`
 - `Microsoft => Windows => DNS` (only the DNS node)
- Highlight the node and click the `Security` button
- Click the `Advanced` button
- Click the `Add` button to add the service account to the list
- Ensure the `Applies to` field is set to `This namespace and subnamespaces`
- Set the following permissions:
 - Enable Account
 - Remote Enable
 - Execute Methods
- Finally, restart Windows Management Instrumentation Service on the server. This is required for the change in permissions to take place.

Additional support reference: https://support.morpheusdata.com/s/article/How-to-give-C?language=en_US

(Optional) Prepare Jump Box

In some environments, Morpheus may not be allowed to access the DNS servers directly, as they may be on segregated networks. In this case, Morpheus can utilize a member server as a “jump box” that can access the DNS servers directly, the jump box will be used to interact with the DNS server instead. If this is a requirement, follow the below process to prepare the jump box.

- Add the service account to the Remote Management Users group of the jump box, which will allow WinRM to access
- Verify the firewall allows WinRM from Morpheus
- Create or edit the following registry key by running `regedit`:
 - Navigate to: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Protect\Providers\df9d8cd0-1501-11d1-8c7a-00c04fc297eb`
 - Create or edit ProtectionPolicy DWORD (32-bit) Value
 - Set ProtectionPolicy value to 1
- Finally, `winrm quickconfig` may need to be run to enable WinRM, if the server is an older operating system.

Add Microsoft DNS Integration

Important: The Morpheus Microsoft DNS integration works over `http/5985` by default. If you have turned off the `http` listener on `5985` and only enabled `https/5986`, be sure to configure the correct `WINRM PORT`.

Note: Depending on the version of Morpheus, some settings may only be available by installing the Microsoft DNS plugin from [Morpheus Marketplace](#). Newer versions of Morpheus should contain this plugin by default.

Microsoft DNS can be added in the Administration or Infrastructure sections:

1. In Administration > Integrations, select + *New Integration*
2. In Infrastructure > Networks > Integrations, select + *Add*
3. Provide the following:

TYPE Microsoft DNS

NAME Name for the Integration in Morpheus

WINRM PORT Port WinRM should use. By default, HTTP (port 5985) is used, which is the default on Windows Server. If HTTPS has been configured by the organization, then specifying port 5986 may be appropriate.

DNS SERVER IP or resolvable hostname of DNS server morpheus will connect to. If using a jump box, specify the IP or resolvable hostname of the jump box here, and the main DNS Server in the COMPUTER NAME field below.

USERNAME DNS provider username

PASSWORD DNS provider user password

ZONE FILTER Comma separated filter for specific zones to be imported. Example entries: `example.morpheus.com`, `*.morpheus.com`, `*.10.in-addr.arpa`, `d*.us.morpheus.com`. Additional explanations can be found at the [plugin source code readme](#).

COMPUTER NAME If the DNS SERVER specified is not the main DNS server but rather a jump box, enter the Computer Name of the main DNS Server here. If the DNS SERVER specified above is the main DNS server and not a jump box, leave COMPUTER NAME blank.

CREATE POINTERS Enable to create PTR (Pointer/Reverse Lookup) records during provisioning

4. Once saved the Integration will be added and visible in both Administration > Integrations and Infrastructure > Networks > Services

Note: All fields can be edited after saving.

Domains

Once the integration is added, Microsoft DNS Domains will sync and listed under Infrastructure > Networks > Domains.

Note: Default Domains can be set on Networks and Clouds, and can be selected when provisioning. Additional configuration options are available by editing a domain in Networks > Domains

Configuring Microsoft DNS with Clouds and Groups

DNS Integrations are available in the DNS Integration dropdown in Cloud and Group settings. Morpheus will register Instances with the DNS provider when provisioned into a Cloud or Group with a DNS Integration added.

Add DNS Integration to a Cloud

1. In Infrastructure > Clouds edit the target Cloud.
2. Expand the Advanced Options section.
3. In the DNS Integration dropdown, select an available DNS Integration.
4. Save Changes

Add DNS Integration to a Group

1. In Infrastructure > Groups select the target Group.
2. Select the Edit button for the Group
3. Expand the Advanced Options section.
4. In the DNS Integration dropdown, select an available DNS Integration.
5. Save Changes

Note: Instances provisioned into a Cloud or Group with a DNS Integration added will be registered as instance-name.domain with the DNS Provider during provisioning, and de-registered at teardown.

PowerDNS

Overview

Morpheus integrates directly with PowerDNS to automatically create DNS entries for Instances provisioned to a configured Cloud or Group. Morpheus also syncs in PowerDNS Domains for easy selection while provisioning, or setting as the default Domain on a Cloud or Network.

Add PowerDNS Integration

PowerDNS can be added in the *Administration* or *Infrastructure* sections:

1. In *Administration > Integrations*, select *+ New Integration*
2. In *Infrastructure > Networks > Services*, select *Add Service*
3. Provide the following:
 - TYPE** PowerDNS
 - NAME** Name for the Integration in Morpheus
 - API HOST** URL of PowerDNS API. Example: `http://10.30.20.10:8081`
 - Token** PowerDNS API Token
 - Version** PowerDNS API Version
4. Once saved the Integration will be added and visible in both *Administration > Integrations* and *Infrastructure > Networks > Services*

Note: All fields can be edited after saving.

Domains

Once the integration is added, PowerDNS Domains will sync and listed under *Infrastructure > Networks > Domains*.

Note: Default Domains can be set on Networks and Clouds, and can be selected when provisioning. Additional configuration options are available by editing a domain in *Networks > Domains*

Configuring PowerDNS with Clouds and Groups

DNS Integrations are available in the *DNS Integration* dropdown in Cloud and Group settings.

Morpheus will register Instances with the DNS provider when provisioned into a Cloud or Group with a DNS Integration added.

Add DNS Integration to a Cloud

1. In **Infrastructure** > **Clouds** edit the target Cloud.
2. Expand the *Advanced Options* section.
3. In the *DNS Integration* dropdown, select an available DNS Integration.
4. Save Changes

Add DNS Integration to a Group

1. In **Infrastructure** > **Groups** select the target Group.
2. Select the *Edit* button for the Group
3. Expand the *Advanced Options* section.
4. In the *DNS Integration* dropdown, select an available DNS Integration.
5. Save Changes

Note: Instances provisioned into a Cloud or Group with a DNS Integration added will be registered as instance-name.domain with the DNS Provider during provisioning, and de-registered at teardown.

5.13.21 Identity Management

Active Directory

Overview

Active Directory is Microsoft's primary authentication service. It is widely used in enterprise organizations and even in Microsoft cloud services. While Active Directory also supports LDAP protocol support (which Morpheus can integrate with as well), Morpheus includes a dedicated identity integration type specifically for Active Directory. By integrating Active Directory, Morpheus administrators can fully offload the work of managing the user lifecycle to Active Directory. Creating new users, applying roles to users, updating basic user data, disabling users, and more can be handled in Active Directory and automatically filtered down to Morpheus. This section includes an example integration walkthrough as well as additional details on the integration feature set.

Note: Caution should be used when integrating more than one Active Directory identity source with the same Morpheus Tenant. You must ensure the users on each identity source are unique users or that the two domains use different naming conventions for users.

How It Works

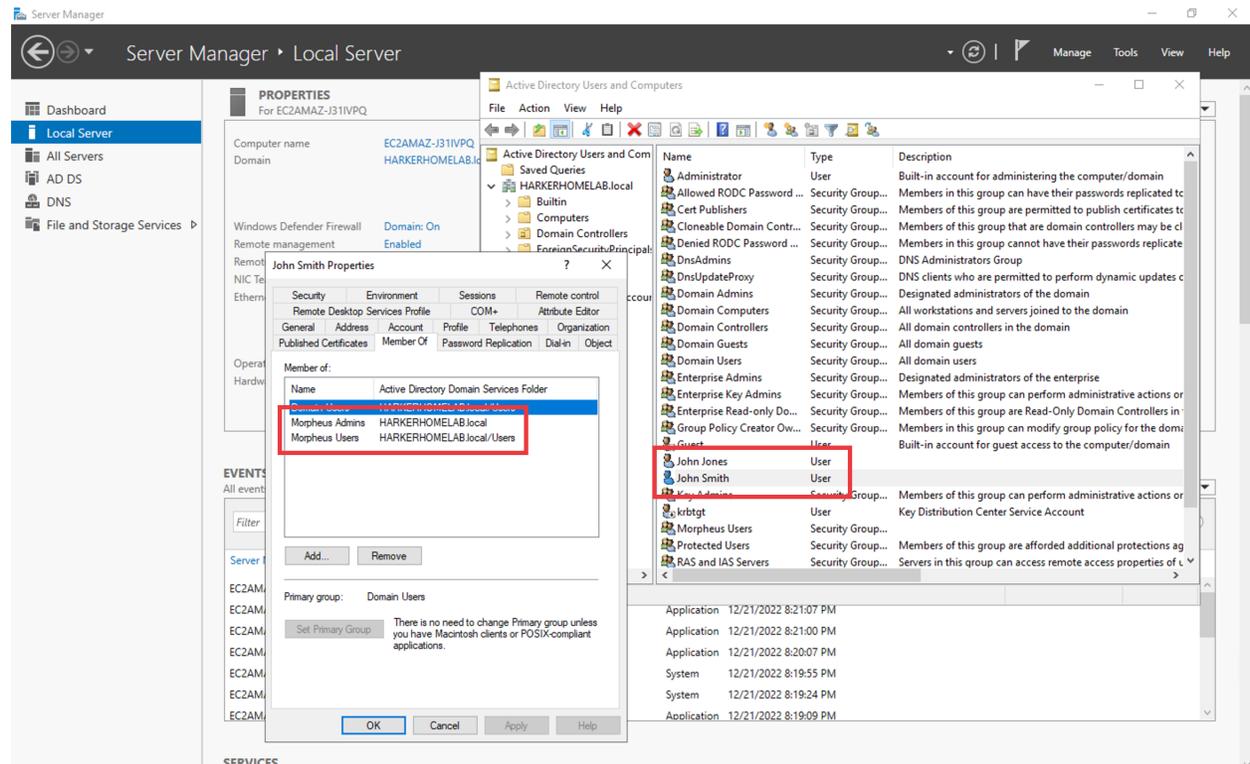
In Morpheus, identity sources (if used) are configured per Tenant. In order to see an identity integration or create a new one, navigate to a Tenant detail page (Administration > Tenants > Selected Tenant) and click *IDENTITY SOURCES*. From this page we can add a new identity source or click the pencil () icon next to an existing identity source to view or edit it. Any configured identity source will apply to just one Tenant and they cannot be shared. One scenario where this is especially useful is in an MSP appliance where each Tenant is a siloed environment for a specific customer. The customer's own existing Active Directory server and groups can be leveraged to build Morpheus user accounts with correct role mapping automatically.

When configuring an Active Directory integration in Morpheus, AD groups are mapped to Morpheus roles. When the user logs in for the first time, Morpheus adds the new user account with correct name, email address, and applies one or more roles depending on configuration. Going forward, Morpheus will sync down any changes to the user, including any role changes based on changes to the user's associated AD groups or updated passwords. Additionally, disabling a user in AD will prevent them from accessing Morpheus.

Important: Morpheus will connect over port 389 for non-secure LDAP and port 636 for secure LDAP. Ensure that Morpheus is able to talk to the AD server on the proper port.

Example Integration

In a simple example, we have an Active Directory server which has two groups relevant to Morpheus, "Morpheus Users" and "Morpheus Admins." We will configure Morpheus so that only users in the "Morpheus Users" group can access Morpheus in any capacity. Users who are also in the "Morpheus Admins" group will take on the System Admin role in Morpheus. We'll see later when the integration is configured how Morpheus roles can be mapped to AD groups. In the same AD server, I have two users. John Smith is in groups "Morpheus Users" and "Morpheus Admins". John Jones is only in the "Morpheus Users" group.



Knowing the AD scheme and the requirements for Morpheus user roles, we can begin the process of creating the integration. Identity integrations are specific to each Tenant so begin by navigating to the Tenant detail page (Administration > Tenants > Selected Tenant) and clicking *IDENTITY SOURCES*. On setting the type to “Active Directory,” the form will update with the needed fields. Note the following basic fields:

- **AD SERVER:** The IP address or hostname for the Active Directory Server
- **DOMAIN:** The AD domain in which the relevant users and groups reside
- **BINDING USERNAME:** A server user which has access to relevant objects on the AD server. In my example, I’ve used the in-built Administrator user which is the easiest option. Other users may be used depending on your organization’s IT security policies but the integration with Morpheus will not work properly if the user does not have the needed access
- **BINDING PASSWORD:** The password for the user in the prior field

With the basic configurations completed, the remaining configurations will affect Morpheus user and role generation. A **REQUIRED GROUP** is optional and is a group the user *must* be in to have any Morpheus access. Here we require that users be in the “Morpheus Users” group. A **DEFAULT ROLE** is required and will be assigned to all users regardless of any additional roles they may be assigned based on their AD group membership. Beyond that, all other Morpheus roles will be listed here and an AD group name can be associated with as many as you required. In this example, we are giving users in the “Morpheus Admins” group the Morpheus System Admin role. Though we did not use them, it’s worth pointing out that **ENABLE ROLE MAPPING PERMISSION** will give administrators in the Tenant the ability to update the AD role mappings (though they will not have access to the core integration fields such as **AD SERVER**, **DOMAIN**, or binding user details). **MANUAL ROLE ASSIGNMENT** allows users to manually update Morpheus roles outside of the automatic mappings created by the AD integration.

EDIT IDENTITY SOURCE
✕

Identity Source

TYPE

ACTIVE

NAME

DESCRIPTION

Active Directory Connection

AD SERVER

DOMAIN

USE SSL Yes No

BINDING USERNAME

BINDING PASSWORD

REQUIRED GROUP

Include Member Groups

Role Mappings

DEFAULT ROLE

AZRA CATALOG USER

MULTI-TENANT ADMIN

SYSTEM ADMIN

TEST MULTITENANT

USER ADMIN

Role Options

ENABLE ROLE MAPPING PERMISSION

MANUAL ROLE ASSIGNMENT

DEACTIVATE
DELETE
SAVE CHANGES

With the above integration steps completed, users can now log into Morpheus and a user account with correct roles will automatically be created. In our example case, John Smith has logged in and we can see he is assigned the default role as well as the System Admin role based on his AD group associations. Going forward, Morpheus will continue to sync any changes to these users. For example, Morpheus roles may be updated based on changing AD groups or user access may be completely revoked by disabling the user in AD.

The screenshot shows the Morpheus Administration interface. The top navigation bar includes 'Operations', 'Provisioning', 'Library', 'Infrastructure', 'Backups', 'Monitoring', 'Tools', and 'Administration'. The 'Administration' section is active, showing 'Users' and 'User Groups'. A search bar is present, along with buttons for 'IDENTITY SOURCES', '+ CREATE USER', and a settings icon. Below the navigation is a table of users:

DISPLAY NAME	USERNAME	EMAIL	ROLE
Alex Harker	aharker	aharker@morpheusdata.com	System Admin
John Smith	jsmith	jsmith@example.com	Azra Catalog User, System Admin

Adding an Active Directory Integration

1. Navigate to Administration > Tenants
2. Select a Tenant
3. Select *IDENTITY SOURCES*
4. Select + *ADD IDENTITY SOURCE*
5. Set the TYPE to “Active Directory”
6. Populate the following:

Name A friendly name in Morpheus for the AD integration

AD Server The Hostname or IP address of AD Server

Domain The AD domain in which the relevant user and group objects reside

USE SSL Indicates whether SSL should be used for communication with the AD server. Morpheus will connect over port 389 for non-secure LDAP and port 636 for secure LDAP, ensure Morpheus can connect to the AD server over the correct port

Binding Username A username for a service account which has access to relevant objects (users, groups, etc.). For ease, the “Administrator” user may be used

Binding Password The password for the above account

Required Group The AD group users must be in to have access (optional, see example in the prior section)

Default Role The default role a user is assigned when they are in the required group or if no specific group mapping applies to the user (see example in prior section)

ENABLE ROLE MAPPING PERMISSION When selected, Tenant users with appropriate rights to view and edit Roles will have the ability to set role mapping for the Identity Source integration. This allows the Tenant user to edit only the role mappings without viewing or potentially editing the basic Identity Source configuration (AD server, domain, binding user details, etc)

MANUAL ROLE ASSIGNMENT When selected, administrators can manually edit Roles for users created through this identity source integration from the user detail page (Administration > Users > Selected user)

Note: For more on Identity Source role mapping permissions, see the [associated guide](#) in our KnowledgeBase.

7. Select *SAVE CHANGES*.

Now allowed AD users can login to Morpheus via their Active Directory credentials and a User will be automatically generated to Morpheus with matching metadata and mapped Role permissions.

Note: Sub-tenant Morpheus API authentication for Active Directory generated users is not currently supported.

Troubleshooting

If you're unable to get the Active Directory integration to work, the following troubleshooting steps may be useful to ensure your appliance can talk to the Active Directory server.

1. Open firewall ports

Source: Morpheus appliance

Destination: AD server's FQDN or IP address

Non-SSL AD integration: TCP-389

SSL AD integration: TCP-636

2. Checking open LDAP connections from the Morpheus appliance

Connect to a Morpheus appliance box and run the following:

```
$ sudo lsof i- | grep :ldap
```

3. Check LDAP connectivity from the Morpheus appliance

Connect to a Morpheus appliance box and run the following. Be sure to replace the placeholder values in the command with the correct values for your environment:

```
$ ldapsearch -x -h xx.xx.xx.xx -D "binding-user@acme.com" -W -b "cn=users,  
↪dc=acme,dc=com"
```

4. Run tcpflow from the Morpheus appliance for non-SSL enabled AD identity Integrations

Use tcpflow from the Morpheus appliance and then start the identity source configuration once again. Keep in mind this will only work for AD servers which are not SSL enabled:

```
$ sudo tcpflow -i any -c -v port 389
```

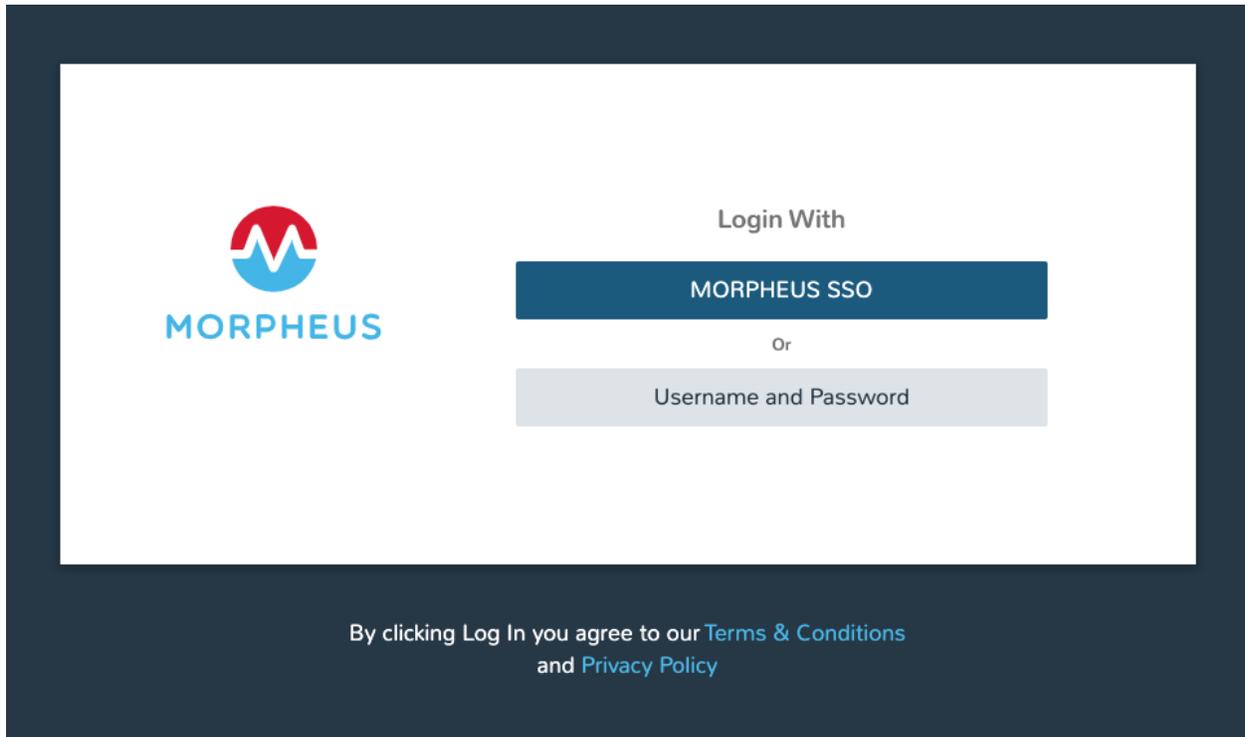
5. Check the AD and domain controllers event logs

Check the event logs for LDAP queries from the Morpheus appliance to ensure network connectivity.

SAML Integration

Overview

The Morpheus SAML identity source integration allows customers to add user SSO to Morpheus, authenticated by external login SAML providers.



Adding a SAML Integration

To add a SAML integration:

1. Navigate to Administration > Tenants
2. Select a tenant.
3. Select *IDENTITY SOURCES* in the Tenant detail page
4. Select + *ADD IDENTITY SOURCE*.
5. Select *SAML SSO* from the *TYPE* field
6. Add a Name and optional Description for the SAML integration

NEW IDENTITY SOURCE ✕

Identity Source

TYPE

NAME

DESCRIPTION

SAML SSO Configuration

LOGIN REDIRECT URL

SAML LOGOUT REDIRECT URL

INCLUDES SAML REQUEST PARAMETER

SAML REQUEST

SAML RESPONSE

Copy

Advanced Validation Options

Assertion Attribute Mappings Show/Hide

Role Mappings

DEFAULT ROLE

ROLE ATTRIBUTE NAME

REQUIRED ROLE ATTRIBUTE VALUE

LITTLE ACCESS

SUPER ADMIN - MULTI-TENANT

SYSTEM ADMIN - MULTI-TENANT

Role Options

ENABLE ROLE MAPPING PERMISSION

MANUAL ROLE ASSIGNMENT

SAVE CHANGES

There are 4 sections with fields that need to be populated depending on the desired configuration:

- SAML Configuration
- Role Mappings
- Role Options
- Assertion Attribute Mappings

SAML Configuration

LOGIN REDIRECT URL This is the SAML endpoint Morpheus will redirect to when a user signs into Morpheus via SAML

SAML LOGOUT REDIRECT URL The URL Morpheus will POST to when a SAML user logs out of Morpheus

INCLUDES SAML REQUEST PARAMETER Yes (recommended) - the AuthN request will be sent via the ?SAMLRequest= parameter in the URL (GET)

No - the AuthN request will be submitted in the body of the request (POST)

Note: The SAML SP documentation should mention which binding to use but GET is most common

SAML REQUEST No Signature - No signature is used on the SAML request

Self Signed - A self-signed X.509 Certificate is generated after clicking *SAVE CHANGES*. This signature value can be used by the SAML SP to verify the authenticity of the request

Custom RSA Signature - Import a custom RSA Private Key and respective X.509 Certificate. This signature value can be used by the SAML SP to verify the authenticity of the request

SAML RESPONSE Do Not Validate Assertion Signature - The SAML response signature from the SAML SP will not be validated

Validate Assertion Signature - The SAML response signature from the SAML SP will be validated. Enter the SAML SP X.509 certificate in the **Public Key** field. This must be in PEM format

Important: Setting SAML REQUEST to “No Signature” and SAML RESPONSE to “Do Not Validate Assertion Signature” is allowed but not recommended for security reasons.

Role Mappings

DEFAULT ROLE Role any SAML user will be assigned by default

ROLE ATTRIBUTE NAME The name of the attribute/assertion field that will map to Morpheus roles, such a MemberOf

REQUIRED ROLE ATTRIBUTE VALUE Attribute/assertion value that a user must be assigned/a member of to be authorized, such as group or role in the SAML SP. This is obtained from the attribute/assertion defined in the ROLE ATTRIBUTE NAME field

<Morpheus ROLE NAME> Additional roles that can be mapped to a user, which will add to the DEFAULT ROLE. Attribute value that a user must be assigned/a member of to be authorized, such as group or role in the SAML SP. This is obtained from the attribute/assertion defined in the ROLE ATTRIBUTE NAME field

Note: For more on Identity Source role mapping permissions, see the [associated guide](#) in our KnowledgeBase.

Role Options

ENABLE ROLE MAPPING PERMISSION When selected, Tenant users with appropriate rights to view and edit Roles will have the ability to set role mapping for the Identity Source integration. This allows the Tenant user to edit only the role mappings without viewing or potentially editing the Identity Source configuration.

MANUAL ROLE ASSIGNMENT When selected, administrators can manually edit Roles for users created through this identity source integration from the user detail page (Administration > Users > Selected user).

Assertion Attribute Mappings

GIVEN NAME ATTRIBUTE NAME SAML SP field value to map to Morpheus user First Name

SURNAME ATTRIBUTE NAME SAML SP field value to map to Morpheus user Last Name

EMAIL ATTRIBUTE SAML SP field value to map to Morpheus user email address

SAML SSO Configuration

LOGIN REDIRECT URL

SAML LOGOUT REDIRECT URL

INCLUDES SAML REQUEST PARAMETER

SAML REQUEST

SAML RESPONSE

Copy

Advanced Validation Options

Assertion Attribute Mappings Show/Hide

GIVEN NAME ATTRIBUTE NAME

SURNAME ATTRIBUTE

Once populated, select *SAVE CHANGES* and the SAML identity source integration will be added.

In the *Identity Sources* section, important information for configuration of the SAML integration is provided. Use the SP ENTITY ID and SP ACS URL for configuration on the external login SAML provider side.

Note: In some cases, the SAML provider may need these values before providing the LOGIN REDIRECT URL and other values. When creating the integration, the NAME and LOGIN REDIRECT URL can contain any values, then selecting SAVE CHANGES will generate the above values. The NAME and LOGIN REDIRECT URL can be edited later, once the SAML configuration is created in the SAML provider.

- ENTITY ID
- SP ACS URL
- LOGIN REDIRECT URL

- SP METADATA

SAML SSO	Okta - SAML	ENTITY ID:	https://[REDACTED]/saml/eDKL60P25
		SP ACS URL:	https://[REDACTED]/externalLogin/callback/eDKL60P25
		LOGIN REDIRECT URL:	https://[REDACTED] [REDACTED]/sso/saml
		SP METADATA:	VIEW

Sample Metadata code output:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><EntityDescriptor entityID=
↪ "https://someip.com/saml/eDKL60P25" xmlns="urn:oasis:names:tc:SAML:2.0:metadata">
↪ <SPSSODescriptor AuthnRequestsSigned="false" WantAssertionsSigned="true"
↪ protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"><NameIDFormat>
↪ urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</NameIDFormat>
↪ <AssertionConsumerService index="0" isDefault="true" Binding=
↪ "urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://someip.com/
↪ externalLogin/callback/eDKL60P25"/></SPSSODescriptor></EntityDescriptor>
```

Note: Different SAML providers will have different field names and requirements. An Okta SAML Dev environment was used for the example integration in this article.

Okta SAML SSO

For Okta SAML integration, the following fields are mapped:

- LOGIN REDIRECT URL : Identity Provider Single Sign-On URL
- ENTITY ID: Audience URI (SP Entity ID)
- SP ACS URL: Single sign on URL

Onelogin SAML SSO

For Onelogin SAML integration, the following fields are mapped:

- LOGIN REDIRECT URL : SAML 2.0 Endpoint (HTTP)
- SAML LOGOUT REDIRECT URL : SLO Endpoint (HTTP)
- SIGNING PUBLIC KEY : X.509 Certificate
- ENTITY ID: ACS (Consumer) URL Validator
- SP ACS URL: ACS (Consumer) URL

Azure Active Directory SSO (SAML)

Azure Active Directory Single Sign-on can be added as a Identity Source in Morpheus using the SAML Identity Source Type. The Azure AD SSO configuration is slightly different than other SAML providers, and this guide will assist in adding a Azure AD SSO Identity Source.

Create Azure Enterprise Application

1. Login to the [Azure Portal](#)
2. Navigate to: **Azure Active Directory > Enterprise Applications**
3. Click the **+ New application** button at the top
4. Click the **+ Create your own application** button at the top
5. Ensure **Integrate any other application you don't find in the gallery (Non-gallery)** is selected and enter a name for the app. Common examples are: **MorpheusSSO**
6. Click the **Create** button at the bottom and wait for it to complete
7. Once created, you'll be in the **Overview** of the application created. Navigate to the **Single sign-on** section from the left pane
8. Choose **SAML** as the Single sign-on method
9. Copy both the **Login URL** and **Logout URL** in Step 4, we'll need these in some of the next steps
10. Before we can continue configuring the application, the configuration needs to be generated in Morpheus for more data

Create an Azure AD SAML Integration in Morpheus

Azure requires inputting the **Identifier (Entity ID)** and **Reply URL (Assertion Consumer Service URL)** in the Azure SSO configuration before it provides the **Endpoints** and **Certificate** necessary to add the Integration into Morpheus. In order to get the **Identifier (Entity ID)** and **Reply URL (Assertion Consumer Service URL)** to input into Azure SSO configuration, we need to create a **Azure AD SAML SSO** integration in Morpheus first.

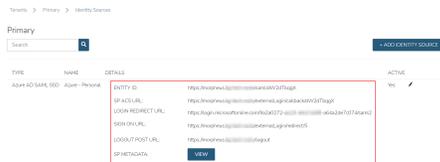
To add the integration:

1. Login to Morpheus
2. Navigate to **Administration > Tenants**
3. Click a tenant hyperlink
4. Click the **IDENTITY SOURCES** button in the Tenant detail page
5. Click the **+ ADD IDENTITY SOURCE** button
6. Select **Azure AD SAML SSO** from the **TYPE** dropdown
7. Add
 - Name
 - (Optional) Description
 - Paste the **Login URL** copied from Azure into the **LOGIN REDIRECT URL** field
 - Paste the **Logout URL** copied from Azure into the **SAML LOGOUT REDIRECT URL** field

8. This is the minimum information needed for now, which will let us generate the details needed from Morpheus. We'll return to this configuration page later to enter more information.
9. Click the *SAVE CHANGES* button

Important: Setting SAML REQUEST to “No Signature” and SAML RESPONSE to “Do Not Validate Assertion Signature” is allowed but not recommended for security reasons.

Upon saving, the *Entity ID* (Identifier (Entity ID)) and *SP ACS URL* (Reply URL (Assertion Consumer Service URL)) will be provide in the Identity Source list view. Copy these for use in Azure SSO configuration.



Configure Azure Enterprise Application

This guide assumes an Azure AD Enterprise Application has already been created. Please refer to documentation above, if this has not already been configured.

1. Navigate to: **Azure Active Directory > Enterprise Applications > Single sign-on**
2. Choose SAML as the Single sign-on method
3. On Step 1 (Basic SAML Configuration), click the Edit button and enter the following:
 - **Identifier (Entity ID)** Enter the Entity ID URL from the Morpheus Identity Source Integration above
 - **Reply URL (Assertion Consumer Service URL)** Enter the SP ACS URL from the Morpheus Identity Source Integration above
 - **Logout URL** Enter the following format: `https://yourUrl/login/` If this is a sub tenant, the format may instead be the following: `https://yourUrl/login/account/1` The login URL can be found under *IDENTITY SOURCES* in the tenant
4. On Step 2 (Attributes and Claims), click the Edit button
5. Click the Add a group claim button at the top
6. Choose All groups and ensure Group ID is selected for the Source attribute dropdown

Note: You can also choose Security groups, which ever makes more sense for the organization

7. Close the pane and return to the Enterprise Application in the Single sign-on section
8. On Step 3 (SAML Certificates), click the Download link next to Certificate (Base64) and Federation Metadata XML

Note: The files will download, keep them available for later configuration in Morpheus

9. Navigate to Users and Groups in the left pane
10. Click the Add user/group button

11. Add Azure groups to this application that will be able to login to Morpheus

Note: Note the object ID for each of these groups, as they will be used later when configuring Morpheus to map the group to roles

12. Once groups have been added, click the Assign button at the bottom

Configure the Azure AD SAML Integration in Morpheus

1. Login to Morpheus using Username and Password, as usual
2. Navigate to Administration > Tenants
3. Click a tenant hyperlink
4. Select *IDENTITY SOURCES* in the Tenant detail page
5. Click the pencil (edit) next to the integration created previously
6. Ensure the SAML REQUEST field is set to Self Signed

Note: A custom RSA signature can be used here if needed, if required by the organization

7. Ensure the SAML RESPONSE field is set to Validate Assertion Signature

Note: With this setting, if the assertion signature ever changes in the Azure Enterprise Application, this would need to be updated to match

8. Edit/view the downloaded Federation Metadata XML (.xml extension) file from the previous section

Note: It is recommended to use Microsoft Edge, or another browser, to view the contents

9. In the Federation Metadata XML file, locate the <X509Certificate> </X509Certificate> under the <Signature> section. Copy the entire contents between the <X509Certificate> and </X509Certificate>, it is very long
10. Paste the value copied from the Federation Metadata XML file into the Public Key (Optional) box, below the SAML RESPONSE dropdown

Configure Role Mappings

Role mappings will map Azure AD Groups to Morpheus Roles. Azure AD users will be assigned Roles in Morpheus upon signing in based on their Group Membership in Azure AD.

Important: Use an Azure Groups Object ID, not Group name, when entering Role Mappings. Example: 7626a4a2-b388-4d9b-a228-72ce9a33bd4b

DEFAULT ROLE Role a Azure AD user will be assigned by default upon signing in to Morpheus using this Identity Source.

REQUIRED AZURE AD GROUP OBJECT ID Object ID of Azure AD Group a user must be a member of to be authorized to sign in to Morpheus. Users not belonging to this Group will not be authorized to login to Morpheus. This field is optional, and if left blank, any user from the Azure AD App will be able to sign in to Morpheus and will be assigned the Default Role if no Role Mappings match AD Group membership.

GROUP ASSERTION ATTRIBUTE NAME Enter `http://schemas.microsoft.com/ws/2008/06/identity/claims/groups` for Azure AD SSO

Additional Role Mappings The existing Roles in Morpheus will be listed. To map a Morpheus Role to an Azure AD Group, enter the Object ID of the desired Azure AD Group in the *Role Attribute Value* field for the corresponding Morpheus Role.

Important: Use an Azure Groups Object ID, not Group name, when entering Role Mappings. Example: 7626a4a2-b388-4d9b-a228-72ce9a33bd4b

ENABLE ROLE MAPPING PERMISSION When selected, Tenant users with appropriate rights to view and edit Roles will have the ability to set role mapping for the Identity Source integration. This allows the Tenant user to edit only the role mappings without viewing or potentially editing the Identity Source configuration.

MANUAL ROLE ASSIGNMENT When selected, administrators can manually edit Roles for users created through this identity source integration from the user detail page (Administration > Users > Selected user).

Note: For more on Identity Source role mapping permissions, see the [associated guide](#) in our KnowledgeBase.

Once populated, select *SAVE CHANGES* and the SAML identity source integration will be added. The Identity Source can be edited anytime to deactivate or change Role Mappings or other values.

Note: If Role mappings are edited after Azure AD SSO users have signed into Morpheus, currently logged in users will need to log out of Morpheus for the new Role mappings to take effect, when applicable.

- Under the Role Azure Group Mappings section, verify the DEFAULT ROLE dropdown has the role in Morpheus selected that all users will be assigned by default
 - It is recommended that this role contains no permissions, which ensures that anyone who authenticates gets no access
- Under the Role Azure Group Mappings section, you will see role names listed. Next to these are text boxes with Assertion Attribute Mappings inside. Enter group object IDs from Azure into these text boxes. This will map the Azure AD groups to specific roles in Morpheus
- Finally, click Save Changes at the bottom of the page

Here is an example of the configuration above:



Azure Group Lookups

When a user in azure ad has more that 150 group attributes, Azure does not include the group claims in the SAML response, and Morpheus is required to query Microsoft Graph to obtain the users group attribute values. When there are users that are members of more that 150 groups, populate the Azure Group Lookups section in order for those users to be able to use the Azure AD SAML SSO integration, otherwise no groups will be obtained and proper role mappings cannot occur.

AZURE TENANT ID Add Azure AD Tenant ID if user group membership will exceed 150. See *Copy Directory (tenant) and Application (client) IDs* for information on obtaining an Azure AD Tenant ID

AZURE APP ID Add Azure AD Application (Client) ID if user group membership will exceed 150. See *Copy Directory (tenant) and Application (client) IDs* for information on obtaining an Azure AD Application (Client) ID

AZURE APP SECRET Add Azure Application (Client) Secret if user group membership will exceed 150. See *Generate a Client Secret* for information on creating an Azure Application (Client) Secret

ROLE LINK ATTRIBUTE NAME default: <http://schemas.microsoft.com/claims/groups.link>. This is not normally changed.

Logging Into Morpheus with Azure AD SAML

1. Navigate to the Morpheus URL
2. A new button will appear to allow sign-in using Azure AD SAML, with the same name as the integration. Click the button .. image:: /images/integration_guides/identity_sources/azure_ad_saml/sign_in_page.png
width 30%
3. Sign-in with your Microsoft/Azure account .. image:: /images/integration_guides/identity_sources/azure_ad_saml/ms_signin.png
width 20%

Note: If no local users other than the System Admin have been created, “USERNAME AND PASSWORD” option will not be displayed, only the SAML option.

OneLogin

Adding OneLogin Identity Source Integration

1. Navigate to Administration > Tenants
2. Select the Tenant to add the Identity Source Integration
3. Select *IDENTITY SOURCES*
4. Select + *IDENTITY SOURCE*
5. Enter the following:
TYPE OneLogin
NAME Name of the Identity Source Integration in Morpheus
DESCRIPTION Optional Description of the Identity Source
ONELOGIN SUBDOMAIN

example: morpheus-dev

Warning: Please verify the subdomain carefully. An invalid subdomain will cause authentication attempts by OneLogin users to fail.

ONELOGIN REGION Specify US or EU region

API CLIENT SECRET OneLogin API Client Secret from the Settings - API section in OneLogin portal

API CLIENT ID OneLogin API Client ID from the Settings - API section in OneLogin portal

REQUIRED ROLE Enter a role of OneLogin users logging into Morpheus must have at least this OneLogin role to gain access to Morpheus.

DEFAULT ROLE The default Morpheus Role applied to users created from OneLogin Integration if no other role mapping is specified below

ROLE MAPPINGS Existing Morpheus Roles will be listed with fields to enter OneLogin Roles to map to. Users with OneLogin roles matching the role mappings will be assigned the appropriate Role(s) in Morpheus when signing in.

ENABLE ROLE MAPPING PERMISSION When selected, Tenant users with appropriate rights to view and edit Roles will have the ability to set role mapping for the Identity Source integration. This allows the Tenant user to edit only the role mappings without viewing or potentially editing the Identity Source configuration.

MANUAL ROLE ASSIGNMENT When selected, administrators can manually edit Roles for users created through this identity source integration from the user detail page (Administration > Users > Selected user).

Note: For more on Identity Source role mapping permissions, see the [associated guide](#) in our KnowledgeBase.

1. Select *SAVE CHANGES* and the OneLogin Integration will be added.

Users can now login to Morpheus with OneLogin credentials. The first Login will create a user in Morpheus matching the Username, email and Password from OneLogin. If a **REQUIRED ROLE** is specified in the Identity Source settings, only users with that Role in OneLogin will be able to login to Morpheus.

Important: OneLogin users will not authenticate in Morpheus if there is an existing Morpheus User with matching username or email address.

Okta

Overview

Morpheus allows users to integrate an Okta deployment for user management and authentication. In Morpheus, identity sources are added on a per-Tenant basis and Morpheus allows you to map Okta user groups to Morpheus user groups. User accounts are automatically created with matching metadata and role permissions when users are authenticated.

Adding an Okta Integration

1. Navigate to Administration > Tenants
2. Select a Tenant
3. Select *IDENTITY SOURCES*
4. Select + *IDENTITY SOURCE*
5. Choose TYPE: “Okta”
6. Populate the following, then select *SAVE CHANGES*:

Name Unique name for authentication type

Description A description for your new Okta Identity Source

Okta URL Your Okta URL

Administrator API Token Your Okta Administrator API Token

Required Group The Okta group that users must be in to have access (optional)

Default Role The default role a user is assigned if no group is listed under an Okta user that maps within the Morpheus Role Mappings section

ENABLE ROLE MAPPING PERMISSION When selected, Tenant users with appropriate rights to view and edit Roles will have the ability to set role mapping for the Identity Source integration. This allows the Tenant user to edit only the role mappings without viewing or potentially editing the Identity Source configuration.

MANUAL ROLE ASSIGNMENT When selected, administrators can manually edit Roles for users created through this identity source integration from the user detail page (Administration > Users > Selected user).

Note: For more on Identity Source role mapping permissions, see the [associated guide](#) in our KnowledgeBase.

Now, allowed Okta users can log into Morpheus via their Okta credentials and a user will be automatically generated within Morpheus with matching metadata and mapped Role permissions.

Note: If you’ve created multi-tenant roles, these will also appear here and can be mapped to Okta user groups allowing you to map users to equivalent user groups in Morpheus.

5.13.22 ITSM

ServiceNow

Overview

IT Service Management (ITSM) is an important area of focus for many organizations. Organizations invested in ServiceNow as an ITSM provider will find that Morpheus integrates tightly with some of the most-used features. After integrating ServiceNow with Morpheus, both environments can be used interchangeably and the results are synced to both places. This guide walks administrators through the process of integrating ServiceNow with Morpheus and how Morpheus can be used to effectively leverage the best of ServiceNow.

Tip: The ServiceNow integration guide is also available as a [PDF download](#), which includes additional example use cases and screenshots.

Important: Only one Morpheus appliance should be integrated with a single ServiceNow instance at any given time. Integrating multiple appliances with the same ServiceNow instance can cause issues with sharing Morpheus Catalog Items through to ServiceNow for ordering from the ServiceNow console. Additional details on Catalog Item sharing are included in this guide.

Add ServiceNow Integration

1. Navigate to Administration > Integrations
2. Select + *NEW INTEGRATION*
3. Select “ServiceNow” from the dropdown list
4. Add the following:

NAME A friendly name to describe the ServiceNow integration in Morpheus.

ENABLED Check “Enabled” to allow consumption of this ServiceNow integration in Morpheus.

SERVICENOW HOST URL of the ServiceNow instance (ex: <https://your.instance.service-now.com>), keep in mind you can create multiple ServiceNow integrations in Morpheus if needed.

API PROXY If necessary, select a configured proxy (Infrastructure > Network > Proxies) to route traffic through to the ServiceNow API. If a proxy is not configured here, ServiceNow API traffic will be routed through the global proxy if one is configured on the appliance.

CREDENTIALS Supply credentials for a user in ServiceNow that is able to access the REST interface and create/update/delete incidents, requests, requested items, item options, catalog items, workflows, etc. The list of necessary roles includes `x_moda_morpheus_ca.integration` (available if the Morpheus ServiceNow plugin is installed from the ServiceNow Store), `catalog_admin`, `itil`, `rest_service`, `web_service_admin` and `import_transformer`.

Morpheus supports simple and OAuth 2.0 authentication with ServiceNow. See the next section for additional details on configuring the ServiceNow appliance for OAuth 2.0 authentication if you intend to use it. When supplying credentials to Morpheus users may opt to integrate with a saved set of username and password credentials, a saved set of OAuth 2.0 credentials, a new set of username and password credentials (not saved), or a new set of credentials (OAuth 2.0 or username/password) which will also be saved to the Morpheus credential store for later use. Once the credential type is selected, the fields in the modal will adjust to correspond to the chosen credential type. Once again, see the next section for more details on configuring OAuth 2.0 authentication.

CMDB CUSTOM MAPPING If needed, administrators can opt to populate a specific field in the ServiceNow table and such mapping is identified here with a JSON code snippet. Below is an example that populates the `object_id` field in the CM database with the Morpheus instance name and two other field examples:

```
{
  "object_id": "<%=instance.name%>",
  "SN_field_id2": "<%=morph.varname2%>",
  "SN_field_id3": "<%=morph.varname3%>"
}
```

CMDB CLASS MAPPING Define the mapping between Morpheus server types and ServiceNow CI classes. Select a Morpheus server type from the dropdown menu and a new field will appear in the list. Enter a ServiceNow CI class into the text field to create the association

DEFAULT CMDB BUSINESS CLASS Allows the user to define the table CMDB records are written to if they prefer this over Morpheus default. By default, Morpheus writes to the `cmdb_ci_vm_instance` table.

CMDB MODE Choose “Table Based” or “Identification & Reconciliation API”. Selecting “Table Based” will utilize the older Table-based API mode for CMDB sync. Organizations utilizing IRE should select “Identification & Reconciliation API”.

5. Save Changes

Important: Morpheus supports integration with single-domain and multi-domain ServiceNow appliances. In multi-domain installations, a selected ServiceNow company can be mapped to a selected Morpheus Tenant for purposes of exposing Morpheus Library items only to users within a certain company. In this configuration, ServiceNow integrations should be added in each relevant Morpheus Tenant. Further setup steps for exposing Morpheus library items to ServiceNow are included in a later section below.

Configuring ServiceNow for OAuth 2.0 Authentication

Before configuring Morpheus to use OAuth 2.0 authentication with ServiceNow, ensure your ServiceNow appliance is configured correctly. OAuth must be set up and activated, you must also create a new endpoint for the client. See the following relevant parts of ServiceNow documentation to properly configure your appliance:

1. [Set up OAuth](#)
2. [Create a new application endpoint](#) for Morpheus to access the ServiceNow instance

With ServiceNow correctly configured, we can integrate ServiceNow using either a stored OAuth 2.0 credential set or we can create one on the fly during integration. When creating one on the fly Morpheus will save it as a stored credential set for later use. Whether storing one ahead (Infrastructure > Trust > Credentials) or storing one at integration time, configure your credentials as follows. Note that all fields are required for a ServiceNow Integration unless specifically mentioned otherwise:

Note: Some of the fields below may not be present if creating an OAuth credential set on the fly as opposed to the Infrastructure > Trust section of Morpheus.

- **CREDENTIAL STORE:** Select “Internal” or (if present) an external Cypher store
- **NAME:** A name for the stored credential set in Morpheus
- **DESCRIPTION:** An optional description for the credential set
- **ENABLED:** If enabled, this credential set will be selectable for creating various integrations in Morpheus
- **GRANT TYPE:** Use “Password Credentials”
- **ACCESS TOKEN URL:** Should be the appliance domain with the path of “/oauth_token.do”. For example, “https://mydomain.service-now.com/oauth_token.do”
- **CLIENT ID:** The client ID (potentially auto-generated) set when the endpoint was created in ServiceNow
- **CLIENT SECRET:** The client secret set when the endpoint was created in ServiceNow
- **USERNAME:** The username for a ServiceNow service account, note the required permissions this user must have in the section above

- **PASSWORD:** The password for the ServiceNow account
- **SCOPE:** Left empty
- **CLIENT AUTHENTICATION:** Use “Send client credentials in body”

If storing these credentials for later use, click *ADD CREDENTIALS*. If creating this credential set on the fly at the time of integration, complete the rest of the new integration modal as discussed in the prior section.

EDIT CREDENTIALS ✕

CREDENTIAL STORE: Internal

NAME: ServiceNow OAuth

DESCRIPTION:

ENABLED

GRANT TYPE: Password Credentials

ACCESS TOKEN URL: https://d[REDACTED]/oauth_token.do

CLIENT ID: [REDACTED]

CLIENT SECRET: [REDACTED]

USERNAME: morpheusintegration

PASSWORD: [REDACTED]

SCOPE:

CLIENT AUTHENTICATION: Send client credentials in body

SAVE CHANGES

ServiceNow Configuration Management Database (CMDB)

ServiceNow CMDB is central to maintaining a complete record of IT infrastructure at many organizations. The Morpheus ServiceNow integration can create and update configuration item (CIs) records as new services are provisioned or existing services are reconfigured. Once a ServiceNow integration is set as the CMDB for a Cloud or Group, CI records are created and managed by Morpheus.

Setting a CMDB on a Group

When adding or editing a Morpheus Group, any active ServiceNow integration can be set as the CMDB.

1. Navigate to Infrastructure > Groups
2. Select an existing Group name from the list
3. Click *EDIT*
4. Under “Advanced Options”, select an active ServiceNow integration from the CMDB dropdown menu
5. If desired, select “CMDB DISCOVERY” to create CMDB CI records for discovered (unmanaged) servers that Morpheus automatically onboards to this Group

This setting is also available when creating a Group. Rather than selecting an existing Group in step two above, click + *CREATE* to make a new Group.

Setting a CMDB on a Cloud

When adding or editing a Morpheus Cloud, any active ServiceNow integration can be set as the CMDB.

1. Navigate to Infrastructure > Clouds
2. Select an existing Cloud name from the list
3. Click *EDIT*
4. Under “Advanced Options”, select an active ServiceNow integration from the CMDB dropdown menu
5. If desired, select “CMDB DISCOVERY” to create CMDB CI records for discovered (unmanaged) servers that Morpheus automatically onboards to this Cloud

This setting is also available when creating a Cloud. Rather than selecting an existing Cloud in step two above, click + *ADD* to make a new Cloud.

Provisioning and CI Records

With a ServiceNow instance integrated with Morpheus and the instance set as the CMDB for a Morpheus Group or Cloud, we will see CI records created as new resources are provisioned to the Cloud or Group in Morpheus. After the provisioning process has completed, a CI record should exist with a name value equal to the Instance name in Morpheus.

Provisioned and active Instances in Morpheus will have CI records with an “On” state in ServiceNow. After they are deleted in Morpheus, the state value will be rolled to “Terminated” in ServiceNow as expected.

Morpheus will also populate a number of additional fields in ServiceNow including IP address, FQDN and more. Custom views can be created in ServiceNow to expose these fields.

ServiceNow Approval Policies

Morpheus offers its own approval engine out of the box, but some organizations prefer ServiceNow to be their final approval authority. With a ServiceNow instance integrated with Morpheus, administrators can create provision approval policies and tie them to an active ServiceNow integration. With the policy in place, any new provisioning within the policy scope (Global, Group, Cloud, User, or Role) is sent to ServiceNow for approval before provisioning will go ahead in Morpheus. Approvals are synced between the two applications every minute.

Add ServiceNow Provision Approval Policy to a Cloud

Note: Any Instance provisioned into a Cloud with an approval policy enabled will not proceed without the required approval.

To add a ServiceNow Approval policy to a Cloud:

1. Navigate to **Infrastructure > Clouds**
2. Select a Cloud by clicking on the desired Cloud name link
3. Select the **POLICIES** tab
4. Click + **ADD POLICY**
5. Select **Provision Approval** from the type dropdown
6. Optionally enter a description for the Policy
7. Configure the following:

APPROVAL INTEGRATION Select the ServiceNow Integration already configured in **Administration > Integrations** to use for the approval policy.

WORKFLOW Select the ServiceNow workflow for the approval in ServiceNow (if desired). These workflows are configured and synced in from the ServiceNow Integration.

TENANTS (if applicable) Only required for multi-tenant permission scoping. For the policy to apply to a Subtenant, type the name of the tenant(s) and select the Tenant(s) from the typeahead list.

8. Save Changes

Add ServiceNow Provision Approval Policy to a Group

Note: Any Instance provisioned into a Group with an approval policy enabled will not proceed without the required approval.

To add a ServiceNow Approval policy to a Group:

1. Navigate to **Infrastructure > Groups**
2. Select a Group by clicking on the Group name
3. Select the **POLICIES** tab
4. Click + **ADD POLICY**
5. Select **Provision Approval**

6. Optionally enter a description for the Policy

7. Configure the following:

APPROVAL INTEGRATION Select the ServiceNow Integration already configured in Administration > Integrations to use for the approval policy.

WORKFLOW Select the ServiceNow workflow for the approval in ServiceNow (if desired). These workflows are configured and synced in from the ServiceNow Integration.

TENANTS (if applicable) Only required for multi-tenant permission scoping. For the policy to apply to a Subtenant, type the name of the tenant(s) and select the Tenant(s) from the typeahead list.

8. Save Changes

Using ServiceNow Approval Policies

Any Instance provisioned into a Cloud or Group with an approval policy enabled will be in a PENDING state until the request is approved.

Instances pending a ServiceNow approval will show “Waiting for Approval” with the Requested Item number and Request number, ex: `Waiting for Approval [RITM0010002 - REQ0010002]`.

ServiceNow approval requests are displayed in **Operations > Approvals**. Instances pending a ServiceNow approval must be approved in ServiceNow for provisioning to initiate. Approval requests from a ServiceNow approval policy cannot be approved in Morpheus, only approvals originating from Morpheus.

ServiceNow approval requests are displayed in Morpheus under **Operations > Approvals**. Pending ServiceNow approval requests can be cancelled in Morpheus by selecting the request and then selecting **ACTIONS > Cancel**.

Once a pending ServiceNow approval request is approved in ServiceNow, the Instance(s) will begin to provision in Morpheus within one minute of being approved in ServiceNow.

ServiceNow Monitoring Integration Settings

Note: A ServiceNow integration must be already configured in Administration > Integrations to enable ServiceNow monitoring.

The ServiceNow monitoring integration is enabled and configured in Administration > Settings > Monitoring. As long as the “Enabled” switch is activated, Morpheus will report monitoring data to ServiceNow. Configuration selections are described below:

Enabled Enables the ServiceNow monitoring integration

Integration Select from an existing ServiceNow integration in `|AdmInt|`

New Incident Action The ServiceNow action to take when a Morpheus incident is created

Close Incident Action The Service Now action to take when a Morpheus incident is closed

Incident Severity Mapping

Morpheus Severity	ServiceNow Impact
Info	Low/Medium/High
Warning	Low/Medium/High
Critical	Low/Medium/High

Once finished working with configuration, click *APPLY*

▼ servicenow ENABLED

Enabled

Integration

New Incident Action
The Service Now action to take when a Morpheus incident is created.

Close Incident Action
The Service Now action to take when a Morpheus incident is closed.

MORPHEUS SEVERITY	SERVICE NOW IMPACT
Info	<input type="text" value="Low"/>
Warning	<input type="text" value="Medium"/>
Critical	<input type="text" value="High"/>

ServiceNow Service Catalog Integration

In addition to integrating with key ServiceNow features, Morpheus offers a free plugin directly from the ServiceNow Store. Once the plugin is installed, Morpheus Self-Service Catalog Items can be presented as provisioning options in the ServiceNow catalog for ordering.

Note: Surfacing Catalog Items made with Forms to ServiceNow is not yet supported. If planning to use ServiceNow to order Catalog Items you should not use Forms on any Catalog Items until it is supported.

The Morpheus plugin supports integration with ServiceNow whether it's configured for a single tenant or for multiple domains. When both Morpheus and ServiceNow are configured for multiple Tenants, we can create ServiceNow integrations in any relevant Morpheus Tenant and map those to specific companies in ServiceNow. Any exposed library items would only be shared with users in the relevant ServiceNow company. The Morpheus plugin will automatically detect whether the *ServiceNow Domain Support–Domain Extensions Installer plugin* has been installed and respond accordingly. Additionally, the *User Criteria Scoped API plugin* must also be enabled on the ServiceNow instance for multi-tenant use.

Depending on the scenario, setup steps for the Morpheus plugin will be slightly different. Setup steps for both single and domain-separated ServiceNow environments are included below.

Important: A valid SSL Certificate is required on the Morpheus Appliance for the ServiceNow plugin to be able to communicate with the appliance.

Important: As described below, the Morpheus ServiceNow plugin requires the use of a Morpheus service account to integrate back with the Morpheus appliance. Some symbol characters, specifically “%” and “&” are valid for use in Morpheus account passwords but aren't passed correctly when ServiceNow makes its API calls to Morpheus. It is best not to use these characters in the password for Morpheus accounts which may be used in the ServiceNow plugin to interface back with Morpheus. Authentication errors will occur and the plugin will not work. This is a ServiceNow issue which Morpheus has no control over.

Single-Domain ServiceNow Configuration

1. Install the Morpheus plugin from the ServiceNow store, refer to the [Morpheus Data plugin for ServiceNow](#) installation instructions for additional help with the installation steps
2. Navigate to Morpheus Catalog > Properties
3. Set the following properties:

MID Server If desired, specify the name of an existing MID server

Morpheus Appliance Endpoint The full URL to your Morpheus appliance

Username Morpheus user that the plugin will connect as to the Morpheus API

Password Password to the above Morpheus account

Morpheus Manage Workflows? Indicate whether Morpheus should manage workflows. If this option is checked, Morpheus will overwrite the workflow and set it to “Morpheus (Internal) Catalog Item Provision Instance” on sync

Important: The Morpheus service account integrated with the plugin interacts with the Morpheus appliance through Morpheus API and must have the appropriate Role permissions to complete all provisioning requests from the ServiceNow plugin. Often it’s easiest to make a service account with full administrator rights to avoid failed provisioning. If you’d prefer to create a minimal service account for security reasons, ensure the Role for the service account User has the following permissions:

- Personas: Standard: Full
- Personas: Service Catalog: Full
- Features: Provisioning: Instances: Full
- Features: Provisioning: Apps: Full
- Groups: Full rights to all Groups containing Clouds you will expose to ServiceNow
- Instance Types: Full rights to all Instance Types you will expose to ServiceNow
- Blueprints: Full rights to all Blueprints you will expose to ServiceNow
- Catalog Item Types: Full rights to all Catalog Item Types you will expose to ServiceNow

Users created from SAML Identity Sources cannot authenticate with the Morpheus API and cannot be used for the ServiceNow plugin.

Properties to control the Morpheus integration

The Morpheus Appliance endpoint [?](#)

Username to authenticate with the Morpheus Appliance [?](#)

The password to authenticate with the Morpheus Appliance [?](#)

Specify the name of a MID Server for Morpheus to utilize [?](#)

[Save](#)

Multi-Domain ServiceNow Configuration

1. Install the Morpheus plugin from the ServiceNow store, refer to the [Morpheus Data plugin for ServiceNow](#) installation instructions for additional help with the installation steps
2. Navigate to Morpheus Catalog > Multi-Tenant Credentials
3. Set the following properties:

Morpheus Appliance Endpoint The full URL to your Morpheus appliance

Morpheus Tenant ID The integer database ID for the selected Tenant

Username Morpheus user that the plugin will connect as to the Morpheus API. This user must exist within the Morpheus Tenant being linked to the chosen ServiceNow company

Password The password for the above user

ServiceNow Company Select a company from the list to link with the Tenant whose ID was entered above

MID Server If desired, specify the name of an existing MID server. Note that configuring a multi-domain MID server requires the `glide.ecc.enable_multidomain_mid` property in `sys_properties.list` be set to `true` prior to creating the MID server in the global domain. This allows the MID server to explore any domain for which it has the credentials. The ServiceNow user (which the MID server authenticates with) must be in the global domain as well. For more, see [this section of ServiceNow documentation](#).

Morpheus Manage Workflows? Indicate whether Morpheus should manage workflows. If this option is checked, Morpheus will overwrite the workflow and set it to “Morpheus (Internal) Catalog Item Provision Instance” on sync

Important: The Morpheus service account integrated with the plugin interacts with the Morpheus appliance through Morpheus API and must have the appropriate Role permissions to complete all provisioning requests from the ServiceNow plugin. Often it’s easiest to make a service account with full administrator rights to avoid failed provisioning. If you’d prefer to create a minimal service account for security reasons, ensure the Role for the service account User has the following permissions:

- Personas: Standard: Full
- Personas: Service Catalog: Full
- Features: Provisioning: Instances: Full
- Features: Provisioning: Apps: Full
- Groups: Full rights to all Groups containing Clouds you will expose to ServiceNow
- Instance Types: Full rights to all Instance Types you will expose to ServiceNow
- Blueprints: Full rights to all Blueprints you will expose to ServiceNow
- Catalog Item Types: Full rights to all Catalog Item Types you will expose to ServiceNow

Users created from SAML Identity Sources cannot authenticate with the Morpheus API and cannot be used for the ServiceNow plugin.

Adding to ServiceNow Catalog

Once the ServiceNow plugin is installed and configured, Service Catalog items can be exposed to the ServiceNow catalog from Morpheus. Follow the guide below to expose Morpheus Clouds, Library Items, and Blueprints to users in the ServiceNow catalog.

1. Navigate to Administration > Integrations
2. Select the relevant ServiceNow integration
3. Within the “EXPOSED CATALOG ITEMS” section is a list of currently-exposed Service Catalog items
4. To expose a new item, click + *ADD CATALOG ITEM*
5. Select an available item from the dropdown menu and click *SAVE CHANGES*
6. Back in ServiceNow, access the Morpheus plugin from the Service Catalog
7. Exposed Morpheus Service Catalog items are visible here for ServiceNow users with sufficient role permissions

Integrations > Labs ServiceNow

 Labs ServiceNow

[EDIT](#)
[DELETE](#)



Host: <https://ven01435.service-now.com/> Last Update: 12/20/2021 04:07 PM

EXPOSED CATALOG ITEMS

Select which catalog items you would like to have exposed to ServiceNow for provisioning. [+ ADD CATALOG ITEM](#)

TYPE	NAME	ENABLED	
	Amazon Options	<input checked="" type="checkbox"/>	

Cherwell

Add Cherwell Integration

1. Navigate to Administration > Integrations
2. Select + **NEW INTEGRATION**
3. Select Cherwell from the dropdown.
4. Add the following:
 - NAME** Name of the Integration in Morpheus.
 - ENABLE** Leave checked to enable the Integration.
 - HOST** Url of the Cherwell Instance
 - USER** Enter in username
 - PASSWORD** Above Cherwell user’s password
 - CLIENT KEY** Provide your Cherwell client key
 - CREATED BY USER** This is the full name of a user in the Cherwell system. When a new change management record is created in the Cherwell system, this user will be added to the record as the user that created it.

START DAYS FROM NOW Number of days from now to set proposed start date

END DAYS FROM NOW Number of days from now to set proposed end date

CUSTOM MAPPING This is an optional json object that allows the custom setting of the Cherwell fields on the Change Request object.

Note: The keys in the map correspond to the name of the field on the Change Request in Cherwell that you would like to set (see <https://bertram.d.pr/1Ziuhy> for a reference). In addition, the value in the map corresponds to the value you wish to use. Within the value, Morpheus variables may be used. Here is an example for setting the Description is:

```
{
  "Description": "Created from Morpheus by ${instance.createdByUsername}
  ↪ in ${zone.name}"
}
```

5. Save Changes

Remedy

PreRequisites

The user used for this integration need to be an Administrator in Remedy or have all the permissions to the form that is outlined in the table below.

API Endpoint	Action	BMC Form
/api/arsys/v1/entry/CTM:People	GET	CTM:People
/api/arsys/v1/entry/COM:Company?q=%27Status%27=%22Enabled%22&fields=values(Company)	GET	COM:Company
/api/arsys/v1/entry/User	GET	User
/api/arsys/v1/entry/Group	GET	Group
/api/arsys/v1/entry/CHG:Infrastructure%20Change	POST	CHG:Infrastructure
/api/arsys/v1/entry/CHG:Infrastructure%20Change	PUT	CHG:Infrastructure
/api/arsys/v1/entry/CHG:Infrastructure%20Change	GET	CHG:Infrastructure
/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_DiskDrive	POST	BMC.CORE:BMC
/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_DiskDrive	PATCH	BMC.CORE:BMC
/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_DiskDrive	GET	BMC.CORE:BMC
/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_DiskDrive	DELETE	BMC.CORE:BMC
/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_IPEndpoint	POST	BMC.CORE:BMC
/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_IPEndpoint	PATCH	BMC.CORE:BMC
/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_IPEndpoint	GET	BMC.CORE:BMC
/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_IPEndpoint	DELETE	BMC.CORE:BMC
/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_Memory	POST	BMC.CORE:BMC
/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_Memory	PATCH	BMC.CORE:BMC
/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_Memory	GET	BMC.CORE:BMC
/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_Memory	DELETE	BMC.CORE:BMC
/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_Processor	POST	BMC.CORE:BMC
/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_Processor	PATCH	BMC.CORE:BMC
/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_Processor	GET	BMC.CORE:BMC
/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_Processor	DELETE	BMC.CORE:BMC

Table 24 – continued from previous page

API Endpoint	Action	BMC Form
/api/arsys/v1/entry/AST:ComputerSystem	GET	AST:ComputeSystem
/api/arsys/v1/entry/AST:ComputerSystem	PUT	AST:ComputeSystem
/api/arsys/v1/entry/AST:ComputerSystem	POST	AST:ComputeSystem
/api/arsys/v1/entry/AST:IPEndpoint	GET	AST:IPEndpoint
/api/arsys/v1/entry/AST:IPEndpoint	PUT	AST:IPEndpoint
/api/arsys/v1/entry/AST:IPEndpoint	POST	AST:IPEndpoint
/api/arsys/v1/entry/AST:DiskDrive	GET	AST:DiskDrive
/api/arsys/v1/entry/AST:DiskDrive	PUT	AST:DiskDrive
/api/arsys/v1/entry/AST:DiskDrive	POST	AST:DiskDrive
/api/arsys/v1/entry/AST:Processor	GET	AST:Processor
/api/arsys/v1/entry/AST:Processor	PUT	AST:Processor
/api/arsys/v1/entry/AST:Processor	POST	AST:Processor
/api/arsys/v1/entry/AST:Memory	GET	AST:Memory
/api/arsys/v1/entry/AST:Memory	PUT	AST:Memory
/api/arsys/v1/entry/AST:Memory	POST	AST:Memory
/api/jwt/login	POST	

Add Remedy Integration

1. Navigate to Administration > Integrations
2. Select + NEW INTEGRATION
3. Select Remedy from the dropdown.
4. Add the following:

NAME Name of the Integration in Morpheus.

ENABLE Leave checked to enable the Integration.

REMEDY HOST Url of the Remedy Instance. e.g: <http://xx.xx.xx.xx:8008>

USER Enter in username

PASSWORD Above Remedy user's password

COMPANY The dropdown will populate with values as soon as the auth using the above creds are successful

APPROVAL USER Full name of the user as it appear in Remedy. E.g: userid 'anish' would have full name as "Anish Abraham"

5. Save Changes

5.13.23 Load Balancers

AzureLB

Add Azure Load Balancer

1. Navigate to *Infrastructure > Load Balancers*
2. Select + *ADD*
3. Select *Azure Load Balancer*
4. Fill in the following:
 - CLOUD** Select the Cloud the Load Balancer will be available for
 - NAME** Name of the Load Balancer in Morpheus
 - DESCRIPTION** Identifying information displayed on the Load Balancer list page.
 - VISIBILITY** Define Multi-Tenant permissions
 - RESOURCE GROUP** Select the Resource Group the Load Balancer will be linked to
5. Save changes

F5 Load Balancers

Add F5 Load Balancer

To add a F5 Load Balancer Integration:

1. Navigate to *Infrastructure > Load Balancers*
 2. Select + *ADD*
 3. Select *F5 BigIP*
 4. Fill in the following:
 - GROUP** Select the Group the Load Balancer will be available for
 - CLOUD** Select the Cloud the Load Balancer will be available for
 - NAME** Name of the Load Balancer in Morpheus
 - DESCRIPTION** Identifying information displayed on the Load Balancer list page.
 - VISIBILITY** Define Multi-Tenant permissions
 - API HOST** IP or resolvable hostname url.
 - API PORT** Typically 8443
 - USERNAME** API user
 - PASSWORD** API user password
 - MANAGEMENT URL** Example: `https://10.30.20.31:8443/xui/`
- Advanced Options (optional)**
- VIRTUAL NAME
 - POOL NAME

- SERVER NAME

5. Save Changes

Note: The Morpheus integration with F5 only supports basic authorization. Token-based authorization is not currently supported.

Virtual Servers

Instances attached to an F5 will be listed in the Virtual servers tab. Virtual servers can also be manually added in this section.

Add Virtual Server

1. Navigate to *Infrastructure > Load Balancers*
2. Select F5 Integration name to drill into the detail page
3. Select + *ADD* in the VIRTUAL SERVERS tab
4. Fill in the following:
 - **NAME** Name of the Virtual Server in Morpheus
 - **DESCRIPTION** Description of the Virtual Server in Morpheus
 - **Enabled** Uncheck to keep the configuration but disable F5 availability in Morpheus
 - **VIP TYPE**
 - Standard
 - Forwarding (Layer 2)
 - Forwarding (IP)
 - Performance (HTTP)
 - Performance (Layer 4)
 - Stateless
 - Reject
 - DHCP
 - Internal
 - Message Routing
 - **VIP HOSTNAME** Enter Hostname of the VIP (optional)
 - **VIP ADDRESS** Enter IP address for the VIP
 - **VIP PORT** Enter port used for the VIP
 - **SOURCE ADDRESS** Enter Virtual Server source address
 - **PROTOCOL** tcp, udp, or sctp
 - **PROFILES** Search for and select from available PROFILES
 - **POLICIES** Search for and select from available POLICIES

- **IRULES** Search for and select from available RUEL SCRIPTS
- **PERSISTENCE**
 - cookie
 - dest-addr
 - global-settings
 - hash
 - msrdp
 - sip
 - source-addr
 - ssl
 - universal
- **DEFAULT POOL** Select from available POOLS

5. Select *SAVE CHANGES*

Policies

Policies will be synced and listed in the Policies tab. These policies will be available options when creating Virtual Servers.

Pools

Create Pool

NAME Name of the POOL in Morpheus

DESCRIPTION Description of the POOL in Morpheus

BALANCE MODE

- Round Robin
- Least Connections

SERVICE PORT Specify SERVICE PORT for the POOL

MEMBERS Search for and select from available NODES

MONITORS Search for and select from available Monitors

Profiles

SSL Profiles are synced and will be created when an SSL Certificate is assigned in the Load balancer section when provisioning or editing a Load balancer on an Instance.

Monitors

Create Monitor

NAME Name of the MONITOR in Morpheus

DESCRIPTION Description of the MONITOR in Morpheus

PARENT MONITOR Select from available MONITORS

DESTINATION Specify Destination, such a *:443. Default is *:*

INTERVAL Specify Monitor Interval. Default is 5

TIMEOUT Specify Monitor Timeout. Default is 15

MONITOR CONFIG Enter monitor config.

Nodes

Create Node

NAME Name of the NODE in Morpheus

DESCRIPTION Description of the NODE in Morpheus

ADDRESS Enter node address

MONITOR Select from available MONITORS

SERVICE PORT Specify SERVICE PORT for the NODE

Rule Scripts

Rule Scripts will be synced and listed in the RULE SCRIPTS tab. These rules will be available options when creating Virtual Servers.

Citrix NetScaler



Add NetScaler Integration

To add a NetScaler Load Balancer Integration:

1. Navigate to *Infrastructure > Load Balancers*
2. Select + *ADD*
3. Select *Citrix NetScaler*
4. Fill in the following:

GROUP * Select the Group the Load Balancer will be available for.

CLOUD * Select the Cloud the Load Balancer will be available for.

NAME * Name of the Load Balancer in Morpheus.

DESCRIPTION Identifying information displayed on the Load Balancer list page.

VISIBILITY

Define Tenant Visibility

- Public: Available to all Tenants.
- Private: Only available to specified Tenant.

Tenant If Visibility is set to private, define the Tenant the Load Balancer will be available in.

API URL *

URL of the NetScaler API

- Example: <http://10.30.21.55>

API PORT *

NetScaler API port

- Example: 80

USERNAME * NetScaler service account username

PASSWORD * NetScaler service account password

VIRTUAL NAME

Naming Pattern for new NetScaler Virtual Servers

- If blank, defaults to `morph_lb_{{loadBalancer.id}}`

SERVICE NAME

Naming Pattern for new NetScaler Services

- If blank, defaults to `morph_service_{{container.id}}`

SERVER NAME

Naming Pattern for new NetScaler Servers

- If blank, defaults to `morph_server_{{server.id}}`

Add Load Balancer to Instance

Load Balancers can be added to Instances during Provisioning or to existing Instances. For Load Balancer settings to appear during provisioning, or for the scale tab to be available on an Instance, the instances Node Type must have a LB port defined.

Note: For Load Balancer settings to appear during provisioning, or for the scale tab to be available on an Instance, the instances Node Type must have a LB port defined.

Add Load Balancer during Provisioning

In the Instance Provisioning wizard, Load Balancers can be configured in the Automation > Load Balancer section.

1. Navigate to Provisioning > Instances.
2. Select + *ADD*.
3. Select an Instance Type that supports scaling. (ENABLE SCALING (HORIZONTAL) flagged on Instance Type configuration)
4. Proceed with Instance configuration to the Automation section.
5. Fill in the following:

VIP ADDRESS

Define IP Address for the Virtual Server

- Example: 10.30.23.191

VIP PORT

Define port for the Virtual Server

- Example: 80

VIP HOSTNAME

Define hostname that will resolve to the VIP IP.

- Example: jwDemoHaApp59.den.example.com

VIRTUAL SERVICE NAME Define name for the Virtual Service. Defaults to `${instance.name}`

BALANCE MODE

Specify balance mode for the VIP

- Least Connections
- Round Robin

STICKY MODE

Specify sticky session options for the VIP

- Source IP
- Cookie

SHARED VIP ADDRESS Select if VIP is shared, then enter DIRECT VIP ADDRESS

SSL CERT

SSL Certificate that will be applied to the VIP.

- No SSL
- Select existing Certificate from Infrastructure > Keys & Certs or from a Trust Provider Integration.

USE EXTERNAL ADDRESS FOR BACKEND NODES

- Select if traffic from LB to Backend Nodes needs to be sent to the External Addresses, or leave deselected to use Internal Addresses for Backed Nodes.
6. Optionally configure auto-scaling configuration in the Scale section
 7. Select *NEXT* and provision the Instance.

After all nodes in the Instance are provisioned, the LB configuration will be added to the Instance and Virtual Servers, Services and Servers will be created and configured on the NetScaler. The Load Balancer settings and status will be visible in the Instance details page LOAD BALANCER section, with additional details, links, and configurations options available in the SCALE tab.

5.13.24 Logs

Syslog

Adding Syslog Integration

1. Navigate to Administration > Settings > Logs
2. Expand the Morpheus logging section
3. Add the Syslog forwarding rules
4. Select *QUICK ADD*

5.13.25 Monitoring

ServiceNow Monitoring Integration

Note: A ServiceNow Integration must be already configured in Administration > Integrations to enable the ServiceNow Monitoring Integration. Refer to the *ServiceNow* configuration guide for more information.

Enabled Enables the ServiceNow Monitoring Integration

Integration Select from a ServiceNow Integration added in Administration > Integrations

New Incident Action The Service Now action to take when a Morpheus incident is created.

Close Incident Action The Service Now action to take when a Morpheus incident is closed.

Incident Severity Mapping

Morpheus Severity	ServiceNow Impact
Info	Low/Medium/High
Warning	Low/Medium/High
Critical	Low/Medium/High

NewRelic

Configuring The NewRelic Integration

1. Navigate to Administration > Settings > Monitoring
2. Expand the NewRelics section
3. Toggle the Enable slider
4. Enter License Key to be used when installing the New Relic agent in order for the agent to report data to your New Relic account

Note: The License Key is the 40-character hexadecimal string that New Relic provides when you sign up for your account.

5.13.26 Networking Integrations

Infoblox

Features

- Network Pools synchronization
- DNS Zone & Zone record synchronization
- Host Record synchronization
- Total & Free IP status bar for networks
- Network Grid and List view with IP Status and records, date and user tracking
- Automatic and manual IP Reservations, DNS A/PTR record creation and deletion
- Use script variables like `<%= variableX %>` for evaluation of the key data in extended attributes

Adding Infoblox Integration

Note: Making full use of the Morpheus Infoblox integration requires credentials for an Infoblox user account with API access granted, access to list the pools and zones you wish to work with, and rights to create and destroy records. See Infoblox documentation for more information on user rights administration in that product.

1. Navigate to Infrastructure - Network - Integrations
2. Select + ADD > IPAM > Infoblox
3. Enter the following:

ADD IPAM INTEGRATION ✕

NAME

ENABLED

URL

USERNAME

PASSWORD

THROTTLE RATE ms

DISABLE SSL SNI VERIFICATION

INVENTORY EXISTING

NETWORK FILTER

ZONE FILTER

TENANT MATCH ATTRIBUTE

IP MODE ▾

EXTRA ATTRIBUTES

Accepts a JSON input of custom attributes that can be saved on Host Record in Infoblox. These Must be first defined as extra attributes in Infoblox and values can be injected for the user creating the record and the date of assignment. The available injectable attributes are: `userId`, `username`, and `dateCreated`. They can be injected with `<%=>`.

SAVE CHANGES

NAME Name of the Integration in Morpheus

Enabled Deselect to disable the Integration

URL Infoblox wapi url. Example: <https://yourInfobloxHost/wapi/v2.10.5>

Tip: The target Infoblox host wapi version can be referenced at <https://yourInfobloxHost/wapidoc/>

USERNAME Infoblox user username

PASSWORD Infoblox user password

THROTTLE RATE In milliseconds (ms)

DISABLE SSL SNI VERIFICATION Leave selected to disable SSL SNI Verification

INVENTORY EXISTING Mark this option to inventory existing network pools from Infoblox

NETWORK FILTER Filter which networks are synced into Morpheus. Example: Network Filter: `network_view=default&*Building=work`

ZONE FILTER Filter terms for Zone Records

TENANT MATCH ATTRIBUTE This can be set to the name of the extended attribute in Infoblox where Morpheus will check for the id of a morpheus tenant. This allows for setting the tenant's Morpheus id to an extended attribute field on a network view or network in Infoblox, and when the network or view is discovered by morpheus, it will be auto assigned to the right tenant.

IP MODE Static IPs or DHCP Reservations

EXTRA ATTRIBUTES Accepts a JSON input of custom attributes that can be saved on host records in Infoblox. These Must be first defined as extra attributes in Infoblox and values can be injected for the user creating the record and the date of assignment. The available injectable attributes are: `userId`, `username`, and `dateCreated`.

```
{
  "Date Assigned": "<%=dateCreated%>",
  "Requestor": "<%=username%>",
  "Request Number": "<%=userId%>"
}
```

4. Select *SAVE CHANGES*

Upon save the Infoblox IPAM integration will be created and the following will sync:

- Infoblox networks will be synced in and populate in the *Infrastructure - Network - IP Pools* tab and in the Infoblox detail page under the *NETWORK POOLS* tab
- Host Records will sync and populate in the Network Pool detail view (select an IP Pool name to view)
- DNS Zones will sync and populate under *Infrastructure - Network - Domains* and in the Infoblox detail page under the *HOSTS* tab
- DNS Zone Records will sync and populate

Adding IP Pools to Networks

Morpheus can automatically assign the next available Infoblox IP in an IP/Network Pool and create the corresponding DNS records, as well as remove the records upon teardown. To enable this, add an Infoblox IP/Network Pool to the *Network Pool* section on a Network(s).

1. Navigate to *Infrastructure > Network > Networks*
2. Select a Network name and click *EDIT*
3. In the *NETWORK POOL* section, search for and select the name of the IP/Network Pool.
 - Gateway, DNS and CIDR must be populated for static/pool IP assignment
 - Select *Allow IP Override* to allow selecting between DHCP, Static entry and Pool Selection at provision time (if desired)
 - Deselect DHCP server if a DHCP server will not be used on the network (only static and/or IP Pool IP assignment)
4. Select *SAVE CHANGES*

Creating Host Records

1. Select a Network Pool from *Infrastructure > Network > IP Pools* or *Infrastructure > Network > Services > Infoblox*
2. Select + *ADD*
3. Enter the following

CREATE HOST RECORD ✕

HOSTNAME

IP ADDRESS

DOMAIN ▾

Create DNS Records

SAVE CHANGES

HOSTNAME Hostname for the record

IP ADDRESS IP address for the Host Record

DOMAIN Select an Infoblox Zone

Create DNS Records Select to create DNS A and PTR Records in Infoblox

4. Select *SAVE CHANGES*

Creating Zone Records

1. Select a Domain from *Infrastructure > Network > Domains* or *Infrastructure > Network > Services > Infoblox > Zones*
2. Select + *ADD*
3. Enter the following

CREATE ZONE RECORD
✕

NAME

TYPE

CONTENT

TTL

SAVE CHANGES

NAME Name for the record, such as Hostname

Type A, AAAA, CNAME, MX, NS, PTR, SOA, or TXT

CONTENT Content of the record, such as IP or A Record

TTL Time To Live value

4. Select *SAVE CHANGES*

phpIPAM

Configuration

1. Within phpIPAM dashboard, enable api in Administration > phpIPAM settings > feature settings. Toggle API switch to on and save.
2. Go to Admin > API > create API key.
3. Create unique App ID.
4. Enable read/write/admin access under **App Permissions**.
5. Under **App Security** select SSL with User Token.

Add phpIPAM integration to Morpheus

1. Navigate to **Infrastructure > Network > Integrations**
2. Select **+ ADD > IPAM > phpIPAM**
3. Enter the following:
 - Name
 - **URL** Add `/api/` to end of URL ex. `http://10.30.20.196/api/`
 - **App ID** From phpIPAM API Key
 - Username
 - Password
 - Enable or Disable SSL SNI Verification
 - Enter Network Filter
4. Select **SAVE IPAM INTEGRATION**

NSX-T

Overview

VMware NSX-T offers network virtualization allowing for creation and management of software-based virtual networks in an efficient and programmatic way. Morpheus offers a full-featured integration with NSX-T, including Project scoping for NSX-T 4+ integrations. Morpheus will ingest and expose its networking abstractions in the following sections of the Morpheus NSX-T integration:

- SUMMARY
- TRANSPORT ZONES
- DHCP
- SEGMENTS
- FIREWALL
- TIER-1 GATEWAYS
- TIER-0 GATEWAYS
- EDGE CLUSTERS
- GROUPS

This guide goes through the process of integrating an existing NSX-T installation with Morpheus and working with the associated objects synced in with the integration. For more on installing NSX-T and an overview of its concepts, please review the [NSX-T overview documentation](#) provided by VMware.

NSX-T Projects

Projects in NSX-T are analogous to tenants in other products and are a part of NSX-T version 4+. Projects allow for the isolation of networking abstractions into individual tenants within a single NSX-T appliance. If your organization is already utilizing NSX-T Projects, you are probably very familiar with their concept and execution but others can find high-level details about them [here](#).

Morpheus supports a full-featured integration with NSX-T, including the ability to scope the Morpheus integration to a specific Project the service user can access. Using Project-scoped integrations allows multiple NSX-T integrations to be made to the same NSX-T appliance and ensures Morpheus users are siloed to only the NSX-T Projects they can access.

Add NSX-T Integration to Morpheus

1. Navigate to Infrastructure > Network > Integrations
2. Select Select + *ADD* > VMWare NSX
3. Enter the following:
 - **NAME:** Name for the NSX Integration in Morpheus
 - **VISIBILITY:** Public (available to all Morpheus Tenants) or Private (available only to the current Tenant). This option is shown only in the Morpheus Master Tenant
 - **API HOST:** URL of the NSX Manager (ex. <https://x.x.x.x/api>)
 - **CREDENTIALS:** A pre-stored credential set can be used to create this integration. If “Local Credentials” is selected, USERNAME and PASSWORD fields are presented and must be filled
 - **USERNAME:** NSX service account username. Prior to NSX version 4, this is likely an admin account with access to all networking constructs. In NSX version 4 and higher, this could be an admin for access to default space constructs or it could be a Project-specific user depending on the access needs of the integration being created
 - **PASSWORD:** The password for the NSX service account entered above
 - **PROJECT:** As soon as an API HOST and credentials are provided, Morpheus will attempt to authenticate with the NSX-T appliance. When authentication is successful and a NSX-T 4+ appliance is detected, a PROJECT field will appear and the dropdown will be pre-populated with Projects accessible to the service user account
 - **VMWARE CLOUD:** Select the existing VMware cloud associated with this NSX integration
4. Select *ADD NETWORK INTEGRATION*

Once the NSX Integration is added Morpheus will sync in existing Transport Zones, DHCP servers and relays, Segments, firewall groups and rules, Gateways, Edge Clusters, and Groups. We can manage these synced items from within Morpheus UI, including the ability to create, edit, and delete them.

Note: The available tabs on the integration detail page will be dependent on the Project selected when the integration was created. Just like in NSX-T, the default view (and thus integrations scoped to the default Project) will have access to all constructs whereas individual Projects will not. Integrations scoped to individual Projects can view the DHCP, Segments, Firewall, Tier-1 Gateways, and Groups tabs but not the other tabs described here. These limitations are identical to those in the NSX-T console UI. More information on NSX-T Projects is available [here](#).

Summary View

The SUMMARY tab contains the default view when accessing an NSX-T integration. From the summary view we can see the status of the NSX-T server, and details about interfaces and group status.

Transport Zones

Access Transport Zones by selecting the Transport Zones tab. The default view of the Transport Zones tab lists Transport zones and presents some detail about them such as name, traffic type, status, and more. The integration allows for creation of new Transport Zones, editing and deleting.

✓ Labs NSX 4

EDIT
DELETE
ACTIONS ▾

Host: <https://prod-dc2-nsx-mgr01.prod.dc2.den.morpheusdata.com> Last Update: 12/18/2023 01:47 PM

Summary
Transport Zones
DHCP
Segments
Firewall
Tier-0 Gateways
Tier-1 Gateways
Edge Clusters
Groups

+ CREATE NSX TRANSPORT ZONE

NAME	DESCRIPTION	TRAFFIC TYPE	STATUS	VISIBILITY	TENANT	
dd-tz-01		OVERLAY	⊖	Private	Morpheus QA	✎ 🗑 ▾ MORE
morpheus-overlay-tz		OVERLAY	✓	Private	Morpheus QA	✎ 🗑 ▾ MORE
morpheus-vlan-tz		VLAN	✓	Private	Morpheus QA	✎ 🗑 ▾ MORE
nsx-overlay-transportzone		OVERLAY	⊖	Private	Morpheus QA	✎ 🗑 ▾ MORE
nsx-vlan-transportzone		VLAN	⊖	Private	Morpheus QA	✎ 🗑 ▾ MORE
prod-dc2-edge-tz		VLAN	✓	Private	Morpheus QA	✎ 🗑 ▾ MORE
prod-dc2-tz		OVERLAY	⊖	Private	Morpheus QA	✎ 🗑 ▾ MORE

DHCP

DHCP servers and relays are displayed on the DHCP tab. View information such as names and server addresses. The integration allows for creation of new servers and relays, editing and deleting.

✓ Labs NSX 4

EDIT DELETE ACTIONS

Host: https://prod-dc2-nsx-mgr01.prod.dc2.den.morpheusdata.com Last Update: 12/18/2023 01:47 PM

- Summary
- Transport Zones
- DHCP**
- Segments
- Firewall
- Tier-0 Gateways
- Tier-1 Gateways
- Edge Clusters
- Groups

DHCP SERVERS

CREATE DHCP SERVER

NAME	SERVER ADDRESS	
prod-dc2-nsx-65114	172.16.14.5/24	
dd-dhcp-test-2	100.96.0.1/30	
dd-dhcp-server-3	100.96.0.1/30	

DHCP RELAYS

CREATE DHCP RELAY

NAME	SERVER ADDRESSES	
prod-dc2-vlan2-dhcp-relay	10.32.21.5	

Segments

Access Segments by from the Segments tab. The summary view includes high-level information such as status, name, network name and CIDR. The integration allows for creating, editing and deleting NSX-T Segments

✓ Labs NSX 4

EDIT DELETE ACTIONS

Host: https://prod-dc2-nsx-mgr01.prod.dc2.den.morpheusdata.com Last Update: 12/18/2023 01:47 PM

- Summary
- Transport Zones
- DHCP
- Segments**
- Firewall
- Tier-0 Gateways
- Tier-1 Gateways
- Edge Clusters
- Groups

+ CREATE NSX SEGMENT

STATUS	NAME	NETWORK NAME	DESCRIPTION	CIDR	
✓	dd-nsx4-seg	dd-nsx4-seg		192.168.10.1/24	
✓	prod-dc2-edge-vlan6	prod-dc2-edge-vlan6		10.32.26.115/24	
✓	prod-dc2-nsx-65114	prod-dc2-nsx-65114		172.16.14.1/24	
✓	prod-dc2-vlan2	prod-dc2-vlan2			
✓	prod-dc2-vlan22	prod-dc2-vlan22			
✓	prod-dc2-vlan6	prod-dc2-vlan6			
✓	segment	segment		10.255.24.2/23	

Firewall

Firewall Groups and Rules are accessible from the Firewall tab. From the summary view, Groups can be expanded to view Rules within. From the ACTIONS menu, create new Groups by selecting “Create Group”. When a Group has been expanded, the “Create Rule” selection within the ACTIONS menu will also be accessible and a new rule can be created within the selected Group. The integration allows for viewing, creating, editing and deleting Firewall Groups and Rules.

✓ Labs NSX 4
EDIT
DELETE
ACTIONS ▾

Host: https://prod-dc2-nsx-mgr01.prod.dc2.den.morpheusdata.com Last Update: 12/18/2023 01:47 PM

Summary
Transport Zones
DHCP
Segments
Firewall
Tier-0 Gateways
Tier-1 Gateways
Edge Clusters
Groups

Enabled: Yes
Auto Save Drafts : true
Identity Firewall Enabled : No

RULES

Search Q ACTIONS ▾

NAME	PRIORITY	CATEGORY	ENABLED	POLICY	DIRECTION	SOURCE	DESTINATION	SERVICES	VISIBILITY	TENANT	
▶ tmtest	20	Infrastructure							Private	Morpheus QA	▾ MORE
▶ Default Layer2 Section	5000000	Ethernet							Private	Morpheus QA	▾ MORE
▶ Default Malicious IP Block Rules	19999999	Infrastructure							Private	Morpheus QA	▾ MORE
▶ Default Layer3 Section	2147483647	Application							Private	Morpheus QA	▾ MORE

Tier-0 Gateways

Access Tier-0 Gateways from the Tier-0 Gateways tab. The integration allows creating, editing and deleting Tier-0 Gateways.

✓ Labs NSX 4
EDIT
DELETE
ACTIONS ▾

Host: https://prod-dc2-nsx-mgr01.prod.dc2.den.morpheusdata.com Last Update: 12/18/2023 01:47 PM

Summary
Transport Zones
DHCP
Segments
Firewall
Tier-0 Gateways
Tier-1 Gateways
Edge Clusters
Groups

Search Q
+ CREATE NSX TIER-0 GATEWAY

STATUS	NAME	DESCRIPTION	VISIBILITY	TENANT	
	prod-dc2-t0		Private	Morpheus QA	▾ MORE

Tier-1 Gateways

Access Tier-1 Gateways from the Tier-1 Gateways tab. The integration allows creating, editing and deleting Tier-1 Gateways.

✓ Labs NSX 4
EDIT
DELETE
ACTIONS ▾

Host: <https://prod-dc2-nsx-mgr01.prod.dc2.den.morpheusdata.com> Last Update: 12/18/2023 01:47 PM

[Summary](#)
[Transport Zones](#)
[DHCP](#)
[Segments](#)
[Firewall](#)
[Tier-0 Gateways](#)
[Tier-1 Gateways](#)
[Edge Clusters](#)
[Groups](#)

+ CREATE NSX TIER-1 GATEWAY

STATUS	NAME	DESCRIPTION	VISIBILITY	TENANT	
✓	prod-dc2-t1		Private	Morpheus QA	MORE

Edge Clusters

View Edge Clusters from the Edge Clusters tab. The default view lists each Edge Cluster with name, member type, cluster profile, and more. The integration allows viewing and limited editing of Edge Clusters.

✓ Labs NSX 4
EDIT
DELETE
ACTIONS ▾

Host: <https://prod-dc2-nsx-mgr01.prod.dc2.den.morpheusdata.com> Last Update: 12/18/2023 01:47 PM

[Summary](#)
[Transport Zones](#)
[DHCP](#)
[Segments](#)
[Firewall](#)
[Tier-0 Gateways](#)
[Tier-1 Gateways](#)
[Edge Clusters](#)
[Groups](#)

NAME	DESCRIPTION	MEMBER TYPE	CLUSTER PROFILE	TRANSPORT NODES	VISIBILITY	TENANT	
prod-dc2-edge-cl1		Edge Node	nsx-default-edge-high-availability-profile	0	Private	Morpheus QA	MORE

Groups

NSX-T Groups are viewed from the Groups tab. The default view lists each Group alone with member details. The Morpheus NSX-T integration allows for creating, editing and deleting Groups.

✓ Labs NSX 4

EDIT DELETE ACTIONS ▾

Host: https://prod-dc2-nsx-mgr01.prod.dc2.den.morpheusdata.com Last Update: 12/18/2023 01:47 PM

Summary Transport Zones DHCP Segments Firewall Tier-0 Gateways Tier-1 Gateways Edge Clusters **Groups**

Search

+ CREATE GROUP

NAME	DESCRIPTION	MEMBERS	VISIBILITY	TENANT
dd-test-group			Private	
DefaultMaliciousIpGroup	Default Malicious IP group	1 Criteria	Private	
Edge_NSGroup	Edge VMs are added to DFW exclude list via Edge_NSGroup	1 Criteria	Private	
IP Address Group		1 IPS / MACS	Private	
MetricsSystem_NSGroup	List of Transport Nodes and Edge Nodes in MetricsSystemVM_NSGroup	2 Criteria	Private	
ServiceInsertion_NSGroup	SVMs deployed by ServiceInsertion are added to this NSGroup		Private	
SystemVM_NSGroup	System VMs are added to DFW exclude list via SystemVM_NSGroup		Private	

Cisco ACI

Overview

✓ ACI Sandbox

EDIT DELETE



Host: https://sandboxapicdc.cisco.com

SUMMARY TENANTS **APPLICATION PROFILES** ENDPOINT GROUPS CONTEXTS BRIDGE DOMAINS FILTERS CONTRACTS

APPLICATION PROFILES

Search

+ CREATE ACI APP PROFILE

NAME	DESCRIPTION	DN	
A1_API		uni/tn-A1/ap-A1_API	Remove
A2_API		uni/tn-A2/ap-A2_API	Remove
access		uni/tn-infra/ap-access	Remove
apBankPac		uni/tn-BankPac/ap-apBankPac	Remove
appProfile		uni/tn-zxdf/ap-appProfile	Remove

Add ACI as a network and security integration. Inventory your existing ACI configurations. Create networks, bridge domains, application profiles, tenants, endpoint groups, contexts, filters and contracts. Provision instances into new endpoint groups and define security groups that apply contracts on provision.

From Morpheus below can be created:

- Tenants
- ANP's
- EPG's
- Contexts
- Bridge Domains

- Filters
- Contracts

Note: Morpheus to ACI Sync Job Schedule: Every 5 minutes

Note: Morpheus connects to ACI APIC over port 443

Add Network Integration

1. Navigate to Infrastructure > Networks > Integrations
2. Select *+ADD > Networking > Cisco ACI*
3. Populate the following:

NAME

ACI Integration Name/Label in Morpheus This is unique to Morpheus and not part of authentication

URL ACI fabric url, eg `https://apicdc.company.com`

USERNAME ACI aaaUser name attribute

PASSWORD ACI aaaUser pwd attribute

TENANT Populates upon authentication, tenant selection not required

4. Select *ADD NETWORK INTEGRATION*

Configure Cloud Network Mode

For your ACI Integration to be available during provisioning, ACI needs to be defined on a Cloud or multiple Clouds **NETWORK MODE** advanced options.

1. Select an existing VMware vCenter Cloud
2. Select *EDIT*
3. Expand the *Advanced Options* section
4. Select ACI Integration in **NETWORK MODE** dropdown
5. Select *SAVE*

Instance Provisioning

▼ ACI Options

ENDPOINT GROUP

APP PROFILE

▼ ACI Security

CONSUMES

PROVIDES

Once ACI is integration to a cloud, it can be used during instance provisioning:

1. From the EPG drop down, either an existing EPG can be selected or a new one can be created. It is the same for ANP, either create a new one or choose an existing.
2. Under ACI security consumes and provides, contracts can be searched when you enter a name. When the provisioning wizard is completed, it will provision the instance and apply the ACI options and Security. This can be viewed under the instance page, or via REST API and CLI.

Blueprint Configuration

▼ ACI Options

ENDPOINT GROUP

APP PROFILE

▼ ACI Security

CONSUMES

PROVIDES

- In a Blueprint, you can define the ANP and EPG of each Tier
- Variables can be used for EPG and ANP names.
- This could be useful to create blueprints for dev testing to isolate from prod networks.
- This can be hybrid based on the VMM domains in APIC.

Bluecat

Overview

Morpheus integrates with Bluecat IPAM to scope pools to networks for static IP assignment from Bluecat to your Morpheus Instances.

Adding Bluecat to Morpheus

1. Navigate to **Infrastructure > Network > Integrations**
2. Click + *ADD*
3. Select *Bluecat*
4. Enter in the following information

Name Name of the Bluecat Integration in Morpheus

Enabled Uncheck to disable sync with the Bluecat endpoint

URL URL of the Bluecat server, ex: `http://10.30.20.10`

Username Username of Bluecat API User. API and root level propagating read access required, read/write access required for target Networks and Domains.

Password Bluecat User password

Network Filter Optionally enter a comma-separated list of `entityIds` for config(s), block(s) or network(s) (ex. `entityId1,entityId2...`) to filter the synced IP pools, networks and domains

5. Click *SAVE CHANGES*

The Bluecat Integration will be saved, IP pools will sync in and populate under **Infrastructure > Network > IP Pools**, and Domains will populate in **Infrastructure > Network > Domains**. Pools and Domains can also be found in the Bluecat Integration details page, which can be accessed by clicking on the name of the added Bluecat Integration in **Infrastructure > Network > Services**.

Important: *Quick Deployments* must be enabled in Bluecat for Morpheus to create instantly available DNS records when using Bluecat DNS.

Adding IP Pools to Networks

Morpheus can automatically assign the next available Bluecat IP in an IP/Network Pool and create the corresponding DNS records, as well as remove the records upon teardown. To enable this, add an Bluecat IP/Network Pool to the *Network Pool* section on a Network(s).

1. Navigate to **Infrastructure - Network- Networks**
2. Select a Network name and **EDIT**, or select *ACTIONS - Edit*
3. In the *NETWORK POOL* section, search for and select the name of the IP/Network Pool.
 - Gateway, DNS and CIDR must be populated for static/pool IP assignment
 - Select *Allow IP Override* to allow selecting between DHCP, Static entry and Pool Selection at provision time

- Deselect DHCP server if a DHCP server will not be used on the network (only static and/or IP Pool IP assignment)
4. Select *SAVE CHANGES*

SolarWinds IPAM

Features

- Automate static IP assignment across environments using Solarwinds IPAM
- Network pool sync: Network pools can be set on networks in Morpheus for automated IP allocation and record creation
- Optional Network Pool allocation and record sync. **Inventory Existing** option syncs all individual IP records and corresponding status. Inventory is not required for provisioning
- Grid and list displays with IP record overlays and color coding for static, available, reserved and transient status
- Manual IP Host record creation from Network Pool detail pages

Adding SolarWinds to Morpheus

1. Navigate to **Infrastructure > Network > Integrations**
2. Click + *ADD*
3. Select *SolarWinds*
4. Enter in the following information

Name Name of the SolarWinds Integration in Morpheus

Enabled Deselect to disable sync with the SolarWinds endpoint

URL URL of the SolarWinds server, ex: `http://10.30.20.10:17778`

Username Username of SolarWinds API User. See the NOTE box below for information on minimum rights requirements

Password SolarWinds User password

5. Click *SAVE CHANGES*

Note: At minimum you will need credentials for a user with API and root-level propagating read access, as well as read/write access for target networks and domains. For a quicker solution, you can also use an account with the Power User role in situations where you aren't concerned with providing only the minimum required access.

Consuming SolarWinds in Morpheus

On saving your new integration, SolarWinds networks will be synced and can be viewed by navigating to Infrastructure > Network > IP POOLS. They're also viewable from the detail section of the SolarWinds integration at Infrastructure > Network > INTEGRATIONS > (your SolarWinds integration) > NETWORK POOLS.

✓ Labs SolarWinds EDIT DELETE

solarwinds

Host: https://10.30.22.213:17778 Last Update: 01/30/2020 10:00 AM

NETWORK POOLS NETWORKS ZONES

NETWORK POOLS

Search

NAME	TYPE	NETWORK
vLAN0002 - Internal Server	SolarWinds	
OneNeck	SolarWinds	
vLAN0000 - SolarWinds Managed Pool	SolarWinds	

Host records can also be viewed here by clicking on the name of a SolarWinds network.

✓ vLAN0002 - Internal Server

solarwinds

Pool Service: Labs SolarWinds

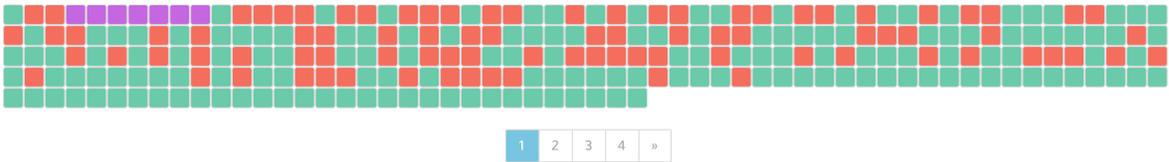
▼ INFO

Pool Type: SolarWinds CIDR: 10.30.20.0/22 Netmask: 255.255.252.0
Gateway: DHCP:

▼ HOST RECORDS

Grid List + ADD

10.30.20.1 - 10.30.20.255



Note: Morpheus SolarWinds integration does not support zone record syncing despite the presence of the ZONES tab on the integration detail page. This is a UI feature carried over from other networking integrations and is not supported at this time.

Adding IP Pools to Networks

Morpheus can automatically assign the next available SolarWinds IP in an IP/Network Pool and create the corresponding DNS records. Morpheus will also remove the records upon teardown. To enable this, add an SolarWinds IP/Network Pool to the *Network Pool* section on a Network(s).

1. Navigate to **Infrastructure > Network > Networks**
2. Select a Network name and click *EDIT*
3. In the NETWORK POOL typeahead field, search for and select the name of the selected IP/Network Pool.
 - Gateway, DNS, and CIDR must be populated for static/pool IP assignment
 - Select **ALLOW IP OVERRIDE**, if desired, to allow selecting between DHCP, static IP address entry, and pool address selection at provision time
 - Deselect DHCP server if a DHCP server will not be used on the network (only static and/or IP Pool IP assignment)
4. Select *SAVE CHANGES*

Creating Host Records

1. Select a Network Pool from **Infrastructure > Network > IP POOLS**
2. Click + *ADD*
3. Enter the following
 - HOSTNAME** Hostname for the record
 - IP ADDRESS** IP address for the Host Record
4. Select *SAVE CHANGES*



CREATE HOST RECORD

HOSTNAME

IP ADDRESS

SAVE CHANGES

Unisys Stealth

Introduction

Unisys Stealth is a network security tool for safeguarding sensitive information across shared networks. By creating pre-defined communities of interest (COIs) and curating user access to these groups, the need to create separate networks for handling restricted data is reduced.

Morpheus includes a full integration with Stealth allowing administrators to create and manage COIs, work with configurations and roles, and provision new endpoints into COIs.

Stealth Concepts

- **Communities of Interest (COIs):** A collection of endpoints cryptographically separated so they can only communicate to each other
- **Endpoints:** Any system or device with a Stealth agent
- **Configuration:** Tells the endpoints which authorization methods and services to use for obtaining COI membership
- **Role:** Defines COI membership. Users and groups are assigned to a role which grants access to that role's COIs
- **Filters:** Customizes Stealth communication. More specifically, filters constrain Stealth communication to specific addresses, ports, or protocols and allow Stealth endpoints to communicate with non-Stealth endpoints

Integrating Stealth with Morpheus

1. Navigate to Infrastructure > Network > Integrations
2. Click + *ADD*
3. Complete the following fields:
 - **NAME:** A name for this Stealth integration in Morpheus
 - **API HOST:** The address for the server hosting Stealth (ex: <https://x.x.x.x:8448>)
 - **USERNAME:** A username for the portaladmin, or the user who logs into the web console
 - **PASSWORD:** A password for the account
 - **MANAGER USERNAME:** A username for the manager account
 - **MANAGER PASSWORD:** A password for the manager account
4. Click *ADD SECURITY INTEGRATION*

ADD SECURITY INTEGRATION ✕

NAME

API HOST

USERNAME

PASSWORD

MANAGER USERNAME

MANAGER PASSWORD

▶ Advanced Options

ADD SECURITY INTEGRATION

Summary View

The default view when accessing a Stealth integration in Morpheus is the Summary view. In addition to basic information about the Stealth server itself, we can see system status, license and service information.

Networks > Security Services > Labs Stealth

✓ Labs Stealth

EDIT

DELETE

ACTIONS ▾

Last Update: 06/10/2020 09:47 AM

Summary

Endpoints

Configurations

Roles

COIs

Filters



Name: Labs Stealth

Type: Stealth

Date Created: 2020-05-08 18:50:43.0

▼ STEALTH SYSTEM STATUS

STATUS	NAME	LOCATION	ROLES
!			License, Authorization
!			Management, License, Authorization
✓			License, Authorization

▼ STEALTH LICENSE

STATUS	TYPE	USED	AVAILABLE	MAX
✓	Client	0	25	25
✓	Server	2	23	25
✓	Mobile	0	25	25
✓	VGD	0	25	25
✓	AWSMarketplace	0	0	0

Endpoints

Endpoints are any system or device with a Stealth agent. Stealth endpoints can be provisioned in Morpheus in the same way other cloud resources are provisioned. With Stealth integrated, workloads provisioned on the appointed networks are assigned Stealth configuration and a Stealth role during the provisioning process. Based on a user's assigned roles and COIs assigned to those roles, only workloads on the appointed COIs will be visible to the user. Additionally, workloads can only see other workloads within their COI.

Note: Machines on the same network which are not Stealth-managed will be able to see and communicate with each other but will not be able to see workloads which are assigned to a Stealth COI.

Endpoints View

The endpoints view will display all available Stealth-managed resources. Endpoints are not created here but will be synced into this view as they are created (through Morpheus provisioning or outside creation). Stealth-managed endpoints can be deleted by clicking the trash can icon at the end of each row in this view.

The following fields are exposed in the endpoints list view:

- **DISPLAY NAME**
- **NAME**

- **DESCRIPTION**

Configurations

Configurations in Stealth are the top-level construct and house COIs, roles, groups, users and endpoints. Your Stealth integration will include at least one configuration but often they will include more.

Configurations are primarily created and managed from the Stealth console but we can opt to remove them from Morpheus by clicking the trash can icon at the end of each row on the configurations list page. Configurations are selected along with a Stealth role at provision time in Morpheus.

Configurations View

The following fields are exposed in the configurations list view:

- **NAME:** The name of the configuration
- **DESCRIPTION:** A description value for the configuration

Roles

Users are placed into roles which are associated with COIs. A user's role(s) determine which COIs he or she will be able to see and access. Roles are synced into Morpheus once the integration process is complete and can be viewed in the Roles list view. Roles can also be created from the Morpheus integration as described later in this section.

Roles View

The following fields are exposed in the roles list view:

- **NAME:** The name of the role
- **DESCRIPTION:** A description value for the role

Note: More detail on each item in the roles list can be revealed by clicking on the (i) icon in each row, including the COIs associated with the role.

Creating Stealth Roles

1. Navigate to Infrastructure > Network > Integrations > (Your Stealth integration) > Roles
2. Click + *CREATE ROLE*
3. Complete the following fields:
 - **NAME:** The name for the new role
 - **DESCRIPTION:** A description value for the new role
 - **CONFIGURATION:** Select an existing Stealth configuration to associate with the role
 - **ROLE TYPE:** Identifies how the role is used. Can be Global (for roles used to isolate endpoints and users), Service (for roles used by endpoints in service mode to access an authorization service) or WorkGroup (for roles used by endpoints in normal operation)

- **FILTER SET:** Choose a filter set to apply to the role to allow or deny clear text communication with non-Stealth-managed endpoints
- **COIs:** Select the COIs to be associated with the role
- **PROVISION CHANGES:**

4. Click *ADD ROLE*

ADD ROLE [Close]

NAME []

DESCRIPTION []

CONFIGURATION [Management]

ROLE TYPE [Global]

FILTER SET [Labs Networking]

COIS [Select] +

PROVISION CHANGES

ADD ROLE

COIs (Communities of Interest)

COIs exist within configurations and create a logical separation between endpoints in separate COIs. Communication between endpoints in the COI is encrypted and those outside the COI are unable to see or access endpoints despite being on the same network.

On completing the integration, Morpheus will sync in existing COIs. COIs can also be created from Morpheus UI which is described later in this section. COIs are deleted by clicking on the trash can icon at the end of each row in the COIs list view.

COIs View

The following fields are exposed in the roles list view:

- **NAME:** The name of the COI
- **DESCRIPTION:** A description value for the COI

Creating Stealth COIs

1. Navigate to Infrastructure > Network > Integrations > (Your Stealth integration) > COIs
2. Click + *CREATE COI*
3. Complete the following fields:
 - **NAME:** The name for the new COI
 - **DESCRIPTION:** A description value for the new COI
 - **TYPE:** Workgroup or Service
 - **DIRECTION:** Default (enables COI to accept inbound and initiate outbound tunnels), Initiate (restricts the COI to only initiate outbound tunnels), or Accept (restricts the COI to only accept inbound tunnels)
4. Click *CREATE COI*

CREATE COI

NAME

DESCRIPTION

TYPE

DIRECTION

CREATE COI

Filters

Filters customize Stealth communication. More specifically, filters constrain Stealth communication to specific addresses, ports, or protocols and allow Stealth endpoints to communicate with non-Stealth endpoints.

Filters are synced into Morpheus when integrating with Stealth and are viewable from the filters list view. They are created and managed from within the Stealth console itself.

When accessing the filters list view, all filter sets are displayed. Each filter set can be expanded to view the individual filters within. Information on each filter is displayed once the filter set has been expanded to reveal the individual filters.

Provisioning with Stealth

In order to provision new Stealth-managed endpoints, Stealth must be integrated with Morpheus as described above. In addition, Stealth must be selected as the Security Server for the cloud you're provisioning into. Security servers can be selected at the time a new Cloud integration is created or by editing an existing Cloud integration.

Choosing a Cloud Security Server

Assuming the Cloud is already integrated with Morpheus, use the steps below to set the security server and activate Stealth prompts at provision time. The steps to set the security server during the time the cloud is initially integrated with Morpheus is very similar.

1. Navigate to Infrastructure > Clouds > (Your Selected Cloud)
2. Click *EDIT*
3. Click on Advanced Options to reveal additional selections
4. In the dropdown for SECURITY SERVER, choose an existing Stealth integration

Provisioning to a Stealth-enabled Cloud

Once we have selected our Stealth integration as the security server for at least one Cloud in Morpheus, new Instances (endpoints) can be provisioned and managed by Stealth.

1. Navigate to Provisioning > Instances
2. Click + *ADD*
3. Select the Instance Type, Cloud, and Group making sure to choose a Cloud that has been set up for an existing Stealth integration
4. On the Configure tab of the provisioning wizard, choose a Stealth Configuration and a Stealth Role according to the needs of the machine(s) being provisioning
5. Once the provisioning process is complete, the new Stealth-managed endpoints will be available and restricted based on the Stealth implementation

CREATE INSTANCE
✕

TYPE >
GROUP >
CONFIGURE >
AUTOMATION >
REVIEW

Configuration Options

VERSION

7.5

▼

LAYOUT

VMware VM

▼

PLAN

1 CPU, 512MB Memory

▼

Cores: 1 Memory: 512 MB Price: \$33.50 / Month ⓘ

RESOURCE POOL

[blurred]

▼

VOLUMES

root

10

GB

SCSI 0:0

▼

Auto - Datastore

+

NETWORKS

[blurred]

▼

VMXNET 3

▼

Network Default

+

HOST

Select

▼

FOLDER

Tests

▼

STEALTH CONFIGURATION

Select

▼

STEALTH ROLE

Select

▼

▶ User Config

▶ Advanced Options

PREVIOUS

NEXT

EfficientIP SOLIDserver

Features

- Network Pools synchronization
- DNS Zone & Zone record synchronization
- Host Record synchronization
- Total & Free IP status bar for networks
- Network Grid and List view with IP Status and records, date and user tracking
- Automatic and manual IP Reservations, DNS A/PTR record creation and deletion

Required Role Permissions

Add, edit and remove EfficientIP SOLIDserver integrations

- Infrastructure: Network Integration: Full

View and edit synced IP pools

- Infrastructure: Network IP Pools: Full

View networks and add synced IP pools to networks

- Infrastructure: Networks: Full or Group

View and edit synced DNS zones, including creation of zone records

- Infrastructure: Network Domains: Full

Adding EfficientIP SOLIDserver Integration

The EfficientIP SOLIDserver integration type is a plugin that must be added to Morpheus before the option to create one will be available. In the future, users will be able to download this and other plugin types from a centralized marketplace. For now, the plugin jar file can be compiled from a public Github repository or can be requested from your account team. See the [Plugins Section](#) of Morpheus documentation for more on the process of uploading the plugin JAR to your appliance.

1. Navigate to Infrastructure > Network > Integrations and click + *ADD*
2. Under the IPAM section, select EfficientIP SOLIDserver
3. Configure the following:
 - **NAME:** Friendly name for this EfficientIP SOLIDserver integration
 - **ENABLED:** When checked, this integration will be accessible in Morpheus
 - **API URL:** The FQDN for the EfficientIP server, not a specific path
 - **USERNAME:** The username for an EfficientIP service account. Bear in mind this account will need API access as well as the rights to work with pools, zones, and records you wish to consume from Morpheus
 - **PASSWORD:** The password for the above named account
 - **THROTTLE RATE:** In larger environments, it may be necessary to introduce a rate limit on calls to the EfficientIP API from Morpheus. If the EfficientIP console UI becomes less responsive than it was prior to integration with Morpheus, it may be due to a high number of API calls in the background from Morpheus. In such a case, start with a 50ms throttle rate and adjust accordingly depending on performance
 - **DISABLE SSL SNI VERIFICATION:** If necessary, disable the check for a valid SSL certificate on the EfficientIP server
 - **INVENTORY EXISTING:** When checked, used IP space will be continually synced between Morpheus and EfficientIP. If left unchecked, only IP space claimed (and freed) from Morpheus is shown on the detail page for the EfficientIP pool
4. Click *SAVE CHANGES*

Once saved, Morpheus will begin to onboard data from EfficientIP. EfficientIP networks are viewable in Infrastructure > Network > IP Pools under the IP Pools tab. Depending on EfficientIP configuration, you may see up to two “types” of Network Pools sync from EfficientIP, SOLIDserver Subnet and SOLIDserver Pool. In EfficientIP, “pools” are an optional construct that subdivides subnets. In Morpheus, both constructs are synced which gives an additional layer of organization when linking Network Pools with Networks (described in the next section) for organizations that use the pools construct. Within a selected IP Pool, host records will also sync and can be viewed in a grid or list layout. DNS

Zones are synced under Infrastructure > Network > Domains. By clicking into the domain, DNS Zone records can be viewed.

The screenshot shows the Morpheus web interface. The top navigation bar includes the Morpheus logo, a search bar, a notification bell with a red '7', a 'Support' dropdown, and a user profile for 'Alex Harker'. Below this is a secondary navigation bar with tabs for Operations, Provisioning, Library, Infrastructure (selected), Backups, Monitoring, Tools, and Administration. A third navigation bar contains icons for Groups, Clouds, Clusters, Compute, Network (selected), Load Balancers, Storage, Trust, and Boot. The main content area shows the breadcrumb 'Network Pools > VLAN13 EfficientIP Segment'. A green checkmark icon is next to the title 'VLAN13 Network - VLAN13 EfficientIP Segment'. Below the title is the 'efficient iP' logo and the text 'Pool Service: Denver EIP'. A dropdown menu is open under 'INFO', showing fields for Pool Type (SolidServer Pool), Gateway, CIDR, DHCP, and Netmask. Another dropdown menu is open under 'HOST RECORDS', showing a grid view of IP addresses. The grid shows a range of 10.32.36.10 - 10.32.36.73. The first few addresses are highlighted in red, and the rest are green. There are buttons for 'Grid', 'List', and '+ ADD'.

Adding IP Pools to Networks

At provision time, Morpheus can automatically assign the next available IP address in an EfficientIP pool and create the corresponding DNS records. Morpheus can also clean up DNS records and free up IP address space on teardown. In order to enable this functionality, add an EfficientIP IP Pool as the Network Pool for an existing network (or networks).

1. Navigate to Infrastructure > Network > Networks
2. Select a network to view the network detail page and click *EDIT*
3. In the typeahead field for NETWORK POOL, search for and select the EfficientIP pool
4. Click *SAVE CHANGES*

Note: Gateway, DNS and CIDR must be populated for static/pool IP assignment. If desired, select “Allow IP Override” to allow selecting between DHCP, Static entry, and pool selection at provision time. Finally, deselect “DHCP server” if a DHCP server will not be used on the network (only static and/or IP Pool assignment).

Creating Host Records

1. Select an EfficientIP Network Pool from Infrastructure > Network > IP Pools
2. Select + *ADD*
3. Configure the following:
 - **HOSTNAME:** The hostname for the record
 - **IP ADDRESS:** The IP address for the host record
 - **DOMAIN:** Select an EfficientIP zone
 - **CREATE DNS RECORDS:** If selected, DNS A and PTR records will be created in EfficientIP

1. Click *SAVE CHANGES*

CREATE HOST RECORD
×

HOSTNAME

IP ADDRESS

DOMAIN

CREATE DNS RECORDS

SAVE CHANGES

Creating Zone Records

1. Select an EfficientIP zone from the domains list at Infrastructure > Network > Domains
2. Click + *ADD* on the Zone Records tab
3. Configure the following:
 - **NAME:** The name for the records (hostname)
 - **TYPE:** The record type: A, AAAA, CNAME, MX, NS, PTR, SOA, or TXT
 - **CONTENT:** The content of the record, such as IP address or A record
 - **TTL:** The time to live value
1. Click *SAVE CHANGES*

CREATE ZONE RECORD ✕

NAME

TYPE

CONTENT

TTL

5.13.27 Storage

3Par

Adding 3Par Storage Server

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the SERVERS tab, Click the + *ADD* button.
4. From the ADD STORAGE SERVER wizard input the following:
 - NAME** Name of the Storage Server in Morpheus
 - TYPE** Select *3Par*
 - URL** URL Of 3Par Server Example : *https://192.168.190.201:8008*
 - USERNAME** Add your administrative user account.
 - PASSWORD** Add your administrative password.
5. Select *SAVE CHANGES*

The 3Par Storage Server will be added and displayed in the Buckets tab.

Buckets, Files Shares and Storage Groups will be synced in.

AzureStorage

To Add Azure Storage

1. Navigate to *Infrastructure > Storage*
2. Select + *ADD*
3. From the New Storage Provider Wizard input the following:
 - Name** Name of the storage provider.
 - Provider Type** Azure
 - Storage Account** Add Storage Account

Storage Key Add Storage Key

Share Name Add Share Name

Targets

- Default Backup Target
- Default Deployment Archive Target
- Default Virtual Image Store

4. *Save Changes*

Dell ECS

Overview

Morpheus integrates with DELL EMC ECS via the ECS api. This allows Morpheus to talk directly to the ECS services.

When you add a ECS Server, Morpheus will sync in the following.

- Storage Groups
- Buckets
- File shares

Users will be able to create the following items within ECS without direct access to the ECS console.

- Buckets
- File shares

Storage Servers

The first step in the Dell EMC ECS integration is to add a Dell EMC ECS Storage Server. Once added, Buckets, Files Shares and Storage Groups will be synced in and can be access and managed in Morpheus.

Adding Dell EMC ECS Storage Server

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the SERVERS tab, Click the + *ADD* button.
4. From the ADD STORAGE SERVER wizard input the following:

NAME Name of the Storage Server in Morpheus

TYPE Select *Dell EMC ECS*

URL URL Of DELL EMC ECS Server Example : *https://192.168.190.200:4443*

Tip: The port 4443 is the api port for ECS api. This may be different depending on your configuration

USERNAME Add your administrative user account.

PASSWORD Add your administrative password.

S3 SERVICE URL (Optional) Add your S3 service url Example: <http://192.168.190.220:9020>

Note: S3 SERVICE URL is not required if you are not planning on using ECS S3.

5. Select *SAVE CHANGES*

The Dell EMC ECS Storage Server will be added and displayed in the Buckets tab.

Buckets, Files Shares and Storage Groups will be synced in.

Buckets

- **Buckets** will be listed in *Infrastructure - Storage - Buckets*
 - Buckets can be created and deleted with *Infrastructure - Storage* Role Permissions
 - Buckets can be browsed with *Infrastructure: Storage Browser* Role permissions
 - File and folders can be uploaded, downloaded and deleted with Full *Infrastructure: Storage Browser* Role permissions.

Adding Dell EMC ECS Buckets

Note: A Dell ECS Storage Server must be configured in *Infrastructure - Storage - Servers* prior to adding a Dell ECS Bucket.

To Add a Dell ECS Storage Bucket:

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the BUCKETS tab, Click the + *ADD* button.
4. Select *Dell EMC ECS Bucket* from the dropdown list
5. From the NEW BUCKET Wizard input the following:

NAME Name of the Bucket in Morpheus.

STORAGE SERVICE Select existing Dell EMC ECS Storage Server (configured in *Infrastructure - Storage - Servers*)

BUCKET NAME Enter a name for the new Dell ECS bucket.

USER Your Dell EMC ECS S3 user account

SECRET KEY

Your Dell EMC ECS S3 Secret Example: `jW+pFyAPtSS5FuEqKwt44xlpM/2`

NAMESPACE Select Dell EMC ECS Namespace for the Bucket

STORAGE GROUP Select a Dell EMC ECS Storage Group

Default Backup Target Sets this bucket as the default backup target when creating Backups. If selected the option to update existing Backup configuration to use this Bucket will be presented.

Archive Snapshots Enabled to export VM snapshots to this Bucket when creating VMware Backups, after which the snapshot will be removed from the target hypervisor.

Default Deployment Archive Target Sets this Bucket as the default storage target when uploading Deployment files in the *Deployments* section.

Default Virtual Image Store Sets this bucket as the default storage target when uploading Virtual Images from the *Virtual Images* section, importing Images from Instance Actions, creating Images with the *Image Builder* and when creating new images from *Migrations*.

RETENTION POLICY

None Files in the Bucket will not be automatically deleted or backed up.

Backup Old Files

This option will backup files after a set amount of time and remove them from the bucket.

DAYS OLD Files older than the set number of days will be automatically backed up to the selected Backup Bucket.

BACKUP BUCKET Search for and select the Bucket the files will be backed up to.

DELETE OLD FILES

This option will delete files from this bucket after a set amount of days.

DAYS OLD Files older than the set number of days will be automatically deleted from the Bucket.

6. Select *SAVE CHANGES*

The Bucket will be created and displayed in the Buckets tab.

- To browse, upload, download, or delete files from this Bucket, select the name of the Bucket.
- To edit the Bucket, select the edit icon or select the name of the Bucket and select *ACTIONS - EDIT*.

Warning: Repointing a bucket that is in use may cause loss of file references. Ensure data is mirrored first.

- To delete a Bucket, select the trash icon or select the name of the Bucket and select *DELETE*.

Warning: When deleting a Bucket, all Deployment Versions and Backups associated with the Bucket will be deleted.

Add Dell EMC ECS File Shares

To Add a Dell EMC ECS File Share:

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the FILE SHARES tab, Click the + *ADD* button.
4. Select *Dell EMC ECS Share* from the dropdown list
5. From the NEW FILE SHARE Wizard input the following:
 - NAME** Name of the File Share in Morpheus.

STORAGE SERVICE Select existing Dell EMC ECS Storage Server (configured in *Infrastructure - Storage - Servers*)

SHARE PATH

Enter Dell EMC ECS Share Path Example: `ecs-file-share-1`

USER Dell EMC ECS User

SECRET KEY Dell EMC ECS Secret key

Volume Size Specify volume size for the File Share (in MB)

Allowed IP's

Specify IP Addresses to limit accessibility to the File Share

Leave blank for open access Click the + symbol to the right of the first ALLOWED IPS field to add multiple IP's

NAMESPACE Select Dell EMC ECS Namespace (synced)

STORAGE GROUP Select Dell EMC ECS Storage Group (synced)

Default Backup Target Sets this File Share as the default backup target when creating Backups. If selected the option to update existing Backup configuration to use this File Share will be presented.

Archive Snapshots Enabled to export VM snapshots to this File Share when creating VMware Backups, after which the snapshot will be removed from the source Cloud.

Default Deployment Archive Target Sets this File Share as the default storage target when uploading Deployment files in the *Deployments* section.

Default Virtual Image Store Sets this File Share as the default storage target when uploading Virtual Images from the *Virtual Images* section, importing Images from Instance Actions, creating Images with the *Image Builder* and when creating new images from *Migrations*.

RETENTION POLICY

None Files in the File Share will not be automatically deleted or backed up.

Backup Old Files

This option will backup files after a set amount of time and remove them from the File Share.

DAYS OLD Files older than the set number of days will be automatically backed up to the selected Backup File Share.

BACKUP File Share Search for and select the File Share the files will be backed up to.

DELETE OLD FILES

This option will delete files from this File Share after a set amount of days.

DAYS OLD Files older than the set number of days will be automatically deleted from the File Share.

6. Select *SAVE CHANGES*

The File Share will be created and displayed in the File Shares tab.

- To browse, upload, download, or delete files from this File Share, select the name of the File Share.
- To edit the File Share, select the edit icon or select the name of the File Share and select *ACTIONS - EDIT*.

Warning: Repointing a File Share that is in use may cause loss of file references. Ensure data is mirrored first.

- To delete a File Share, select the trash icon or select the name of the File Share and select *DELETE*.

Warning: When deleting a File Share, all Deployment Versions and Backups associated with the File Share will be deleted.

Isilon

Add Dell EMC Isilon Storage Server

Important: Enable insecure mode on the NFS settings. This allows non-root ports to be used. Setting the insecure/privileged mode will require a restart of the Isilon nodes.

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the SERVERS tab, Click the + *ADD* button.
4. From the ADD STORAGE SERVER wizard input the following:
 - NAME** Name of the Storage Server in Morpheus
 - TYPE** Select *Dell EMC Isilon*
 - URL** URL Of Dell EMC Isilon Server Example : *https://192.168.190.202:8080*
 - USERNAME** Add your administrative user account.
 - PASSWORD** Add your administrative password.
 - PROVISION USER** Select Provision User
 - PROVISION GROUP** Select Provision Group
 - ROOT PATH**
 - Enter Root Path** Example : \
5. Select *SAVE CHANGES*

The Dell EMC Isilon Storage Server will be added and displayed in the Buckets tab.

Buckets, Files Shares and Storage Groups will be synced in.

Add Dell EMC Isilon File Share

To Add a Dell EMC Isilon File Share:

1. Select the Infrastructure link in the navigation bar.
2. Select the Storage link in the sub navigation bar.
3. In the FILE SHARES tab, Click the + *ADD* button.
4. Select *Dell EMC Isilon Share* from the dropdown list
5. From the NEW FILE SHARE Wizard input the following:

NAME Name of the File Share in Morpheus.

STORAGE SERVICE Select existing Dell EMC Isilon Storage Server (configured in *Infrastructure - Storage - Servers*)

SHARE PATH

Enter Dell EMC Isilon Share Path Example: `ecs-file-share-1`

Volume Size Specify volume size for the File Share (in MB)

Allowed IP's

Specify IP Addresses to limit accessibility to the File Share

Leave blank for open access Click the + symbol to the right of the first ALLOWED IPS field to add multiple IP's

NAMESPACE Select Dell EMC Isilon Namespace (synced)

STORAGE GROUP Select Dell EMC Isilon Storage Group (synced)

Default Backup Target Sets this File Share as the default backup target when creating Backups. If selected the option to update existing Backup configuration to use this File Share will be presented.

Archive Snapshots Enabled to export VM snapshots to this File Share when creating VMware Backups, after which the snapshot will be removed from the source Cloud.

Default Deployment Archive Target Sets this File Share as the default storage target when uploading Deployment files in the *Deployments* section.

Default Virtual Image Store Sets this File Share as the default storage target when uploading Virtual Images from the *Virtual Images* section, importing Images from Instance Actions, creating Images with the *Image Builder* and when creating new images from *Migrations*.

RETENTION POLICY

None Files in the File Share will not be automatically deleted or backed up.

Backup Old Files

This option will backup files after a set amount of time and remove them from the File Share.

DAYS OLD Files older than the set number of days will be automatically backed up to the selected Backup File Share.

BACKUP File Share Search for and select the File Share the files will be backed up to.

DELETE OLD FILES

This option will delete files from this File Share after a set amount of days.

DAYS OLD Files older than the set number of days will be automatically deleted from the File Share.

6. Select *SAVE CHANGES*

The File Share will be created and displayed in the File Shares tab.

- To browse, upload, download, or delete files from this File Share, select the name of the File Share.
- To edit the File Share, select the edit icon or select the name of the File Share and select *ACTIONS - EDIT*.

Warning: Repointing a File Share that is in use may cause loss of file references. Ensure data is mirrored first.

- To delete a File Share, select the trash icon or select the name of the File Share and select *DELETE*.

Warning: When deleting a File Share, all Deployment Versions and Backups associated with the File Share will be deleted.

5.13.28 Supported Integration Versions

Morpheus supports an extensive range of software integrations and versions past and present. Current iterations of Amazon AWS, Microsoft Azure, Google Cloud Platform, Digital Ocean, OpenTelekom Cloud, IBM Bluemix, Softlayer and UpCloud are all supported.

In addition, Morpheus is verified to work with, but not limited to:

Integrations

Note: Current iterations of Amazon AWS, Microsoft Azure, Google Cloud Platform, Digital Ocean, OpenTelekom Cloud, IBM Bluemix, Softlayer and UpCloud are all supported.

Table 25: Integration Compatibility

Integration	Supported Version(s)	Notes
Ansible	2.7.x or higher	
Ansible Tower	3.8.x	
App Dynamics	4.5.x	
Ava-mar	18	
Azure Stack	2002 back to 1908	2019-03-01-hybrid api-profile version used which is supported in 1908 and later Azure Stack versions
Cisco ACI	3.x,4.x,5.x	
Citrix Netscaler	v12.1	
Comm-vault	v11 sp 19	
F5 Big-IP	11.4+	
In-foblox	Latest Versions Supported	
Kubernetes	1.21+	
Microsoft Hyper-V	2012R2, 2016, 2019	
Nutanix AHV	5.0+	In 5.5 - 5.7 if Prism Central is managing Prism Element, image creation will not function due to PC Image Management.
Openstack	Latest Versions Supported	When creating an OpenStack integration, select the latest available from the OS Version dropdown menu when running a later version
Puppet	6+	Puppet Agent version will be latest 6 version from yum.puppetlabs.com or apt.puppetlabs.com
Rubrik	Up to 7.x	
SCVMM	2016, 2019	
ServiceNow	Per servicenow plugin version	Refer to the servicenow store for Morpheus Catalog compatibility
Terraform	Latest	When Morpheus is handling Terraform installation, the version used will be 1.5.5 for licensing reasons. If a later version is required, you will need to manually handle Terraform installation. See the “Terraform Installation” section of Morpheus docs for more details.
vCloud Director	10.0, 10.2, 10.3, 10.4	When upgrading a vCD environment, you should update the API Version setting on the Morpheus Cloud configuration first

Note: Non-listed versions may be compatible but are not verified.

Note: Non-listed versions may be compatible but are not verified.

If you have any specific requirements please contact support@morpheusdata.com

5.13.29 Trust

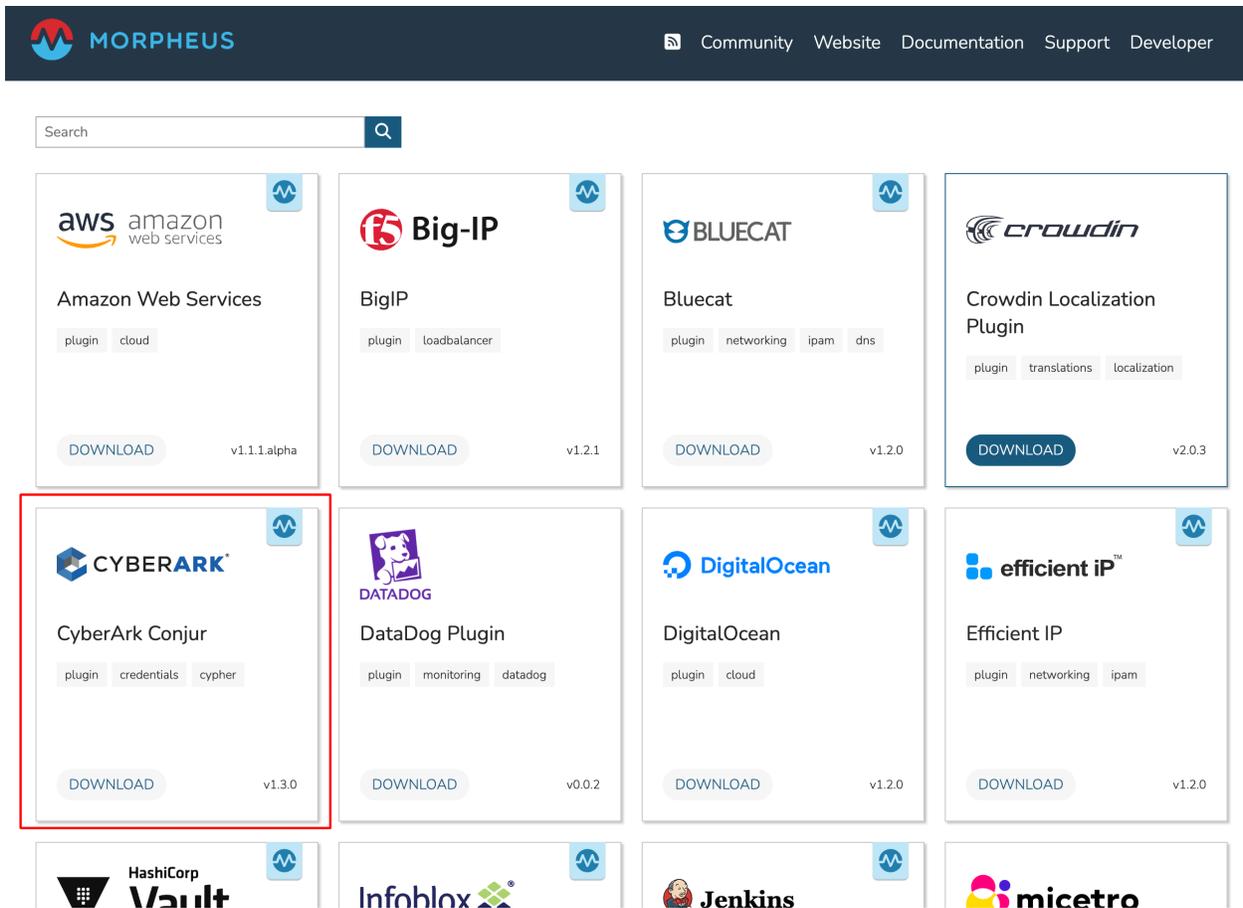
CyberArk Conjur

Morpheus supports integration with CyberArk Conjur for stored credential sets as well utilization of Conjur-stored secret values via Morpheus Cypher service. This works identically to built-in credential storage and built-in Cypher functionality with the exception that your Conjur server is the storage backend rather than the Morpheus appliance itself. In some cases, this may help to consolidate secret management into one central place or it even may be a requirement to satisfy internal security policies at some organizations.

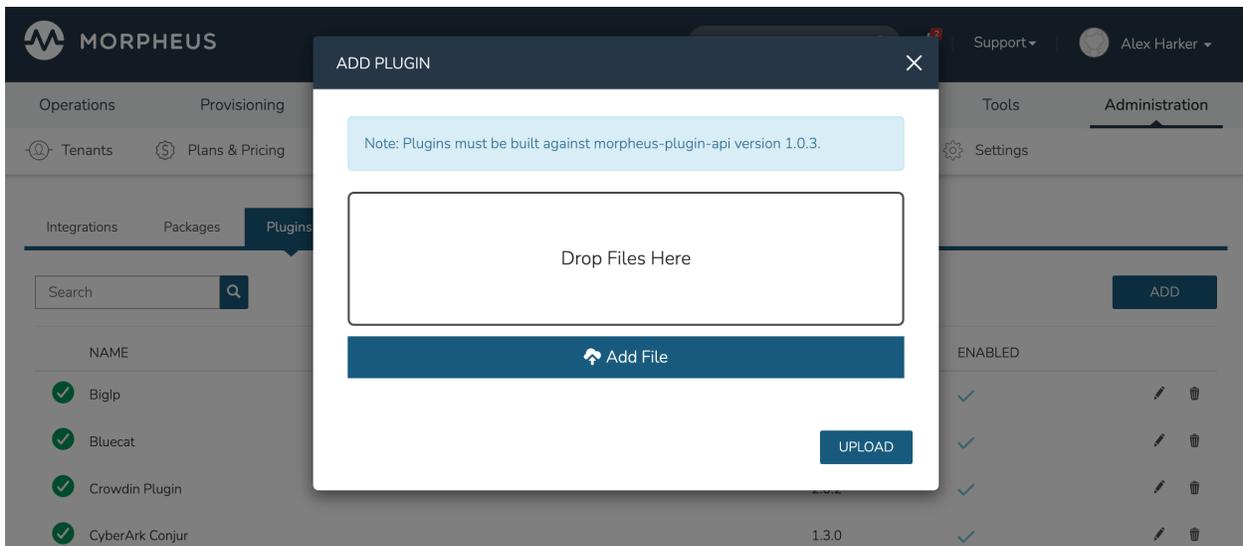
CyberArk Conjur integration is not included with Morpheus out of the box. It is developed as an official plugin and may be applied to any Morpheus appliance that needs it. See the Morpheus plugin repository website for access to the latest version of the plugin.

Applying the Plugin

The latest version of the CyberArk Conjur plugin is available at the Morpheus [plugin share site](#). Select the pane for the correct plugin and click the *DOWNLOAD* button. The plugin JAR file will be downloaded to the local computer you're working on.



With the plugin downloaded, head back to Morpheus to apply the plugin to your appliance. Plugins are added by navigating to Administration > Integrations > Plugins and clicking **ADD**. Drop the plugin JAR file onto the target and click **UPLOAD**. It may take a few moments for the plugin to be applied successfully and for the modal to be dismissed after clicking the upload button.



Once the plugin is added, we configure the connection details here. Edit the new Conjur plugin by clicking the pencil icon from its row in the list. Configure the following fields:

- **NAME:** A friendly name for the new integration to identify it within Morpheus
- **ENABLED:** Mark the box to make this integration active and available
- **CONJUR API URL:** The full URL for the Conjur server, usually listening on port 8443
- **CONJUR USERNAME:** The Conjur user account you wish to authenticate with. This user should have read, write, and execute access to the variables configured within the Conjur policy
- **CONJUR USERNAME API KEY:** The API key for user entered in the prior field
- **CONJUR ORGANIZATION:** The Conjur account in which the user and variables reside
- **CLEAR SECRET ON DELETION:** When marked, secret values will be cleared when deleted from Morpheus

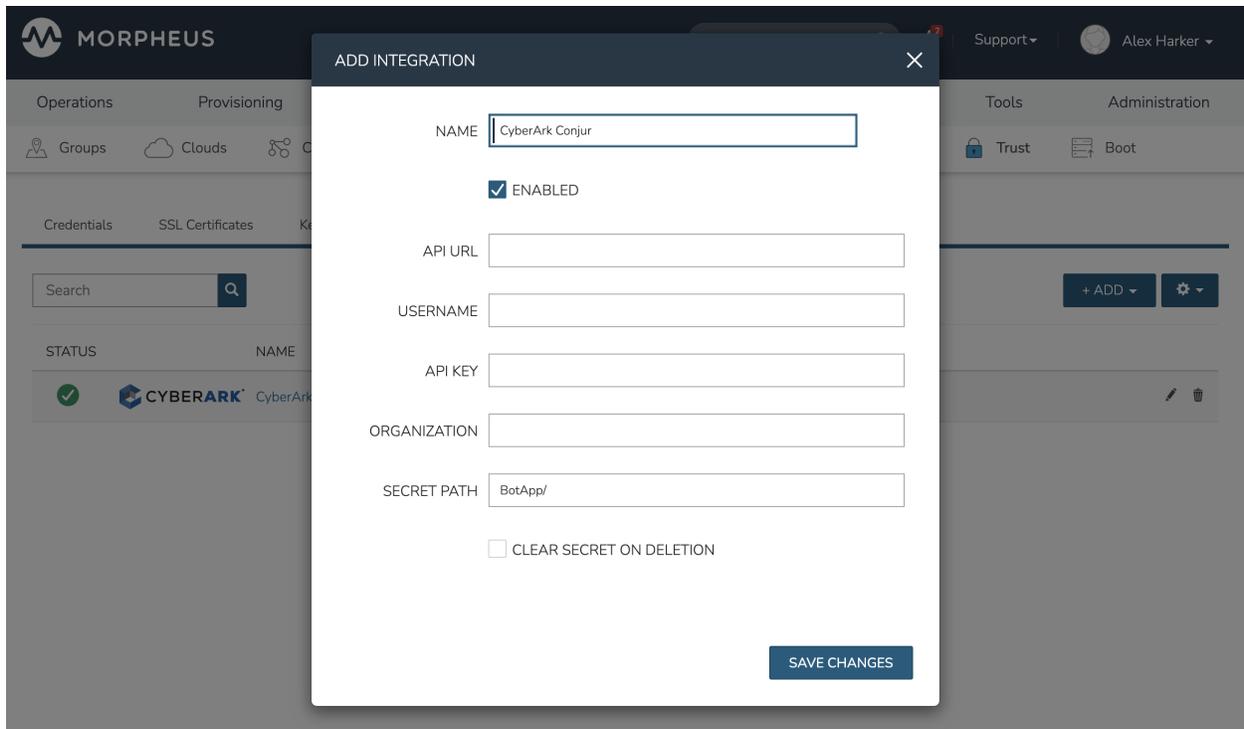
Important: Variables are predefined by Conjur policies. This integration does not allow users to create variables but does allow you to read and update their values. If the plugin is configured to clear secrets that are deleted from Morpheus, the variable value is updated with an empty value as there is no concept of fully deleting secrets in the Conjur API. If the plugin is configured not to clear secrets on deletion, the secret object is deleted from Morpheus without altering it on the Conjur backend.

By applying and configuring the plugin, the appliance now has the Conjur/Cypher integration available. If you don't intend to use the credential store integration, you can stop here and look ahead to the feature demonstration further ahead in this guide. If you do intend to use the credential store integration as well, there's one additional configuration step to take in the next section.

Adding A Conjur Trust Integration

In order to use Conjur with Morpheus credentials, we must also add a Conjur-type integration in Infrastructure > Trust > Integrations. By applying the plugin in the previous step, we now have the ability to add this type of integration. From the Trust integrations list page, click + *ADD* and select "Conjur." It's important to note that you **do not** need to again configure the API URL and Conjur login credentials as we just did in the previous step. You may configure them here if you wish, such as if you were making multiple integrations with multiple Conjur appliances, but if you're simply integrating with the same appliance and user account, this is unnecessary. Simply give the new Trust integration a friendly name for reference in Morpheus and indicate the desired mount point for secrets you wish to consume as done in the screenshot below.

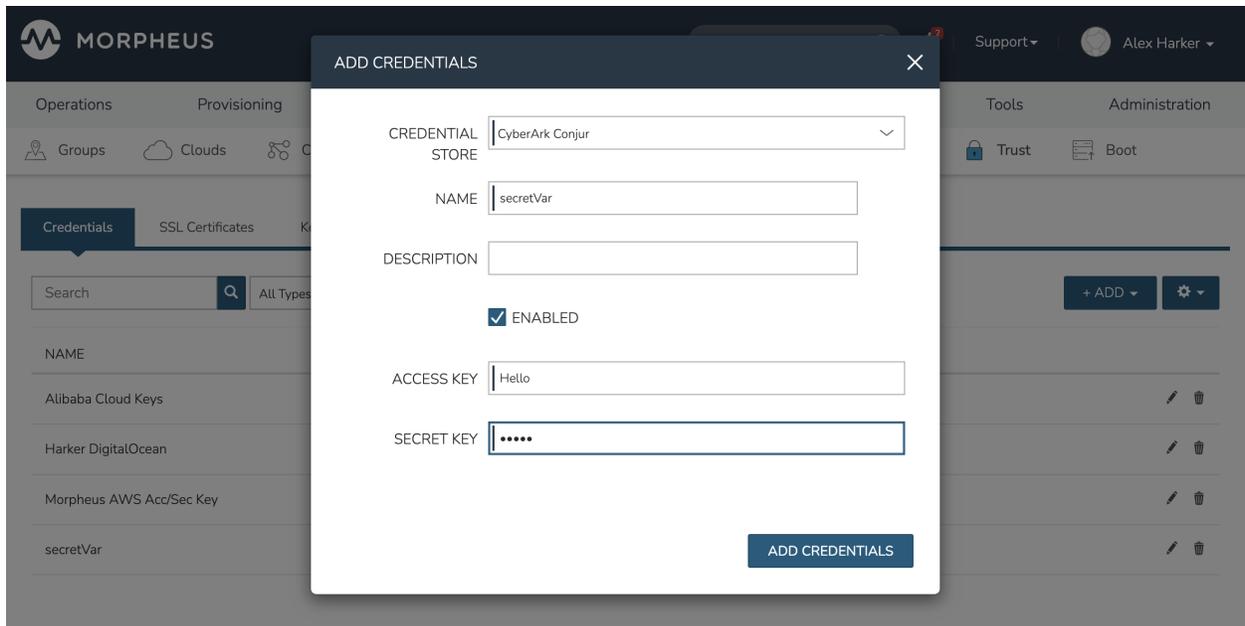
Note: In many cases, you don't have to configure the connection details when adding a Conjur Trust integration. See the paragraph above.



Conjur and Credential Stores

With configurations completed, we can take a look at how Morpheus is able to interface with Conjur as a credential store. As mentioned at the beginning of this guide, variables are created by Conjur policies and not by Morpheus. We can reference existing variables and update the secret values stored within.

As an example, there is a variable named `BotApp/secretVar` in the target Conjur appliance. When configuring the Trust integration (in the prior section), the `BotApp/` mount point was referenced (see the screenshot in the last section). When creating stored credential sets in Morpheus against this Conjur credential store, reference the rest of the variable path as the NAME value on the credential set. In this case, the credential set will be named `secretVar` to complete the `BotApp/secretVar` path as shown in the screenshot below. Be sure to also select Conjur as the CREDENTIAL STORE value rather than the default "Internal."



After saving and going back to the Conjur appliance, the value stored at the referenced variable has been updated. This credential set is also available for use anywhere Morpheus allows users to utilize them, such as when integrating new Clouds, authenticating REST calls to populate Option Lists, and many other places.

Conjur and Cypher

In addition to stored credential sets, the CyberArk Conjur plugin also enables integration with Morpheus Cypher. Just as calls can be made to internally-stored secret values with Cypher, Conjur variable endpoints may be referenced to onboard secret strings into automation Tasks.

In the screenshot below, see a simple shell script Task which echoes out a secret value stored in Conjur. As you can see, the `conjur/` mount point is referenced along with the rest of the path to the variable which should be accessed. After executing the Task, we can check the execution history and see that the Conjur-stored value is indeed echoed back out.

NEW TASK

NAME

CODE

LABELS

A comma separated list of labels that can be used to group/organize items.

TYPE

RESULT TYPE

SUDO

SOURCE

CONTENT

```
echo "<%= cypher.read('conjur/BotApp/secretVar') %>"
```

If desired, the value stored in Conjur could also be updated through the Morpheus Cypher UI. Navigate to Tools > Cypher and click + ADD. Note the `conjur` mount point is included in the example list of available mount points shown in the modal. By creating a new Cypher entry at the `conjur` mount point which references an existing Conjur variable path, the value in Conjur can be overwritten. If you've configured the plugin to clear secrets on deletion, you can also update the stored value with an empty string when you delete the Cypher object in Morpheus. For our example case, we could update the value in Conjur by creating a Cypher entry at the mount point `conjur/BotApp/secretVar`.

ADD KEY

KEY

Keys can have different behaviors depending on the specified mountpoint.

Available Mountpoints:

- password - Generates a secure password of specified character length in the key pattern (or 15) with symbols, numbers, upper case, and lower case letters (i.e. password/15/mypass generates a 15 character password).
- tfvars - This is a module to store a tfvars file for terraform
- conjur - This allows secret data to be fetched from a CyberArk Conjur integration. This can be configured in the plugin integration settings.
- secret - This is the standard secret module that stores a key/value in encrypted form.
- sys - This is the standard secret module that stores a key/value in encrypted form.
- uuid - Returns a new UUID by key name when requested and stores the generated UUID by key name for a given lease timeout period.
- key - Generates a Base 64 encoded AES Key of specified bit length in the key pattern (i.e. key/128/mykey generates a 128-bit key)
- vault - This allows secret data to be fetched to/from a HashiCorp Vault server. This can be configured in the Plugin settings.

VALUE

5.14 v7.0.0 LTS Release Notes

- Compatible Plugin API version: 1.1.0
- Compatible Morpheus Worker version: 5.4.8+
- Minimum upgrade version: v6.0.0

Note: Items appended with ^{6.x.x} are also included in that version

Release Dates

- v7.0.0 Mar 20 2024

5.14.1 New Features

API & CLI

- Added API support for optionally specifying a stack name when provisioning from CloudFormation templates ^{6.2.8}
- Added API support for specifying an S3 bucket to read CloudFormation templates from during provisioning. This is necessary when provisioning from CF templates greater than 50 KB ^{6.2.8}

CloudFormation

- Provisioning from CloudFormation templates now includes a “STACK NAME” configuration. By default, this will be the same as the Instance or App name but can be overridden ^{6.2.8}
- When provisioning from CloudFormation Spec Templates, added a configuration to specify an S3 bucket to read the Spec Template from. This is required for CF templates greater than 50 KB ^{6.2.8}

Dashboard

- Added support for Spanish-language localizations for Morpheus Dashboard ^{6.2.8}

Identity Sources

- “Post RelayState” field added for For SAML SSO Identity Sources using “Post Binding Mode” for defining RelayState post parameter. ^{6.2.8}

Import/Export

- Forms can now be used with the import/export feature. Export Forms as code into an integrated Git repository and import them back into any other appliance with the same repository integrated

Kubernetes

- System Kubernetes 1.29 Layouts added ^{6.2.8}

Policies

- The Roles and Policies list pages have been updated to give the user control over visible output columns and page size

Roles

- Added a Cluster Types tab to the Role detail page to control the Cluster types available to the Role

Security

- Upgraded `spring-web` to version 5.3.32 to mitigate CVE-2024-22243 ^{6.2.8}

Terraform

- For licensing reasons, automated Terraform installs handled by Morpheus are now capped at version 1.5.5. Other versions may be utilized in Morpheus through manual installation ^{6.2.8}

VMware

- When Snapshot names are changed in VMware, the name change is now reflected in Morpheus following the next Cloud sync ^{6.2.8}

5.14.2 Fixes

API & CLI

- Adding or updating Identity Sources via Morpheus API, will no longer fail when adding a mapping for a built-in role (such as System Admin) ^{6.2.8}
- After updating the CODE value on a Cloud (via UI or API), GET calls to `/api/billing/zones` no longer return the old `zoneCode` value ^{6.2.8}
- Fixed situations where a 200 status and a `success=false` message could be received in the same response when creating virtual servers for load balancers via Morpheus API ^{6.2.8}
- Service Plans created via API without a `config.ranges` value in the payload can now be updated normally via API ^{6.2.8}
- Updating Catalog Item icon logos via Morpheus API now works correctly ^{6.2.8}

Agent

- Fixed an issue with the Windows Agent that could cause CPU usage to be reported incorrectly for VMs within the UI ^{6.2.8}
- Linux Agent upgrades now update the appliance URL (if applicable) to account for situations in which this has changed since the last Agent upgrade ^{6.2.8}

Amazon

- Fixed an issue causing actions such as adding nodes or reconfiguring at the Instance or VM level from working correctly for discovered nodes which were converted to managed ^{6.2.8}
- Fixed an issue that prevented AWS Clouds scoped to all regions from reading from region-scoped S3 buckets ^{6.2.8}
- Fixed an issue with snapshot sync when more than 1000 snapshots exist in an aws account. ^{6.2.8-2}
- When creating RDS Instances on AWS Clouds, the subnet groups now populate correctly for Clouds scoped to all regions ^{6.2.8}
- When scoping AWS Clouds to all VPCs, DB subnets available on the AWS Cloud are now discovered properly and are available for selection during RDS provisioning. Previously, this only worked correctly when scoping to a specific VDC ^{6.2.8}

Apps

- When adding an Instance to an existing App, the display name of the Instance is now shown rather than the Instance name ^{6.2.8}

Catalog

- The default FORM TYPE value for catalog items is now “Select” (that is, unselected) rather than “Form” ^{6.2.8}

Cluster Layouts

- Hidden-type Inputs on Cluster Layouts now show the `inputName` value along with “(hidden)” when the Cluster Layout is later edited ^{6.2.8}
- On custom Cluster Layouts, Inputs are no longer shown in an order when many Inputs are used (approximately five or more) ^{6.2.8}

Clusters

- When additional nodes are added to Clusters, they now have the custom options values (Input values) that nodes created at initial cluster provision had ^{6.2.8}

Costing

- Fixed costs shown on Instance detail pages within Subtenants showing costs in USD rather than the configured Tenant currency ^{6.2.9}
- Fixed the “Cost this Month” figure displayed for a Cloud integrated in the Master Tenant and shared down with a Subtenant ^{6.2.8}

Cypher

- Removed the “sys” mountpoint tool tip from the “ADD KEY” modal that appears when adding a new entry to Cypher ^{6.2.8}
- When File Templates containing calls to Cypher are set on a Node Type created in the Master Tenant, the Cypher values are now properly decrypted when provisioning involving the Node Type takes place in a subtenant ^{6.2.8}

Hosts

- Improved the backend logic for handling server records moved from one Cloud type to another (Change Cloud functionality) ^{6.2.8}

Import/Export

- Fixed a situation where “Export All” could cause the UI to become unresponsive and require restart ^{6.2.8}

Inputs

- Fixed Input dependency and visibility not working as configured when the Inputs were created by different methods (API vs UI) ^{6.2.8}

Instances

- Fixed Layout and Version fields not appearing on “Convert to Instance” modal (for converting servers to managed Instances) ^{6.2.9}
- Fixed reconfigures to add disks dropping the name(s) of the new disk(s) under specific conditions ^{6.2.8}
- Improved how CPU representation graphs are displayed to prevent misinterpretation of total CPU capacity being used ^{6.2.8}

Kubernetes

- Fixed 500 errors being thrown when Kubernetes Apps were deleted ^{6.2.8}

NSX-T

- Fixed network resources being visible in other Subtenants when NSX-T integrations created in one Subtenant were scoped to a public Cloud integrated from the Master Tenant ^{6.2.9}

- Load balancer virtual server protocol configuration can now be successfully updated via Morpheus API and CLI ^{6.2.8}

Network

- A friendly UI error message is now given when attempting to save a network display name larger than 255 characters ^{6.2.8}

Nutanix Prism Central

- Discovered hypervisor hosts now correctly display the OS rather than defaulting statically to display “ESXi” ^{6.2.8}
- Fixed Instances not deleting when they were provisioned to Nutanix networks utilizing Morpheus IP Pools ^{6.2.8}
- Network interfaces on Prism Central VMs no longer show a null MAC address until after the first Cloud sync following provisioning ^{6.2.8}

Option Lists

- Setting a proxy that requires authentication no longer causes 407 errors and fetch failures for REST-populated Option Lists ^{6.2.8}

Plugins

- Updated Ansible Tower and Bluecat plugins to honor global proxy settings ^{6.2.8}

Proxies

- The Morpheus Windows Agent now uses a proxy if one is configured ^{6.2.8}

SAML

- There are no longer two POST BINDING MODE fields on Add/Edit modals for SAML SSO Identity Sources. Additionally, the INCLUDES SAML REQUEST PARAMETER field no longer toggles back to “Yes” on edit ^{6.2.8}

SCVMM

- Fixed SCVMM Clouds scoped to specific clusters discovering workloads from outside that scoped cluster ^{6.2.8}
- Fixed an issue that could cause cloned Veeam backups for SCVMM Instances to not be created properly ^{6.2.8}

Security

- Fixed cache files exposing cloud credentials in plaintext under certain conditions when local credentials (not stored credentials) were used to authenticate the cloud integration ^{6.2.8}

VMware

- Any changes to minimum memory values for an image in vCenter are now properly synced over to Morpheus on the next Cloud sync ^{6.2.8}
- Fixed tag values being updated via Morpheus API being wiped out on the next Cloud sync ^{6.2.8}

5.14.3 Appliance Updates

Agents

- Linux Agent: Morpheus linux agent updated to v2.5.4
- Windows Agent: Morpheus Windows Agent updated to v2.5.0.0

Appliance

- mysql: Updated default jdbc url used for db cluster connections to include connectTimeout, maxReconnects, queriesBeforeRetrySource and secondsBeforeRetrySource. Default values can be updated in morpheus.rb. ^{6.2.8}
- nginx: Logging requests to the /ping endpoint of Morpheus app nodes are now disabled by default. Logging can be reenabled by adding nginx['ssl_access_ping_log'] = true &

nginx['access_ping_log'] = true to morpheus.rb ^{6.2.8}

- Package repo cleanup: The appliance installer has been updated to clean /var/opt/morpheus/package-repos/ directory after package install when it is larger than 5GB. The current Morpheus package repo files will be added during reconfigure or supplemental package install ^{6.2.8}
- SLES 15 FIPS: - Added a FIPS-compliant Morpheus installer for SLES 15 ^{6.2.8}

Embedded Plugins

- Bluecat: bluecat-plugin updated to v1.2.1
- DigitalOcean: digital-ocean-plugin updated to v1.2.3
- Infoblox: infoblox-plugin updated to v1.3.5
- Solarwinds: solarwinds-plugin updated to v1.1.2
- phpIPAM: phpipam-plugin updated to v1.2.3
- Efficient IP: efficient-ip-plugin updated to v1.2.1

Node Package

- Morpheus Node & VM Node Packages updated to 3.2.22

5.14.4 Worker

Distributed Worker

- Morpheus Worker v7.0.0 now available
- Distributed worker v7.0.0 can now be deployed in a highly-available configuration. See distributed worker documentation for additional details

5.15 v7.0.0 Compatibility & Breaking Changes

When installing and upgrading to Morpheus v7.0.0, refer to the following to ensure compatibility.

5.15.1 Breaking Changes

- 6.3.0: Version 6.3.0 is the first version to require Plugin API 1.0.0+. Small changes will need to be made in order to make plugins created for prior versions of Morpheus compatible with 6.3.0+. See the [related article in our KnowledgeBase](#) on the small changes that will need to be made to ensure plugin compatibility
- 6.2.2, 6.0.7 v7.0.0 contains embedded MySQL v8 upgrade when upgrading from v6.0.0 - v6.0.6 or 6.1.0 - 6.2.1. **BACKUP YOUR DATABASE PRIOR TO UPGRADE** when using embedded MySQL (all-in-one appliances)
- 6.2.2, 6.0.7 Minimum v6.x required to upgrade to v7.0.0 for environments using embedded RabbitMQ. Environments running 5.5.x or earlier using embedded RabbitMQ must upgrade to v6.0.0 - v6.0.6, or 6.1.0 - 6.2.1 prior to upgrading to v7.0.0
- 6.2.2, 6.0.7 Rolling upgrades for HA environments using embedded RabbitMQ and/or embedded Elasticsearch services are not supported when upgrading from v6.0.0 - v6.0.6 or 6.1.0 - 6.2.1
- 6.1.1 contains database datatype modifications on account_invoice and account_invoice_item that may cause long initial ui start up times while the modifications are ran in MySQL for environments with over 100k invoice records when upgrading from any version other than 6.0.3
- 6.1.1 relocates the embedded plugin folder and remove the previous folder. For HA environments using shared storage, rolling upgrades from any version other than 6.0.3 are not advised as the embedded plugins will not be found on non-upgraded nodes after one node is upgraded.
- 6.1.1: NSX-V networking integration support is removed and no longer supported as of Morpheus 6.1.1
- 6.0.3 contains database datatype modifications on account_invoice and account_invoice_item that may cause long initial ui start up times while the modifications are ran in MySQL for environments with over 100k invoice records.
- 6.0.3 relocates the embedded plugin folder and remove the previous folder. For HA environments using shared storage, rolling upgrades are not advised as the embedded plugins will not be found on non-upgraded nodes after one node is upgraded.
- 6.0.0: NSX-V support is deprecated though still supported as of Morpheus 6.0.0. It will be removed and unsupported in 6.1.1 and higher.
- 6.0.0+: In Morpheus 6.0.0+, many third party integrations have been moved out of the core installer package and converted to Morpheus plugins. As a result, during the upgrade process your appliance will need to be able to access share.morpheusdata.com, the online repository for all Morpheus plugins. Where this is not possible, users may instead apply the supplemental installer package which is also available at Morpheus Hub alongside the main installer package.
- 6.0.0+: In Morpheus 6.0.0+, older service specific system provided Instance Types and Layouts were deprecated and disabled. Updating to 6.0.0 will not affect existing Instances that are associated with the disabled types, however existing catalog item configurations, blueprints and api requests that use disabled Instance Types and layouts will need to be updated.
- 5.5.2: VM Node Packages: Due to build java version requiremnets, the i386.deb and i386.rpm (32-bit) VM Node Packages can no longer be updated, and remain on v3.2.9.
- 5.4.12: Guacd: Guacd is now compiled iwth libssh2-1.10.0 on all platforms. Appliances on SLES15 may need openssl-devel manually installed for guacd to succesfully compile.
- 5.4.12: Session Manager: Morpheus features a new session manager that was necessary in order to resolve expiring connections from the agents due to a Spring framework update. This new session manager no longer requires Sticky Sessions and they can now be turned off at the load balancer if so desired. However, keeping them on is totally reasonable as well as it reduces overall system load. Rolling restarts no longer kick you out of your session if sticky sessions are off as it distributes your session data across the morpheus nodes in an HA environment. Additionally, overall system load is reduced as a result of the new session manager.

- 5.4.9: Morpheus 5.4.9 adds the “Provisioning: State” Role permission. This permission determines access to the State tab for Terraform-backed Instances and is set to “None” by default. On upgrade, only System Admin users will be able to see the State tab for these Instances. For other users who should have this access, edit their Roles to include “Provisioning: State” permissions.
- 5.4.5: Warning: Database indexes added for account_usage and metadata_tag tables. Customers with very large account_usage and/or metadata_tag tables (10 million+) may experience slower initial morpheus-ui loading time after upgrading to 5.4.5, as well as additional database load.
- 5.4.5: ‘AVI Load Balancer’ renamed to ‘NSX Advanced Load Balancer’
- 5.4.5: Cloud Types disabled by default: Dell, HPE (NOT HPE Oneview), Supermicro and Cloud Foundry. Users would still be able to re-enable these clouds in the appliance settings. Does not affect existing Clouds.
- 5.4.5: A10 Load Balancer type has been disabled, and will no longer be an option when adding new Load Balancers. Contact Morpheus if you need to re-enable A10 Load Balancer option. This does not affect existing Load Balancers.
- 5.4.5: Morpheus Cluster type “Combo Cluster” renamed to “KVM/Docker Cluster”
- 5.4.5: Greenfield managed vm’s (provisioned with Morpheus) can no longer be deleted in Morpheus without removing the actual vm/infrastructure. Restriction does not apply to brownfield vm’s that have been converted to managed.
- 5.4.4: The Venafi and AppDynamics integrations are deprecated in v5.4.4 and will be removed in v5.4.5. App-Dynamic will return as a plugin at a later date.
- 5.4.4: The morpheus-ui logging configuration file has changed from logback.groovy to logback.xml in v5.4.4 (/opt/morpheus/conf/logback.xml). The logback.groovy file from previous versions can be removed, and any updates to logback.groovy will not result in any logging configuration changes.
- 5.4.3: vCloud Director: Support for integrations with vCD 9 ended
- 5.4.3: Morpheus Worker/Gateway v5.4.3 packages are now available. Existing Worker & Gateway nodes must be upgraded to v5.4.3 for compatibility with Morpheus v5.4.3 Appliances.
- 5.4.2: vCloud Director: vCD 9.x will no longer be supported by Morpheus
- 5.4.2: ServiceNow: Instance and Blueprint specific exposures will be removed from ServiceNow plugin support. More advanced configurations of Instances and Blueprints, in addition to Workflows, can be exposed utilizing Catalog Items
- 5.4.2: After upgrading, it is recommended that you manually perform one “Daily” refresh Amazon Clouds to ensure availability of Amazon Service Plans for each region. To manually refresh a Cloud, navigate to Infrastructure > Clouds > (Selected Amazon Cloud) and select “Daily” from the REFRESH dropdown menu. If this is not done, Morpheus may not show Amazon Service Plans in the provisioning wizard until after Midnight UTC following the upgrade when the next automatic Daily sync would run.
- 5.3.4: Major UI navigation structure changes. Refer to the Navigation Updates reference table
- 5.3.3: Support for OpenStack v2 Identity API is removed
- 5.3.2+: The local code repository path moved from /var/opt/morpheus/morpheus-ui/repo to /var/opt/morpheus/morpheus-local/repo to reduce potential shared storage issues and performance restrictions. The reconfigure process creates the folders and sets the paths in application.yml, no manual intervention is needed unless symlinks exist on /var/opt/morpheus/morpheus-ui/repo/git which will need to be removed prior to reconfiguring - 5.3.2+ The deprecated /var/opt/morpheus/morpheus-ui/repo path will be automatically deleted in a future release but can be manually recursively deleted at any time for storage reclamation.
- 5.3.2+: *Provisioning* → *Deployments* has been moved to *Provisioning* → *Code* → *Deployments*
- 5.2.9: OpenStack v2 Identity API is deprecated as of v5.2.9 (and is removed as of v5.3.3)

- 5.2.6, 5.3.1: Appliance & Agent java version updated to 8u292-b10. jdk8u292 disables TLS 1.0 and 1.1 by default
- 5.2.3+: codeready (codeready-builder-for-rhel-8-x86_64-rpms) repo access required for RHEL 8+ Appliances, replacing the previous PowerTools/powertools requirement
- 5.2.1 & 4.2.5: API: Metadata: Metadata tags now referred to as `tags` and labels now referred to as `labels`. Previously metadata tags were referred to as `metadata` and labels were referred to as `tags`
- 5.0.0+: When upgrading to 5.0.0+ from 4.x.x, any bearer tokens that have been generated are deleted which requires users to request new bearer tokens
- 4.2.4: For appliances with externalized MySQL databases, due to MySQL deprecation of the “EDT” timezone you may need to update your database timezone to UTC or another compatible value. If this is not done, you will receive errors referencing timezone and Morpheus will not start. Morpheus should handle this change automatically for all-in-one appliances.
- 4.2.1+: Tasks: Python: Virtual environment are now used for Python Tasks. **Note:** `virtualenv` is required on all Appliance App nodes
- 4.2.1+: Puppet: Morpheus integration now supports version 6+. Puppet versions prior to 6 are no longer supported
- 4.2.1+: Clouds: VirtualBox, VirtuSteam, and MetaCloud Cloud Types are no longer supported or available
- 4.2.1+: Appliance: OS: Ubuntu 14.04 has reached its end of life (EOL) and is no longer supported as a Morpheus Appliance Host Operating System. Any Morpheus Appliance running on 14.04 must be upgraded to 16.04, 18.04, 20.04 or 22.04 BEFORE upgrading to 4.2.1+. Upgrades on 14.04 will not succeed

5.15.2 Morpheus Application OS

Morpheus can be installed on the following platforms. Please note the table below is for Morpheus Application OS support, not Morpheus Agent OS Support.

Note: If CentOS 8.2 is pinned to 8.2.2004 vault, the PowerTools repository will need to be pinned to 8.2.2004 to access freerdp-libs 2.0.0

Table 26: **Supported Appliance Operating Systems**

OS	Version(s)	Notes
Amazon Linux	2	
CentOS	7.x, 8.x (stream) 9.x (stream)	
Debian	10, 11	
RHEL	7.x, 8.x, 9.x	
Oracle Enterprise Linux (OEL)	7.x, 8.x	
SUSE SLES	12, 15	
Ubuntu	18.04, 20.04, 22.04	14.04 is no longer supported for Appliance OS. Note: 14.04 is still supported by the Morpheus Agent.

5.15.3 Services

v7.0.0 Service Versions & Compatibility

Table 27: v7.0.0 Service Versions & Compatibility

Service	Compatible Branch	Morpheus Installer Version	Updated in v7.0.0
Plugin API	1.1.0	1.1.0	
Morpheus Worker	5.4.8+		
MySQL	v5.7, v8.0	v8.0.35	
MySQL (FIPS)	v5.7, v8.0	v8.0.34	
Elasticsearch	v8.11+	v8.11.2	
RabbitMQ	v3.5-3.12	v3.12.9	
Tomcat		v9.0.83	
Nginx		v1.25.1	
OpenSSL		1.1.1t, 1.0.2u (FIPS)	
Java		11.0.20+8	
Java (macOS agent)		11.0.14+9	

Important: Elasticsearch 8.10.1 was released within days of Morpheus v7.0.0. It's expected to be compatible but has not been tested. Customers managing their own Elasticsearch clusters should be aware that internal testing of the Morpheus v7.0.0 was conducted on Elasticsearch 8.9.2 and we cannot recommend deploying a higher version at this time.

Morpheus Agent & Node Package Versions

Table 28: v7.0.0 Agent & Node Package Versions

Package	Version	v7.0.0 changes from v6.3.4
Morpheus Node and VM Node Packages	3.2.22	Node and VM Node Packages update to 3.2.22 with v2.5.4 Linux Agent
Morpheus Linux Agent	v2.5.4	Morpheus Linux Agent updated to v2.5.4
Morpheus Windows Agent	v2.5.0.0	Morpheus Windows Agent updated to v2.5.0.0
Morpheus macOS Agent	v2.4.0	No changes

5.15.4 Security

Security Advisories

Advisory ID	Severity	Description	Updated On
MOR20220721-01	Critical	Morpheus through 5.4.3 (which run Java 8) are confirmed to be impacted, Morpheus through 5.5.1-1 (for customers on 5.5.x Standard installations) and 5.4.8-2 (for customers on 5.4.x LTS installations) are potentially impacted if the vulnerability is found on Java 11.	07-21-2022
MOR20220524-01	High	An XXE issue was discovered in Morpheus through 5.2.16 and 5.4.x through 5.4.4. A successful attack requires a SAML identity provider to be configured.	06-08-2022

5.15.5 Upgrade Paths & Methods

The following table shows supported version upgrade paths and methods.

* Some Features and Fixes in the From version may not be included in the To version due to From version being released after the To version.

5.15.6 Integrations

Note: Current iterations of Amazon AWS, Microsoft Azure, Google Cloud Platform, Digital Ocean, OpenTelekom Cloud, IBM Bluemix, Softlayer and UpCloud are all supported.

Table 29: Integration Compatibility

Integration	Supported Version(s)	Notes
Ansible	2.7.x or higher	
Ansible Tower	3.8.x	
App Dynamics	4.5.x	
Ava-mar	18	
Azure Stack	2002 back to 1908	2019-03-01-hybrid api-profile version used which is supported in 1908 and later Azure Stack versions
Cisco ACI	3.x,4.x,5.x	
Citrix Netscaler	v12.1	
Comm-vault	v11 sp 19	
F5 Big-IP	11.4+	
In-foblox	Latest Versions Supported	
Kubernetes	1.21+	
Microsoft Hyper-V	2012R2, 2016, 2019	
Nutanix AHV	5.0+	In 5.5 - 5.7 if Prism Central is managing Prism Element, image creation will not function due to PC Image Management.
Openstack	Latest Versions Supported	When creating an OpenStack integration, select the latest available from the OS Version dropdown menu when running a later version
Puppet	6+	Puppet Agent version will be latest 6 version from yum.puppetlabs.com or apt.puppetlabs.com
Rubrik	Up to 7.x	
SCVMM	2016, 2019	
ServiceNow	Per servicenow plugin version	Refer to the servicenow store for Morpheus Catalog compatibility
Terraform	Latest	When Morpheus is handling Terraform installation, the version used will be 1.5.5 for licensing reasons. If a later version is required, you will need to manually handle Terraform installation. See the "Terraform Installation" section of Morpheus docs for more details.
vCloud Director	10.0, 10.2, 10.3, 10.4	When upgrading a vCD environment, you should update the API Version setting on the Morpheus Cloud configuration first

Note: Non-listed versions may be compatible but are not verified.

5.16 Morpheus v7 Highlights

This section includes feature change highlights and key enhancements for all releases within Morpheus version 6. Many other features changes and enhancements were added to each of these versions and the release notes page for each individual version can be reviewed for complete details.

5.16.1 6.3.1

Distributed Worker

- Updated Distributed Worker such that all Morpheus Agent communications can be routed to the Morpheus appliance via the Worker

Kubernetes

- Added enhancements for utilizing Provisioning Workflows with custom Cluster Layouts
- Added the ability to send Kube commands to MKS servers via Morpheus Agent when the Morpheus appliance cannot connect directly to the cluster

5.16.2 6.3.0

Cluster Packages Added new UI area (Library > Templates > Cluster Packages) for creating Cluster Packages which can be associated with Cluster Layouts. See the appropriate areas of Morpheus Documentation for more on Cluster Packages and Cluster Layouts

Image Builder Updated the Image Builder form into a single form rather than a paged wizard. See the Image Builder section of Morpheus documentation for example scripts and help getting started

VMware Added support for versioned templates from VMware Content Library Added the ability to set vApp Property values on VMware Node Types. See Node Type docs for more

5.16.3 6.2.5

Distributed Worker

- Updated Distributed Worker such that all Morpheus Agent communications can be routed to the Morpheus appliance via the Worker

5.16.4 6.2.4

NSX-T Official support added for NSX-T 4.1

ServiceNow Added the ability to switch back to the older table-based API mode for CMDB sync

5.16.5 6.2.3

Identity Sources

- SAML SSO identity sources using HTTP-POST binding are now working as expected when integrated with Morpheus Tenants

Kubernetes

- Updated default Kubernetes Cluster Layouts to version 1.28

5.16.6 6.2.2

Catalog

- Added support for saving Catalog items without first passing a check for valid JSON in the config

Inputs

- Added “REMOVE NO SELECTION” attribute for Select List-based Inputs. This defaults the Input to the first selection in the list rather than to an empty selection

Layouts

- Added Display Order property for Layouts. Layouts are listed in high-to-low order based on the Display Order in the Layouts dropdown of the provisioning wizard

5.16.7 6.2.1

Forms

- Additional quality of life features added for Forms

XaaS

- When Teardown-phase Tasks fail following an attempt to delete an XaaS Instance, the remaining Tasks are stopped which prevents the deletion from taking place. This allows users to correct the failing Tasks and ensure the object is deleted gracefully. Non-XaaS Instances already supported this.

5.16.8 6.2.0

Import/Export

- Configure code repositories (ex. integrated Github repositories) as import and export targets. Export Morpheus items as code into repositories and import them into other Morpheus appliances.

Workflows

- When running Workflows on-demand against an Instance, users can now select a specific phase of Tasks to be run if a Provisioning Workflow is selected

5.16.9 6.1.2

Forms

- Added Text Array input type for Forms which allows the user to enter a list of values separated by a delimiter. Once entered, the values are parsed out and may be individually deleted prior to submitting the form
- Added new ability to filter available Cloud types on Forms. Select a Cloud type from the LIMIT TO CLOUD TYPE dropdown or select FILTER FROM RESOURCE. The option to filter from resource reads the Cloud type from the Catalog Item Instance config

5.16.10 6.1.1

Amazon

- Added ability to scope Amazon AWS Clouds to all regions

Instances

- Both the Name and Display Name property for Instances can now be edited. Previously, only the Display Name could be edited

5.16.11 6.1.0

Forms

- Added a Form builder tool to aid in creating robust order Forms for Catalog Items

5.16.12 6.0.10

Distributed Worker

- Updated Distributed Worker such that all Morpheus Agent communications can be routed to the Morpheus appliance via the Worker

5.16.13 6.0.9

NSX-T Official support added for NSX-T 4.1

ServiceNow Added the ability to switch back to the older table-based API mode for CMDB sync

5.16.14 6.0.8

Kubernetes

- Upgrade default Kubernetes Cluster Layouts to version 1.28

5.16.15 6.0.7

Layouts

- Added Display Order property for Layouts. Layouts are listed in high-to-low order based on the Display Order in the Layouts dropdown of the provisioning wizard

5.16.16 6.0.6

Costing

- The date filter on the Invoices list page now defaults to the last three months to ensure quicker page loads

5.16.17 6.0.5

Clouds

- IBM PowerVC Cloud support is now officially added. This Cloud type has existed in prior versions but is officially out of Alpha state with 6.0.5

Kubernetes

- Added Kubernetes 1.25, 1.26 and 1.27 layouts for vCloud Director
- Added default Kubernetes 1.25, 1.26, and 1.27 layouts for Google Cloud Platform

Workflows

- When running a Workflow on demand against an Instance, users can now select a phase of Tasks to run when a Provisioning Workflow is selected

5.16.18 6.0.4

Workflows

- Added Scale Down phase for Provisioning Workflows. Tasks in this phase are run on nodes being deleted when Instances are scaled down (horizontally). This phase is invoked during both manual and automatic scale down events

5.16.19 6.0.3

Instances

- Instances now have a Name and Display Name field when editing. Previously editing the Name only updated the Display Name database property which created confusion when duplicate name warnings were received in future provisioning

Logs

- Morpheus Agent logs can now be disabled on a per-server basis in addition to the global enable/disable setting which is already in the product

5.16.20 6.0.2

Plans & Pricing

- Added the ability to set a cores per socket range on VMware-type Service Plans

Policies

- Added Max VM Snapshot Policies to allow users to limit the number of stored snapshots per VM which allows greater control over storage
- Max Policies (Max Cores, Storage, and Memory) now include the option to include or exclude container resources in the Policy

ServiceNow

- Refactored API calls to ServiceNow which provide integration functionality within Morpheus. This results in greater fault prevention and some performance improvements

5.16.21 6.0.1

Labels

- Run Tasks, Operational Workflows, or Jobs against a group of Workloads (Instances or servers) with a common Label

Kubernetes

- Added Kubernetes 1.26 support

Oracle Cloud

- Added two-way tag sync for Oracle Cloud workloads, similar to tag sync capability with other public Clouds

Policies

- Added Workflow execution approval Policies. When Operational Workflows are executed

Workflows

- Added Workflow stop capability, such as if you realize a long-running Task will fail and do not wish to wait out the expected failure

5.16.22 6.0.0

Dashboard

- The main Morpheus Dashboard landing page (Operations > Dashboard) has been completely redesigned

Instances

- Instance detail pages now include a Resources tab which shows VMs, containers, Apps, and other constructs which may be associated with the Instance. Previously this information was on the main detail page, not inside its own tab
- The Instance detail page header has been redesigned to move more of the most important information to the top of the page
- The Instance detail page headers has been redesigned to move more of the most important information to the top of the page

- The Instance detail page now includes a costing tab. This tab pulls and aggregates Instance and host invoices, pricing history charts, pricing trends, and lists associated metered prices
- The Instances detail page now includes a Summary tab which holds information that was previously in the Info section of the page and was always present (regardless of which subtab the user was looking at)
- The Instances detail page now includes a monitoring tab which holds memory, storage, CPU, disk I/O and network stats. This information can be shown over a maximum of 90 days depending on your appliance stats retainment setting

Policies

- Many Policy types can now be scoped to Service Plans

Workflows

- Nested Workflows have been added. Create modular Workflow pieces to build out larger Workflows
- Retry failed Workflows from the point where the first Task failed