<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3</td>
<td>Plans &amp; Pricing</td>
<td>170</td>
</tr>
<tr>
<td>4.4</td>
<td>Roles</td>
<td>174</td>
</tr>
<tr>
<td>4.5</td>
<td>Users &amp; User Groups</td>
<td>179</td>
</tr>
<tr>
<td>4.6</td>
<td>Integrations</td>
<td>183</td>
</tr>
<tr>
<td>4.7</td>
<td>Policies</td>
<td>183</td>
</tr>
<tr>
<td>4.8</td>
<td>Provisioning Settings</td>
<td>189</td>
</tr>
<tr>
<td>4.9</td>
<td>Monitoring Settings</td>
<td>191</td>
</tr>
<tr>
<td>4.10</td>
<td>Backup Settings</td>
<td>193</td>
</tr>
<tr>
<td>4.11</td>
<td>Logging Settings</td>
<td>194</td>
</tr>
<tr>
<td>4.12</td>
<td>Appliance Settings</td>
<td>195</td>
</tr>
<tr>
<td>5</td>
<td>Monitoring</td>
<td>203</td>
</tr>
<tr>
<td>5.1</td>
<td>Overview</td>
<td>203</td>
</tr>
<tr>
<td>6</td>
<td>Logs</td>
<td>209</td>
</tr>
<tr>
<td>6.1</td>
<td>Logs</td>
<td>209</td>
</tr>
<tr>
<td>7</td>
<td>Backups</td>
<td>215</td>
</tr>
<tr>
<td>7.1</td>
<td>Initial Backups Setup</td>
<td>215</td>
</tr>
<tr>
<td>7.2</td>
<td>Configuring Backups during Provisioning</td>
<td>216</td>
</tr>
<tr>
<td>7.3</td>
<td>Summary</td>
<td>217</td>
</tr>
<tr>
<td>7.4</td>
<td>Backups</td>
<td>217</td>
</tr>
<tr>
<td>7.5</td>
<td>Managing Backups</td>
<td>219</td>
</tr>
<tr>
<td>8</td>
<td>Operations</td>
<td>223</td>
</tr>
<tr>
<td>8.1</td>
<td>Dashboard</td>
<td>223</td>
</tr>
<tr>
<td>8.2</td>
<td>Reports</td>
<td>223</td>
</tr>
<tr>
<td>8.3</td>
<td>Analytics</td>
<td>225</td>
</tr>
<tr>
<td>8.4</td>
<td>Guidance</td>
<td>228</td>
</tr>
<tr>
<td>8.5</td>
<td>Budgets</td>
<td>229</td>
</tr>
<tr>
<td>8.6</td>
<td>Scheduling</td>
<td>233</td>
</tr>
<tr>
<td>8.7</td>
<td>Approvals</td>
<td>237</td>
</tr>
<tr>
<td>8.8</td>
<td>Usage</td>
<td>240</td>
</tr>
<tr>
<td>8.9</td>
<td>Activity</td>
<td>240</td>
</tr>
<tr>
<td>8.10</td>
<td>Health</td>
<td>241</td>
</tr>
<tr>
<td>9</td>
<td>Services</td>
<td>245</td>
</tr>
<tr>
<td>9.1</td>
<td>Cypher</td>
<td>245</td>
</tr>
<tr>
<td>9.2</td>
<td>Archives</td>
<td>247</td>
</tr>
<tr>
<td>10</td>
<td>Integration Guides</td>
<td>251</td>
</tr>
<tr>
<td>10.1</td>
<td>Automation</td>
<td>251</td>
</tr>
<tr>
<td>10.2</td>
<td>Backups</td>
<td>272</td>
</tr>
<tr>
<td>10.3</td>
<td>Clouds</td>
<td>275</td>
</tr>
<tr>
<td>10.4</td>
<td>Containers</td>
<td>369</td>
</tr>
<tr>
<td>10.5</td>
<td>Deployment</td>
<td>377</td>
</tr>
<tr>
<td>10.6</td>
<td>DNS</td>
<td>377</td>
</tr>
<tr>
<td>10.7</td>
<td>Identity Management</td>
<td>381</td>
</tr>
<tr>
<td>10.8</td>
<td>ITSM</td>
<td>391</td>
</tr>
<tr>
<td>10.9</td>
<td>Keys and Certificates</td>
<td>397</td>
</tr>
<tr>
<td>10.10</td>
<td>Load Balancers</td>
<td>398</td>
</tr>
<tr>
<td>10.11</td>
<td>Logs</td>
<td>401</td>
</tr>
<tr>
<td>10.12</td>
<td>Monitoring</td>
<td>402</td>
</tr>
<tr>
<td>10.13</td>
<td>Networking</td>
<td>403</td>
</tr>
<tr>
<td>10.14</td>
<td>Service Discovery</td>
<td>415</td>
</tr>
</tbody>
</table>
## 11 Troubleshooting

- **11.1 Ansible Troubleshooting** .................................................. 425
- **11.2 Attaching Logs to Case** .................................................. 425
- **11.3 Blank Dashboard** ......................................................... 427
- **11.4 Cannot Login** .............................................................. 427
- **11.5 CLI Troubleshooting** ...................................................... 428
- **11.6 Common Ports & Requirements** ......................................... 428
- **11.7 How to un-manage an Instance/VM/Host** .............................. 430
- **11.8 Deleting Instances** ...................................................... 431
- **11.9 Morpheus Agent Install Troubleshooting** .................. 434
- **11.10 Morpheus UI not loading after upgrade or reconfigure** ...... 438
- **11.11 Remote Console** .......................................................... 440
- **11.12 Restart a Morpheus Installation** ....................................... 444
- **11.13 Unable to Delete Tenant** .............................................. 445
- **11.14 Unable to Provision a Custom Image** ............................... 445
- **11.15 Variables** ................................................................. 446

## 12 Release Notes

- **12.1 v3.6.2** ................................................................. 453
- **12.2 v3.6.1** ................................................................. 458
- **12.3 v3.6.0** ................................................................. 460
- **12.4 v3.5.3** ................................................................. 468
- **12.5 v3.4.4** ................................................................. 472
- **12.6 v3.5.2** ................................................................. 474
- **12.7 v3.4.3** ................................................................. 476
- **12.8 v3.5.1** ................................................................. 477
- **12.9 v3.4.2** ................................................................. 479
- **12.10 v3.5.0** ................................................................. 481
- **12.11 v3.4.1** ................................................................. 485
- **12.12 v3.4.0** ................................................................. 486

## 13 Getting Started

- **13.1 Installation** ............................................................... 489
- **13.2 Setup Appliance** ......................................................... 490
- **13.3 Get Started** ............................................................... 490
- **13.4 Provisioning** .............................................................. 492
- **13.5 List Arguments** .......................................................... 494

## 14 Setup New Appliance

## 15 Managing Instances

- **15.1 Listing Instances** ....................................................... 501
- **15.2 Creating Instances** ..................................................... 503
- **15.3 Removing Instances** ................................................... 504

## 16 Shell

- **16.1 Open a shell** ........................................................... 505

## 17 Morpheus Commands

- **17.1 morpheus access-token** ............................................. 509
- **17.2 morpheus alias** ....................................................... 510
- **17.3 morpheus apps** ....................................................... 512
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.4 morpheus archives</td>
<td>522</td>
</tr>
<tr>
<td>17.5 morpheus benchmark</td>
<td>531</td>
</tr>
<tr>
<td>17.6 morpheus blueprints</td>
<td>533</td>
</tr>
<tr>
<td>17.7 morpheus clouds</td>
<td>547</td>
</tr>
<tr>
<td>17.8 morpheus containers</td>
<td>554</td>
</tr>
<tr>
<td>17.9 morpheus cypher</td>
<td>560</td>
</tr>
<tr>
<td>17.10 morpheus datastores</td>
<td>564</td>
</tr>
<tr>
<td>17.11 morpheus deploy</td>
<td>566</td>
</tr>
<tr>
<td>17.12 morpheus deployments</td>
<td>567</td>
</tr>
<tr>
<td>17.13 morpheus edit-profile</td>
<td>570</td>
</tr>
<tr>
<td>17.14 morpheus edit-rc</td>
<td>570</td>
</tr>
<tr>
<td>17.15 morpheus execute-schedules</td>
<td>571</td>
</tr>
<tr>
<td>17.16 morpheus execution-request</td>
<td>577</td>
</tr>
<tr>
<td>17.17 morpheus file-copy-request</td>
<td>579</td>
</tr>
<tr>
<td>17.18 morpheus groups</td>
<td>582</td>
</tr>
<tr>
<td>17.19 morpheus hosts</td>
<td>588</td>
</tr>
<tr>
<td>17.20 morpheus image-builder</td>
<td>600</td>
</tr>
<tr>
<td>17.21 morpheus instance-types</td>
<td>606</td>
</tr>
<tr>
<td>17.22 morpheus instances</td>
<td>608</td>
</tr>
<tr>
<td>17.23 morpheus key-pairs</td>
<td>637</td>
</tr>
<tr>
<td>17.24 morpheus library-file-templates</td>
<td>640</td>
</tr>
<tr>
<td>17.25 morpheus library-instance-types</td>
<td>644</td>
</tr>
<tr>
<td>17.26 morpheus library-layouts</td>
<td>649</td>
</tr>
<tr>
<td>17.27 morpheus library-node-types</td>
<td>653</td>
</tr>
<tr>
<td>17.28 morpheus library-option-lists</td>
<td>657</td>
</tr>
<tr>
<td>17.29 morpheus library-option-types</td>
<td>661</td>
</tr>
<tr>
<td>17.30 morpheus library-scripts</td>
<td>665</td>
</tr>
<tr>
<td>17.31 morpheus library-upgrades</td>
<td>669</td>
</tr>
<tr>
<td>17.32 morpheus license</td>
<td>673</td>
</tr>
<tr>
<td>17.33 morpheus load-balancers</td>
<td>675</td>
</tr>
<tr>
<td>17.34 morpheus login</td>
<td>679</td>
</tr>
<tr>
<td>17.35 morpheus logout</td>
<td>679</td>
</tr>
<tr>
<td>17.36 morpheus monitor-apps</td>
<td>680</td>
</tr>
<tr>
<td>17.37 morpheus monitor-checks</td>
<td>687</td>
</tr>
<tr>
<td>17.38 morpheus monitor-contacts</td>
<td>696</td>
</tr>
<tr>
<td>17.39 morpheus monitor-groups</td>
<td>699</td>
</tr>
<tr>
<td>17.40 morpheus monitor-incidents</td>
<td>706</td>
</tr>
<tr>
<td>17.41 morpheus network-domains</td>
<td>716</td>
</tr>
<tr>
<td>17.42 morpheus network-groups</td>
<td>720</td>
</tr>
<tr>
<td>17.43 morpheus network-pool-servers</td>
<td>724</td>
</tr>
<tr>
<td>17.44 morpheus network-pools</td>
<td>728</td>
</tr>
<tr>
<td>17.45 morpheus network-proxies</td>
<td>732</td>
</tr>
<tr>
<td>17.46 morpheus network-services</td>
<td>736</td>
</tr>
<tr>
<td>17.47 morpheus networks</td>
<td>737</td>
</tr>
<tr>
<td>17.48 morpheus passwd</td>
<td>742</td>
</tr>
<tr>
<td>17.49 morpheus policies</td>
<td>743</td>
</tr>
<tr>
<td>17.50 morpheus power-schedules</td>
<td>749</td>
</tr>
<tr>
<td>17.51 morpheus process</td>
<td>756</td>
</tr>
<tr>
<td>17.52 morpheus recent-activity</td>
<td>759</td>
</tr>
<tr>
<td>17.53 morpheus remote</td>
<td>759</td>
</tr>
<tr>
<td>17.54 morpheus roles</td>
<td>763</td>
</tr>
<tr>
<td>17.55 morpheus security-group-rules</td>
<td>773</td>
</tr>
<tr>
<td>17.56 morpheus security-groups</td>
<td>775</td>
</tr>
<tr>
<td>17.57 morpheus shell</td>
<td>778</td>
</tr>
<tr>
<td>23.27 Policies</td>
<td>976</td>
</tr>
<tr>
<td>23.28 Power Schedules</td>
<td>986</td>
</tr>
<tr>
<td>23.29 Preseed Scripts</td>
<td>991</td>
</tr>
<tr>
<td>23.30 Process History</td>
<td>995</td>
</tr>
<tr>
<td>23.31 Provision Types</td>
<td>1004</td>
</tr>
<tr>
<td>23.32 Provisioning</td>
<td>1011</td>
</tr>
<tr>
<td>23.33 Resource Folders</td>
<td>1014</td>
</tr>
<tr>
<td>23.34 Resource Pools</td>
<td>1017</td>
</tr>
<tr>
<td>23.35 Roles</td>
<td>1021</td>
</tr>
<tr>
<td>23.36 Security Group Rules</td>
<td>1029</td>
</tr>
<tr>
<td>23.37 Security Groups</td>
<td>1032</td>
</tr>
<tr>
<td>23.38 Setup</td>
<td>1034</td>
</tr>
<tr>
<td>23.39 SSL Certificates</td>
<td>1035</td>
</tr>
<tr>
<td>23.40 Storage Buckets</td>
<td>1037</td>
</tr>
<tr>
<td>23.41 Tasks</td>
<td>1041</td>
</tr>
<tr>
<td>23.42 Task Types</td>
<td>1043</td>
</tr>
<tr>
<td>23.43 User Settings</td>
<td>1049</td>
</tr>
<tr>
<td>23.44 User Sources</td>
<td>1052</td>
</tr>
<tr>
<td>23.45 Users</td>
<td>1057</td>
</tr>
<tr>
<td>23.46 Virtual Images</td>
<td>1060</td>
</tr>
<tr>
<td>23.47 Whoami</td>
<td>1064</td>
</tr>
<tr>
<td>23.48 Workflows</td>
<td>1067</td>
</tr>
</tbody>
</table>
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 vm’s and containers. To get started some base requirements are recommended:

1.1.1 Base Requirements

- **Operating System:** Ubuntu 14.04 /16.04 or CentOS/RHEL greater than 7.0.
- **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)
- **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.**
- **Access to base yum and apt repos**
- **An Appliance License is required for any operations involving provisioning.**
- **Internet Connectivity (optional)**
  - To download from Morpheus’ public docker repositories and system Virtual Image catalog
  - Offline installation require installing the offline package in addition to the regular installation package.

**Note:** Access to base yum and apt repos is still required for offline installations.
• VM and Host Agent Install (optional)
  – Inbound connectivity access from provisioned vm’s and container hosts on ports 443 (Agent install and communication) and 80 (Linux Agent installs via yum and apt)
  – 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 Admin->Settings.

**Note:** Ubuntu 16.10 and Amazon Linux are not supported.

### 1.1.2 Storage Considerations

Upon initial installation Morpheus takes up less than 10 GB of space, however Morpheus Services, Virtual Images, Backups, Logs and stats and user uploaded and imported data require adequate space on the Morpheus Appliance(s) per Appliance Configuration and activity.

**Important:** It is the customers responsibility to ensure adequate storage space per configuration and use case.

#### Default Paths

/\texttt{/opt/morpheus} Morpheus Application and Services Files

/\texttt{/var/opt/morpheus} User, Application and Services Data, including default config Elasticsearch, RabbitMQ and Database data, and default Virtual Image path.

/\texttt{/var/log} Morpheus Service logs

/\texttt{/tmp/morpheus} Working directory for Backups

#### 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 /\texttt{/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 /\texttt{/tmp/morpheus/backups/}. When using none NFS Storage providers, the backup file(s) must be written to /\texttt{/tmp/morpheus/working/} before they can be zipped, sent to the destination Storage provider such as S3, and removed from /\texttt{/tmp/morpheus/working/}. Please ensure adequate space in /\texttt{/tmp/morpheus/} when offloading Backups.

#### Migrations

When performing a Hypervisor to Hypervisor migration, such as VMware to AWS, Virtual Images are written to local storage before conversion and/or upload to the target hypervisor. Please ensure adequate space in /\texttt{/var/opt/morpheus/morpheus-ui/vms} or other configured local Storage Provider paths when performing Migrations.
VM Logs and Stats

When using a Morpheus configuration with locally installed ElasticSearch, VM, Container, Host and Appliance logs and stats are are stored in Elasticsearch. Please ensure adequate space in /var, specifically /var/opt/morpheus/elasticsearch in relation to the number or Instances reporting logs, log frequency, and log retention count.

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.

1.1.3 Network Connectivity

Morpheus primarily operates via communication with its agent that is installed on all managed vm’s 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 Admin->Settings). When the agent is started it automatically makes this connection and securely authenticates. Therefore, it is necessary for all vm’s and docker based hosts that are managed by morpheus to be able to reach the appliance server ip on port 443.

Morpheus also utilizes SSH (Port 22) and Windows Remote Management (Port 5985) to initialize a server. This includes sending remote command instructions to install the agent. It is actually possible for Morpheus to operate without agent connectivity (though stats and logs will not function) and utilize SSH/WinRM to perform operations. Once the agent is installed and connections are established SSH/WinRM communication will stop. This is why an outbound requirement exists for the appliance server to be able to utilize port 22 and 5985.

Note: In newer versions of morpheus this outbound connectivity is not mandatory. The agent can be installed by hand or via Guest Process API's on cloud integrations like VMware.

1.1.4 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)
- Redis (Cache 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.

1.1.5 Common Ports & Requirements

The following chart is useful for troubleshooting Agent install, Static IP assignment, Remote Console connectivity, and Image transfers.
Table 1: Common Ports & Requirements

<table>
<thead>
<tr>
<th>Feature</th>
<th>Method</th>
<th>OS</th>
<th>Source</th>
<th>Destination</th>
<th>Port</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent Communication</td>
<td>All</td>
<td>All</td>
<td>Node</td>
<td>Appliance</td>
<td>443</td>
<td>DNS Resolution from node to appliance url</td>
</tr>
<tr>
<td>Agent Install</td>
<td>All</td>
<td>Linux</td>
<td>Node</td>
<td>Appliance</td>
<td>80</td>
<td>Used for appliance yum and apt repos</td>
</tr>
<tr>
<td></td>
<td>SSH</td>
<td>Linux</td>
<td>Node</td>
<td>Appliance</td>
<td>22</td>
<td>DNS Resolution from node to appliance url</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Virtual Images configured</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SSH Enabled on Virtual Image</td>
</tr>
<tr>
<td>WinRM</td>
<td>Windows</td>
<td>Appliance</td>
<td>Node</td>
<td></td>
<td>5985</td>
<td>DNS Resolution from node to appliance url</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Virtual Images configured</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>WinRM Enabled on Virtual Image (winrm quickconfig)</td>
</tr>
<tr>
<td>Cloud-init</td>
<td>Linux</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cloud-init installed on template/image</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cloud-init settings populated in User Settings or in Admin -&gt; Provisioning</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Agent install mode set to Cloud-Init in Cloud Settings</td>
</tr>
<tr>
<td>Cloudbase-init</td>
<td>Windows</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cloudbase-init installed on template/image</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cloud-init settings populated in User Settings or in Admin -&gt; Provisioning</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Agent install mode set to Cloud-Init in Cloud Settings</td>
</tr>
<tr>
<td>VMtools</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VMtools installed on template</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cloud-init settings populated in Morpheus user settings or in Administration -&gt; Provisioning when using Static IP’s</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Existing User credentials entered on Virtual Image when using DHCP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RPC mode set to VMtools in VMware cloud settings.</td>
</tr>
<tr>
<td>Static IP Assignment &amp; IP Pools</td>
<td>Cloud-Init</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
<td>Network configured in Morpheus (Gateway,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Primary and Secondary DNS, CIDR populated,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DHCP disabled)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cloud-init/Cloudbase-init installed on template/image</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cloud-init settings populated in Morpheus user settings or in Administration -&gt; Provisioning</td>
</tr>
<tr>
<td>VMware Tools</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Network configured in Morpheus (Gateway,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Primary and Secondary DNS, CIDR populated,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DHCP disabled)</td>
</tr>
</tbody>
</table>
1.2 Installation

Morpheus comes packaged as a debian or yum based package. It can be installed on a single on/off premise linux based host or configured for high availability and horizontal scaling. Morpheus is currently only supported on Ubuntu 14.04, Ubuntu 16.04, CentOS 7.0 or newer, and RHEL 7.0 or newer based hosts.

Note: You can view our offline installation guide at offline-installer.

1.2.1 Ubuntu

To get started installing Morpheus on Ubuntu (14.04 currently) a few prepratory items should be addressed first.

1. First make sure the apt repository is up to date by running `sudo apt-get update`. It might also be advisable to verify that 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.

2. Next simply download the relevant .deb package for installation. This package can be acquired from your account rep or via a free trial request from.

   Tip: Use the `wget` command to directly download the package to your appliance server. i.e. `wget https://downloads.gomorpheus.com/path/to/package.deb` THIS IS NOT THE PACKAGE URL. The package URL can be acquired from your account rep or via a free trial request from.

3. 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
   ```

4. 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 post install settings that can be viewed in the Advanced section of the guide.

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.
1.2.2 CentOS

To get started installing Morpheus on CentOS a few preparatory items should be addressed first.

1. Configure firewalld to allow access from users on port 80 or 443 (Or remove firewall if not required).
2. Make sure the machine is self resolvable to its own hostname.

**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.

3. Next simply download the relevant `.rpm` package for installation. This package can be acquired from your account rep or via a free trial request from .

**Tip:** Use the `wget` command to directly download the package to your appliance server, i.e. `wget https://downloads.gomorpheus.com/path/to/package.rpm` THIS IS NOT THE PACKAGE URL. The package URL can be acquired from your account rep or via a free trial request from.

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

```
sudo rpm -i morpheus-appliance-x.x.x-1.x86_64.rpm
sudo morpheus-ctl reconfigure
```

5. 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 post install settings that can be viewed in the Advanced section of the guide.

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`. 

---

1.2. Installation
1.2.3 RHEL

To get started installing Morpheus on RHEL 7 a few prerequisite items are required.

1. Configure firewalld to allow access from users on port 80 or 443 (Or remove firewall if not required).
2. Make sure the machine is self resolvable to its own hostname.
3. For RHEL, In order for the guacamole service (remote console) to properly install some additional optional repositories first need added.
   - RHEL 7.x Amazon: `yum-config-manager --enable rhui-REGION-rhel-server-optional`
   - 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.

**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.

The RedHat Enterprise Linux 7 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.el7 python-rhsm: 1.10.12-2.el7
```

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.el7 python-rhsm: 1.10.12-2.el7
```

If the subscription manager re-turns 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 your account rep or via a free trial request from .

**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 post install settings that can be viewed in the Advanced section of the guide.

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.

---

1.2. Installation
1.2.4 Additional Options

There are several additional configuration options during installation that may be performed. For example, Morpheus provides convenient options for uploading your own SSL certificates as well as externalizing several dependent services.

System Defaults

Morpheus follows several install location conventions. Below is a list of system defaults for convenient management:

- **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
  - Redis: /var/log/morpheus/redis
- **User-defined install/config:** /etc/morpheus/morpheus.rb

SSL Certificates

The default installation generates a self-signed SSL certificate. To implement a third-party certificate:

1. Copy the private key and certificate to /etc/morpheus/ssl/your_fqdn_name.key and /etc/morpheus/ssl/your_fqdn_name.crt respectively.

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

   ```ruby
   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.

3. Next simply reconfigure the appliance and restart nginx:

   ```bash
   sudo morpheus-ctl reconfigure
   sudo morpheus-ctl restart nginx
   ```
**Additional Configuration Options**

There are several other options available to the `/etc/morpheus/morpheus.rb` file that can be useful when setting up external service integrations or high availability:

```ruby
mysql['enable'] = false
mysql['host'] = '52.53.240.28'
mysql['port'] = 10004
mysql['morpheus_db'] = 'morpheusdb01'
mysql['morpheus_db_user'] = 'merovingian'
mysql['morpheus_password'] = 'Wm5n5gXqXCe9v52'
rabbitmq['enable'] = false
rabbitmq['vhost'] = 'zion'
rabbitmq['queue_user'] = 'dujour'
rabbitmq['queue_user_password'] = '5tfg9n2iBifzW5c'
rabbitmq['host'] = '54.183.196.152'
rabbitmq['port'] = '10008'
rabbitmq['stomp_port'] = '10010'
redis['enable'] = false
redis['host'] = '52.53.240.28'
redis['port'] = 10009
elasticsearch['enable'] = false
elasticsearch['cluster'] = 'nebuchadnezzar'
elasticsearch['es_hosts'] = {'52.53.214.68' => 10003}
```

These settings allow one to externally configure and scale mysql, elasticsearch, redis, and rabbitmq which is critical for a high availability setup.

**1.3 Initial Appliance Setup**

**1.3.1 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 longs 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](https://example.morpheusdata.com).

The Appliance URL can be changed later, and also set to different url per cloud integration.
1. 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 Admin -> Settings page, where you will add your License Key.

### 1.3.2 Login Methods

**Master Tenant**
- Enter username or email. 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.

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Details</th>
<th>Active</th>
</tr>
</thead>
</table>

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

### 1.3.3 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 - Provisioning. Its is recommended to define these settings after installation unless credentials are defined per Virtual Image for Provisioning.
1.3.4 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.

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.

1.4 Upgrading

Morpheus provides a very simple and convenient upgrade process. In most cases it is simply a matter of installing the new package on top of itself and reconfiguring the services.

**Important:** All services except the morpheus-ui must be running during a reconfigure. The morpheus-ui also must be restarted or stopped and started during an upgrade. Failure to do so will result in errors.

1.4.1 Debian / Ubuntu

Simply download the latest package or request the latest package from your account service representative.
Then run the install process as follows:

```bash
sudo dpkg -i morpheus-appliance_x.x.x-1.amd64.deb
sudo morpheus-ctl stop morpheus-ui
sudo morpheus-ctl reconfigure
sudo morpheus-ctl start morpheus-ui
```

This typically is enough to complete a full upgrade. Databases will automatically be migrated upon restart of the application and service version upgrades will automatically be applied.

### 1.4.2 CentOS / RHEL

Yum based package upgrades are a little different. In this case we want to run a `rpm -U` command as the package manager is slightly different.

```bash
sudo rpm -U morpheus-appliance-x.x.x-1.x86_64.rpm
sudo morpheus-ctl stop morpheus-ui
sudo morpheus-ctl reconfigure
sudo morpheus-ctl start morpheus-ui
```

**Tip:** Sometimes it may be necessary to restart all appliance services on the host. In order to do this simply type `sudo morpheus-ctl restart`. This will restart ALL services.

### 1.4.3 Deploy WAR file

**Download the war file**

```bash
wget <url>
```

**Move the file**

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

**Change permissions**

```bash
chown morpheus-app.morpheus-app /opt/morpheus/lib/morpheus/morpheus-ui.war
```

**Restart UI**

```bash
morpheus-ctl restart morpheus-ui
```

### 1.5 Advanced Configuration

Morpheus provides more advanced configuration capabilities, including High Availability configurations, and support for tougher network environments with offline installation and Proxy configurations.
1.5.1 Offline Installer

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

Offline Installer 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 offline packages must be downloaded from another system and transferred to the Morpheus Appliance server.

Note: The offline package is linked 1-to-1 to the appliance release. For example the offline package for 2.12.2-1 should be used with the appliance package 2.12.2-1

Offline Install

Ubuntu/Debian

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

```
wget http://example_url/morpheus-appliance_package_url.deb
wget http://example_url/morpheus-appliance_package_offline_url.deb
```

2. Install the appliance package. DO NOT run morpheus-ctl reconfigure yet.

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

3. Install the offline package using dpkg -i morpheus-appliance-offline_2.12.2-rc1-1_all.deb.

```
sudo dpkg -i morpheus-appliance-offline_version_all.deb
```

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

```
sudo vi /etc/morpheus/morpheus.rb
edit appliance_url to resolvable url (if not configured correctly by default)
```

5. 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.

1.5. Advanced Configuration
CentOS/RHEL

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

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

2. Install the appliance package. DO NOT run morpheus-ctl reconfigure yet.

```
sudo rpm -i morpheus-appliance_version_amd64.rpm
```

3. Install the offline package using rpm -i morpheus-appliance-offline_2.12.2~rc1-1_all.rpm

```
sudo rpm -i morpheus-appliance-offline_version_all.rpm
```

4. 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
```

5. 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.

1.5.2 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 Admin -> 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 aa 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 hosts 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.

### 1.5.3 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:
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 10.1.2.10 morpheus -p < /tmp/morpheus_backup.sql
Enter password:
```

The data form 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
```

### 1.6 High Availability Configuration

#### 1.6.1 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 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 App 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.
Important: This is a sample configuration only. Customer configurations and requirements will vary.

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 5.6.10) 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. (8gb) 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 clustered 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. 4–8gb of Memory is recommended to start.

**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, Tomcat, and Redis (to be phased out at a later date).
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.
1.6.2 Database Tier

Morpheus needs a database to connect to. Out of the box Morpheus uses MySQL but Morpheus supports any MySQL compliant database. There are many ways to set up a highly available, MySQL dialect based database. One which has found favor with many of our customers is Percona’s XtraDB Cluster. Percona’s product is based off of Galera’s WSREP Clustering, which is also supported.

If you’re not as familiar with WSREP and prefer replication, some of our customers prefer to configure a failover connection to a MariaDB or MySQL based Master/Master Replication cluster. Less often used, though still a viable option, is MySQL based NDB Clustering. Wonderful guides for each of these HA and DR based database management strategies can be found here: https://www.percona.com/doc/percona-xtradb-cluster/LATEST/index.html

Requirements

*Note:* Morpheus idiomatically connects to database nodes over 3306

Once you have your database installed and configured:
1. Create the Database you will be using with morpheus.

```
mysql> CREATE DATABASE morpheusdb;
mysql> show databases;
```

2. Next create your morpheus database user. The user needs to be either at the IP address of the morpheus application server or use '@%' within the user name to allow the user to login from anywhere.

```
mysql> CREATE USER '$morpheus_db_user_name'@'$source_ip' IDENTIFIED BY '$morpheus___db_user_pw';
```

3. Next Grant your new morpheus user permissions to the database.

```
mysql> GRANT ALL PRIVILEGES ON morpheus_db_name.* TO 'morpheus_db_user'@'$source_ip' IDENTIFIED BY 'morpheus__db_user_pw' with grant option;
mysql> GRANT SELECT, PROCESS, SHOW DATABASES, SUPER ON *.* TO 'morpheus_db_user'@'$source_ip' IDENTIFIED BY 'morpheus_db_user_pw';
mysql> FLUSH PRIVILEGES;
```

4. Checking Permissions for your user.

```
SHOW GRANTS FOR '$morpheus_db_user_name'@$source_ip';
```

### 1.6.3 RabbitMQ Cluster

An HA deployment will also include a Highly Available RabbitMQ. This can be achieved through RabbitMQ’s HA-Mirrored Queues on at least 3, independent nodes. To accomplish this we recommend following Pivotal’s documentation on RabbitMQ here: [https://www.rabbitmq.com/ha.html](https://www.rabbitmq.com/ha.html) and [https://www.rabbitmq.com/clustering.html](https://www.rabbitmq.com/clustering.html)

Install RabbitMQ on the 3 nodes and create a cluster.

**Note:** For the most up to date RPM package we recommend using this link: [https://www.rabbitmq.com/install-rpm.html#downloads](https://www.rabbitmq.com/install-rpm.html#downloads)

**Important:** Morpheus connects to AMQP over 5672 or 5671(SSL) and 61613 or 61614(SSL)

```
rabbitmq-plugins enable rabbitmq_stomp
rabbitmqctl set_policy -p morpheus --apply-to queues --priority 1 statCommands -="statCommands.*" '{expires:1800000}''
rabbitmqctl set_policy -p morpheus --apply-to queues --priority 1 morpheusAgentActions "morpheusAgentActions.*" '{expires:1800000}''
```

### 1.6.4 Elasticsearch

Install 3 node Elasticsearch Cluster on Centos 7
Important: This is a sample configuration only. Customer configurations and requirements will vary.

Requirements

1. Three Existing CentOS 7+ nodes accessible to the Morpheus Appliance
2. Install Java on each node
   
   You can install the latest OpenJDK with the command:

   ```
   sudo yum install java-1.8.0-openjdk.x86_64
   ```

   To verify your JRE is installed and can be used, run the command:

   ```
   java -version
   ```

   The result should look like this:

   ```
   Output of java -version
   openjdk version "1.8.0_65"
   OpenJDK Runtime Environment (build 1.8.0_65-b17)
   OpenJDK 64-Bit Server VM (build 25.65-b01, mixed mode)
   ```

Installation

To install Elasticsearch please use the following instructions


Once installed, to make sure Elasticsearch starts and stops automatically, add its init script to the default runlevels with the command:

```
sudo systemctl enable elasticsearch.service
``` 

Configuring Elastic

Now that Elasticsearch and its Java dependencies have been installed, it is time to configure Elasticsearch.

The Elasticsearch configuration files are in the /etc/elasticsearch directory. There are two files:

```
sudo vi /etc/elasticsearch/elasticsearch.yml
``` 

- **elasticsearch.yml** Configures the Elasticsearch server settings. This is where all options, except those for logging, are stored, which is why we are mostly interested in this file.

- **logging.yml** Provides configuration for logging. In the beginning, you don’t have to edit this file. You can leave all default logging options. You can find the resulting logs in /var/log/elasticsearch by default.

The first variables to customize on any Elasticsearch server are node.name and cluster.name in elasticsearch.yml. As their names suggest, node.name specifies the name of the server (node) and the cluster to which the latter is associated.

1.6. High Availability Configuration
Important: Make sure to uncomment each of the following listed below in /etc/elasticsearch/elasticsearch.yml

Node 1

```yaml
cluster.name: morpheusha1
node.name: "morpheuses1"
network.host: enter the IP of the node ex: 10.30.22.130
http.port: 9200
discovery.zen.ping.unicast.hosts: ["10.30.20.91","10.30.20.149","10.30.20.165"]
```

Node 2

```yaml
cluster.name: morpheusha1
node.name: "morpheuses2"
network.host: enter the IP of the node ex: 10.30.22.130
http.port: 9200
discovery.zen.ping.unicast.hosts: ["10.30.20.91","10.30.20.149","10.30.20.165"]
```

Node 3

```yaml
cluster.name: morpheusha1
node.name: "morpheuses3"
network.host: enter the IP of the node ex: 10.30.22.130
http.port: 9200
discovery.zen.ping.unicast.hosts: ["10.30.20.91","10.30.20.149","10.30.20.165"]
```

For the above changes to take effect, you will have to restart Elasticsearch with the command:

```
sudo service elasticsearch restart
```

Next restart the network with the command:

```
sudo service network restart
```

Testing

To make sure Elasticsearch is running use the following commands


1.6.5 Application Tier

Morpheus configuration is controlled by a configuration file located at /etc/morpheus/morpheus.rb. This file is read when you run morpheus-ctl reconfigure after installing the appliance package. Each section is tied to a deployment tier: database is mysql, message queue is rabbitmq, search index is elasticsearch. There are no entries for the web and application tiers since those are part of the core application server where the configuration file resides.

1. Download and install the Morpheus Appliance Package
2. Next we must install the package onto the machine and configure the morpheus services:
3. After installing and prior to reconfiguring, edit the `morpheus.rb` file

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

Change the values to match your configured services:

**Note:** The values below are examples. Update hosts, ports, usernames and password with your specifications. Only include entries for services you wish to externalize.

```ruby
mysql['enable'] = false
mysql['host'] = '10.30.20.139:3306,10.30.20.153:3306,10.30.20.196'
mysql['morpheus_db'] = 'morpheusdb'
mysql['morpheus_db_user'] = 'dbuser'
mysql['morpheus_password'] = 'dbuserpassword'
rabbitmq['enable'] = false
rabbitmq['vhost'] = 'morpheus'
rabbitmq['queue_user'] = 'lbuser'
rabbitmq['queue_user_password'] = 'lbuserpassword'
rabbitmq['host'] = 'rabbitvip'
rabbitmq['port'] = '5672'
rabbitmq['stomp_port'] = '61613'
rabbitmq['heartbeat'] = 50
elasticsearch['enable'] = false
elasticsearch['cluster'] = 'esclustername'
elasticsearch['es_hosts'] = {'10.30.20.91' => 9200, '10.30.20.149' => 9200, '10.30.20.165' => 9200}
```

1. Reconfigure Morpheus

```
sudo morpheus-ctl reconfigure
```

### 1.6.6 Storage

When Morpheus is in a High Availability configuration the required Local Storage File Shares will need to be copied to a shared file system so that all nodes within the Morpheus cluster is able to connect to assets.

**Assets**

- White label images
- Uploaded virtual images
- Deploy uploads
- Ansible Plays
- Terraform
- Morpheus backups.
Tip: Backups, deployments and virtual images can be overridden within the Morpheus-UI. You can find more information on storage here: Storage

To copy the `morpheus-ui` directory to the shared storage follow the below steps:

1. SSH into the Appliance
2. sudo su (or login as root)
3. cd into `/var/opt/morpheus/`
4. Backup morpheus-ui directory by running the command below. This will create a new directory in `/var/opt/morpheus/` called morpheus-ui-bkp and copy the contents of morpheus-ui into the new directory
   
   ```
   cp -r morpheus-ui morpheus-ui-bkp
   ```

5. Move morpheus-ui to your shared storage. Example below:

   ```
   mv morpheus-ui /nfs/appliance-files/
   ```

6. Mount your shared storage volume to `/var/opt/morpheus/morpheus-ui`. How you mount it is dependent on what kind of storage it is. If you mount the volume after the package install, but before the reconfigure then you don’t need to copy anything to a backup.

7. SSH into the second Appliance and then Backup morpheus-ui directory by running

   ```
   cp -r morpheus-ui morpheus-ui-bkp
   ```

Tip: when adding additional nodes you will only need to run step 6 and 7

1.6.7 3 Node with Externalized DB Configuration

Assumptions

This guide assumes the following:

- There is an externalized database running for Morpheus to access.
- The database service is a MySQL dialect (MySQL, MariaDB, Galera, etc…)
- A database has been created for Morpheus as well as a user and proper grants have been run for the user. Morpheus will create the schema.
- The Baremetal nodes cannot access the public internet
- The base OS is RHEL 7.x
- Shortname versions of hostnames will be resolvable
- All nodes have access to a shared volume for `/var/opt/morpheus/morpheus-ui`. This can be done as a post startup step.
- This configuration will support the complete loss of a single node, but no more. Specifically the Elasticsearch tier requires at least two nodes to always be clustered.
Steps

1. First begin by downloading the requisite Morpheus packages either to the nodes or to your workstation for transfer. These packages need to be made available on the nodes you wish to install Morpheus on.

```
[root@app-server-1 ~]# wget https://example/path/morpheus-appliance-ver-1.el7.x86_64.rpm
[root@app-server-1 ~]# wget https://example/path/morpheus-appliance-offline-ver-1.noarch.rpm
```

2. Once the packages are available on the nodes they can be installed. Make sure that no steps beyond the rpm install are run.

```
[root@app-server-1 ~] rpm -i morpheus-appliance-ver-1.el7.x86_64.rpm
[root@app-server-1 ~] rpm -i morpheus-appliance-offline-ver-1.noarch.rpm
```

3. Next you will need to edit the Morpheus configuration file `/etc/morpheus/morpheus.rb` on each node.

```
Node 1

appliance_url 'https://morpheus1.localdomain'
elasticsearch['es_hosts'] = {'10.100.10.121' => 9200, '10.100.10.122' => 9200, '10.100.10.123' => 9200}
elasticsearch['node_name'] = 'morpheus1'
elasticsearch['host'] = '0.0.0.0'
rabbitmq['host'] = '0.0.0.0'
rabbitmq['nodename'] = 'rabbit@node01'
mysql['enable'] = false
mysql['host'] = '10.100.10.111'
mysql['morpheus_db'] = 'morpheusdb'
```

(continues on next page)
Node 2

appliance_url 'https://morpheus2.localdomain'
elasticsearch['es_hosts'] = {'10.100.10.121' => 9200, '10.100.10.122' => 9200, '10.100.10.123' => 9200}
elasticsearch['node_name'] = 'morpheus2'
elasticsearch['host'] = '0.0.0.0'
rabbitmq['host'] = '0.0.0.0'
rabbitmq['nodename'] = 'rabbit@node02'
mysql['enable'] = false
mysql['host'] = '10.100.10.112'
mysql['morpheus_db'] = 'morpheusdb'
mysql['morpheus_db_user'] = 'morpheus'
mysql['morpheus_password'] = 'password'

Node 3

appliance_url 'https://morpheus3.localdomain'
elasticsearch['es_hosts'] = {'10.100.10.121' => 9200, '10.100.10.122' => 9200, '10.100.10.123' => 9200}
elasticsearch['node_name'] = 'morpheus3'
elasticsearch['host'] = '0.0.0.0'
rabbitmq['host'] = '0.0.0.0'
rabbitmq['nodename'] = 'rabbit@node03'
mysql['enable'] = false
mysql['host'] = '10.100.10.113'
mysql['morpheus_db'] = 'morpheusdb'
mysql['morpheus_db_user'] = 'morpheus'
mysql['morpheus_password'] = 'password'

Note: If you are running MySQL in a Master/Master configuration we will need to slightly alter the mysql['host'] line in the morpheus.rb to account for both masters in a failover configuration. As an example: mysql['host'] = '10.100.10.111:3306,10.100.10.112'. Morpheus will append the ‘3306’ port to the end of the final IP in the string, which is why we leave it off but explicitly type it for the first IP in the string. The order of IPs matters in that it should be the same across all three Morpheus Application Servers. As mentioned, this will be a failover configuration for MySQL in that the application will only read/write from the second master if the first master becomes unavailable. This way we can avoid commit lock issues that might arise from a load balanced Master/Master.

4. Run the reconfigure on all nodes

[root@app-server-1 ~] morpheus-ctl reconfigure

Morpheus will come up on all nodes and Elasticsearch will auto-cluster. The only item left is the manual clustering of RabbitMQ.

5. Select one of the nodes to be your Source Of Truth (SOT) for RabbitMQ clustering. We need to copy the secrets for RabbitMQ, copy the erlang cookie and join the other nodes to the SOT node.

Begin by copying secrets from the SOT node to the other nodes.
```
[root@app-server-1 ~] cat /etc/morpheus/morpheus-secrets.json
{
  "rabbitmq": {
    "morpheus_password": "***REDACTED***",
    "queue_user_password": "***REDACTED***",
    "cookie": "***REDACTED***"
  }
}
```

Then copy the erlang.cookie from the SOT node to the other nodes
```
[root@app-server-1 ~]# cat /opt/morpheus/embedded/rabbitmq/.erlang.cookie
# 754363AD864649RD63D28
```

6. Once this is done run a reconfigure on the two nodes that are NOT the SOT nodes.
```
[root@app-server-2 ~] morpheus-ctl reconfigure
```

**Note:** This step will fail. This is ok, and expected. If the reconfigure hangs then use Ctrl+C to quit the reconfigure run and force a failure.

7. Subsequently we need to stop and start Rabbit on the NOT SOT nodes.

**Important:** The commands below must be run at root

**Note:** If you receive an error unable to connect to epmd (port 4369) on app-server-1: nxdomain (non-existing domain) make sure to add all IPs and hostnames to the etc/hosts file like so:
```
127.0.0.1   localhost   localhost.localdomain   localhost4   localhost4.localdomain4
::1         localhost   localhost.localdomain   localhost6   localhost6.localdomain6
127.0.0.1   app-server-1.localdomain   app-server-2   localhost
127.0.0.1   container16
10.100.10.113 app-server-1
10.100.10.114 app-server-2
10.100.10.115 app-server-3
```
```
[root@app-server-2 ~]# morpheus-ctl stop rabbitmq
[root@app-server-2 ~]# morpheus-ctl start rabbitmq
[root@app-server-2 ~]# PATH=/opt/morpheus/sbin:/opt/morpheus/sbin:/opt/morpheus/˓
→embedded/sbin:/opt/morpheus/embedded/bin:$PATH
[root@app-server-2 ~]# rabbitmqctl stop_app
Stopping node 'rabbit@app-server-2' ...
[root@app-server-2 ~]# rabbitmqctl join_cluster rabbit@app-server-1
Clustering node 'rabbit@app-server-2' with 'rabbit@app-server-1' ...
[root@app-server-2 ~]# rabbitmqctl start_app
Starting node 'rabbit@app-server-2' ...
```

1.6. High Availability Configuration 29
8. Now make sure to reconfigure

```
[root@app-server-2 ~] morpheus-ctl reconfigure
```

9. Once the Rabbit services are up and clustered on all nodes they need to be set to HA/Mirrored Queues:

```
[root@app-server-2 ~]# rabbitmqctl set_policy -p morpheus --priority 1 --apply-to all ha ".*,.*" '{"ha-mode": "all"}'
```

10. The last thing to do is restart the Morpheus UI on the two nodes that are NOT the SOT node.

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

If this command times out then out then run:

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

11. You will be able to verify that the UI services have restarted properly by inspecting the logfiles. A standard practice after running a restart is to tail the UI log file.

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

12. Lastly, we need to ensure that Elasticsearch is configured in such a way as to support a quorum of 2. We need to do this step on EVERY NODE.

```
[root@app-server-2 ~]# echo "discovery.zen.minimum_master_nodes: 2" >> /opt/morpheus/embedded/elasticsearch/config/elasticsearch.yml
[root@app-server-2 ~]# morpheus-ctl restart elasticsearch
```

**Note:** For moving /var/opt/morpheus/morpheus-ui files into a shared volume make sure ALL Morpheus services on ALL three nodes are down before you begin.

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

13. Permissions are as important as is content, so make sure to preserve directory contents to the shared volume.

14. Subsequently you can start all Morpheus services on all three nodes and tail the Morpheus UI log file to inspect errors.

### Database Migration

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

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

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

2. 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
```
3. Take note of this 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.

4. 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-1 ~]# morpheus-ctl stop morpheus-ui
[root@app-server-2 ~]# morpheus-ctl stop morpheus-ui
[root@app-server-3 ~]# morpheus-ctl stop morpheus-ui
```

5. 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-1 ~]# /opt/morpheus/embedded/mysql/bin/mysql -u morpheus -h 10.130.2.38 morpheus -p < /tmp/morpheus_backup.sql
Enter password:
```

**Recovery**

If a node happens to crash most of the time Morpheus will start upon boot of the server and the services will self-recover. However, there can be cases where RabbitMQ and Elasticsearch are unable to recover in a clean fashion and it require minor manual intervention. Regardless, it is considered best practice when recovering a restart to perform some manual health checks.

```
[root@app-server-1 ~]# morpheus-ctl status
run: check-server: (pid 17808) 7714s;
run: log: (pid 549) 8401s
```

(continues on next page)
run: elasticsearch: (pid 19207) 5326s;
run: log: (pid 565) 8401s
run: guacd: (pid 601) 8401s;
run: log: (pid 573) 8401s
run: morpheus-ui: (pid 17976) 7633s;
run: log: (pid 555) 8401s
run: nginx: (pid 581) 8401s;
run: log: (pid 544) 8401s
run: rabbitmq: (pid 17850) 7708s;
run: log: (pid 542) 8401s
run: redis: (pid 572) 8401s;
run: log: (pid 548) 8401s

But, a status can report false positives if, say, RabbitMQ is in a boot loop or Elasticsearch is up, but not able to join
the cluster. It is always advisable to tail the logs of the services to investigate their health.

| [root@app-server-1 ~]# morpheus-ctl tail rabbitmq |
| [root@app-server-1 ~]# morpheus-ctl tail elasticsearch |

To minimize disruption to the user interface, it is advisable to remedy Elasticsearch clustering first. Due to write
locking in Elasticsearch it can be required to restart other nodes in the cluster to allow the recovering node to join.
Begin by determining which Elasticsearch node became the master during the outage. On one of the two other nodes
(not the recovered node):

| [root@app-server-2 ~]# curl localhost:9200/_cat/nodes |
| app-server-1 10.100.10.121 7 47 0.21 d * morpheus1 |
| localhost 127.0.0.1 4 30 0.32 d m morpheus2 |

The master is determined by identifying the row with the ‘*’ in it. SSH to this node (if different) and restart Elastic-
search.

| [root@app-server-1 ~]# morpheus-ctl restart elasticsearch |

Go to the other of the two ‘up’ nodes and run the curl command again. If the output contains three nodes then
Elasticsearch has been recovered and you can move on to re-clustering RabbitMQ. Otherwise you will see output that
contains only the node itself:

| [root@app-server-2 ~]# curl localhost:9200/_cat/nodes |
| localhost 127.0.0.1 4 30 0.32 d * morpheus2 |

If this is the case then restart Elasticsearch on this node as well:

| [root@app-server-2 ~]# morpheus-ctl restart elasticsearch |

After this you should be able to run the curl command and see all three nodes have rejoined the cluster:

| [root@app-server-2 ~]# curl localhost:9200/_cat/nodes |
| app-server-1 10.100.10.121 9 53 0.31 d * morpheus1 |
| localhost 127.0.0.1 7 32 0.22 d m morpheus2 |
| app-server-3 10.100.10.123 3 28 0.02 d m morpheus3 |

The most frequent case of restart errors for RabbitMQ is with epmd failing to restart. Morpheus’s recommendation is
to ensure the epmd process is running and daemonized by starting it:

| [root@app-server-1 ~]# /opt/morpheus/embedded/lib/erlang/erts-5.10.4/bin/epmd -daemon |
And then restarting RabbitMQ:

```
[root@app-server-1 ~]# morpheus-ctl restart rabbitmq
```

And then restarting the Morpheus UI service:

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

Again, it is always advisable to monitor the startup to ensure the Morpheus Application is starting without error:

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

**Recovery Thoughts/Further Discussion:** If Morpheus UI cannot connect to RabbitMQ, Elasticsearch or the database tier it will fail to start. The Morpheus UI logs can indicate if this is the case.

Aside from RabbitMQ, there can be issues with false positives concerning Elasticsearch’s running status. The biggest challenge with Elasticsearch, for instance, is that a restarted node has trouble joining the ES cluster. This is fine in the case of ES, though, because the minimum_master_nodes setting will not allow the un-joined singleton to be consumed until it joins. Morpheus will still start if it can reach the other two ES hosts, which are still clustered.

The challenge with RabbitMQ is that it is load balanced behind Morpheus for requests, but each Morpheus application server needs to bootstrap the RabbitMQ tied into it. Thus, if it cannot reach its own RabbitMQ startup for it will fail.

Similarly, if a Morpheus UI service cannot reach the database, startup will fail. However, if the database is externalized and failover is configured for Master/Master, then there should be ample opportunity for Morpheus to connect to the database tier.

Because Morpheus can start even though the Elasticsearch node on the same host fails to join the cluster, it is advisable to investigate the health of ES on the restarted node after the services are up. This can be done by accessing the endpoint with curl and inspecting the output. The status should be “green” and number of nodes should be “3”:

```
[root@app-server-1 ~]# curl localhost:9200/_cluster/health?pretty=true
{
"cluster_name" : "morpheus",
"status" : "green",
"timed_out" : false,
"number_of_nodes" : 3,
"number_of_data_nodes" : 3,
"active_primary_shards" : 110,
"active_shards" : 220,
"relocating_shards" : 0,
"initializing_shards" : 0,
"unassigned_shards" : 0,
"number_of_pending_tasks" : 0,
"number_of_in_flight_fetch" : 0
}
```

If this is not the case it is worth investigating the Elasticsearch logs to understand why the singleton node is having trouble joining the cluster. These can be found at:

```
/var/log/morpheus/elasticsearch/current
```

Outside of these stateful tiers, the “morpheus-ctl status” command will not output a “run” status unless the service is successfully running. If a stateless service reports a failure to run, the logs should be investigated and/or sent to Morpheus for additional support. Logs for all Morpheus embedded services are found in `var/log/morpheus`.  

1.6. High Availability Configuration 33
1.7 Morpheus Agent

The Morpheus Agent is an important and powerful facet of Morpheus as an orchestration tool. Though it is not required (one unique capability of our platform vs. some of the competitors out there), it is recommended for use as it brings with it a lot of insightful benefits. Not only does it provide statistics of 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. The Morpheus Agent is very lightweight and secure.

**Note:** The agent is not required by Morpheus to become a managed instance. If you don’t have the agent installed we try to aggregate stats but it can vary based on the cloud and can be limited or inaccurate.

The Morpheus Agent 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 things 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 up with regards to the agent while demonstrating its security benefits as well as 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.

1.7.1 Key Agent Features

- Provides key enhanced statistics (disc usage, CPU usage, network, disc IO)
- Handles log aggregation
- Provides a command bus where Morpheus doesn’t need to get credentials to access a box. Can still run workflows if credentials are changed
- SSH agent can be disabled and still get access to the box
- Agent can be installed over Cloud Init for internetless situations

**The Morpheus agent is optional**

- Makes a single connect that’s persistence over HTTPS web socket and runs as a service
- Health checks for Linux (not available on windows)

**No inbound Ports**

- Agent buffers and compresses logs and sends them in chunks to minimize packets
- Can be configured to collect logs and send them somewhere
- Linux agent can be shrunk and should be less then .2% peak (Windows less 97 kb)
- Run workflows, Have expiration/shutdown policies and can help reign in environments amongst other things
- Accepts commands, can execute commands, write files, and manipulate firewall

1.7.2 Morpheus Agent Support

Microsoft Windows
Note: if you require tls 1.2 then .net 4.5 should be installed.

- Windows Server 2008R2 (Requires .Net 4.3 framework)
- Windows Server 2012
- Windows Server 2012R2
- Windows Server 2016
- Windows Server 2019
- Windows 10 PRO

**Redhat Based linux Distrubution:**

- Redhat 6.6
- Redhat 7.x
- CentOS 6.x
- CentOS 7.x
- Oracle 6.x
- Oracle 7.x

**Debian Based linux Distrubutions:**

- Ubuntu 14.04.x
- Ubuntu 16.04.x
- Ubuntu 18.04.x (Only supported for VM, not docker host)
- Debian 8.x
- Debian 9.x

**Unix Based Operating Systems:**

- MacOS Mojave
- MacOS High Sierra
- MacOS Sierra

### 1.8 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: Work loads 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)

### 1.9 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** This list 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, redis, 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:

<table>
<thead>
<tr>
<th>General Commands:</th>
</tr>
</thead>
<tbody>
<tr>
<td>cleanse</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Delete <em>all</em> morpheus data, and start from scratch.</td>
</tr>
<tr>
<td>help</td>
</tr>
<tr>
<td>Print this help message.</td>
</tr>
<tr>
<td>reconfigure</td>
</tr>
<tr>
<td>Reconfigure the application.</td>
</tr>
<tr>
<td>show-config</td>
</tr>
</tbody>
</table>

(continues on next page)
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:

   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.
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.

2.1 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 lots 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.

2.1.1 VM Provision Steps

While an infinite number of steps can happen when provisioning an Instance or App using a 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 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 longs 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.

2.1.2 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 scrip 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 upon 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 Images that has Cloud Init Enabled? checked on the Virtual Record in Morpheus but does not have cloud-init install 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 -> 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 n 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 configure 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 deslected.

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 of the Windows Image has been sys-prepped. If enabled Morpheus will inject Unattend.xml through the Nutanix API (v3+ only)

Important: Provisioning a Virtual Images that has Cloud Init Enabled? checked on the Virtual Record in Morpheus but does not have cloud-init install will result in immediate provisioning failure.

Important: For Linux images without Cloud-Init, and 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.

### 2.1.3 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 Admin -> 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.

In addition to the main appliance_url in Admin -> 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 (Admin -> Settings) is both reachable and resolvable from the provisioned node.

The appliance_url begins with to 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 Provisioning - 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 configure, 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 Provisioning -> 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 Provisioning -> 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 Provisioning -> Virtual Images section.
Cloud-Init agent install mode

- Cloud-Init is configured in Admin -> Provisioning section
- Provisioned image/blueprint has Cloud-Init (linux) or Cloudbase-Init (windows) installed

2.2 Provisioning Concepts

Morpheus is a powerful infrastructure agnostic Cloud Application Management Platform. As a result of this there are some differing concepts compared to other CMP platforms in the space. It is here that it is important to notice the qualification difference between Morpheus and other platforms.

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 based 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.

2.2.1 Instances

Morpheus starts with provisioning Instances. In some platforms an Instance is representative of a singular object like a “Virtual Machine” in Amazon. 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 even several docker containers.

For example, in the morpheus Instance wizard Mongo is an option and contains several “Instance Configurations”. One of these configurations is a full Mongo 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. 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 horizontally scaled out to handle traffic.

When viewing an instance detail page, one is able to look at details/statistics specific to a virtual machine or container. Morpheus simply helps simplify the management model for tracking these services.

2.2.2 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 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).

2.2.3 Hosts / Servers

This concept is mostly tailored to users of morpheus responsible for managing and maintaining the underlying infrastructure integrations. A Host typically refers to a Docker Host in which a container in an instance is running, or a hypervisor 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 (5 minutes typically) with real-time information and changes. This, in essence, provides CMDB based 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 also optionally be installed to provide more refined guest operating system level statistics and logging.

### 2.2.4 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 like Consul.

### 2.2.5 Blueprints

A blueprint is typically referred to as an Application Blueprint. It 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.

### 2.3 Instances

Instances are a great starting point for taking advantage of self-service features and spinning up both VM’s 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.

**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.

The Instance list contains important information about each instance, including the instance name, environment tag, instance type icon, IP 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.
2.3.1 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 thereby restricting access to non sanctioned catalog items, as well as added to via the Provisioning -> 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. These cloud integrations can also be in multiple groups and is also useful for using role based access control 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 the main section on Getting started by setting up some cloud integrations and infrastructure first. 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 option that was chosen and 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 initial username. 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 Mongo 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), 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).

2.3.2 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 differ between instance types and user permissions.

There are several sections within an Instance page that provide useful capabilities to the user.

**Summary** 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 (i.e. mysql presents the my.ini)

**Network** Useful for configuring security groups and access to the instance.

**Monitoring** Quick summary of the monitoring system and all checks that were configured to test the state of the instance

**Backups** Quick backup dashboard. Useful for viewing historical backups as well as kicking off new ones.

**Logs** View all aggregated logs from the containers or VM’s representing the instance.

**Environment** View the environment variables presented to the instances or exported by the instances via Apps (more on this in the Apps section). Even see Imported environment variables that may be referenced by the running instance.

**Scale** For instances that support load balancing and auto scaling. Easily configure auto scaling thresholds and load balancer settings that pertain to a particular instance.

**Console** Access the instance or container via a client-less Console supporting SSH, RDP, VNC, and even hypervisor level remote consoles.

### 2.3.3 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 associated VM’s or Containers and cannot be undone. To delete instances without deleting associated VM’s, delete the VM from the Infrastructure section and uncheck “Remove Infrastructure” while checking “Remove Associated Instances” in the delete modal options.

**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.

**Lock/Unlock Instance** A locked instance cannot be deleted until it is unlocked.

**Import As Image** Creates a Virtual Image Blueprint from the Instance at its current state and adds it to the Virtual Image library with corresponding metadata.
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 Provisioning -> Automation section.

Run Script  Presents Script options and immediately executes selected Script on the Instance. Scripts can be created in the Provisioning -> Library section.

Apply Template  Presents Template options and immediately applies selected Template to the Instance. Templates can be created in the Provisioning -> 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 Notes section for adding useful information about the Instance. Notes can be added by selecting the ADD NOTES button on the bottom of Instance Detail pages. Existing notes can be edited by selecting the EDIT NOTES

Tip:  Markdown Syntax is supported in Instance Notes.

2.4 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 Provisioning -> Blueprints or from Existing Apps.

### 2.4.1 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:** Blueprints must be created in Provisioning -> Blueprints. to appear as options when creating 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.
2.4.2 Creating Apps fromExisting 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.
2.4.3 Exporting JSON or YAML

1. Navigate to Provisioning -> Apps
2. On an existing App, select ACTIONS -> EXPORT
3. Select YAML or JSON in the top right.
4. Select the Configurations to include in the Export by clicking on a Configuration. Selected Configurations will be highlighted.
5. Select the DOWNLOAD CONFIGURATION button.

6. The Blueprint Export file will be downloaded to your computer as `{app_name}-config.json` or `{app_name}-config.yaml`.

**Tip:** JSON or YAML can also be Viewed, Edited, Copied, or Exported by selecting ACTIONS → EDIT on an App and clicking the RAW section in the Edit App modal.
2.4.4 Provisioning Apps via API

A quick example of how this work: https://d.pr/i/yxsW7t
2.5 Blueprints

2.5.1 Overview

With the release of Morpheus version 3, a new app blueprint builder was developed from scratch and extended to support a vast array of configurations even with programmatic markup or (Infrastructure as Code) capabilities.

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.

Some unique capabilities of the YAML/JSON based Morpheus blueprint structure is it’s 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 or group the app is being provisioned in as a target. The environment can also be used as a scope. Maybe the “development” environment doesn’t need as many horizontally scaled nodes (for example) 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 Cloud Formation 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:

```yaml
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.

```yaml
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:

```yaml
name: Simple Nginx
type: morpheus
tiers:
  Web:
    linkedTiers: []
    instances:
      - instance:
          type: docker
          name: Sample Nginx
          clouds:
            AWS Cali:
              instance:
                layout:
```

(continues on next page)
Theres 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 option types 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.
2.5.2 Creating App Blueprints

1. Navigate to Provisioning -> 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 by selecting the Lock icon next to the Field. Locking prevent the field from being editable when provisioning an App using the Blueprint.
   • 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.
Blueprints can be created, edited, or exported in the RAW section when creating or editing a blueprint.

To Export a Blueprint as JSON or YAML:

1. Create or edit a blueprint.
2. Select the RAW section on the top of the APP BLUEPRINT modal.
3. Select JSON or YAML in the top right of the RAW section.
4. Select the EXPORT button.
5. Select the configurations to include in the export by clicking on a configuration. Selected configurations will be highlighted.
6. Select the DOWNLOAD CONFIGURATION button.
7. The Blueprint Export file will be downloaded to your computer as `{template_name}-config.json` or `{template_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 evars 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.

### 2.6 Automation

**Provisioning -> 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**.

#### 2.6.1 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.
## Task Types

<table>
<thead>
<tr>
<th>Task Type</th>
<th>Task Description</th>
<th>Task Target</th>
<th>Configuration Requirements</th>
<th>Role Permissions Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ansible</strong></td>
<td>Runs an Ansible playbook. Ansible Integration required</td>
<td>Instance or Host</td>
<td>Existing Ansible Integration</td>
<td>Provisioning: Tasks</td>
</tr>
<tr>
<td><strong>Chef bootstrap</strong></td>
<td>Executes Chef bootstrap and run list. Chef Integration required</td>
<td>Instance or Host</td>
<td>Existing Chef Integration</td>
<td>Provisioning: Tasks</td>
</tr>
<tr>
<td><strong>Groovy script</strong></td>
<td>Executes Groovy Script locally (on Morpheus app node)</td>
<td>Local</td>
<td>None</td>
<td>Provisioning: Tasks, Tasks - Script Engines</td>
</tr>
<tr>
<td><strong>HTTP</strong></td>
<td>Executes REST call for targeting external API’s.</td>
<td>URL specified in Task</td>
<td>None</td>
<td>Provisioning: Tasks</td>
</tr>
<tr>
<td><strong>Javascript</strong></td>
<td>Executes Javascript locally (on Morpheus app node)</td>
<td>Local</td>
<td>None</td>
<td>Provisioning: Tasks, Tasks - Script Engines</td>
</tr>
<tr>
<td><strong>jRuby Script</strong></td>
<td>Executes Ruby script locally (on Morpheus app node)</td>
<td>Local</td>
<td>None</td>
<td>Provisioning: Tasks, Tasks - Script Engines</td>
</tr>
<tr>
<td><strong>Library Script</strong></td>
<td>Creates a Task from an existing Library Script (Provisioning -&gt; Library -&gt; Scripts)</td>
<td>Instance or Host</td>
<td>Existing Library Script</td>
<td>Provisioning: Tasks</td>
</tr>
<tr>
<td><strong>Library Template</strong></td>
<td>Creates a Task from an existing Library Template (Provisioning -&gt; Library-&gt; Templates)</td>
<td>Instance or Host</td>
<td>Existing Library Templates</td>
<td>Provisioning: Tasks</td>
</tr>
<tr>
<td><strong>Local Shell Script</strong></td>
<td>Executes Bash script locally (on Morpheus app node)</td>
<td>Local</td>
<td>None</td>
<td>Provisioning: Tasks, Tasks - Script Engines</td>
</tr>
<tr>
<td><strong>Puppet Agent Install</strong></td>
<td>Executes Puppet Agent bootstrap, writes puppet.conf and triggers agent checkin. Puppet Integration required</td>
<td>Instance or Host</td>
<td>Existing Puppet Integration</td>
<td>Provisioning: Tasks</td>
</tr>
<tr>
<td><strong>Python Script (jython)</strong></td>
<td>Executes Python script locally (on Morpheus app node)</td>
<td>Local</td>
<td>None</td>
<td>Provisioning: Tasks, Tasks - Script Engines</td>
</tr>
<tr>
<td><strong>Remote Shell Script</strong></td>
<td>Executes Bash script against the Instance or Host the Task or Workflow is ran on</td>
<td>Instance or Host</td>
<td>None</td>
<td>Provisioning: Tasks</td>
</tr>
<tr>
<td><strong>Restart</strong></td>
<td>Restarts target VM/Host/Container and confirms status before executing next task in Workflow</td>
<td>Instance or Host</td>
<td>None</td>
<td>Provisioning: Tasks</td>
</tr>
<tr>
<td><strong>SSH Script</strong></td>
<td>Execute Bash script against IP specified in Task.</td>
<td>IP specified in Task</td>
<td>None</td>
<td>Provisioning: Tasks</td>
</tr>
<tr>
<td><strong>WinRM Script</strong></td>
<td>Execute Powershell script against IP specified in Task.</td>
<td>IP specified in Task</td>
<td>None</td>
<td>Provisioning: Tasks</td>
</tr>
</tbody>
</table>
Ansible Playbook

Description  Runs an Ansible playbook. Ansible Integration required
Target  Instance or Host
Role Permissions  Provisioning: Tasks

Task Configuration

NAME  Name of the Task
CODE  Unique code name for api, cli, and variable reference
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 Ref’s 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.

Chef Bootstrap

Description  Executes Chef bootstrap and run list. Chef Integration required
Target  Instance or Host
Role Permissions  Provisioning: Tasks

Task Configuration

NAME  Name of the Task
CODE  Unique code name for api, cli, and variable reference
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 uon 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 {}

Chapter 2. Provisioning
Groovy script

**Description**  Executes Groovy Script locally (on app node)

**Target**  Local App Node

**Role Permissions**  Provisioning: Tasks Provisioning: Tasks - Script Engines

**Task Configuration**

- **NAME**  Name of the Task
- **CODE**  Unique code name for api, cli, and variable reference
- **RESULT TYPE**
  - Single Value
  - Key/Value Pairs
  - JSON
- **SCRIPT**  Contents of Groovy Script to execute

HTTP (api)

**Description**  Executes REST call for targeting external API’s.

**Target**  URL specified in Task

**Role Permissions**  Provisioning: Tasks

**Task Configuration**

- **NAME**  Name of the Task
- **CODE**  Unique code name for api, cli, and variable reference
- **RESULT TYPE**
  - Single Value
  - Key/Value Pairs
  - JSON
- **URL**  http or https url for 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
### Table 2: Http Header examples

<table>
<thead>
<tr>
<th>Authorization</th>
<th>Bearer <strong>token</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Type</td>
<td>application/json</td>
</tr>
</tbody>
</table>

#### Javascript

**Description**  
Executes Javascript locally (on app node)

**Target**  
Local App Node

**Role Permissions**  
Provisioning: Tasks Provisioning: Tasks - Script Engines

**Task Configuration**

- **NAME**  
  Name of the Task
- **CODE**  
  Unique code name for api, cli, and variable reference
- **RESULT TYPE**
  - Single Value
  - Key/Value Pairs
  - JSON
- **SCRIPT**  
  Contents of Javascript to execute

#### jRuby Script

**Description**  
Executes Ruby script locally (on app node)

**Target**  
Local App Node

**Role Permissions**  
Provisioning: Tasks Provisioning: Tasks - Script Engines

**Task Configuration**

- **NAME**  
  Name of the Task
- **CODE**  
  Unique code name for api, cli, and variable reference
- **RESULT TYPE**
  - Single Value
  - Key/Value Pairs
  - JSON
- **SCRIPT**  
  Contents of jRuby Script to execute
Library Script

**Description** Creates a Task for an existing Library Script (Provisioning -> Library -> Scripts)

**Target** Instance or Host

**Role Permissions** Provisioning: Tasks

**Task Configuration**

- **NAME** Name of the Task
- **CODE** Unique code name for api, cli, and variable reference
- **RESULT TYPE**
  - Single Value
  - Key/Value Pairs
  - JSON
- **SCRIPT** Search for and select existing Library Script

Library Template

**Description** Creates a Task for an existing Library Template (Provisioning -> Library-> Templates)

**Target** Instance or Host

**Role Permissions** Provisioning: Tasks

**Task Configuration**

- **NAME** Name of the Task
- **CODE** Unique code name for api, cli, and variable reference
- **TEMPLATE** Search for and select existing Library Template

Local Shell Script

**Description** Executes Bash script locally (on Morpheus app node)

**Target** Local App Node

**Role Permissions** Provisioning: Tasks Provisioning: Tasks - Script Engines

**Task Configuration**

- **NAME** Name of the Task
- **CODE** Unique code name for api, cli, and variable reference
- **RESULT TYPE**
Puppet Agent Install

**Description**  Executes Puppet Agent bootstrap, writes puppet.conf and triggers agent checkin. Puppet Integration required

**Target**  Instance or Host

**Role Permissions**  Provisioning: Tasks

**Task Configuration**

- **NAME**  Name of the Task
- **PUPPET MASTER**  Select Puppet Master from existing Puppet Integration
- **PUPPET NODE NAME**  Enter Puppet Node Name. Variables supported eg. 
  
  ```
  <%= instance.name %>
  ```

- **PUPPET ENVIRONMENT**  Enter Puppet Env. eg. production

Python Script (jython)

**Description**  Executes Python script locally (on app node)

**Target**  Local App Node

**Role Permissions**  Provisioning: Tasks - Script Engines

**Task Configuration**

- **NAME**  Name of the Task
- **CODE**  Unique code name for api, cli, and variable reference
- **TYPE**  Python Script (jython)
- **RESULT TYPE**
  - Single Value
  - Key/Value Pairs
  - JSON
- **SCRIPT**  Python Script (jython) Script to execute
Remote Shell Script

**Description**  Executes Bash script against the Instance or Host the Task or Workflow is ran on

**Target**  Instance or Host

**Role Permissions**  Provisioning: Tasks

**Task Configuration**

- **NAME**  Name of the Task
- **CODE**  Unique code name for api, cli, and variable reference
- **RESULT TYPE**
  - Single Value
  - Key/Value Pairs
  - JSON
- **SCRIPT**  Enter Bash Script to execute

Restart

**Description**  Specifically for use in Workflows after a task that requires a restart, the Restart task executes a restart on the target Instance or Host. Morpheus will wait until the restart is complete to execute the next task in the workflow phase.

**Target**  Instance or Host

**Role Permissions**  Provisioning: Tasks

**Task Configuration**

- **NAME**  Name of the Task
- **CODE**  Unique code name for api, cli, and variable reference

SSH Script

**Description**  Execute Bash script against IP specified in Task.

**Target**  IP specified in Task

**Role Permissions**  Provisioning: Tasks

**Task Configuration**

- **NAME**  Name of the Task
- **CODE**  Unique code name for api, cli, and variable reference
- **RESULT TYPE**
  - Single Value
**Key/Value Pairs**

- **IP ADDRESS** IP Address of the ssh task target
- **PORT** SSH port for ssh task target (22 default)
- **KEY** Select existing Keypair for key auth
- **USERNAME** Username for ssh task target
- **PASSWORD** Password for ssh task target
- **SCRIPT** Enter Bash Script to execute

**WinRM Script**

- **Description** Execute Powershell script against IP specified in Task.
- **Target** IP specified in Task
- **Role Permissions** Provisioning: Tasks

**Task Configuration**

- **NAME** Name of the Task
- **CODE** Unique code name for api, cli, and variable reference
- **RESULT TYPE**
  - Single Value
  - Key/Value Pairs
  - JSON

- **IP ADDRESS** IP Address of the WinRM task target
- **PORT** SSH port for WinRM task target (5985 default)
- **USERNAME** Username for WinRM task target
- **PASSWORD** Password for WinRM task target
- **SCRIPT** Enter Script to execute

**Task Management**

**Adding Tasks**

1. Select the Provisioning link in the navigation bar.
2. Select Automation from the sub-navigation menu.
3. Click the *Add* button.
4. From the New Task Wizard input a name for the task.
5. Select the type of task from from the type dropdown.
6. Input the appropriate details dependent on the task type you selected from the dropdown.
7. Save

**Editing Tasks**

1. Select the Provisioning link in the navigation bar.
2. Select Automation from the sub-navigation menu.
3. Click the Edit icon on the row of the task you wish to edit.
4. Modify information as needed.
5. Click the Save Changes button to save.

**Deleting Tasks**

1. Select the Provisioning link in the navigation bar.
2. Select Automation from the sub-navigation menu.
3. Click the Delete icon on the row of the task you wish to delete.

**Task Results**

**Overview**

Task Results allow Tasks to use the output from preceding Tasks in the same Workflow via results variables.

**Configure Tasks**

In script type tasks, if RESULT TYPE is set, Morpheus will store the Task’s output as a variable.

**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.
Examples

Single Value using Task Code

Source Task Config

NAME  Var Code (single)
CODE  single
RESULT TYPE  Single Value
SCRIPT  echo "string value"

Source Task Output  string value

Results Task using task code in variable

Results Task Script  echo "single: <%=results.single%>"
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  keyval
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.keyval%>"
Results Task Output  keyval:  [flash:bang, ping:pong]

Results Task for a single value

Results Task Script  echo "keyval value: <%=results.keyval.flash%>"
Results Task Output  keyval value:  bang

JSON

Source Task Config
NAME Var Code (json)
CODE json
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.json%>"
Results Task Output json: [ping:pong, flash:bang]

Results Task for a single value

Results Task Script \echo\ "json value: <%=results.json.ping%>"
Results Task Output json value: pong

Python

Python tasks can access the results dictionary directly: results[key]. To pass the results of a Python task to the next task within a workflow add the results as a value within the results dictionary. For example results['taskcode'] = 'Hello World' where 'taskcode' is the code of the task.

Results are available for all tasks executed in a workflow. For example, instead of using just one Tasks results in another Task, we can use all of the Task Results from the tasks above in a single task inside a workflow. Multiple Task Results:

Results Task Script
\begin{verbatim}
\echo "single: <%=results.single%>"
\echo "task name: <%=results["Var Code"]%>"
\echo "keyval: <%=results.keyval%>"
\echo "keyval value: <%=results.keyval.flash%>"
\echo "json: <%=results.json%>"
\echo "json value: <%=results.json.ping%>"
\end{verbatim}

Results Task Output
\begin{verbatim}
single: string value
task name: string value
keyval: [flash:bang, ping:pong ]
keyval value: bang
json: [ping:pong, flash:bang]
json value: pong
\end{verbatim}

2.6.2 Workflows

Add Workflow

1. Select the Provisioning 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 button.
5. From the New Workflow Wizard input a name for the workflow.
6. Optionally input a description.
7. Expand the execution phases to add tasks to, and type the name of a created task and click the task when it appears to add.
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. Click the Save Changes button to save.

**Workflow Execution Phases**

For VM’s, Pre-Provision and Provision execute after the VM is running. Pre-Provision can be used for a blueprint so it is added before a script 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 up. Post-Provision will execute after the entire provisioning process is complete.

**Edit Workflow**

1. Select the Provisioning 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 Provisioning 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.

## 2.7 Virtual Images

*Provisioning -> Virtual Images*

### 2.7.1 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.
Important: Invalid Image Settings cause provisioning failures. Morpheus syncs in as much meta-data as possible for synced images, they still need to be properly configure 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.

2.7.2 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.

2.7.3 Configuring Virtual Images

System Images

System 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. It is highly recommended to populate the Administration -> Provisioning -> Cloud-Init section with user data prior to provisioning, as the user and password/key will be added to all Instances provisioned from System Images. Users can also be added during provisioning in the Add User provisioning wizard section.

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 Actions - Edit in the Virtual Images list, or 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 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.

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 or Guest Customizations are used, credentials are defined in Administration -> 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 n 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.

Create Image Select FILE to select or drag and drop image file, or 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. No impact on single-tenant environments.

Auto Join Domain? Enable to have instances provisioned with this image auto-join configured domains (Windows only, domain controller must be configure 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.

Trial Version Enable to automatically re-arm the expiration on Windows Trial Images during provisioning.

Enabled Sysprep? Applicable to Nutanix Only. Enable of the Windows Image has been sys-prepped. If enabled Morpheus will inject Unattend.xml through the Nutanix API (v3+ only)

3. 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.
2.7.4 Provisioning Images

When provisioning a System Image for the first time, Morpheus will download and stream the image from S3 to the source Cloud if the image is not local to the Cloud. The Image will also be cached on the Morpheus Appliance under /var/opt/morpheus/vm/vmcache. Subsequent provisions of the image will use the created template in the Cloud or the cached local Image if the images does not exist in the selected Cloud, in which case the cached Image will be copied to the Cloud.

When using Images that already exist in the destination cloud, such as synced, marketplace, or previously copied images, no image transfer between the Morpheus Appliance and destination cloud will take place.

**Note:** The Morpheus Appliance must be able to download from Amazon S3 when provisioning System Images for the first time.

**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.

2.7.5 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.

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
   - VirtualBox
• VirtualBox (vdi)
• 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 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.

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 or Guest Customizations are used, credentials are defined in Administration -> Provisioning and User Settings. If credentials are defined on the Image and Cloud-Init is enabled, lmorpheus] 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.

Create Image  Select FILE to select or drag and drop image file, or 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. No impact on single-tenant environments.

Auto Join Domain?  Enable to have instances provisioned with this image auto-join configured domains (Windows only, domain controller must be configure 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.
**Trial Version** Enable to automatically re-arm the expiration on Windows Trial Images during provisioning.

**Enabled Sysprep?** Applicable to Nutanix Only. Enable of the Windows Image has been sys-prepped. If enabled Morpheus will inject Unattend.xml through the Nutanix API (v3+ only)

**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 -> 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.**
2.8 Library

2.8.1 Overview

The Library section is used to add virtual images as custom instances to the provisioning catalog. The Library Section is composed of:

- Instance Types
- Layouts
- Node Types
- Option Types
- Option Lists
- File Templates
- Scripts

Uploaded or synced images from the virtual images section are added to nodes, a node or multiple nodes are added to layouts, and layouts are added to Instance Types. Scripts and File Templates can be attached to nodes, with phased execution options for scripts.
Adding an Instance Type creates a new Library Item category. Multiple layouts can be added to an instance type, and these layout can have different nodes attached. The instance 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**  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**  Suggested Dimensions: 150 x 51

**Visibility**

• Private- Only accessibly by assigned Accounts/Tenants
• Public- accessible by all Accounts/Tenants

**Environment Prefix** Used for exportable environment variables when tying instance types together environment variables in app contexts. If not specified a name will be generated

**Enable Scaling (Horizontal)** Enables load balancer assignment and auto-scaling features

**Supports 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.

---

**Note:** Custom Instance Types do not display logs on the instance page. Logs will only show on the VM page.
2.8.3 Node Types

The following fields are for all node technology types:

- Name
- Short Name
- Version
- Category

- Technology * Alibaba * Amazon * Azure * Docker * ESXi * Fusion * Google * Hyper-V * KVM * Nutanix * OpenStack * Oracle VM * SCVMM * UpCloud * vCloud Director * VMware * Xen
- Environment Variables

The Options fields will change depending on the Technology option selected.

For VM provisioning technology options, select an image from the VM Image dropdown, which is populated from the Virtual Images Section and will include images uploaded into Morpheus, and 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.

For Docker, type in the name and version of the Docker Image and select the integrated registry.

**Expose Ports** To open port on the node, select “Add Port” and enter the name and port to expose. The Load Balancer http, https or tcp setting is only required when attaching to load balancers.

Example port configuration:

<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT</th>
<th>LB</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>8000</td>
<td>HTTP</td>
</tr>
<tr>
<td>Collector</td>
<td>8088</td>
<td>None</td>
</tr>
<tr>
<td>Forwarder</td>
<td>9997</td>
<td>None</td>
</tr>
<tr>
<td>KVstore</td>
<td>8191</td>
<td>None</td>
</tr>
<tr>
<td>TCP</td>
<td>1514</td>
<td>None</td>
</tr>
<tr>
<td>Custom</td>
<td>1515</td>
<td>None</td>
</tr>
</tbody>
</table>

**2.8.4 Scripts & File Templates**

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.
Upon save the Node Type will be created, and available for adding to layouts.

2.8.5 Layouts

Layouts are added to Instance types, and will be presented under the Configuration Options dropdown in the Provisioning Wizard for that Instance type.

**Instance Type** Select the Instance Type to add the new Layout to. Custom Instance Types must already be created and one layout cannot be added to multiple instance types, or change Instance Types after creation.

**Note:** Layouts cannot be added to Morpheus provided library items at this time.

**Name** The name the layout will present as in the Configuration Options dropdown in 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 in selected while provisioning.

**Description** Description of the layout

**Technology** Technology determines which cloud this layout will be available for.

**Environment Variables** Custom evars to be added to the instance when provisioned.
Nodes  Single or multiple nodes can be added to a Layout by searching for and selecting the node(s). An example of a layout with multiple nodes is the Hyper-V MySQL Master/Slave layout pictured below (note this is the Layout detail screen after the layout has been created.)

Multi-node Layout example:

<table>
<thead>
<tr>
<th>VM TYPES</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
</tr>
<tr>
<td>MySQL 5.6 Master</td>
</tr>
<tr>
<td>MySQL 5.6 Slave</td>
</tr>
</tbody>
</table>

Upon save, the layout will be attached to the selected Instance Type, and available when provisioning that Instance Type for the appropriate cloud technology.

2.8.6 Option Types

Option Types allow you to create additional fields within the provisioning wizard.
These field entries can then be used in scripts and templates using our variable naming convention (more here).
2.8.7 Option List

Much like Option Types, 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 either be manually entered CSV or JSON or they can be dynamically compiled from REST calls via GET or POST requests.

**Note:** JSON entries must be formatted like the following example: 
```
["name":"Test", "value":1],
{"name":"Testing", "value":2}]
```
Your new Library Item is now ready for provisioning. Multiple Layouts, Versions and Technology types can be added to a single Instance Type.
2.9 Migrations

2.9.1 Migration Types

**Hypervisor to Hypervisor**

*Store* Morpheus will create a snapshot of existing VM and upload the snapshot to virtual image directory. Images that have been uploaded to the Virtual Images library can be converted to VHD, QCOW2, RAW and VMDK formats and then re-provisioned.

*New* Morpheus will create a snapshot of an existing VM, convert from source format to required destination format, and then provision the VM into the target environment.

**Source** VMWare, Openstack, Xen, Nutanix* Azure* Hyper-V* ( *in-development)

**Destination** Softlayer, Openstack, Metapod, Xen, Amazon, VMWare, ESXi, Nutanix, Hyper-V Supported OS Type: Windows or Linux

**Service Impact** Disruptive Migration

**Virtual Image Extract**

The Virtual Image extract capabilities allow for a virtual image to be extracted and stored in the virtual image repository or the image can be migrated into a cloud.

**Source** Any Cloud

**Destination** SoftLayer (Only)

**Supported OS Type** Windows

**Service Impact** Non Disruptive

**Requirements** Requires a separate disk or network share to store the image during conversion process. Capacity of the disk or network share should be sized appropriately to support the data that will be exported.

**Live Stream**

**Note:** Live Stream is deprecated

Live Stream is a linux only streaming process that will take a snapshot of a volume and allow it to be streamed to a destination linux system that is either existing or new. The destination linux must already exist and it can either be a managed or unmanaged VM in Morpheus. The destination will be overwritten from a root level perspective.

**Source** Any Cloud

**Destination** Morpheus

**Supported OS Type** Linux (Only)

**Service Impact** Non Disruptive

**Requirements** Requires the Linux host/guest to be configured for LVM and that free space of the capacity to be streamed is available. A destination linux host/guest must be available to receive the stream.
2.9.2 Add Migration

1. Select the Provisioning link in the navigation bar.
2. Select the Migrations link in the sub-navigations bar.
3. Click the Add Migration button.
4. From the Create Migration Wizard select the type of migration, then click the Next button.

Depending on the Migration Type selected input the following, then click the Next button.

- Hypervisor to Hypervisor * Select Cloud, and Server * Input Host, Remote Port, Username, and Password
- Live Stream * Select Platform, Existing or New, Cloud, and Server * Input Host, SSH Port, SSH User, SSH Password, Public Key, and Logical Volume Device. * Enter Destination details, then click the Next button.

5. Finalize your configuration if needed, then click the complete button.

2.9.3 Manually Start Migration

If you chose to not run your migration in the Create Migration Wizard then you will be able to manually start the migration.

1. Select the Provisioning link in the navigation bar.
2. Select the Migrations link in the sub-navigations bar.
3. Click the actions dropdown of the row of the migration you wish start, and select Run.

2.9.4 Remove Migration

1. Select the Provisioning link in the navigation bar.
2. Select the Migrations link in the sub-navigations bar.
3. Click the actions dropdown of the row of the migration you wish remove, and select Remove.

2.9.5 VMware to AWS Migration

Requirements

When performing a Hypervisor to Hypervisor migration from VMware to AWS, there are some requirements that must be met:

1. Add S3 Storage Provider to Morpheus
2. Set Image Transfer Store in you AWS cloud(s) settings in Morpheus
3. Create VM Import Service roles in your AWS account (not in Morpheus )
4. Storage Provider selected for migration destination must be set as a Local Storage Provider (not AWS)
Add S3 Storage Provider

In the `Infrastructure-> Storage` section, select `+ADD` and enter the same AWS Access key and Secret Key used to create your AWS cloud in Morpheus. Then add the S3 bucket name that the migration image(s) will be added to.

Set Image Transfer Store

Under `Infrastructure -> Clouds`, select your AWS cloud and click `EDIT`. Expand the Advanced Options section and for `IMAGE TRANSFER STORE` select the AWS S3 Storage Provider you created previously and then Save.

Add VM Import Service

**Tip:** Refer to the AWS document below to add the required VM Import Service role in AWS: [http://docs.aws.amazon.com/vm-import/latest/userguide/import-vm-image.html](http://docs.aws.amazon.com/vm-import/latest/userguide/import-vm-image.html)

VM Import requires a role to perform certain operations in your account, such as downloading disk images from an Amazon S3 bucket. You must create a role named vmimport with a trust relationship policy document that allows VM Import to assume the role, and you must attach an IAM policy to the role.

**To create the service role**

Create a file named `trust-policy.json` with the following policy:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": { "Service": "vmie.amazonaws.com" },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:Externalid": "vmimport"
                }
            }
        }
    ]
}
```

You can save the file anywhere on your computer. Take note of the location of the file, because you’ll specify the file in the next step.

Use the create-role command to create a role named vmimport and give VM Import/Export access to it. Ensure that you specify the full path to the location of the `trust-policy.json` file.

```bash
aws iam create-role --role-name vmimport --assume-role-policy-document file://trust-policy.json
```

Create a file named `role-policy.json` with the following policy, where disk-image-file-bucket is the bucket where the disk images are stored:
Use the following put-role-policy command to attach the policy to the role created above. Ensure that you specify the full path to the location of the `role-policy.json` file.

```
aws iam put-role-policy --role-name vmimport --policy-name vmimport --policy-document file:///role-policy.json
```

For more information about IAM roles, see IAM Roles in the IAM User Guide.

### Storage Providers

Set the “Storage Provider” in the migration wizard destination as a Local Storage type, or leave as Select to use the Morpheus Appliance.

A local image must be created by Morpheus prior to S3 upload. A Local Storage provider can be used if one had been added in the Infrastructure-> Storage section. Simply leaving the Storage Provider setting as “select” will create an image on the Morpheus appliance, provided sufficient storage existing on the Morpheus appliance drive.

**Important:** Setting AWS as the Destination Storage Provider will result in a migration failure.

These settings will allow a successful migration from VMware to AWS using the Morpheus migration wizard.
2.10 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.

2.10.1 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 -> 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, its 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.

2.10.2 Deploying to an Instance

Now that a version has been added to a deployment it is easy to push that deploy out to any instance provisioned within Morpheus. Simply navigate to the specific Instance that 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.

**2.10.3 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 Admin -> Provisioning once a storage provider is created within the account.

**2.10.4 Add Deployment**

1. Select the Provisioning link in the navigation bar.
2. Select the Deployments link in the sub-navigation bar.
3. Click the Add button.
4. Enter a Name for the deployment and a description (optional)
5. Click the Save Changes button to save.

**2.10.5 Add Version**

1. Select the Provisioning link in the navigation bar.
2. Select the Deployments link in the sub-navigation bar.
3. Click the Name of the deployment you would like to add a version to.
4. Click the Add Version button.
5. From the Add Version Wizard select the deployment type.
6. Input the Version of the deployment.
7. Depending on the type of deployment selected perform one of the following:
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.

8. Click the Save Changes button to save.

2.10.6 Edit Deployment

1. Select the Provisioning link in the navigation bar.
2. Select the Deployments link in the sub-navigation bar.
3. Click the Edit Deployment icon on the row of the deployment you wish to edit.
4. Modify information as needed
5. Click the Save Changes button to save.

2.10.7 Delete Deployment

1. Select the Provisioning link in the navigation bar.
2. Select the Deployments link in the sub-navigation bar.
3. Click the Delete Deployment icon on the row of the deployment you wish to delete.
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.

3.1 Groups

3.1.1 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

---

- 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`
3.1.2 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

Minimal values: `groups add CLITest` There would be prompt to provide optional values for code and location.
3.1.3 Managing Groups

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.

3.1.4 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.

### 3.1.5 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

### 3.1.6 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.

### 3.2 Clouds

#### 3.2.1 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)
- 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)
• HPE OneView
• Huawei
• Hyper-V
• IBM Cloud
• IBM Cloud Platform
• Kubernetes
• MacStadium
• Metacloud
• Morpheus (Generic Cloud type for PXE and manually added Hosts)
• Nutanix
• Open Telekom Cloud
• OpenStack
• Oracle Public Cloud
• Oracle VM
• Platform 9
• SCVMM
• SoftLayer
• Supermicro (Cloud type for PXE and manually added Supermicro Hosts)
• UCS
• UpCloud
• VMWare ESXi
• VMware Fusion
• VMware vCenter
• VMware vCloud Air
• VMware vCloud Director
• VirtualBox
• Virtustream
• XenServer

Information on each cloud type can be found in the Integration Guides section.

3.2.2 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 Integration Guides section.
3.2.3 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 Unmanaged resources.

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. Last sync date, time and duration is shown under the Cloud name.

**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.
**Hosts** The hosts tab panel displays available hosts in the cloud and displays power, os, name, type, cloud, ip address, nodes, disc space, memory, and status. You can add a container host from this by clicking the Container Hosts button, add a hypervisor host by clicking the HyperVisor button, or perform actions actions by click the Actions button.

**Virtual Machine** Displays an Inventory of Existing Instances in your cloud configuration and provides details such as power, os, name, type, cloud, ip address, nodes, disc space, memory, and status.

**Bare Metal** Setup PXE Boot in the Boot section to add bare metal servers. Once setup you can view information such as power, os, name, type, cloud, ip address, nodes, disc space, memory, and status.

**Security Groups** The Security Groups tab panel displays a list of existing Security groups in the cloud. You can add a security group to this cloud by clicking the Edit Security Groups button.

**Load Balancers** The load balancers tab panel displays available load balancers in the cloud and displays the name, description, type, cloud and host. You can add a load balancer from this tab by clicking the Add Load Balancer button.

**Networks** Displays Networks synced or added to the Cloud.

**DataStores** Displays Datastores synced or added to the Cloud.


**Policies** Manages Policies enforced on the Cloud.

+ **Container Host** Provisions a Docker host into the Cloud, or adds an existing Docker Host (manual) to the Cloud. KVM hosts are also available for Morpheus and Bare Metal cloud types.

+ **Hypervisor** Add an existing Hypervisor to the Cloud. Not available for all Cloud types.

### 3.2.4 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 and managed Hosts and VM’s must be removed prior to deleting a cloud. To remove Instances, Hosts and VM’s from Morpheus without deleting them in the actual Cloud, select Delete on the Host or VM, unselect “Remove Infrastructure” and select “Remove Associated Instances” if Instance are associated with the Hosts or VMs.

### 3.3 Hosts

#### 3.3.1 Overview

The *Infrastructure -> Hosts* section provides a universal stage for viewing and managing Hosts and Virtual Machines from all of your Clouds.

In this section you can:

- View & Manage all Hosts, Virtual Machines & Bare Metal
- Provision Docker & KVM Hosts
• Convert existing hosts to Docker & KVM Hosts
• Add Hypervisors
• Convert Hosts, Virtual Machines and Bare Metal to Managed

**Important:** When local firewall management is enabled, Morpheus will automatically set an IP table rule to allow incoming connections on tcp port 22 from the Morpheus Appliance.

### 3.3.2 Hosts

Hosts in Morpheus are Hypervisors and Docker Hosts that your VM’s and Container are hosted on, such as ESXi, Hyper-V and Docker Hosts. These Hosts are populated from integrated clouds, hosts provisioned from Morpheus, or manually added Hosts.

### 3.3.3 Virtual Machines

The Virtual Machines tab lists all Managed and Unmanaged VM’s across Morpheus. Managed VM’s are either provisioned by Morpheus, or inventoried VM’s that were converted to managed. Unmanaged VM’s are from Cloud integrations with “Inventory Existing Instances” enabled in the Cloud settings.

### 3.3.4 Bare Metal

Bare Metal hosts are from PXE Boot or manually added in this section. Bare Metal hosts that are also Hypervisors will be listed in both the Bare Metal and Hypervisor sections.

### 3.3.5 Docker Hosts

**Overview**

Morpheus can provision Docker Hosts into any cloud, convert existing Hosts to Docker Hosts, or even make itself a Docker Host.

**To add a Docker Host to any cloud:**

1. Navigate to Infrastructure -> Hosts
2. Click the **+CONTAINER HOST** button
3. Select a container host type
4. Select a Group
5. Enter the following:
   - Name
   - Description
   - Visibility
   - Select a Cloud
   - Enter tags (optional)

Then click NEXT.

6. Configure the host options

   Select a Service Plan (Volume, Memory and CPU count fields may not be shown if selected service plan does not have custom options enabled).

   - Add and set size the volumes
   - Set memory size
   - Set the CPU count
   - Select a network

   Optionally configure the following:

   - OS username
   - OS password
   - Domain name
   - Hostname (default is the name previously provided for the container host)
Then click the NEXT button

7. Optionally add any Automation Workflows and configure for Backups.

8. Review and click Complete to save
Your new container host will begin provisioning, and soon be running and ready for containers.

**Add an existing Docker Host**

Morpheus can manage and inventory existing/brownfield Docker Hosts by using the *Manual Docker Host* option.

**Note:** Adding a Docker Host that was previously managed by another Morpheus Appliance will disable management of the host on that Appliance as the Morpheus Agent settings will be reconfigured.

**Note:** *Container Mode* on the Cloud settings where the Host is being added must be set to Morpheus for non-Kubernetes/Swarm hosts.

1. Navigate to Infrastructure -> Hosts
2. Select *+CONTAINER HOST* button
3. Select *Manual Docker Host*
4. In the CREATE HOST Wizard, enter the following:
   - **GROUP**
   - **GROUP** Select the Group this Host will be available for
   Select *NEXT*
   - **NAME**
CLOUD  Select the Cloud the Host will be assigned to
NAME    Enter name for the Docker Host in Morpheus
DESCRIPTION Enter optional description for the Docker Host
VISIBILITY Select Tenant Visibility
TAGS Add optional Morpheus tags (these are not meta-data tags)
Select NEXT
CONFIGURE
SSH HOST Enter IP or resolvable hostname of the target host
SSH USER Enter existing username on the target host
SSH PASSWORD Enter password for SSH User
PUBLIC KEY For key auth (recommended), copy and add the displayed Public Key to the authorized_keys file on the target host.
PLAN Default Manual
LVM ENABLED? Deselect if target host is not LVM enabled (required when using Morpheus provided docker images)
DATA VOLUME Enter path of the target data volume on the target host
SOFTWARE RAID? Enable for software RAID (disabled by default)
NET INTERFACE Enter network interface name of target host’s target network
Select NEXT
AUTOMATION
POST PROVISION Select a workflow to execute after Host is added (optional).
Select NEXT
REVIEW Review settings and select COMPLETE to add the Manual Docker Host.
Your new container host will begin provisioning, and soon be running and ready for containers.

Note: Existing containers will be inventoried after the Hosts is successfully added.

3.4 Network

3.4.1 Networks

Infrastructure -> Network -> Networks

Overview

The Networks section is for configuring networks across all clouds in Morpheus. Existing networks from the 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. Networks can also be set as inactive and unavailable for provisioning use.

### 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 (Scan Network)

When adding or editing a network there is an option to scan network. If checked scan network will ping the IP’s in the network range, and if ping is successful Morpheus will quickly check for listening ports on the IP.

Important: Network scanning may cause network monitoring or other alerts

3.4.2 Services

Overview

The Network Services 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

DNS
• Microsoft DNS
• Power DNS
• Route 53

Service Registry
• Consul

Security
• Cisco ACI

Add a Service

To configure any of the services, select ADD SERVICE, and fill out the required fields.

Infoblox
• Name
• URL (wapi url + version)
  – example https://x.x.x.x/wapi/v2.2.1
• Username
• Password

Bluecat
• Name
• URL
• Username
• Password

Microsoft DNS
• Name
• DNS Server
• Username
• Password
• Zone

PowerDNS
• Name
• API Host
• Token

Route 53
• Region
• Access Key
• Secret Key

After Saving, your Network Service integrations will be available for use. These integrations must be scoped to the appropriate sections in Morpheus:

**Scoping Services**

**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.

**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.

**3.4.3 IP Pools**

**Infrastructure -> Network -> IP Pools**

**Overview**

The Networks IP Pools sections allows you to create Morpheus IP Pools, which is an IP Range Morpheus can use to assign available static IP addresses to instances. The IP Pool section also displays pools from IPAM integrations like Infoblox and Bluecat.

**To add a Morpheus Network Pool**

1. Select *ADD* in the **Infrastructure -> Network -> IP Pools** section
2. Enter the following:
   - **Name** Name of the IP Pool in Morpheus. The name is presented when selecting an IP Pool for a Network, so use a name that easily identifies the IP Pool.
   - **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.

**3.4.4 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 Windows Instances to Domains, and creating A Records with DNS Integrations. The Domains section is also a multi-tenant endpoint for managing domain settings across multiple accounts.

- Added and synced 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 Provisioning -> Virtual Images 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**: Example demo.example.com
   - **Description**: Descriptive meta-data for use in Morpheus
   - **Public Zone**: Check for Public Zones, leave unchecked for Private Zones.
   - **Join Domain Controller**: Enable to have Windows instances join a Domain Controller
   - **Username**: Admin user for Domain Controller
   - **Password**: Password for DC Username
   - **DC Server**: (optional) Specify the URL or Path of the DC Server
   - **OU Path**: (optional) Enter the OU Path for the connection string.
   - **Permissions**: Configure Tenant permissions in Morpheus for the Domain (only applicable in Multi-tenant Morpheus setups)
   - **Tenant**: Select the Tenant to set permissions to for the Domain.

Visibility

- **Private**: Only Accessible by the select Tenant
- **Public**: Available for use by all Tenants.

4. Save Changes

The Domain has been added and will be selectable in Domain dropdown during provisioning, and in Cloud and Network settings.

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.
Editing and Removing Domains

- Domains can be edited by selecting the Actions dropdown for the Domain and selecting Edit.
- Added Domains can be removed from Morpheus by selecting the Actions dropdown for the Domain and selecting Remove.

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.

3.4.5 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 Admin -> 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 hosts 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.

3.4.6 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 Host Level Firewall must be enabled for Security Groups to be applied. The Host Level Firewall can be enabled in *Administration -> Settings -> Host Level Firewall Enable/Disable*

**Important:** When local firewall management is enabled, Morpheus will automatically set an IP table rule to allow incoming connections on tcp port 22 from the Morpheus Appliance.

### 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 cloud to add an ACL.
3. Click the Security Groups tab.
4. Click the *Edit Security Groups* button.
5. Click the + (Add) button next to the Security Group(s) in the Available Security Groups list to add to Added Security groups list.
6. Save Changes

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

3.5 Load Balancers

*Infrastructure -> Load Balancers*

### 3.5.1 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.

**Note:** HaProxy VM Load Balancer option, Load Balancer detail pages, Balance Mode, Sticky Mode and Shared VIP address option are available in Morpheus 2.11.3+.

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.
3.5.2 Load Balancers

The Load Balancers tab lists 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, choose 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

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.

3.5.3 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
- AVI
- Amazon ELB
- Amazon ALB
- Azure Load Balancer
- Fortinet
- Openstack Octavia
- HA Proxy

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 AVI), 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 Provisioning -> 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.

3.6 F5 Load Balancers

3.6.1 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

F5 Details Page

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 post 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 as *: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.

3.7 Storage

Note: In v3.5.2 STORAGE PROVIDERS has been split out into BUCKETS and FILE SHARES sections.

3.7.1 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/. Its 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.

3.7.2 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
- Openstack Swift
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 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 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 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.

### 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 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 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 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.

### 3.7.3 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 if 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 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 if 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 font 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. 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 font 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 if 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.

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 if 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.

### 3.7.4 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
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

### 3.7.5 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
   - 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.

### 3.8 Key Pairs & Certificates

#### 3.8.1 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.

To navigate to the Key Pairs section:

1. Select the Infrastructure link in the navigation bar.
2. Select the Key Pairs link in the sub navigation bar.

#### 3.8.2 Add Key Pair

To Add Key Pair:

1. Select the Infrastructure link in the navigation bar.
2. Select the Key Pairs link in the sub navigation bar.
3. Click the Add Key Pair button.
4. From the Add Key Pair Wizard input the following:
   - Name
   - Public Key
   - Private Key

---

**Note:** Certain features do not require storage of the private key.

#### 3.8.3 Delete Key Pair

To Delete Key Pair:

1. Select the Infrastructure link in the navigation bar.
2. Select the Key Pairs link in the sub navigation bar.
3. Click the Delete icon on the row of the Key Pair to delete.

### 3.9 PXE Boot

#### 3.9.1 Overview

Morpheus includes a built in PXE Server to enable easy and rapid bare metal provisioning.
3.9.2 Prerequisites

- DHCP server with following config added to dhcpd.conf

```bash
allow booting;
allow bootp;
option option-128 code 128 = string;
option option-129 code 129 = text;
next-server morpheus-appliance-ip;
filename "pxelinux.0";
```

Note: Replace `morpheus-appliance-ip` in the dhcpd.conf file with your Morpheus appliance IP address.

- Internal Appliance URL (PXE) set in Administration - Settings. 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.

- Mac or IP addresses of PXE target mapped in {morpheus} Infrastructure -> Boot - Mapping

- Target host configured for Network boot in BIOS

Note: On the Morpheus Appliance, PXE is enabled by default and port 69 is forwarded to the Internal PXE port 6969. These settings are configurable in the pxe: section of `/opt/morpheus/conf/application.yml`.

3.9.3 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.

3.9.4 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 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.

3.9.5 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*. 
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.

### 4.1 Tenants

#### 4.1.1 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 besides the Master Tenants are Sub Tenants.

- The Master Tenant is the default Tenant created during the installation of Morpheus.
- All Tenants created after installation are Sub Tenants. Only one Master Tenant can exist.
- The Master Tenant creates and controls all Sub Tenants.

- **Tenants are isolated environments.**
  - Tenants have unique users
  - Tenants have unique workloads
  - Tenants have unique Groups

- The Master Tenant can share or assign Master Tenants resources with/to Sub Tenants
- Sub Tenants cannot share their resources with other tenants
- Sub Tenants cannot see resources from other Sub Tenants
- Sub Tenants can only access Master Tenant resources that have been set to Public visibility or assigned to the Sub Tenant.
Roles

It is important to understand Role types and permission when creating and managing Tenants.

Tenant Roles

Tenant Roles are for capping Sub Tenant permissions by setting the maximum permissions for a Tenant. User Roles in a Tenant cannot exceed the permissions of the Tenant Role assigned to the Tenant.

- Tenant Roles set the maximum permissions for a Tenant
- User Roles in a Tenant cannot exceed the permissions of the Tenant Role assigned to the Tenant.
- Tenants Roles can be set on one or multiple tenants

**Tenant Roles determine Public Cloud access for the tenant.**

- All Clouds in the Master Tenants the have Visibility set to Public will show as options in the Tenant Role Cloud Access.
- Only Master Tenant Clouds given access in a Tenants assigned Tenants role will be accessible in the Sub Tenant.

**Important:** Tenant Roles cap permissions on all Sub Tenant user roles. Sub Tenant user roles can be created in the sub Tenant will lesser permissions than the Tenant Role allows. Tenant Roles are designed for a Master Tenant Admin to set max permissions for a Tenant, and a Sub Tenant Admin to configure User Roles inside the Sub Tenant.

User Roles

User Roles determine Feature, Group and Instance Type access for all Users. For multi-tenancy, there are two types of User Roles.

**Single Tenant User Roles** Single Tenant User Roles only exist in the Tenant they exist in. All Roles created in a Sub Tenant are Single Tenant User Roles.

**Multi Tenant User Roles**

The Master Tenant and only the Master Tenant can create Multi Tenant User Roles. These Roles are for automatically creating base User Roles in Sub Tenants.

- Multi Tenant User Roles will automatically create matching User Roles in all Tenants

**Note:** Multi Tenant User Roles are intended to make Sub Tenant User Role creation easier, so Master Tenant users do not have to re-created the same base Sub Tenant Users Roles for every Sub Tenant. Multi Tenant User Roles are not a single role across Tenants, but more of a template that creates new Sub Tenant User Roles that can then be managed in the Sub Tenant.

- Multi Tenant User Role changes will propagate to all Sub Tenants unless edited by a Sub Tenant
- Once a Multi Tenant User Role is edited inside a Sub Tenant, it is no longer connected to the Multi Tenant User Role and is it own unique Role.
- At least one Multi Tenant User Role is required before any Sub Tenants can be created

**Important:** Deleting a Multi Tenant User Role from the Master tenant will not remove that Role from Sub Tenants.
4.1.2 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, Delete, 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)
   - Base Role Primary role of the Tenant. All User roles within the Tenant cannot exceed the permission of this Role.
   - Limits Restricts the amount of Storage and Memory allocated to the Tenant
5. Click the Save Changes button.

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

**4.1.3 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 create prior to adding sub-tenant users or the user will not save. In previous versions a default multi-tenant role was seeded, but due to customer requests the seeded role was removed and a multi-tenant role must be created by the master tenant for sub-tenant 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 the Create User button.
5. From the New User wizard input the fields below
   - First Name of the user being created
   - Last Name of the user being created
   - Username used to login
   - Email address of the new user
   - Role to be inherited by the user
   - Password
   - Limits
     - Restricts the amount of Storage and Memory the user can provision.
   - Save Changes.
Important: In versions 3.1.1 and 2.12.5 and later, a multi-tenant user role must be create prior to adding sub-tenant users or the user will not save. In previous versions a default multi-tenant role was seeded, but due to customer requests the seeded role was removed and a multi-tenant role must be created by the master tenant for sub-tenant 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 Tenant Name on the row of the Tenant containing the user to be edited.
4. Click the Edit pencil icon of the row of the to edit.
5. Edit User information

Note: Name, Username, Passwords and e-mail addresses cannot be edited on Users created from Identity Source Integrations.

6. 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 on the row of 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. Subusers 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
4.1.4 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, and the Created Date. Click the Tenant Name to drill into the Tenant View where you can again Edit, Delete, as well as Create Users, Edit Users, and Delete Users 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
   - Description (optional)
   - Base Role - Primary role of the Tenant. All roles created within the Tenant must inherit this role as the base role.
   - Currency (for pricing)
   - Limits - These restrict the amount of Storage, Memory, and CPUs that can be collectively provisioned by all users in the Tenant. The default is 0 for these limits, which means no limits will be applied.
5. Click the Save Changes button.
Viewing Tenants

To View an individual Tenant page, select the Tenant name from the main Tenants section.
From inside this Tenant view, the Tenant settings can be edited, existing users can be viewed or edited, and new users for this Tenant can be created.

**Tenant Users**

To create a new user for an Tenant (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.)*

1. Click the **CREATE USER** button. From the New User wizard input the fields below. * First Name of the user being created * Last Name of the user being created * Username used to login * Email address of the new user * Role to be inherited by the user * Password * Storage, Memory and CPU limits for this specific user, if any (0 is no limit)

2. Click **Save Changes**.

**Impersonate Tenant User**

“Impersonate User” enables administrators to login as sub-Tenant users directly from the master Tenant.

To impersonate a user, you must be logged in as a user with the “Impersonate User” feature enabled in assigned role. Navigate to Tenants, select an Tenant, and in the specific user 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 (master Tenant only): Available across all Tenants

Resources in the Master Tenant can also be assigned directly to sub-Tenants. When a resource is assigned to a sub-Tenant, it is only available for that sub-Tenant, and its visibility is automatically set to private. Public Visibility is not an option for any resource assigned to or created in a sub-Tenant.

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
Cloud Visibility & Assignment

Edit Visibility of a Cloud

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 sub Tenants. All sub-Tenants 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 sub Tenant(s).

For Tenants created prior to a Master Tenant cloud being set to public visibility, the sub Tenant will have the option to add that cloud but it will not automatically be added.

While the cloud will be available for Sub-Tenants, the resources available in that cloud to the sub-Tenant(s) depends on the visibility or assignment of the individual resources.

**Note:** A sub-Tenant user must have sufficient role permissions/cloud access to add publicly available clouds. Master Tenant clouds settings cannot be edited from sub Tenants.

### Assign a Cloud to an Tenant

**Important:** When assigning a Cloud to an 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 a 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 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 a 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 that 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.

4.2 Identity Sources

Administration -> Tenants -> Select Tenant -> Identity Sources

4.2.1 Overview

There are several built-in single sign-on integrations included with Morpheus. These can be configured via the Identity Sources button in Admin -> Accounts. These integrations include linking capabilities with LDAP, Active Directory, Okta, and Jump Cloud. One can even map these sign-on tools to equivalent roles in Morpheus so at first log in users are assigned the appropriate role.

4.2.2 Active Directory

Overview

Active Directory is Microsoft’s primary authentication service widely used in Enterprise organizations and even via Microsoft’s cloud services. While Active Directory also supports LDAP protocol support (which Morpheus can integrate with as well), the main Active Directory integration can also be utilized. It is even possible to map Active Directory groups to equivalent Roles within Morpheus.

Note: To use Active Directory, a valid / trusted SSL certificate must be in place on the Active Directory services (self
signed will not work).

Adding an Active Directory Integration

1. Navigate to Administration -> Tenants
2. Select a Tenant
3. Select IDENTITY SOURCES
4. Select + IDENTITY SOURCE
5. Choose “Active Directory”
6. Populate the following:
   - Name  Unique name for authentication type.
   - AD Server  Hostname or IP address of AD Server.
   - Domain  Domain name of AD Domain.
   - Binding Username  Service account username for bind user.
   - Binding Password  Password for bind service account.
   - Required Group  The AD group users must be in to have access (optional)
   - Default Role  The default role a user is assigned if no group is listed under AD user that maps under Role Mappings section.
   - Service Account Holder  This is the admin account type in Morpheus and an AD group can be created and populated to a user that this role should be assigned. Roles are assigned dynamically based on group membership.
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:  Only the username is required with password, not the username@domain.

Note:  Sub-tenant Morpheus API authentication for Active Directory generated users is not currently supported.

4.2.3 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 if 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.

6. 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.

4.3 Plans & Pricing

4.3.1 Overview

The Plans & Pricing page displays a list of all of your available service plans. From the service plans page you will be able to Create, Edit, and Delete service plans, as well as review basic plan details. The list of plans displayed on this page displays planName, Description, Instances Layout, Memory, Storage, and Cost, as well as an action column to edit and delete. A default set of Service Plans are created in Morpheus. 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.

4.3.2 Service Plans

Create Service Plan

To create service plan
1. Select the Administration link in the navigation bar.
2. Select the Plans & Pricing link in the sub navigation bar.
3. Click the Create Service Plan button.
4. From the New Service Plan wizard, input:
   - Name
   - Code used as a unique identifier in the API and CLI.
   - Storage size in megabytes.
   - Memory size in megabytes.
   - Cost is internal cost of plan.
   - Price is what the service offering will be priced at.
   - Instance Types that will be associated with this plan.
   - Click the Save Changes button to save.

**Edit Service Plan**

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. These all can be easily managed from the Admin -> Service Plans section.

To edit service plan:
1. Select the Administration link in the navigation bar.
2. Select the Plans & Pricing link in the sub navigation bar.
3. Click the Edit pencil icon on the row of the plan to edit.
4. Edit the following Edit Service Plan.
5. Click the Save Changes button to save.

**Delete Service Plan**

To delete service plan
1. Select the Administration link in the navigation bar.
2. Select the Plans & Pricing link in the sub navigation bar.
3. Click the Delete trashcan icon on the row of the plan to delete.
4. Confirm

**4.3.3 Pricing**

**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.

**Note:** Month is equivalent to 30 days by default. For AWS, month is 30.5 days. For Azure, month is 30.4 days.

**Types**  Price Set Types determine what prices the Set is composed of.

**Note:** Make note of the Price set options below before creating Prices.

**Everything**  ‘Everything’ price sets require 1 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.

**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.

**Price Types**

• Everything
  – One price for all resources Memory, CPU, RAM, and Disks
• Memory + CPU
• Memory Only
• Cores Only
• Disk Only
• Platform
• Software
Price Units

- Minute
- Hour
- Day
- Month
- Year
- Two Year
- Three Year
- Four Year
- Five Year

Currency

- AUD
- CHF
- DKK
- EUR
- GBP
- IDR
- ILS
- MAD
- NOK
- NZD
- ROL
- SEK
- TRL
- USD
- 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 \(\text{Cost} + (\text{Cost} \times \text{Markup} \%)\)

**Custom Price**  Sets a Price independent from the Cost. If the Cost changes, a Custom Price will not.

### 4.4 Roles

#### 4.4.1 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.

---

**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.

---

#### 4.4.2 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 tenants 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 pre-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 has 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.

Another note about user roles is that when a user role is copied down to a subtenant, the permission scopes cannot supersede the tenants 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.

4.4.3 Roles and Identity Sources

It is very common for large enterprises to have an existing identity source that they would like to plugin 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 found in Admin -> Tenants, and has to be configured on a per tenant basis.

4.4.4 Resource Limits

While it is possible to restrict usages by roles assigned to a tenant or role with max memory utilizations and max storage utilizations, it is preferred to now control this at the Policy level within a group or cloud. Morpheus provides a large swatch of policy types that can be assigned globally or to specific tenants both globally, and per cloud/group entity.

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.

---

**FEATURE ACCESS** Controls Tenant and User access level for sections and features in Morpheus.

**GROUP ACCESS** Controls User access level for Groups. (Groups are not Multi-Tenant.)

**CLOUD ACCESS** Controls Sub-Tenant access level for Master Tenant publicly visible Clouds.

**INSTANCE TYPE ACCESS** Controls Tenant and User access level for Instance Types.
Feature Access Permissions

Feature Access settings control permissions for sections and features in Morpheus. Permission options include:

None  Hidden or inaccessible for user

Read  User can access the section, but cannot edit or create

Full  User has full access

User  User only has access to data from the Instances they have created/own.

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.

- Admin: Appliance Settings (None, Full)
- Admin: Backup Settings (None, Full)
- Admin: Environment Settings (None, Full)
- Admin: Identity Source (None, Full)
- Admin: Integrations (None, Read, Full)
- Admin: License Settings (None, Full)
- Admin: Log Settings (None, Full)
- Admin: Monitoring Settings (None, Full)
- Admin: Provisioning Settings (None, Full)
- Admin: Roles (None, Read, Full)
- Admin: Service Plans (None, Read, Full)
- Admin: Tenant (None, Full)
- Admin: Tenant - Impersonate Users (None, Full)
- Admin: Users (None, Read, Full)
- Admin: Whitelabel Settings (None, Full)
- Administration: Manage Policies (None, Read, Full)
- Backups (None, View, Read, User, Full)
- Billing (None, Read, Full)
- Infrastructure: Boot (None, Read, Full)
- Infrastructure: Certificates (None, Read, Full)
- Infrastructure: Clouds (None, Read, Full)
- Infrastructure: Groups (None, Read, Full)
- Infrastructure: Hosts (None, Read, Full)
- Infrastructure: KeyPairs (None, Read, Full)
- Infrastructure: Load Balancers (None, Read, Full)
- Infrastructure: Networks (None, Read, Full)
- Infrastructure: Security Groups (None, Read, Full)
• Infrastructure: Storage (None, Read, Full)
• Logs (None, Read, User, Full)
• Monitoring (None, Read, User, Full)
• Operations: Analytics (None, Read, Full)
• Operations: Approvals (None, Read, Full)
• Operations: Dashboard (None, Read)
• Operations: Guidance (None, Read, Full)
• Operations: Reports (None, Read, Full)
• Operations: Scheduling - Power (None, Read, Full)
• Operations: Usage (None, Read, Full)
• Provisioning: (None, Read, User, Full)
• Provisioning: Allow Force Delete: (None, Full)
• Provisioning: Apps: (None, Read, User, Full)
• Provisioning: Automation Services (None, Read, Full)
• Provisioning: Deployment Services (None, Read, Full)
• Provisioning: Deployments (None, Read, Full)
• Provisioning: Library (None, Read, Full)
• Provisioning: Migrations (None, Read, Full)
• Provisioning: Tasks (None, Read, Full)
• Provisioning: Tasks - Script Engines (None, Full)
• Provisioning: Blueprints (None, Read, Full)
• Provisioning: Blueprints - ARM (None, Provision, Full)
• Provisioning: Blueprints - Terraform (None, Provision, Full)
• Provisioning: Thresholds (None, Read, Full)
• Provisioning: Virtual Images (None, Read, Full)
• Remote Console (None, Provisioned, Full)
• Remote Console: Auto Login (No, Yes)
• Services: Archives (None, Read, Full)
• Services: Cypher (None, Read, Full, Full Decrypted)
• Services: Image Builder (None, Read, Full)

4.4.5 Adding Roles

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 Mater Account.
To create a Tenant Role:

1. In the Master Account, navigate to Administration -> Roles
2. Select the + CREATE ROLE button
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. Optionally set Limits for Storage, Memory or CPU Count. These limits will apply for any Tenant the Role is assigned to. 0.0 is default and is equal to no limit.

After saving the Role will be created, and you will be redirected to that Roles Permissions settings.

User Roles

User Roles can be single or multi-tenant. Multi-tenant User Roles will automatically be copied to all current and future Tenants.

**Important:** Multi-tenant User Roles are copied to Tenants, but each copied Role becomes its own unique role per tenant and needs to be edited in the Tenant. Changes to a Multi-Tenant User Role at the Master Tenant level will not change existing user roles in Tenants created from the Multi-Tenant Role due to unique Role permissions in each Tenant, such as changes from the overriding Tenant Role and unique Group and Instance Type permissions.

Create a Single Tenant User Role

1. In the Master Account, navigate to Administration -> Roles
2. Select the + CREATE ROLE button
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. Optionally set Limits for Storage, Memory or CPU Count. These limits will apply for any User the Role is assigned to. 0.0 is default and is equal to no limit.

After saving the Role will be created, and you will be redirected to the Roles Permissions settings.

Create a Multi Tenant Role

1. In the Master Account, navigate to Administration -> Roles
2. Select the + CREATE ROLE button
3. Enter a name for the Role and optional Description
4. For TYPE, select “User Role”

5. Select the “Multi-tenant Role” checkbox.

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. Optionally set Limits for Storage, Memory or CPU Count. These limits will apply for any User the Role is assigned to. 0.0 is default and is equal to no limit.

After saving the Role will be created, and you will be redirected to that Roles Permissions settings.

---

**Important:** While a Multi-tenant role is automatically copied into all existing subtenants as well as placed into any new Tenants, the generated roles inside each Tenant should be treated and managed as their own role. The Group Access configuration of a multi-tenant role only applies to the Tenant the role is being edited in, as Groups are unique to each tenant and not shared across Tenants. The purpose of a multi-tenant role is to facilitate an easy method of generating multiple pre-defined user roles for Tenants, NOT manage Tenant User Roles from the master tenant. When editing the permissions for a sub-tenant user, be sure to edit their user role(s) from inside the sub-tenant, not from the Master account, by impersonating a sub-tenant admin with full Role permissions.

---

### 4.5 Users & User Groups

#### 4.5.1 Users

**Overview**

The Users page displays a list of all users. From the users page: Create, Edit, and Delete users. The list of users displayed on this page displays Account, Name, Username, Email, and Role.

**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.

**Create User**

Users can be created from Administration -> Users or Administration -> Tenants -> Select a 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

4.5.2 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).

4.6 Integrations

Administration -> Integrations
To add an integration select + ADD and choose your integration. For more information on each integration, please see the Integration Guides.

4.7 Policies

4.7.1 Overview
Policies add governance, ease of use, cost-savings, and auditing features to Morpheus. Morpheus enables end users to create user, group, cloud, and global policies to give users full control and governance over their environments! Policies can apply towards any instance provisioned by a specific user, globally or into a group or cloud with active policies. Policy generation is a role permission.

4.7.2 Creating Policies
Policies can be created in three different locations.

- Administration -> Policies
- Infrastructure -> Groups -> Group -> Policies
- Infrastructure -> Clouds -> Cloud -> Policies

Policies can be edited and set to active or inactive.

**Important:** Cloud policies will override matching or conflicting group policies during provisioning.

Available Policy Types

**Backup Creation**  Disable or enable the ability to create a backup when provisioning an instance.

**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.
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.

Host Name Pre-populates a fixed or editable name for Hosts and Virtual Machines using ${variable} naming patterns and/or text.

Hostname Pre-populates a fixed or editable name for hostnames/machine names using ${variable} naming patterns and/or text.

Instance Name Pre-populates a fixed or editable name for Instance Names using ${variable} naming patterns and/or text.

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.

Max Hosts Sets the max number of total Hosts in the Group or Cloud the Policy is added to.

Max Memory Sets the max number of total of RAM combined for Instances in the Group or Cloud the Policy is added to.

Max Storage Sets the max number of total of Storage combined for Instances in the Group or Cloud the Policy is added to.

Max VMs Sets the max number of Virtual Machines for the Group or Cloud the Policy is added to.

Power Scheduling Adds a Power Schedule for the Instances in a Group or Cloud. Power Schedules can be created in Operations -> Scheduling

Provision Approval Sets an Approval requirement for Provisioning into a Group or Cloud using Morpheus Approvals or an Approval Integration such a Service Now.

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.

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.

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 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 -> 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 Types sections below for Configuration options.
6. Select **SAVE CHANGES**

#### 4.7.3 Policy Types

**Expiration Policies**

Expiration policies set an expiration timeframe for any instance provisioned into the cloud or group 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 moved into a Group with an Active Policy 
or Instances created when converting an unmanaged 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 ${variable} string interpolation.

Commonly used variables for naming patterns include:

 sausage }
 $(sequence+100) #results in 101
 $(sequence.toString().padLeft(5,'0')) #results in 00001

An example Instance Name Policy using a naming pattern with User Ini-
tials, 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.

Cloud codes and Group codes are fields found in their respective configuration panes.

Note: Static text can also be used in conjunction with ${variable}’s, such as morpheus${cloudCode}${type}${sequence+3000}
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 moved into a Group with an Active Policy or Instances created when converting an unmanaged 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 Operations -> 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 Operations -> 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 and Groups for maximum amount of Memory, Storage, Cores, Hosts, VM’s, or Containers that can be created in the Cloud or Group the Policy is assigned to.

Configuration options for Max Resources 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.

**Max Hosts**  Sets the max number of total Hosts in the Group or Cloud the Policy is added to.

**Max Memory**  Sets the max number of total of RAM combined for Instances in the Group or Cloud the Policy is added to.

**Max Storage**  Sets the max number of total of Storage combined for Instances in the Group or Cloud the Policy is added to.

**Max VMs**  Sets the max number of Virtual Machines for the Group or Cloud the Policy is added to.

**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.
4.8 Provisioning Settings

Administration -> Provisioning

Settings Configure Global Provisioning, Cloud-init and PXE Boot settings.

Environments Create and manage Environment Tags

Licenses Add License to apply to Windows Instances during Provisioning.

4.8.1 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.

Show Pricing Displays or hides Pricing in Provisioning wizard and Instance and Host detail pages.

Deployment Archive Store Default Storage Provider for storing Deployment Archives.

Note: Storage Providers can be configured and managed in the Infrastructure -> Storage section.

4.8.2 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 Infrastructure -> Key Pairs section.

Windows

- Administrator Password: Enter password to be set for the Windows Administrator User during provisioning.

4.8.3 PXE Boot Settings

Default Root Password Enter the default password to be set for Root during PXE Boots.

Environments

Administration -> Provisioning -> Environments
4.8.4 Overview

The Environments section is where you create and manage Environment Tags, which are available in the Environment dropdown during Provisioning to attach to Instances. An instance’s Environment Tag can be changed by editing the instance.

4.8.5 Creating Environments

1. Select + Create Environment
2. Populate the following for the New Environment:
   - **Name**: Name of the Environment
   - **Code**: Shortcode used for API and CLI
   - **Description**: Environment description displayed in Environments list page.
   - **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 Master Tenant.

   **Note:** Existing Environments can be edited or removed using the Actions dropdown in the Environments list.

Licenses

Administration -> Provisioning -> Licenses

4.8.6 Overview

The License section is for automating the application of Licensee 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.

4.8.7 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**: License Version
   - **Copies**: The Number of copies available on the License
   - **Description**: License description displayed in the Licenses list in Morpheus. Helpful for identifying 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 *Provisioning -> 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

### 4.8.8 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.

### 4.8.9 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.

### 4.9 Monitoring Settings

#### 4.9.1 Overview

The *Administration -> Monitoring* section is for configuring Morpheus Monitoring and Monitoring Integrations.

#### 4.9.2 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.

**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.

4.9.3 AppDynamics

AppDynamics Monitoring Integration Settings

Enabled Enables the AppDynamics Integration

Controller Host This is the host name or the IP address of the AppDynamics Controller. This is the same host that you use to access the AppDynamics browser-based user interface.

Controller Port This is the HTTP(S) port of the AppDynamics Controller. This is the same port that you use to access the AppDynamics browser-based user interface. If the Controller SSL Enabled property is set to true, specify the HTTPS port of the Controller; otherwise specify the HTTP port.

Controller SSL Enabled This property specifies whether the agent should use SSL (HTTPS) to connect to the Controller. If SSL Enabled is true, set the Controller Port property to the HTTPS port of the Controller.

Tenant Name This is the account name used to authenticate with the Controller.

Access Key This is the account access key used to authenticate with the Controller.

Controller Version This is the controller version and can be obtained at the bottom of the controller login page.

Application Name This is the name of the logical business application. Note that this is not the deployment name(ear/war/jar) on the application server. (Maximum of 30 numbers or letters)

Tier Name This is the name of the logical tier. (Maximum of 30 numbers or letters)

Controller User A user that can login to the Controller ui and upload a dashboard.

Controller Password Password for the Controller User.

4.9.4 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

<table>
<thead>
<tr>
<th>Morpheus Severity</th>
<th>ServiceNow Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info</td>
<td>Low/Medium/High</td>
</tr>
<tr>
<td>Warning</td>
<td>Low/Medium/High</td>
</tr>
<tr>
<td>Critical</td>
<td>Low/Medium/High</td>
</tr>
</tbody>
</table>
4.9.5 New Relic

New Relic Integration Settings

**Enabled** Enables the New Relic Monitoring Integration

**License Key** 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.

4.10 Backup Settings

Administration -> Backups

4.10.1 Overview

The Backups Settings page allows you enable or disable Scheduled Backups, and select a Default Backup Storage Provider. Backups within Morpheus can always be run manually, however the scheduled backups toggle must be enabled to run jobs automatically. Configure the default storage provider to select the target location for all new backups. (This does not affect existing backups.)

4.10.2 Morpheus Backup Settings

Options:

**Scheduled Backups** Enable automatic scheduled backups for provisioned instances.

**Create Backups** When enabled, Morpheus will automatically configure instances for manual or scheduled backups.

**Copy Snapshots to Store** Copy VMware snapshots to selected Backup Storage Provider

**Backup Appliance** When enabled, a Backup will be created to backup the Morpheus appliance database. Select the Backup Appliance text link to edit Appliance Backup Settings and view existing Appliance Backups.

**Default Backup Provider** Enable/Disable Morpheus as the default backup provider.

**Default Backup Storage Provider** Storage Providers can be configured and managed in the Infrastructure Storage section.

**Backup Retention Count** Default maximum number of successful backups to retain.

4.10.3 Veeam Settings

**Enabled** Enable the Veeam integration

**Default Backup Provider** Sets Veeam as the Default Backup Provider in Morpheus. Backup Providers can also be configured per Backup.

**Visibility** Sets visibility in multi-tenant Morpheus environments:
• Public: Accessible by all Tenants
• Private: Accessible only to the Tenant the Veeam integration is added.

**Host** Host name or the IP address of the Veeam Backup Enterprise Manager. This is the same host that you use to access the Veeam Backup Enterprise Manager browser-based user interface.

**Port** The HTTP(S) port of the Veeam Backup Enterprise Manager API. The default is 9399.

**Username** The username used to authenticate with the Veeam Backup Enterprise Manager.

**Password** The password used to authenticate with the Veeam Backup Enterprise Manager.

**Backup Repositories** Once credentials are authenticated, search will populate available Veeam Repositories to select from.

**Backup Job Templates** The backup jobs configured in the Veeam Backup and Replication Console that can be cloned when creating new backup jobs.

**Refresh Available Jobs** Use to sync newly created Jobs in Veeam.

**Important:** Once a Veeam Integration has been enabled, a VEEAM SERVER setting will be available in Vmware and Hyper-V cloud settings (Infrastructure -> Clouds -> Edit a Cloud). To enabled backups on a Cloud, a Veeam Server must be selected in the VEEAM SERVER dropdown in the Cloud settings and saved. Failure to do so will result in blank Backup Repositories and Backup Job Templates options when configuring Veeam Backups during provisioning.

---

### 4.10.4 Commvault Settings

**Enabled** Enable the Commvault integration

**Default Backup Provider** Sets Commvault as the Default Backup Provider in Morpheus. Backup Providers can be configured per Instance backup settings.

**Host** IP or Hostname of the Commvault server.

**Port** Port configured to access the Commvault server

**Commvault server Username** Admin Username for Commvault

**Password** Password for Username provided (encrypted in Morpheus)

### 4.11 Logging Settings

#### 4.11.1 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 overall 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 sections are only from Managed VM’s and Hosts that have the Morpheus agent installed. 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.
Morpheus also has built in Integrations with 3rd Party solutions. When configured, the Morpheus agent will forward logs to the integrated platforms automatically.

Logging Settings for the build-in Logging, Syslog forwards, and 3rd Party Integrations are configurable in the Administration -> Logs section.

### 4.11.2 Morpheus Logging

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 overall Logs sections. Logs can also be forwarded using Syslog Forward rules to any external solution that supports syslogs.

### 4.11.3 Splunk

To configure Splunk create a syslog listener configuration in Splunk. Then it is simply a matter of expanding the section in Logging settings pertaining to Splunk and filling out the host and port of the appender. Once saved, all hosts managed by Morpheus will be configured to forward logs to the target Splunk listener.

### 4.11.4 LogRhythm

Configuring LogRhythm is much like configuring Splunk. Simply toggle the enabled flag in the LogRhythm section to enabled and fill in the Host, and Port information for the LogRhythm listener.

### 4.12 Appliance Settings

The Administration -> Settings section sets global configuration parameters for the Appliance, Tenant Registration, Email, Proxy and sets which Cloud types are enabled.

#### 4.12.1 Appliance Settings

**Host Level Firewall Enabled** Enables or Disables the host level firewall. This must be Enabled to use Morpheus Security Groups.

**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.

**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.

---

**Note:** The Logs section does not contain Morpheus appliance logs, which can be found in /var/log/morpheus/ and in Operations - Health.
API Allowed Origins  Specifies which origins are allowed to access the Morpheus API.

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.

Email Settings

A default installation of Morpheus uses an online service called postmarkapp. Morpheus api requests to the postmarkapp service to send notification e-mails.

To add your own SMTP server you will need to go to the Administration and Settings of your Morpheus appliance. You will then need to provide Morpheus the following information, your mail server systems administrator should provide you with the below information and the preferred encryption method.

- From Address
- SMTP Server
- SMTP Port
- SSL Enabled
- TLS Encryption
- SMTP User
- SMTP Password

We recommend that you add your Morpheus server to your SMTP white list as well as using user authentication as an additional security measure.

Once you have added your SMTP server information into Morpheus scroll down the Administration and Settings page and press the blue save button which can be found under enabled clouds.

When you have saved your SMTP server settings in the Morpheus appliance you will then need to restart the Morpheus-ui. To restart the Morpheus-ui connection to your Morpheus server via ssh and run the below command.

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

**Important:** If you do not restart the Morpheus-ui the notifications will be sent by the original notification service postmarkapp. Please note it can take up to 3 minutes for the ui to become reachable again. has a built in SMTP server for email notifications and alerts. An alternate SMTP server can be specified below:

Add an alternate SMTP Server:

- From Address
- SMTP Server
- SMTP Port
- SSL Enabled
- TLS Encryption
- SMTP User
- SMTP Password

**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

**Enabled Clouds**

Cloud types can be Enabled or Disabled in this section. When a Cloud type is disabled, it will be removed from the available options when adding new clouds in the *Infrastructure* section.

Available Cloud types:
- Morpheus
- OpenStack
- Amazon
- Metacloud
- VMware vCenter
- VMware vCloud Air
- SoftLayer
- Google Cloud
- Azure (Public)
- Azure Stack (Private)
- DigitalOcean
- VirtualBox
- VMware Fusion
- VMWare ESXi
- Nutanix
- UCS
- XenServer
4.12.2 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. Preferred Image Size (500x76)

Disable Support Menu  Enable this flag to hide the support dropdown menu in the header.

Footer Logo  Footer Logo in bottom left. Preferred Image Size (264x54)

Login Logo  Logo shown on Login screen. Preferred Image Size (228x280)

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 filed 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
Override CSS

Override CSS settings by entering CSS in **Override CSS** field.

Example: (this will add one continues background image to the Header)

```html
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 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.

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 changes to `/etc/morpheus/morpheus.rb` to take effect.

```ruby
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']['timeout_page_title'] = 'Morpheus timeout, please try again...
nginx['loading_pages']['timeout_page_h1'] = 'Timeout waiting for Morpheus to load...
  -click below to try again.'
nginx['loading_pages']['failure_page_title'] = 'Morpheus Server Error'
```

(continues on next page)
4.12.3 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.

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 *

<table>
<thead>
<tr>
<th>Used Space</th>
<th>Unused Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>182.9GB</td>
<td>841.0GB</td>
</tr>
</tbody>
</table>

**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://morpheushub.com](https://morpheushub.com), or by contacting support@ or sales@ morpheusdata.com.

4.12.4 Advanced morpheus.rb Settings

Overview

Morpheus allows for additional advanced customizations to the morpheus.rb file located in `/etc/morpheus/morpheus.rb`. Below is a list of the supported items available in the morpheus.rb file.
### 4.12. Appliance Settings

```ruby
ui['vm_images_cdn_url'] = 'https://morpheus-images.morpheusdata.com'
u['kerberos_config'] = nil
ui['kerberos_login_config'] = nil
ui['max_memory_mb'] = nil
ui['memory_map_threshold'] = 131072
ui['memory_trim_threshold'] = 131072
ui['memory_top_pad'] = 131072
ui['memory_map_max'] = 65536
ui['memory_alloc_arena_max'] = 2
ui['http_client_connect_timeout'] = 10000
ui['http_client_connect_timeout'] = 600000

mysql['enable'] = true
mysql['morpheus_db'] = 'morpheus'
mysql['morpheus_db_user'] = 'morpheus'
mysql['max_active'] = 100
mysql['host'] = '127.0.0.1'
mysql['port'] = 3306
mysql['tmp_dir'] = '/tmp/mysql'

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

rabbitmq['enable'] = true
rabbitmq['vhost'] = 'morpheus'
rabbitmq['queue_user'] = 'queue_user'
rabbitmq['host'] = '127.0.0.1'
rabbitmq['port'] = '5672'
rabbitmq['nodename'] = 'rabbit@localhost'
rabbitmq['stomp_port'] = 61613
rabbitmq['heartbeat'] = nil

elasticsearch['enable'] = true
elasticsearch['host'] = '127.0.0.1'
elasticsearch['es_hosts'] = {'127.0.0.1' => 9200}
elasticsearch['open_files'] = 204800
elasticsearch['memory_map_threshold'] = 131072
elasticsearch['memory_trim_threshold'] = 131072
elasticsearch['memory_top_pad'] = 131072
elasticsearch['memory_map_max'] = 65536
elasticsearch['memory_alloc_arena_max'] = 2
elasticsearch['replica_count'] = 1

nginx['enable'] = true
nginx['workers'] = integer calculated from number of cpus
nginx['worker_connections'] = 10240
nginx['cache_max_size'] = '5000m'
nginx['ssl_country_name'] = "US"
nginx['ssl_state_name'] = "CA"
nginx['ssl_locality_name'] = "San Mateo"
nginx['ssl_company_name'] = "Morpheus, LLC"
nginx['ssl_organizational_unit_name'] = "DevOps"
nginx['ssl_email_address'] = "personal@email.com"
nginx['ssl_ciphers'] = "ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-
SHAC256:DHE-RSA-AES256-GCM-SHA256:DHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-
SHAC256:ECDHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES128-GCM-SHA256:
SHA256:
SHA256:SHA1:HIGH:!aNULL:!eNULL:!EXPORT:!DES:!MD5!PSK!RC4"
```

4.12. Appliance Settings
nginx['ssl_protocols'] = "TLSv1 TLSv1.1 TLSv1.2"
nginx['ssl_session_cache'] = "builtin:1000 shared:SSL:10m"
nginx['ssl_session_timeout'] = "5m"
nginx['loading_pages']['max_loops'] = 60 # seconds
nginx['loading_pages']['timeout_page'] = '/timeout.html'
nginx['loading_pages']['iteration_time'] = 10000 # milliseconds
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']['timeout_page_title'] = 'Morpheus timeout, please try again...'
nginx['loading_pages']['timeout_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.'
repo['repo_host_url'] = 'https://downloads.morpheusdata.com'
CHAPTER 5

Monitoring

5.1 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.

5.1.1 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 3 health states. They are Failed, Warning (Recovering), and Healthy. 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 Failed 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 UI and look for a text Morpheus on the dashboard page.

*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>
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 1 then the check would pass else mark it as critical.

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
5.1.2 Groups & Apps

One great feature of the monitoring system is the ability to organize checks by groups and apps. This provides a nice convenient way to determine what a customer facing impact might be for a single failure as well as representing redundancy via groupings.

It is important to note the relationship of apps, groups, and even checks with regards to instances provisioned within Morpheus. For every Instance that is provisioned: A monitoring Group is created and a Check is added to that group for every Container or Virtual Machine within that Instance. This makes sense such that as an Instance is scaled out horizontally (containers/vms added to it) The monitoring system accurately represents the layers of redundancy. An App simply maps to a Provisioning App and should be pretty straightforward to understand.

Groups

It is also possible to organize custom checks in this hierarchical structure by manually adding or editing a Group or App. Groups 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.

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 checks Max Severity setting.

Affects Availability Whether or not a failed group impacts system wide availability calculations.

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.

Apps

Apps are very useful for seeing an aggregation of failures, or impact based on a set of checks and groups. Apps typically correlate to apps created in provisioning 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. There are a few useful options as well with regards to Apps:

Max Severity The maximum severity incident a failed app may create. This setting overrides check and group Max Severity settings.

Affects Availability Whether or not a failed app impacts system wide availability calculations.
5.1.3 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.

5.1.4 Alerts

There are several ways to configure alerts and notifications within Morpheus. Users can be notified via Email or SMS as well as several other direct integrations. These integrations include PagerDuty, Alert Ops, Victor Ops, and even Slack chat Channel notifications (or optionally via the ServiceNow integration).

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 https://slack.com/[[]Slack] channel.
- **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 targetting 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.
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.

Notifications

Configuring Notification Services

By default Morpheus provides email notification services using the morpheusdata.com email address. It may be advisable to customize these services to use another mail delivery service.

5.1.5 Monitoring Integrations

While Morpheus provides a fantastic means for determining uptime and availability of both services and VMs sometimes more is needed. A good example of this is performance application monitoring. To solve this several external integrations are provided out of the box. Even some external integrations with regards to incident management are provided.

AppDynamics

AppDynamics is a very powerful performance and application monitoring tool. It features advanced correlation features and profiling capabilities for a very wide range of application platforms including native Docker support. Due to the level of capabilities of AppDynamics there are more required settings to integrate it with Morpheus. To get started expand the section in Admin -> Monitoring related to AppDynamics and toggle it to Enabled. There are several fields here that need filled out. Once completed hit save and all hosts will automatically be configured to install the AppDynamics agent.

AppDynamics is capable of begin run as a paid SaaS based service as well as an on premise installation and Morpheus supports both configurations. Most input fields related to connecting to AppDynamics provide helpful tips as to what information exactly needs provided and where to acquire it.

NewRelic

New Relic is a very popular service based performance monitoring tool. It supports a wide variety of application platforms and is a breeze to configure with Morpheus. Another great feature of new relic is its ability to monitor the server applications run on and provide additional stats. To do this an agent needs to be installed and configured on each server. Fortunately, this is performed automatically for every vm and docker host provisioned within Morpheus. To turn on the integration simply go to Admin -> Monitoring and expand the section titled “New Relic”. There it is simply a matter of toggling the Enabled setting to on and entering the New Relic account API Key.

Service Now

Service now integration is provided out of the box with Morpheus. To add a service now integration simply visit the ‘Monitoring Settings’ section in Admin -> Monitoring. This allows one to map incident severity levels to equivalent severities in ServiceNow.

To enable service now simply expand the section labelled “ServiceNow” in Admin -> Monitoring. Toggle the enabled flag and enter the Host, User, and Password information required to connect to ServiceNow. The other options below include behaviors upon new incidents being opened and old incidents closing. It also includes a table for mapping Morpheus incident severity levels to their ServiceNow counterparts.
6.1 Logs

6.1.1 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 Admin -> Logs section. There are a couple useful settings here including customizing the retainment policy (by default 7 days). This could be expanded to years for PCI Compliance purposes or other potential 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.

This area of administration 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 http://www.rsyslog.com/sending-messages-to-a-remote-syslog-server/[]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. Simple 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: [https://lucene.apache.org/core/2_9_4/queryparsersyntax.html](https://lucene.apache.org/core/2_9_4/queryparsersyntax.html)

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.

## 6.1.2 Integrations

While the built in logging solution provided by Morpheus is sufficient for most, there are some scenarios in which a more advanced logging system may be desired or already in place. To facilitate this Morpheus makes it easy to add custom syslog rules as well as built in direct integrations with Splunk and LogRhythm. All integrations pertaining to logging can be configured in the **Administration -> Logging** section.

### Splunk

To configure Splunk simply create a syslog listener configuration in Splunk. Then it is simply a matter of expanding the section in Logging settings pertaining to Splunk and filling out the host and port of the appender. Once saved, all hosts managed by Morpheus will be configured to forward logs to the target Splunk listener.

### LogRhythm

Configuring LogRhythm is much like configuring Splunk. Simply toggle the enabled flag in the LogRhythm section to enabled and fill in the Host, and Port information for the LogRhythm listener.

## 6.1.3 Exporting Logs

### Log Settings

There are three main log areas in Morpheus

- Agent Logs
- Morpheus Server Logs
- Activity / Audit Logs
### 6.1.4 Agent Logs

When instances are deployed through Morpheus, the agent that is installed, captures Application logs and sends them back to the Morpheus Server.

While the built-in logging solution provided by Morpheus is sufficient for most, there are some scenarios in which a more advanced logging system may be desired or already in place. To facilitate this Morpheus makes it easy to add custom syslog rules as well as built in direct integrations with Splunk and LogRhythm. All integrations pertaining to logging can be configured in the Administration -> Logging section.

### Splunk

To configure Splunk simply create a syslog listener configuration in Splunk. Then it is simply a matter of expanding the section in Logging settings pertaining to Splunk and filling out the host and port of the appender. Once saved, all hosts managed by Morpheus will be configured to forward logs to the target Splunk listener.

### LogRhythm

Configuring LogRhythm is much like configuring Splunk. Simply toggle the enabled flag in the LogRhythm section to enabled and fill in the Host, and Port information for the LogRhythm listener.

### 6.1.5 Morpheus Server Logs

The main Morpheus server 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:

```bash
module(load="imfile" PollingInterval="50")
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")
input(type="imfile" File="/var/log/morpheus/guacd/current" Tag="guacd" ReadMode="2" Severity="info")
input(type="imfile" File="/var/log/morpheus/elasticsearch/current" Tag="elasticsearch" ReadMode="2")
input(type="imfile" File="/var/log/morpheus/mysql/current" Tag="mysql" ReadMode="2" Severity="info")
input(type="imfile" File="/var/log/morpheus/nginx/current" Tag="nginx" ReadMode="2" Severity="info")
input(type="imfile" File="/var/log/morpheus/rabbitmq/current" Tag="rabbitmq" ReadMode="2" Severity="info")
input(type="imfile" File="/var/log/morpheus/redis/current" Tag="redis" ReadMode="2" Severity="info")
```

2. Restart rsyslog

The logfiles will now be 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.
6.1.6 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.groovy located at /opt/morpheus/conf/logback.groovy.

2. Copy the below configuration to the bottom of the logback.groovy configuration file, save and then exit.

```groovy
appender("AUDIT", RollingFileAppender) {
    file = "/var/log/morpheus/morpheus-ui/audit.log"
    rollingPolicy(TimeBasedRollingPolicy) {
        fileNamePattern = "/var/log/morpheus/morpheus-ui/audit_%d{yyyy-MM-dd}.%i.log"
        timeBasedFileNamingAndTriggeringPolicy(SizeAndTimeBasedFNATP) {
            maxFileSize = "50MB"
        }
        maxHistory = 30
    }
    encoder(PatternLayoutEncoder) {
        pattern = "[%d] [%thread] %-5level %logger{15} - %maskedMsg %n"
    }
}
logger("com.morpheus.AuditLogService", INFO, ['AUDIT'], false)
```

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

   morpheus-ctl stop morpheus-ui

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

4. 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 moprheus-ui

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

   morpheus-ctl tail moprheus-ui

Once you see the ASCI 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 found in the default Morpheus log path which is /var/log/morpheus/morpheus-ui/

Instead of writing the output to a logile, you could create an Appender definition for your SIEM audit database product

6.1.7 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`
    endscript
}
```
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.

### 7.1 Initial Backups Setup

Global Backup settings, Storage Providers and Execution Schedules should be configured prior to creating backups.

#### 7.1.1 Global Backups Settings

Morpheus Backups can be enabled under Administration -> 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 Operations -> Scheduling -> Execution Schedules

- **Backup Retention Count**  Default maximum number of successful backups to retain.
7.1.2 Backup Schedules

Backup Execution Schedules can be configured and managed in the Operations -> Scheduling -> Execution Schedules. The Default Backup Schedule set in Administration -> Backups will be selected when configuring Backups but other Schedules can be selected during configuration.

7.2 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
- SqlServer
- 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

7.3 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.

7.4 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.
7.4.1 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.

7.4.2 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.
10. Click Complete to save.

7.4.3 Create Server Backup

To create a server backup:

1. Select the Backups link in the navigation bar.
2. Select the Backups link in the sub navigation bar.
3. Click Add Backup.
4. From the Create Backup Wizard select the radio button Server, then click Next.
5. Input the following:
   
   - Name of the backup job being created
   - Server
   - Type of backup you wish to create.
6. Click Next. Different options are presented based upon the type of backup being created.
   
   - File/Directory - input path for the backup.

7. Click Next.

8. Schedule the backup Days, Time, Storage Provider & Retention Count.

9. Click Complete to save.

### 7.5 Managing Backups

#### 7.5.1 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.

#### 7.5.2 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.
10. Click Complete to save.

7.5.3 Create Server Backup

To create a server backup:

1. Select the Backups link in the navigation bar.
2. Select the Backups link in the sub navigation bar.
3. Click Add Backup.
4. From the Create Backup Wizard select the radio button Server, then click Next.
5. Input the following:
   - Name of the backup job being created
   - Server
   - Type of backup you wish to create.
     - File
     - Directory
     - Mongo
     - MySQL
     - Postgres
6. Click Next. Different options are presented based upon the type of backup being created.
   - File/Directory - input path for the backup.

7. Click Next.

8. Schedule the backup Days, Time, Storage Provider & Retention Count.

9. Click Complete to save.
8.1 Dashboard

The Dashboard is a single pane of glass showing quick, easy to read performance and configuration information about the Morpheus Environment.

**Status**  There are four gauges across the top of the dashboard page showing quick system stats for Instances, Monitoring Status, Log Errors, and Backups. Each gauge also serves as a quick link for each section.

**My Instances**  The My Instances section shows quick information about 5 favorite instances like Type, IP and Port. Click View All to be taken directly to the instances page.

**Monitoring**  The Monitoring section displays an overall health, availability statistics, as well as response time and any open incidents requiring action.

**Recent Activity**  Recent Activity is displayed on the right side of the dashboard page. Items like instance provisioning and deletion, backups, and alerts are displayed here.

**Logs**  All Morpheus logs are application aware. Log information from hypervisors, servers, and applications are pushed up into the Morpheus controller node and made searchable and actionable. Choose a timeframe from the Logs pane to view statistics or click List to view all log information.

**Backups**  The backup pane at the bottom of the page shows statistics about Morpheus backups. Information about success and failure rates and the number of backups run versus scheduled is available here. Click on the List button to be taken directly to the backups page where you can view and configure backups.

8.2 Reports

8.2.1 Overview

Customizable Cloud, App and Instance usage reports can be generate in the Operations -> Reports section.
8.2.2 Report Types

INVENTORY REPORTS

Account
- Tenant Inventory Summary

Infrastructure
- Cloud Inventory Summary
- Container Host Inventory Summary
- Group Inventory Summary
- Hypervisor Inventory Summary
- Tenant Resource Allocation

Provisioning
- Instance Inventory Summary
- Virtual Machine Inventory Summary

CLOUD REPORTS

Usage
- Cloud Usage
- Cloud Usage App Summary
- Cloud Usage Instance Type Summary
- Tenant Usage

COST REPORTS

Cost
- Application Cost
- Cloud Cost
- Group Cost
- Instance Cost
- Tenant Cost

8.2.3 Reports History

The landing page of the reports section gives a selection of report types to generate and a history view of generated reports.

8.2.4 Generating Reports

To generate a new Report
1. Select a Report type
2. Set Start and End dates for the report
3. Select Cloud(s)
4. Select other options depending on report type
5. Select run and the report will be generated

8.2.5 Viewing Reports

To view a report, select the Name of the report after the status changes to ready. Generated Reports are available from the Reports History page or in their respective Section.

Note: All reports are saved and accessible until deleted.

8.3 Analytics

8.3.1 Overview

The Morpheus Analytics engine analyzes resource utilization and costs across clouds. This functionality helps users make decisions on where instances and workloads should be provisioned.
8.3.2 Analytics Report Types

Cost

By Application

- Month to Date

By Cloud

- Month to Date

By Group

- Month to Date

By Instance

- Month to Date

By Tenant
• Month to Date

Instances Types

By Cloud
• Count
• Total Memory
• Total Storage
• Total CPUs/Cores

By Group
• Count
• Total Memory
• Total Storage
• Total CPUs/Cores

Instances

By Cloud
• Count
• Total Memory
• Total Storage
• Total CPUs/Cores

By Group
• Count
• Total Memory
• Total Storage
• Total CPUs/Cores

Utilization

CPU vs Price
• Hosts
• Virtual Machines
• Bare Metal

Overall vs Price
• Hosts
• Virtual Machines
• Bare Metal

RAM vs Price
• Hosts
• Virtual Machines
• Bare Metal
8.4 Guidance

8.4.1 Overview

The Operations -> Guidance section show recommendations for Resource and Costs Utilization optimization. By analyzing the CPU, RAM, and Storage activity of Instances and Hosts, Morpheus can recommend actions for Sizing and Power State.

Configuration

Guidance is configured per Cloud and is set to off by default.

To turn on Guidance for a Cloud:

1. Navigate to Infrastructure -> Clouds.
2. Select the Edit icon of the Cloud to configure Guidance for.
3. Expand the Advanced Options section in the Edit Cloud modal.
5. Select Save Changes.

Guidance recommendations will begin to appear in the guidance section when generated.

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.
8.5 Budgets

Budgets provide insight into spending across entire accounts, allowing users to create and plan a budget scoped to their account, clouds, tenants, users, or groups.

8.5.1 Creating A Budget

1. Navigate to Operations > Budget
2. Create a new budget and enter in the following:
   1. Name
   2. Description
   3. Scope: Here you can choose what this budget is tied to
   4. Period
   5. Year: Set future budgets
   6. Interval: Choose Month, Quarter, Year then fill in the budget for that interval
3. SAVE CHANGES
**CREATE BUDGET**

<table>
<thead>
<tr>
<th>NAME</th>
<th>Demo Budget</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESCRIPTION</td>
<td></td>
</tr>
<tr>
<td>Enabled</td>
<td></td>
</tr>
<tr>
<td>SCOPE</td>
<td>Account</td>
</tr>
<tr>
<td>PERIOD</td>
<td>Year</td>
</tr>
<tr>
<td>YEAR</td>
<td>2019</td>
</tr>
<tr>
<td>INTERVAL</td>
<td>Month</td>
</tr>
<tr>
<td>JANUARY</td>
<td>2500</td>
</tr>
<tr>
<td>FEBRUARY</td>
<td>2500</td>
</tr>
<tr>
<td>MARCH</td>
<td>2500</td>
</tr>
<tr>
<td>APRIL</td>
<td>2500</td>
</tr>
<tr>
<td>MAY</td>
<td>2500</td>
</tr>
<tr>
<td>JUNE</td>
<td>2500</td>
</tr>
<tr>
<td>JULY</td>
<td>2500</td>
</tr>
<tr>
<td>AUGUST</td>
<td>2500</td>
</tr>
<tr>
<td>SEPTEMBER</td>
<td>2500</td>
</tr>
<tr>
<td>OCTOBER</td>
<td>2500</td>
</tr>
<tr>
<td>NOVEMBER</td>
<td>2500</td>
</tr>
<tr>
<td>DECEMBER</td>
<td>2500</td>
</tr>
</tbody>
</table>
8.5.2 Cloud Budgets

If you scope a budget to a cloud visit the cloud summary page in Infrastructure > Clouds > Select Cloud > Summary for a detailed breakdown of the costing.

<table>
<thead>
<tr>
<th>SUMMARY</th>
<th>HOSTS</th>
<th>VMS</th>
<th>CONTAINERS</th>
<th>LOAD BALANCERS</th>
<th>NETWORKS</th>
<th>RESOURCES</th>
<th>POLICIES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Appliance URL: default</td>
<td>Time Zone: default</td>
<td>Scale Priority: 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Domain: twx.bertmartins.com</td>
<td>Datacenter ID:</td>
<td>Agent Install Mode: cloudinit</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Container Mode: docker</td>
<td>Storage Mode: lvm</td>
<td>Guidance: manual</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Network Mode: AWS</td>
<td>Security Mode: off</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**AMAZON RESOURCES**

- Instances: 24
- Load Balancers: 2
- Running Instances: 19
- Subnets: 7
- Volumes: 31
- Security Groups: 61

**GUIDANCE**

- Sizing: Memory
- Decrease
- Savings: $20.20
- 06/03/2018 11:05 AM

**COSTING**

- BUDGET: $2,500
- MONTH TO DATE: $7,277
- ESTIMATED SPEND: $11,702
- LAST MONTH: $12,952
- CHANGE FROM LAST MONTH: ($1,250)

<table>
<thead>
<tr>
<th>AWS SPEND</th>
<th>JANUARY</th>
<th>FEBRUARY</th>
<th>MARCH</th>
<th>APRIL</th>
<th>MAY</th>
<th>JUNE</th>
<th>JULY</th>
<th>AUGUST</th>
<th>SEPTEMBER</th>
<th>OCTOBER</th>
<th>NOVEMBER</th>
<th>DECEMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Budget</td>
<td>$2,500.00</td>
<td>$2,500.00</td>
<td>$2,500.00</td>
<td>$2,500.00</td>
<td>$2,500.00</td>
<td>$2,500.00</td>
<td>$2,500.00</td>
<td>$2,500.00</td>
<td>$2,500.00</td>
<td>$2,500.00</td>
<td>$2,500.00</td>
<td></td>
</tr>
<tr>
<td>Actual</td>
<td>$618</td>
<td>$1,829</td>
<td>$1,282</td>
<td>$1,702</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SERVICES BREAKDOWN**

- EC2: $5,400.34
- EFS: $152.01
- OTHER: $759.81
- S3: $303.90
- SUPPORT: $574.45
- TOTAL: $7,276.51

8.5.3 View Budget Summary

To view the budget summary, click into the budget to see the actual vs budgeted spend for the interval selected.
To edit the budget just select **EDIT**
In Operations > Analytics > Budget Analysis select scope (Account, Tenant, Cloud, Group, User) to view the budget analysis.
8.6 Scheduling

8.6.1 Overview

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.

8.6.2 Power Schedules

Create Power Schedule

1. Navigate to Operations -> 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 day’s Schedule. Green sections indicate Power on, red sections indicate Power Off. Time indicated applies to selected Time Zone.
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 -> Hosts -> 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 to a Cloud**

**Note:** Power Schedule Policies apply to Instances created in the Cloud 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**
5. Select **TYPE Power Schedule**
6. Configure the Power Schedule 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 Operations -> Scheduling

**Permissions- TENANTS** Leave blank to apply to all Tenants, or search for and select Tenants to enforce the Policy on specific Tenants.

7. Select **SAVE CHANGES**

### Add Power Schedule to a Group

**Note:** Power Schedule Policies apply to Instances created or moved into the Group 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**
5. Select **TYPE** *Power Schedule*
6. Configure the Power Schedule 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 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 Operations -> Scheduling

   **Permissions- TENANTS** Leave blank to apply to all Tenants, or search for and select Tenants to enforce the Policy on specific Tenants.

### 8.6.3 Execution Schedules

#### Create Execution Schedule

- **NAME**
- **DESCRIPTION**
- **TIME ZONE**
- **Enabled**
- **SCHEDULE**
  - 0 0 *** ***
  - Every day at 00:00
8.7 Approvals

Morpheus and Service Now Approvals

8.7.1 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.

8.7.2 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.

### 8.7.3 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* 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:
• Request: What is being requested
• Type: The type of the approval request
• Request For: Name and link to Instance
• Status
• Date Created
• Requested By
• Actions dropdown * For Internal Approval Requests
  – Approve
  – Deny
  – Cancel
  – For ServiceNow Approval Requests * Cancel

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 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 ServiceNow, 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.

8.8 Usage

8.8.1 Overview

The Operations -> Usage section shows Billing information for Instances and Hosts that have pricing configured on their Service Plan.

Important: Pricing must be enabled in Administration -> Provisioning and Service Plans configured with Price sets in Administration -> Plans & Pricing for Pricing to show in the Usage section.

8.8.2 View Usage

All Instances are listed by default, with the most recent usage information showing first.

Usage details can be filtered by Cloud and Date:

Cloud  Default view is for all Clouds. Select a Cloud to show Instance and Host Usage for only one Cloud.

Date  Default view shows most current Usage. Select the Date filter to scope to a different date range.

8.8.3 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.

8.9 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.

8.9.1 Activity Reports

There are 5 types of activities that are displayed in the Activity Reports:

- Provisioning
- Monitoring
To View a Recent Activity report:

1. Select the Reports link in the navigation bar.
2. Click the tab Recent Activity.

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.

To Filter:

1. Click the filter drop down of type of filter you want to apply.
2. Select the appropriate filter.

8.10 Health

8.10.1 Morpheus Health

The Morpheus health section provides an overview of the health of your Morpheus appliance. It includes data on the following:

- Health Levels
- CPU
- Memory
- Database
- Elastic
- Queues

HEALTH LEVELS include

- Morpheus CPU
- System CPU
- Morpheus Memory
- System Memory
- Used Swap

CPU include

- Processor Count
• Process Time
• Morpheus CPU
• System CPU
• System Load

MEMORY includes
• 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 includes
• 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 includes
• 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 include
• Node
• Master
• Location
• Heap Usage
• Memory Usage
• CPU Usage
• 1M Load
• 5M Load
• 15M Load

**Elastic Indices include**

• Health
• Index
• Status
• Primary
• Replicas
• Doc
• Count
• Primary
• Size
• Total Size

**QUEUES INCLUDE**

• Queue Count
• Busy Queues
• Error Queues

### 8.10.2 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.

### 8.10.3 Morpheus Logs

The Logs displayed in `Operations - 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 `Operations - Health - Morpheus Logs` are filtered for Morpheus services. Complete stack traces can be found in `/var/log/morpheus/morpheus-ui/current`. 
9.1 Cypher

9.1.1 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.

9.1.2 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)

• Key lease times are entered in milliseconds and default to 32 days (2764800000 ms).
  – Quick MS Time Reference:
    – Day: 86400000
    – Week: 604800000
9.1.3 Creating Cypher Keys

1. Navigate to Services - Cypher and select “+ ADD KEY”
2. Configure one of the following types of Keys:

9.1.4 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 filed blank for a password, as it will be generated.

**Lease** Enter lease time in milliseconds (ex. 604800000 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.

9.1.5 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 milliseconds (ex. 604800000 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.

9.1.6 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 milliseconds (ex. 604800000 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.

### 9.1.7 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 milliseconds (ex. 604800000 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.

### 9.1.8 Using Cypher Keys in Scripts

To use a cypher Key in a script, use the following syntax:

```ruby
<%=cypher.read('var_name')%>
```

**Example:** `PASSWORD=<%=cypher.read('secret/myuserpassword')%>`

**Note:** You can reference the original owner of a workflow so that keys can be used in a sub-tenant. 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.

### 9.2 Archives

#### 9.2.1 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. Each bucket has a unique name that is used to identify it in URLs and Scripts.
9.2.2 Storage Provider

Archive buckets are assigned a Storage Provider (Object Store). This is where the bucket will write its files. A Storage Provider can be configured to use the local appliance file system (Local), an Amazon S3 bucket, etc.

Every archive bucket generates and uses a random File Path to store its files under. This ensures two different archive buckets will not contend for the same backend storage location.

9.2.3 Permissions

Visibility

Visibility determines whether your files are secure or not.

**Private**  This secures your files. Only authorized users of the Owner and Tenants account may view the bucket and download its files. This is the default.

**Public**  This makes your files available to the public. Anyone, including anonymous users/scripts can download these files without any authentication.

**Warning:** Be careful not to store sensitive files in a Public archive.

Users of the Owner account may fully manage the files in a bucket.

Tenants

Users of the Owner account may fully manage the files in a bucket. Users of the Tenant account(s) will have read-only access. They may browse and download files in the bucket.

Both Owner and Tenants must have the Services: Archives permission to access a Private bucket. READ level access allows browsing and downloading files in the bucket.

FULL access allows full management of the bucket and its files. This includes modifying files and links, bucket settings and deleting it.

9.2.4 Files

To add a file to a bucket, click on the bucket name, and then click the + ADD FILE button. Once added, click on the file name to access the links, history and script section for the file.

9.2.5 Links

You can create a Link to download a Private file without any authentication. Links may be configured to expire after a period of time.

9.2.6 Scripts

Morpheus automatically generates syntax for creating a link to a file in your Scripts. When the Script is generated, it will create a temporary link to download the file and return the URL of that link. This link is made available to the public. It is accessible to any user or script that can reach the appliance. Downloading the file only requires knowing
the URL, which includes a secret token parameter. You can specify the number of seconds before the link expires. The default value is 1200 (20 minutes).
Integration Guides

10.1 Automation

10.1.1 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 the 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
**Add Ansible Integration**

1. Navigate to *Provisioning -> Automation -> 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.
   - **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 ran 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 Provisioning -> 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 Option type that provides a text box to enter a username. The fieldName of the option type is username. Below is the playbook.

---
- 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:
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.

**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 it’s 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.

**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 but certain os’s or network conditions can prevent automated install. You can run `ansible --version` in the Morpheus Appliance, or in the Ansible integration details page (Administration -> Integrations -> Select Ansible Integration, or 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:

```bash
sudo apt-get install software-properties-common
sudo apt-add-repository ppa:ansible/ansible
sudo apt-get update
sudo apt-get install ansible
```

CentOS:

```bash
sudo yum install epel-release
sudo yum install ansible
```

Then create the working Ansible directory for Morpheus:

```bash
sudo mkdir /opt/morpheus/.ansible
sudo chown morpheus-local.morpheus-local /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 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.

• 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.

10.1.2 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: It 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**: Enabled by default it is enabled. To disable the integration, uncheck this option and save.
- **Ansible Tower URL**: This would be 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 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 will appear and are mandatory.

- **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.

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 see under Instance history.

10.1.3 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 Actions->Run Workflow. 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.

**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

10.1.4 Puppet

Overview

Add Puppet Integration

1. Navigate to Administration -> Integrations and select + New Integration
2. Select Integration type “Puppet”
3. Populate the following fields
   • Name: Name of the Puppet Integration in Morpheus
   • Puppet Master: Hostname
   • Allow Immediate Execution: Yes or No
4. Save Changes

10.1.5 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
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
Scope Salt Integration to Group Or Cloud

Configuration Management integrations like Saltstack apply to the Infrastructure Group abstraction in Morpheus. To tie 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

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

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
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.
If you did not apply a state at provision time now you will be able to run State commands through Morpheus.

In our example the Apache State from a git backend was applied successfully to our newly created vm.
10.1.6 Terraform

Requirements

Role Access

- In order to see the Terraform Blueprint type option and create Terraform App Blueprints in Provisioning -> 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.

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 added to Terraform App Blueprints.

Supported App Provisioning Targets

- VMware
- AWS
- Oracle Cloud
**Note:** Additional clouds will be available in later releases.

## Terraform Installation

Morpheus will automatically install Terraform locally upon the first Terraform App provision. It is possible on some operating system configurations for the automated terraform installation to fail, in which case it can be manually installed (run `terraform --version` to verify).

To manually install and configure terraform on the Morpheus Appliance:

1. Run the following curl on the Morpheus Appliance to install Terraform:

   ```bash
   curl -k -s "https://applianceServerUrl/api/server-script/terraform-install?local=true" | bash
   ``

   **Note:** Replace `applianceServerUrl` with your Morpheus appliance url or ip.

2. Create a working directory for Terraform, and change owner to `morpheus-app`.

   ```bash
   sudo mkdir /var/opt/morpheus/morpheus-ui/terraform
   sudo chown morpheus-app.morpheus-app /var/opt/morpheus/morpheus-ui/terraform
   
   The default location is `/var/opt/morpheus/morpheus-ui/terraform` but can be changed.
   ``

3. Add the Terraform working path to `/opt/morpheus/conf/application.yml`

   ```bash
   sudo vi /opt/morpheus/conf/application.yml
   
   Add the following to the application.yml config below and in-line with the repo section:
   
   ```yaml
   terraform:
       location: '/var/opt/morpheus/morpheus-ui/terraform'
   ```

   Example application.yml config with Terraform location added:

   ```yaml
   repo:
       git:
           location: '/var/opt/morpheus/morpheus-ui/repo/git'
       local:
           location: '/var/opt/morpheus/morpheus-ui/repo/local'
   terraform:
       location: '/var/opt/morpheus/morpheus-ui/terraform'
   bitcan:
       backup:
           destination:
               root: '/var/opt/morpheus/bitcan/backup'
               working: '/var/opt/morpheus/bitcan/working'
   ``

   **Important:** Uses spaces not tabs to indent or ui startup will fail. If you used a different path than the default location, enter that path instead.
4. Restart the morpheus-ui to apply the application.yml config.

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

Terraform is now installed and configured, and Terraform apps can be provisioned from Morpheus.

**Creating Terraform App Blueprints**

In order to provision Terraform apps, Terraform App Blueprints must be created first.

<table>
<thead>
<tr>
<th>Important:</th>
<th>In Morpheus versions 3.3.0 and 3.3.1 VMware cloud types are supported for Terraform App provisioning targets. Additional clouds will be available in later releases.</th>
</tr>
</thead>
</table>

1. Navigate to Provisioning -> 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** Name of the
   - **DESCRIPTION** Description for you App Blueprints shown in the Apps list (optional)
   - **CATEGORY** App Category (optional)
   - **IMAGE** Add reference image/picture for your App Blueprint (optional)
   - **CONFIG TYPE** (select Terraform, Terraform.json, or Git Repository)
     - **Terraform (.tf)**
       - **CONFIG** Paste in the .tf contents in the config section. 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.
     - **Terraform JSON (.tf.json)**
       - Paste in .tf.json contents in the config section. 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.
     - **Git Repository**
       - **SCM Integration** Select a Github SCM integration that has been added in Administration - Integrations. If using a Git Repository integration from Administration - Integrations this filed can be skipped.
       - **Repository** Select repository from selected SCM integration, or Git Repository integration from Administration - Integrations if no SCM/Github Integration is selected.
       - **BRANCH OR TAG** i.e. master (default)
       - **WORKING PATH** Enter the repo path for the .tf files (s). . / is default.
Morpheus Documentation, Release 3.6.2

**CONFIG** .tf files found in the working path will populate in the CONFIG section.

**Note:** If no files are found please ensure your Github or Git integration is configured properly (Private repos need to have a key pair added to Morpheus, the keypair selected on the integration in Morpheus, and the keypair’s public key added to the GitHub users SSH keys in github or to the git repo).

**TFVAR SECRET** Select a tfvars secret for .tf variables. Tfvars secrets can be added in Services -> Cypher using the tfvars/name mountpoint. This allows sensitive data and passwords to be encrypted and securely used with Terraform Blueprints.

**OPTIONS** example -var 'instanceName=sampleTfApp'

6. Select SAVE

Your Terraform App is ready to be provisioned from Provisioning -> Apps.

**Provisioning Terraform Apps**

**Note:** An existing Terraform App Blueprints must be added to Provisioning -> Blueprints before it can be provisioned.

**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 and 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. Populate any required variables in the Terraform Variables section. ..TIP:: If the tf CONFIG data needs to be edited, select the RAW section, edit, and then select the BUILDER section again. The CONFIG changes from the RAW edit will be updated in the CONFIG section.
8. Select COMPLETE

The Terraform App will begin to provision.

Once provisioning is completed, note the TERRAFORM tab in the App details page (Provisioning -> Apps -> select the App). This section contains State and Plan output:

10.1. Automation
10.2 Backups

10.2.1 Commvault

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
   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

10.2.2 Veeam

Adding Veeam Integration

1. Navigate to Backups -> Services
2. Select + ADD
3. Select Veeam
4. Fill in the following:
   
   Name  Name of the Integration in Morpheus
   Enabled  Enable the Veeam integration
   Host  IP or Hostname of the Veeam server.
   Port  Port number configured to access the Veeam server
   Username  Admin Username for Veeam
   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

Important: Once a Veeam Integration has been enabled, a VEEAM SERVER setting will be available in VMWare and Hyper-V cloud settings (Infrastructure -> Clouds -> Edit a Cloud). To enable backups on a Cloud, a Veeam Server must be selected in the VEEAM SERVER dropdown in the Cloud settings and saved. Failure
to do so will result in blank Backup Repositories and Backup Job Templates options when configuring Veeam Backups during provisioning”.

## 10.2.3 Rubrik

### 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.
   - **Username** Admin Username for Rubrik
   - **Password** Password for Username provided (encrypted in Morpheus).
   - **Visibility**
     - **Private** Only Available to the Tenant the Integration is added by
     - **Public** Available to Sub-Tenants (master tenant option only)
5. SAVE

## 10.2.4 Zerto

### Adding Zerto Integration

1. Navigate to Backups -> Services
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 or Zerto Replication Manager
   - **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**

### 10.2.5 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**

### 10.3 Clouds

#### 10.3.1 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

- Virtual Machine Provisioning
- Containers
- Backups / Snapshots
- Resources Groups
- Migrations
- Auto Scaling
- Load Balancing
- AWS Marketplace Search and Provisioning
- Remote Console
- Periodic Synchronization
- Lifecycle Management and Resize
- Restore from Snapshots
- EC2
- RDS
- S3
- ELBs
- ALBs
- Route53
- IAM Profile sync and assignment
- Network Sync
- Security Group Sync (selectable when provisioning, will not appear in Security Groups section)
- Pricing Sync
- Assign Elastic IP’s
- Network Pools
- MetaData Tag creation

Morpheus can provide a single pane of glass and self-service portal for managing instances scattered across both AWS and private cloud offerings like VMWare and Hyper-V.

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 Typical Outbound to Morpheus Appliance: 80, 443
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.

Note: Each AWS Cloud in Morpheus is scoped to an AWS Region and VPC Multiple AWS Clouds can be added and even Grouped. Verify Security groups are properly configured in all Regions Morpheus will scope to.

Adding an AWS Cloud

1. Navigate to Infrastructure -> Clouds
2. Select + Create Cloud
3. Select AWS
4. Enter 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.
   - **Region** Select AWS Region for the Cloud
   - **Access Key** Access Key ID from AWS IAM User Security Credentials.
   - **Secret Key** Secret Access Key associate with the Access Key ID.
5. **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.

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.

Advanced Options

DOMAIN Specify a default domain for instances provisioned to this Cloud.
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.

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 Managed

SECURITY MODE  Defines if Morpheus will control local firewall of provisioned servers and hosts.

| Important: When local firewall management is enabled, Morpheus will automatically set an IP table rule to allow incoming connections on tcp port 22 from the Morpheus Appliance. |

STORAGE MODE  Single Disk, LVM or Clustered

GUIDANCE  Enable Guidance recommendations on cloud resources.

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.

AGENT INSTALL MODE

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud-Init (when available): 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.

API PROXY  Required when a Proxy Server blocks communication between the Morpheus Appliance and the Cloud. Proxies can be added in the Infrastructure -> Networks -> Proxies tab.

Provisioning Options

PROXY  Required when a Proxy Server blocks communication between an Instance and 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 Instance Agent communication to the Appliance URL.

USER DATA (LINUX)  Add cloud-init user data or scripts. Assumes bash syntax.

Note: All fields and options can be edited after the Cloud is created.
Minimum AWS IAM Policies

Below are the AWS IAM Policies for EC2, RDS, and S3 covering the minimum access for Morpheus applying to all resources.


**EC2**

```json
{
"Version": "2012-10-17",
"Statement": [
"Effect": "Allow",
"Action": [
"ec2:AllocateAddress",
"ec2:AssociateAddress",
"ec2:AssignPrivateIpAddresses",
"ec2:AttachVolume",
"ec2:AuthorizeSecurityGroupEgress",
"ec2:AuthorizeSecurityGroupIngress",
"ec2:CancelExportTask",
"ec2:CancelImportTask",
"ec2:CopyImage",
"ec2:CreateImage",
"ec2:CopySnapshot",
"ec2:CreateInstanceExportTask",
"ec2:CreateKeyPair",
"ec2:CreateNetworkAcl",
"ec2:CreateNetworkAclEntry",
"ec2:CreateNetworkInterface",
"ec2:CreateSecurityGroup",
"ec2:CreateSnapshot",
"ec2:CreateTags",
"ec2:CreateVolume",
"ec2:DeleteKeyPair",
"ec2:DeleteNetworkAcl",
"ec2:DeleteNetworkAclEntry",
"ec2:DeleteNetworkInterface",
"ec2:DeleteSecurityGroup",
"ec2:DeleteSnapshot",
"ec2:DeleteTags",
"ec2:DeleteVolume",
"ec2:DeregisterImage",
"ec2:DescribeAccountAttributes",
"ec2:DescribeAddresses",
"ec2:DescribeAvailabilityZones",
"ec2:DescribeClassicLinkInstances",
"ec2:DescribeConversionTasks",
"ec2:DescribeExportTasks",
"ec2:DescribeImageAttribute",
"ec2:DescribeImages",
"ec2:DescribeImportImageTasks",
"ec2:DescribeImportSnapshotTasks",
"ec2:DescribeInstances",
"ec2:DescribeInstanceStatus",
"ec2:DescribeInstances",
"ec2:DescribeInstanceStatus"
]
}
```

(continues on next page)
"ec2:DescribeKeyPairs",
"ec2:DescribeNetworkAcls",
"ec2:DescribeNetworkInterfaceAttribute",
"ec2:DescribeNetworkInterfaces",
"ec2:DescribeRegions",
"ec2:DescribeSecurityGroupReferences",
"ec2:DescribeSecurityGroups",
"ec2:DescribeSnapshotAttribute",
"ec2:DescribeSnapshots",
"ec2:DescribeStaleSecurityGroups",
"ec2:DescribeSubnets",
"ec2:DescribeTags",
"ec2:DescribeVolumeAttribute",
"ec2:DescribeVolumes",
"ec2:DescribeVolumeStatus",
"ec2:DescribeVpcAttribute",
"ec2:DescribeVpcClassicLink",
"ec2:DescribeVpcClassicLinkDnsSupport",
"ec2:DescribeVpcEndpoints",
"ec2:DescribeVpcEndpointServices",
"ec2:DescribeVpcs",
"ec2:DetachNetworkInterface",
"ec2:DetachVolume",
"ec2:DisassociateAddress",
"ec2:ImportImage",
"ec2:ImportInstance",
"ec2:ImportKeyPair",
"ec2:ImportSnapshot",
"ec2:ImportVolume",
"ec2:ModifyImageAttribute",
"ec2:ModifyInstanceAttribute",
"ec2:ModifyNetworkInterfaceAttribute",
"ec2:ModifySnapshotAttribute",
"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"
],
"Resource": "+"
}
RDS:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "rds:AddRoleToDBCluster",
        "rds:AddTagsToResource",
        "rds:ApplyPendingMaintenanceAction",
        "rds:AuthorizeDBSecurityGroupIngress",
        "rds:CopyDBParameterGroup",
        "rds:CopyDBClusterSnapshot",
        "rds:CopyDBSnapshot",
        "rds:CreateDBCluster",
        "rds:CreateDBClusterSnapshot",
        "rds:CreateDBInstance",
        "rds:CreateDBInstanceReadReplica",
        "rds:CreateDBSecurityGroup",
        "rds:CreateDBSnapshot",
        "rds:DeleteDBCluster",
        "rds:DeleteDBInstance",
        "rds:DeleteDBSecurityGroup",
        "rds:DeleteDBSnapshot",
        "rds:DescribeAccountAttributes",
        "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",
      ]
    }
  ]
}
```

(continues on next page)
"rds:RemoveTagsFromResource",
"rds:RestoreDBClusterFromSnapshot",
"rds:RestoreDBClusterToPointInTime",
"rds:RestoreDBInstanceFromDBSnapshot",
"rds:RestoreDBInstanceToPointInTime",
"rds:RevokeDBSecurityGroupIngress"
],
"Resource": "*"
}
]

S3

{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [
"s3:AbortMultipartUpload",
"s3:DeleteObject",
"s3:DeleteObjectVersion",
"s3:GetBucketLocation",
"s3:GetObject",
"s3:GetObjectVersion",
"s3:ListBucket",
"s3:ListBucketMultipartUploads",
"s3:ListBucketVersions",
"s3:ListMultiPartUploadParts",
"s3:PutObject"
],
"Resource": [
"arn:aws:s3:::bucketname",
"arn:aws:s3:::bucketname/*"
]
}
]

Route 53

{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [
"route53:GetHostedZone",
"route53:ListResourceRecordSets"
],
"Resource": "*"
}
]
Resource Filter

If you need to limit actions based on filters you have to pull out the action and put it in a resource based policy since not all the actions support resource filters.


Resource filter example:

```
{
    "Effect": "Allow",
    "Action": [
        "ec2:StopInstances",
        "ec2:StartInstances"
    ],
    "Resource": *
},
{
    "Effect": "Allow",
    "Action": "ec2:TerminateInstances",
    "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
    "Condition": {
        "StringEquals": {
            "ec2:ResourceTag/purpose": "test"
        }
    }
}
```

Amazon Cost and Reservation Sync

If you are enabling costing or costing and reservations sync on an amazon cloud then you will need to enable the following policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ce:*"]
        }
    ]
}
```
10.3.2 Azure

Overview

Azure is Microsoft’s public cloud offering. Offering a full range of services and features across the globe in various datacenters. It is the equivalent of AWS for Microsoft running primarily on the Hyper-V based hypervisor. While it is a great public cloud offering, it can be somewhat difficult to get integrated with which is what this guide aims to cover.

Features

- Virtual Machine Provisioning
- Azure SQL Database
- Backups / Snapshots
- Resource Group Sync & Selection
- Network Sync & Selection
- Security Group Sync & Selection
- Storage Account Sync & Selection
- Marketplace Search and Provisioning
- Azure Marketplace Custom Library Item Support
- 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 on premise solutions like Azure-Stack and Morpheus can provide a single pane of glass and self service portal for managing instances scattered across both public Azure and private Azure Stack offerings.

Note: Morpheus even supports integrating with CSP based accounts in Azure (typically used by managed service providers).
Requirements

- **Azure Active Directory Application & Credentials**
  - Client ID (old portal) / Application ID (new portal)
  - Client Secret (old portal) / Key Value (new portal)
  - Tenant ID (old Portal) / Directory ID (new portal)
  - Azure Subscription ID

- Above Active Directory App added as owner of this Azure Subscription

- Existing Azure Resources
  - Network Security Group(s) * Typical Inbound ports open from Morpheus Appliance: 22, 5985, 3389
    * Typical Outbound to Morpheus Appliance: 80, 443
      - 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.
  - Virtual Network(s)
    * Public IP assignment required for instances if Morpheus Appliance is not able to communicate with Azure instances private ip’s.
  - Resource Group(s)
  - Storage Account(s)

**Note:** Morpheus v2.10.3 added support for multiple Resource Groups and Storage Accounts per cloud, making our Azure integration more capable and easier to configure. Prior versions of Morpheus supported one resource group and one storage account per cloud, with the security group and network selection limited to the scoped Resource Group. If you are on an earlier version of Morpheus, please note you will need to add an Azure cloud integration for each Resource Group and Storage Account you would like to use.

**Azure Active Directory Credentials**

If you do not already have the Azure Active Directory credentials required to add an Azure cloud to Morpheus, use the steps below to obtain them.

**Important:** Microsoft recently added support for Active Directory application configuration in the new Azure portal. Previously, users had to use the old portal to get the required credentials to integrate Azure with Morpheus. The instructions below are updated for the new portal. Microsoft also changed the naming conventions of the credentials:

<table>
<thead>
<tr>
<th>Old Azure Portal Name</th>
<th>New Azure Portal Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tenant ID</td>
<td>Directory ID</td>
</tr>
<tr>
<td>Client ID</td>
<td>Application ID</td>
</tr>
</tbody>
</table>
Creating an Azure Active Directory Application

If you do not have an existing Azure Active Directory application for Morpheus, you will need to create a new one by:

1. Log into the Azure portal
2. Select “Azure Active Directory”
3. Select “App Registrations”
4. Select “New Application Registration”

5. Next, give your new AD 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 Azure Active Directory Application will be created.

Now that we have (or already had) our AD app, we will gather the credentials required for the Morpheus Azure integration.

**Tenant ID/Directory ID**

While still in the Active Directory Section:
1. Select Properties
2. Copy the Directory ID
3. Store/Paste for use as the Tenant ID when Adding your Azure cloud in Morpheus

**Client ID/Application ID**

1. Select App Registrations
2. Select your Active Directory Application
3. Copy the Application ID
4. Store/Paste for use as the Client ID when Adding your Azure cloud in Morpheus
Client Secret/Key Value

While still in your Active Directory Application:

1. Select Keys in the Settings pane
2. Enter a name for the key
3. Select a duration
4. Select save
5. Copy the Key Value
6. Store/Paste for use as the Client Secret when Adding your Azure cloud in Morpheus

**Important:** Copy the key value. You won’t be able to retrieve after you leave this blade.
You now have the 3 Active directory credentials required for Morpheus Azure cloud integration.

**Subscription ID**

The last credential required for the Morpheus Azure cloud integration is the Azure Subscription ID

1. Select Resource Groups
2. Select a Resource Group (instruction below if you do not have an existing resource group)
3. Copy the Subscription ID
4. Store/Paste for use as the Subscription ID when Adding your Azure cloud in Morpheus
Make Azure Active Directory Application owner of Subscription

The Active Directory Application used needs to be an owner of the subscription used for the Azure Morpheus cloud integration.

1. In the Subscription pane, select “Access Control (IAM)”
2. Click “+ Add”, in the pane to the right, select “1 Select a role” and then select “Owner”
3. Select “2. Add Users” and in the search box begin to type the name of the AD Application created earlier.

**Note:** the AD Application will not display by default and must be searched for.

4. Select the Application, then click “Select” at the bottom of the Add Users pane, and the select “OK” at the bottom of the Add Access pane.

**Important:** Be sure to select “OK” at the bottom of the Add Access pane or the user addition will not save.
You now have the required Credentials to add an Azure cloud integration into Morpheus.

Important: You will also need to have existing Network Security Group(s), Virtual Networks(s) and Storage Accounts(s). Instructions for creating these can be found later in this article.

Add Azure cloud in Morpheus

Azure is now ready to be added into Morpheus. Ensure you have the noted Subscription ID, Tenant ID, Client ID, and Client Secret accessible.

1. In Infrastructure - Clouds, select + CREATE CLOUD and select Azure from the cloud widget.

   OR

2. In Infrastructure, Groups- you can select the Clouds tab of a Group and click + ADD next to Azure in the Public Cloud section

3. Enter the following:
   - Name
   - Location (optional)
   - Domain (if not localdomain)
   - Scale Priority
   - Subscription ID (from step 18)
   - Tenant ID (from step 16)
• Client ID (from step 13)
• Client Secret (from step 13)

If everything is entered correctly, the Location dropdown will populate.

4. Select the Location/Region to scope the cloud to (additional Clouds can be added for multiple regions)
5. Select All or specify a Resource Group to scope this cloud to
6. Optionally select “Inventory Existing Instances” (This will inventory your existing vm’s in Azure and list them in Morpheus as unmanaged instances.)
7. Click + Save Changes

Your Azure Cloud will be created.
Creating Resources in Azure

If you do not have existing Network Security Groups, Virtual Networks, or Storage Accounts, you can create them by following the steps below:

Create a Network Security Group

1. In the main Azure toolbar, select the right arrow at the bottom of the toolbar (if collapsed) and search for and select Network Security Groups.
2. Click “+ Add” at the top of the Network security groups pane
3. Enter a unique name for the security group, select the correct subscription, and either select the resource group being used, or create a new one as shown below. Also verify the Location is the same, and then click “Create” at the bottom of the pane.

4. Configure inbound and outbound rules for the security group. Ports 80 (http), 443 (https) 22 (ssh) and 5985 (winrm) need to be open to and from the Morpheus appliance.

Create a Virtual Network

1. In the main Azure toolbar, select the right arrow at the bottom of the toolbar (if collapsed) and search for and select Virtual Networks.
2. Click “+ Add” at the top of the Virtual Networks pane
3. Enter a unique name for the virtual network, the correct subscription, select “Use existing” and select the same resource group as the Network Security Group. Also verify the Location is the same, and then click “Create” at the bottom of the pane.

Create a Storage Account

1. In the main Azure toolbar, select the right arrow at the bottom of the toolbar (if collapsed) and search for and select Storage Accounts.
2. Click “+ Add” at the top of the Storage accounts pane
3. Enter a unique name for the storage account, select “Locally-redundant storage (LRS) for Replication, select the correct subscription, select “Use existing” and select the same resource group as the Network Security Group and Virtual Network. Also verify the Location is the same, and finally click “Create” at the bottom of the pane.

Docker

So far this document has covered how to add the Azure 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 Azure (multiple are needed when dealing with horizontal scaling scenarios).

To provision a Docker Host simply navigate to the Cloud detail page or Infrastructure?Hosts section. From there click the + Container Host button to add a Azure Docker Host. This host will show up in the 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 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.

**Multi-tenancy**

A very common scenario for Managed Service Providers is the need to provide access to Azure resources on a customer by customer basis. With Azure several administrative features have been added to ensure customer resources are properly scoped and isolated. For Azure it is possible to assign specific Networks, and Resource Groups to customer accounts or even set the public visibility of certain resources, therefore allowing all sub accounts access to the resource.

**Azure Scale Sets**

Auto-scaling Azure instances can be done with the native Morpheus scaling service or Azure Scale Sets. When using Azure Scale Sets, Morpheus will configure the scale sets and thresholds, but Azure will be responsible for scaling the instances. The Instances nodes that are added and removed by Azure will be synced in by Morpheus as the instance scales up and down.

**Note:** Instances can only be added to Azure Scale Sets at provision time.

**Adding an Instance to a Scale Set**

1. In **Provisioning - Instances** select **+ ADD**
2. Select an Instance Type that has scaling enabled (Advanced section when editing an Instance Type in **Provisioning - Library**)
3. Configure the Instance as desired
4. In the **AUTOMATION** section under Scale - Scale Type select **Azure Scale Set**
5. Select a default Threshold. Threshold pre-sets can be added in **Provisioning - Automation- Scale Thresholds** (requires Instances - Thresholds permission)
6. Complete the instance configuration and provision the instance.
A Virtual Machine scale set will be created in Azure with the selected threshold and min/max node settings.

**Create Threshold Presets**

1. **In** Provisioning - Automation **select the** SCALE THRESHOLDS **tab**

   **Note:** Access to the SCALE THRESHOLDS section requires Instances - Thresholds Role permissions.
2. Select + ADD

3. Configure Threshold settings.
4. Select **SAVE CHANGES**

The new Threshold will be available for selection in the SCALE section during provisioning or when configuring an App Blueprint.

**Edit Thresholds on an Instance**

1. In **Provisioning - Instances** select the target Instance.
2. Select the **SCALE** tab below the VM’s section
3. In the **THRESHOLDS** section of the SCALE tab, click **EDIT**
4. Update the threshold settings.
5. Select **APPLY**

**Note:** Morpheus will sync in changes to a scale sets threshold settings if the settings are edited in Azure.
10.3.3 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:
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:
   - **NAME** Internal name for the Cloud in Morpheus
   - **LOCATION** (Optional) Can be used to specify the location of the Cloud or add a description.
   - **VISIBILITY** Determines Tenant visibility for the Cloud.
     - Private: Access to the Cloud is limited to the assigned Tenant (Master Tenant by default)
     - Public: Access to the Cloud can be configured for Tenants in their Tenant Role permissions.
   - **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 your 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  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.

HYPER-CONVERGED ENABLED  Not applicable for Azure Stack

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.

AGENT INSTALL MODE

• SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
• Cloud-Init (when available): 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.

API PROXY  Required when a Proxy Server blocks communication between the Morpheus Appliance and the Cloud. Proxies can be added in the Infrastructure -> Networks -> Proxies tab.

Provisioning Options

API PROXY  Required when a Proxy Server blocks communication between an Instance and 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 Instance Agent communication to the Appliance URL.

**USER DATA (LINUX)**  Add cloud-init user data using bash syntax.

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 exiting 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 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.

---

### 10.3.4 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:
   - **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.

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.

6. Select NEXT .. include:: /integration_guides/Clouds/advanced_options.rst

7. Select NEXT

8. 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.

9. Select NEXT

10. 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.

   DOMAIN  Specify a default domain for instances provisioned to this Cloud.

   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.
**APPLIANCE URL** Alternate Appliance url for scenarios when the default Appliance URL (configured in \( \text{admin -> settings} \)) 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 Managed

**SECURITY MODE** Defines if Morpheus will control local firewall of provisioned servers and hosts.

**Important:** When local firewall management is enabled, Morpheus will automatically set an IP table rule to allow incoming connections on tcp port 22 from the Morpheus Appliance.

**STORAGE MODE** Single Disk, LVM or Clustered

**GUIDANCE** Enable Guidance recommendations on cloud resources.

**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.

**AGENT INSTALL MODE**

- **SSH / WINRM:** Morpheus will use SSH or WINRM for Agent install.
- **Cloud-Init (when available):** 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.

**API PROXY** Required when a Proxy Server blocks communication between the Morpheus Appliance and the Cloud. Proxies can be added in the \( \text{Infrastructure -> Networks -> Proxies} \) tab.

**PROXY** Required when a Proxy Server blocks communication between an Instance and the Morpheus Appliance. Proxies can be added in the \( \text{Infrastructure -> Networks -> Proxies} \) tab.

**Bypass Proxy for Appliance URL** Enable to bypass proxy settings (if added) for Instance Agent communication to the Appliance URL.

**USER DATA (LINUX)** Add cloud-init user data or scripts. Assumes bash syntax.

6. Select \( \text{NEXT} \)
7. Review and then Select \( \text{COMPLETE} \)

**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 \( \text{gui label: '+ Add Resource'} \).
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, let's 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:

```yaml
---
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 \texttt{-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 \texttt{-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 \texttt{-path} parameter defines, where relative to the manifest.yml file, your Cloud Foundry application can be found.

• The \texttt{-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 \texttt{Infrastructure \to Clouds} and select your Cloud Foundry Cloud
2. Select the \texttt{MARKETPLACE} tab
3. Select \texttt{+ 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 \texttt{SAVE CHANGES}

A Node Type and layout will be created in the \texttt{Provisioning \to Library} section and the layout will be automatically added to the Instance Type selected when adding the Marketplace Item.

Provisioning Instances

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 \texttt{Provisioning \to Instances} and select an Instance Type with a Cloud Foundry layout (MySQL, Redis and RabbitMQ plus Marketplace additions)
2. Select \texttt{NEXT}
3. Select a Group and PCF Cloud
4. Add an Instance Name
5. Optionally select and Environment Tag and/or add a custom Tag
6. Select \texttt{NEXT}
7. Select Version and Instance Configuration for a Cloud Foundry layout, ex: \textit{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 \texttt{NEXT}
12. Optionally configure Automation options
13. Select \texttt{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.

**Important:** Add Deployments in *Provisioning -> 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 and 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 -> Deployments*

**Important:** If services are specified in a git repo manifest, Morpheus assumes they are 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.

### 10.3.5 Digital Ocean

**Add a Digital Ocean Cloud**

DigitalOcean Cloud Integration Detail fields:

- **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.

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 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 Managed

SECURITY MODE Defines if Morpheus will control local firewall of provisioned servers and hosts.

| Important:  | When local firewall management is enabled, Morpheus will automatically set an IP table rule to allow incoming connections on tcp port 22 from the Morpheus Appliance. |

STORAGE MODE Single Disk, LVM or Clustered

GUIDANCE Enable Guidance recommendations on cloud resources.

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.

AGENT INSTALL MODE
- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud-Init (when available): 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.

API PROXY Required when a Proxy Server blocks communication between the Morpheus Appliance and the Cloud. Proxies can be added in the Infrastructure -> Networks -> Proxies tab.

Provisioning Options

PROXY Required when a Proxy Server blocks communication between an Instance and 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 Instance Agent communication to the Appliance URL.

**USER DATA (LINUX)**  Add cloud-init user data or scripts. Assumes bash syntax.

### 10.3.6 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 Name
   - 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**  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 Managed

**SECURITY MODE**  Defines if Morpheus will control local firewall of provisioned servers and hosts.

**Important:** When local firewall management is enabled, Morpheus will automatically set an IP table rule to allow incoming connections on tcp port 22 from the Morpheus Appliance.
STORAGE MODE  Single Disk, LVM or Clustered

GUIDANCE  Enable Guidance recommendations on cloud resources.

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.

AGENT INSTALL MODE

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud-Init (when available): 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.

API PROXY  Required when a Proxy Server blocks communication between the Morpheus Appliance and the Cloud. Proxies can be added in the Infrastructure -> Networks -> Proxies tab.

Provisioning Options

PROXY  Required when a Proxy Server blocks communication between an Instance and 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 Instance Agent communication to the Appliance URL.

USER DATA (LINUX)  Add cloud-init user data or scripts. Assumes bash syntax.

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.

10.3.7 Google

Requirements

- IAM Service Account with Owner or Compute Admin Role permissions
- project_id, private_key and client_email for the Service Account
- Compute Engine API enabled in GCP API’s and Services

Features

- Provisioning Virtual Machines
- Network tagging
- Private and Local Images
Add a Google Cloud

Tip: All of the required Google Cloud credentials can be found in the .json file created when generating a key for a Google Cloud service account.

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:
   - **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.
   - **Project ID**: Google Cloud Project ID
   - **Private Key**: Service Account Private key, beginning with ‘—BEGIN PRIVATE KEY—’ and ending with ‘—END PRIVATE KEY—’
   - **Client Email**: Service Account Client Email. Ex: morpheus@morpheus.iam.gserviceaccount.com
   - **Region**: Regions will auto-populate upon successful authentication with the above credentials. Select appropriate region for this Cloud.
   - **Inventory Existing Instances**: If enabled, existing Google Instances will be inventoried and appear as unmanaged Virtual Machines in Morpheus.

Note: Morpheus scopes clouds to single regions. Multiple clouds can be added for multi-region support, and then optionally added to the same group.

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** 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 Managed

**SECURITY MODE**  Defines if Morpheus will control local firewall of provisioned servers and hosts.

**Important:** When local firewall management is enabled, Morpheus will automatically set an IP table rule to allow incoming connections on tcp port 22 from the Morpheus Appliance.

**STORAGE MODE**  Single Disk, LVM or Clustered

**GUIDANCE**  Enable Guidance recommendations on cloud resources.

**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.

**AGENT INSTALL MODE**

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud-Init (when available): 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.

**API PROXY**  Required when a Proxy Server blocks communication between the Morpheus Appliance and the Cloud. Proxies can be added in the *Infrastructure -> Networks -> Proxies* tab.

**Provisioning Options**

**PROXY**  Required when a Proxy Server blocks communication between an Instance and 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 Instance Agent communication to the Appliance URL.

**USER DATA (LINUX)**  Add cloud-init user data or scripts. Assumes bash syntax.

Finally, add Google Cloud to an existing Group or create a new Group, and you have now integrated Morpheus with Google Cloud!

**Important:** If you experience difficulties adding a GCP Cloud, ensure you have met all the Requirements above, and have logged into Google Cloud and navigated to the Compute Engine sections as it will not be initialized until navigated to upon Google Cloud account creation.

### 10.3.8 Huawei Cloud

**Adding Huawei Cloud**

**NAME**  Internal name for the Cloud in Morpheus

**CODE**  Short code used for api and variables (Optional)

**LOCATION**  Can be used to specify the location of the Cloud or add a description. (Optional)
VISIBILITY

Determines Tenant visibility for the Cloud.

- Private: Access to the Cloud is limited to the assigned Tenant (Master Tenant by default)
- Public: Access to the Cloud can be configured for Tenants in their Tenant Role permissions.

TENANT  Assigned Tenant when VISIBILITY is set to Private.

Enabled  When unchecked, the cloud will not sync and is not accessible for provisioning actions.

IDENTITY API URL  v2.0 or 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  Target project

USERNAME  Service Username

PASSWORD  Service user password

OS VERSION  Select Openstack Version.

IMAGE FORMAT  Select QCOW2, RAW or VMDK Image Type

LB TYPE

Select LB Type for Huawei 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.

10.3.9 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 and 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
- Containers
- Backups / Snapshots
- Resources Groups
- Migrations
- Auto Scaling
- Load Balancing
- Remote Console
- Periodic Synchronization
- Veeam Integration
Morpheus Documentation, Release 3.6.2

• Lifecycle Management and Resize
• Unique Kerberos Authentication

Morpheus can provide a single pane of glass and self-service portal for managing instances scattered across both Hyper-V and public cloud offerings like Azure.

Getting Started

To get started this 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. This document covers Hyper-V 2008 and Hyper-V 2012.

A user account that is part of the local administrators group on the Hyper-V host is also required.

Understand 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 below Microsoft link:


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:

10.3. Clouds
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
```

(continues on next page)
After creation of the krb5.conf a keytab file is also required. See below on 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 communicated 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.
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 all can be easily managed from the Admin | Service Plans & Pricing section.
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 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 Hyper-V (multiple are needed when dealing with horizontal scaling scenarios).

To provision a Docker Host simply navigate to the Cloud detail page or Infrastructure | Hosts section. From there click the + Container Host button to add a Hyper-V 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.
### 10.3.10 KVM

**Adding VLANs to Morpheus KVM Hosts (CentOS)**

**Overview**

Morpheus KVM is a powerful, cheaper alternative to virtualization when it comes to other hypervisor offerings. It is also very capable of setting up complex shared storage and multiple networks across many hosts. Currently this process is a manual process but will become automated in the coming months. This guide will go over how to configure VLANs on a Morpheus KVM Host.

**Getting Started**

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).

```bash
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`:

```bash
DEVICE=br211
ONBOOT=yes
DEVICETYPE=ovs
```

(continues on next page)
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**:

```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.

### 10.3.11 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 must be scoped to a specific data center and cluster. If the drop downs do not populate, please verify the api url and provided credentials have access to Vcenter.

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 > 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 Provisioning > 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. By default these virtual machines are considered ‘unmanaged’ and do not appear in the Provisioning -> Instances area but rather Infrastructure -> Hosts -> 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 Provisioning -> 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.

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 Cloud detail page or Infrastructure->Hosts section. From there click the + Container Host button 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 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.

**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 Use VNC. 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.

10.3.12 Nutanix

Overview

Nutanix simplifies datacenter infrastructure by integrating server and storage resources allowing applications to run at scale. Morpheus provides and 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 not currently supported
Getting Started

To get started 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.

API URL 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.

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 Hosts section. From there click the + Container Host button 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.

10.3.13 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 ranging from HP’s Helion Cloud to Cisco’s Metapod / Metacloud offering. Morpheus supports integration with all the various platform offerings and ranges in support all the way back to Openstack Icehouse. It leverages the APIs and provides full functionality as a self service portal in front of Openstack.

Features

- Virtual Machine Provisioning
- Backups / 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 / configuration

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

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

Adding an Openstack cloud to Morpheus is one of the simpler cloud integrations to get started with. First go to the Infrastructure -> Clouds section and click add cloud. From here there are several options including Metapod, Helion, and general Openstack. Any of these options will actually work and for the most part the branded Openstack options are represented to make it clearer to the user as to the capabilities of Morpheus.

NAME Internal name for the Cloud in Morpheus

CODE Short code used for api and variables (Optional)

LOCATION Can be used to specify the location of the Cloud or add a description. (Optional)

VISIBILITY

- Determines Tenant visibility for the Cloud.
  - Private: Access to the Cloud is limited to the assigned Tenant (Master Tenant by default)
• Public: Access to the Cloud can be configured for Tenants in their Tenant Role permissions.

**TENANT** Assigned Tenant when VISIBILITY is set to Private.

**Enabled** When unchecked, the cloud will not sync and is not accessible for provisioning actions.

**IDENTITY API URL** v2.0 or 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** Target project

**USERNAME** Service Username

**PASSWORD** Service user password

**OS VERSION** Select Openstack Version.

**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)

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 scopable to domains and projects as needed. The remaining options help Morpheus determine what 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 2 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 -> Hosts -> 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.

### 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. They include OS Volume Type (Local or Volume). This basically 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 instances dashboard itself. It should be noted that this will attempt to acquire a floating IP from the project and if out of capacity it 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 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 Openstack (multiple are needed when dealing with horizontal scaling scenarios).

To provision a Docker Host simply navigate to the Cloud detail page or Infrastructure->Hosts section. From there click the + Container Host button to add an Openstack Docker Host. This host will show up in the 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 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.

### 10.3.14 Oracle VM

**Add an Oracle VM Cloud**

- **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.
- **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.

10.3.15 Oracle Cloud

Add Oracle Public Cloud

Important: A Keypair (both public and private keys) must be added to Morpheus with the Public Key in ssh-rsa format 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

NAME  Internal name for the Cloud in Morpheus

CODE  Short code used for api and variables (Optional)

LOCATION  Can be used to specify the location of the Cloud or add a description. (Optional)

VISIBILITY

Determines Tenant visibility for the Cloud.

• Private: Access to the Cloud is limited to the assigned Tenant (Master Tenant by default)

• Public: Access to the Cloud can be configured for Tenants in their Tenant Role permissions.

TENANT  Assigned Tenant when VISIBILITY is set to Private.

Enabled  When unchecked, the cloud will not sync and is not accessible for provisioning actions.

TENANCY OCID  OCID string from Tenancy Information section in Oracle Cloud

USER OCID  OCID String for OPC API user

SELECT KEY PAIR  Select keypair added to Morpheus matching public key added to specified OPC API user

REGION  Select Region (populates after successful auth validation)

COMPARTMENT  Select Compartment (populates after successful auth validation)

INVENTORY  Turn on for Morpheus to discover and sync existing VM’s

10.3.16 Open Telekom Cloud

Add an Open Telekom Cloud

NAME  Name of the Cloud in Morpheus

CODE
LOCATION Description field for adding notes on the cloud, such as location.

VISIBILITY Description field for adding notes on the cloud, such as location.

TENANT Enabled

Details

IDENTITY API URL

DOMAIN ID 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 openstack domain list.

PROJECT

USERNAME

PASSWORD

OS VERSION

IMAGE FORMAT

LB TYPE Inventory Existing Instances Enable Hypervisor Console

Note: Hypervisor console support for openstack currently only supports novnc. Be sure the novnc proxy is configured properly in your openstack environment.

Advanced Options

Provisioning Command

10.3.17 SCVMM

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 is 5985 in order for Morpheus to communicate to SCVMM. You will also want to make sure SCVMM has WinRM enabled.

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.

SCVMM HOST IP or url of SCVMM host

USERNAME SCVMM Username. ex: svc.scvmm

PASSWORD SCVMM User Password

CLOUD Select a Cloud from the available Clouds in SCVMM.

WORKING PATH Path for Morpheus to write to. ex: c:\Cloud
1. The Cloud can now be added to a Group or configured with additional Advanced options.

### 10.3.18 Softlayer

#### Add a Softlayer Cloud

**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.

**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.

### 10.3.19 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:
   - **Name** Name of the Cloud in Morpheus
   - **Code** Cloud Code for variables
   - **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 which Tenant to scope visibility to when Visibility is set to Private.
   - **Enabled** Unchecking will disable the scheduled cloud-sync job
   - **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

### 10.3.20 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:
   - 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.
   - 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.

**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 exiting 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 added as Virtual Machines (if Inventory is enabled)

• UpCloud Templates (My Templates) will sync and be added to `Provisioning -> 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 and 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 **Provisioning -> 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 and 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

**10.3.21 vCloud Director**

**Configuration**

**Add vCD Cloud From** Infrastructure -> Clouds

1. Navigate to Infrastructure -> Clouds
2. Select + ADD
3. Select VCLOUD DIRECTOR from the Clouds list
4. Select NEXT
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.
   - **API URL**
vCloud Director API Url  Example: https://org.vcd.company.com

USERNAME  vCD Organization Administrator User

NOTE:: User must have an Organizational Administrator Role in the selected Origination for successful provisioning

PASSWORD  vCD Organization Administrator User password

ORGANIZATION  Select Organization. Dropdown populates upon successful authorization.

VDC  Select VDC. Dropdown populates upon successful authorization.

Inventory Existing Instances  If enabled, existing Virtual Machines will be inventoried and appear as unmanaged Virtual Machines in Morpheus.

NOTE: Multiple Organizations/VDC’s can be added by creating additional Clouds in Morpheus.

DOMAIN  Specify a default domain for instances provisioned to this Cloud.

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.

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 Managed

SECURITY MODE  Defines if Morpheus will control local firewall of provisioned servers and hosts.

Important: When local firewall management is enabled, Morpheus will automatically set an IP table rule to allow incoming connections on tcp port 22 from the Morpheus Appliance.

STORAGE MODE  Single Disk, LVM or Clustered

GUIDANCE  Enable Guidance recommendations on cloud resources.

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.

AGENT INSTALL MODE

• SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.

• Cloud-Init (when available): 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.

API PROXY  Required when a Proxy Server blocks communication between the Morpheus Appliance and the Cloud. Proxies can be added in the Infrastructure -> Networks -> Proxies tab.
**PROXY** Required when a Proxy Server blocks communication between an Instance and 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 Instance Agent communication to the Appliance URL.

**USER DATA (LINUX)** Add cloud-init user data or scripts. Assumes bash syntax.

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

### Add vCD Cloud From Infrastructure -> Groups

1. Navigate to Infrastructure -> Groups

2. Select a Group

3. Select the CLOUDS tab

4. Scroll down to VCDIRECTOR 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.
   - **API URL**: vCloud Director API URL Example: https://org.vcd.company.com
   - **USERNAME**: vCD Organization Administrator User
     
       NOTE:: User must have an Organizational Administrator Role in the selected Origination for successful provisioning
   - **PASSWORD**: vCD Organization Administrator User password
   - **ORGANIZATION**: Select Organization. Dropdown populates upon successful authorization.
   - **VDC**: Select VDC. Dropdown populates upon successful authorization.
   - **Inventory Existing Instances**: If enabled, existing Virtual Machines will be inventoried and appear as unmanaged Virtual Machines in Morpheus.
     
       NOTE: Multiple Organizations/VDC’s can be added by creating additional Clouds in Morpheus.
   - **DOMAIN**: Specify a default domain for instances provisioned to this Cloud.
   - **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.
   - **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.

DATA CENTER ID  Used for differentiating pricing among multiple datacenters. Leave blank unless prices are properly configured.

NETWORK MODE  Unmanaged or Managed

SECURITY MODE  Defines if Morpheus will control local firewall of provisioned servers and hosts.

**Important:** When local firewall management is enabled, Morpheus will automatically set an IP table rule to allow incoming connections on tcp port 22 from the Morpheus Appliance.

STORAGE MODE  Single Disk, LVM or Clustered

GUIDANCE  Enable Guidance recommendations on cloud resources.

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.

AGENT INSTALL MODE

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud-Init (when available): 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.

API PROXY  Required when a Proxy Server blocks communication between the Morpheus Appliance and the Cloud. Proxies can be added in the Infrastructure -> Networks -> Proxies tab.

PROXY  Required when a Proxy Server blocks communication between an Instance and 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 Instance Agent communication to the Appliance URL.

USER DATA (LINUX)  Add cloud-init user data or scripts. Assumes bash syntax.

6. Select NEXT

7. Review and then Select COMPLETE

How to create vCloud Director templates for Morpheus

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.

   ```powershell
   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

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 `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 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)

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. [https://support.morpheusdata.com/hc/en-us/articles/115002881228-Creating-a-CentOS-7-Morpheus-VMware-Image](https://support.morpheusdata.com/hc/en-us/articles/115002881228-Creating-a-CentOS-7-Morpheus-VMware-Image)

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 Provisioning > 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.

### 10.3.22 Virtualbox

**Add a VirtualBox Cloud**

1. Navigate to Infrastructure -> Clouds
2. Select + CREATE CLOUD, select Virtual Box, and then click Next.
3. Enter the following into the Create Cloud modal:
   - **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.
   - **VIRTUALBOX HOST** IP or URL of the VirtualBox Host
   - **WORKING PATH** Path Morpheus will write to. ex: ~/virtualbox
   - **USERNAME** Host Username
   - **PASSWORD** Host Password
   - **BRIDGE NAME** Will auto-populate upon successful authentication with the VirtualBox Host (E.X. ‘EN0: ETHERNET’)
   - **VBOXMANAGE EXECUTABLE** Defaults to /usr/local/bin/vboxmanage if left blank
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** 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 Managed

**SECURITY MODE**  Defines if Morpheus will control local firewall of provisioned servers and hosts.

**Important:**  When local firewall management is enabled, Morpheus will automatically set an IP table rule to allow incoming connections on tcp port 22 from the Morpheus Appliance.

**STORAGE MODE**  Single Disk, LVM or Clustered

**GUIDANCE**  Enable Guidance recommendations on cloud resources.

**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.

**AGENT INSTALL MODE**

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud-Init (when available): 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.

**API PROXY**  Required when a Proxy Server blocks communication between the Morpheus Appliance and the Cloud. Proxies can be added in the `Infrastructure -> Networks -> Proxies` tab.

**Provisioning Options**

**PROXY**  Required when a Proxy Server blocks communication between an Instance and 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 Instance Agent communication to the Appliance URL.

**USER DATA (LINUX)**  Add cloud-init user data or scripts. Assumes bash syntax.

### 10.3.23 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

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 must be scoped to a specific data center and cluster. If the drop downs do not populate, please verify the api url and provided credentials have access to Vcenter.

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 > 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 Provisioning > 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. By default these virtual machines are considered ‘unmanaged’ and do not appear in the Provisioning -> Instances area but rather Infrastructure -> Hosts -> 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 Provisioning -> 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.

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 Cloud detail page or Infrastructure -> Hosts section. From there click the + Container Host button 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 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.

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 `Use VNC`. 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**

**Usage**

**VCenter**
- Non-Propagating

**Datacenter**
- Non-Propagating

**Cluster**
- Non-Propagating

**Host**
- Non-Propagating

**Datastore**
- Propagating

**Privileges**

**Datastore**
- 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
10.3.24 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:
   - **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.
   - **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**: 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 Managed
- **SECURITY MODE**: Defines if Morpheus will control local firewall of provisioned servers and hosts.

**Important:** When local firewall management is enabled, Morpheus will automatically set an IP table rule to allow incoming connections on tcp port 22 from the Morpheus Appliance.

- **STORAGE MODE**: Single Disk, LVM or Clustered
- **GUIDANCE**: Enable Guidance recommendations on cloud resources.
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.

AGENT INSTALL MODE

- SSH / WINRM: Morpheus will use SSH or WINRM for Agent install.
- Cloud-Init (when available): 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.

API PROXY  Required when a Proxy Server blocks communication between the Morpheus Appliance and the Cloud. Proxies can be added in the Infrastructure -> Networks -> Proxies tab.

Provisioning Options

PROXY  Required when a Proxy Server blocks communication between an Instance and 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 Instance Agent communication to the Appliance URL.

USER DATA (LINUX)  Add cloud-init user data or scripts. Assumes bash syntax.

10.3.25 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:
   - **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.
   - **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**  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 Managed

**SECURITY MODE**  Defines if Morpheus will control local firewall of provisioned servers and hosts.

---

**Important:** When local firewall management is enabled, Morpheus will automatically set an IP table rule to allow incoming connections on tcp port 22 from the Morpheus Appliance.

---

**STORAGE MODE**  Single Disk, LVM or Clustered

**GUIDANCE**  Enable Guidance recommendations on cloud resources.

**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.

**AGENT INSTALL MODE**

- **SSH / WINRM:** Morpheus will use SSH or WINRM for Agent install.
- **Cloud-Init (when available):** Morpheus will utilize Cloud-Init or Cloudbase-Init for agent install when provisioning images with Cloud-Init/Cloudbase-Init installed. Morpheus will fallback on SSH or WINRM if cloud-init is not installed on the provisioned image.

**API PROXY**  Required when a Proxy Server blocks communication between the Morpheus Appliance and the Cloud. Proxies can be added in the `Infrastructure -> Networks -> Proxies` tab.

---

Provisioning Options

**PROXY**  Required when a Proxy Server blocks communication between an Instance and 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 Instance Agent communication to the Appliance URL.

**USER DATA (LINUX)**  Add cloud-init user data or scripts. Assumes bash syntax.
10.3.26 Windows Image with Cloudbase-Init

Morpheus supports provisioning Windows images with Cloudbase-init to set user data, network setting and other data at boot time. The following is an example of how to prepare a Windows image with cloudbase-init and optionally sysprep it.

Setup

1. On your Windows VM download and install Cloudbase-init from https://cloudbase.it/cloudbase-init/
2. Use the default settings, and do not run sysprep at the end of the install.
3. Under C:\Program Files\Cloudbase Solutions\Cloudbase-Init\conf, edit the cloudbase-init.conf file, referring to the sample configuration below. If the image will be sysprepped, edit cloudbase-init-unattend.conf and unattend.xml as well.

Note: Sample configurations only, user configurations may vary.

cloudbase-init.conf

```ini
[DEFAULT]
# username=Admin
# groups=Administrators
# inject_user_password=true
inject_user_password=false
first_logon_behaviour=no
config_drive_raw_hhd=true
config_drive_cdrom=true
config_drive_vfat=true
bsd.tar_path=C:\Program Files\Cloudbase Solutions\Cloudbase-Init\bin\bsd.tar.exe
mtools_path=C:\Program Files\Cloudbase Solutions\Cloudbase-Init\bin\
verbose=true
dbg=true
logdir=C:\Program Files\Cloudbase Solutions\Cloudbase-Init\log
logfile=cloudbase-init.log
default_log_levels=comtypes=INFO,suds=INFO,iso8601=WARN,requests=WARN
logging_serial_port_settings=
mtu_use_dhcp_config=true
ntp_use_dhcp_config=true
local_script_path=C:\Program Files\Cloudbase Solutions\Cloudbase-Init\LocalScripts\
# servers - tried in order until success
metadata_services=cloudbaseinit.metadata.services.configdrive.ConfigDriveService,
    cloudbaseinit.metadata.services.httpservice.HttpService,
    cloudbaseinit.metadata.services.ec2service.EC2Service,
    cloudbaseinit.metadata.services.maasservice.MaaSHttpService
# What plugins to execute.
plugins=cloudbaseinit.plugins.common.mtu.MTUPlugin,
    cloudbaseinit.plugins.windows.extendvolumes.ExtendVolumesPlugin,
    cloudbaseinit.plugins.common.userdata.UserDataPlugin,
    cloudbaseinit.plugins.common.networkconfig.NetworkConfigPlugin
# disabled plugins
# cloudbaseinit.plugins.common.sethostname.SetHostNamePlugin
```

(continues on next page)
# cloudbaseinit.plugins.windows.createuser.CreateUserPlugin
# cloudbaseinit.plugins.windows.setuserpassword.SetUserPasswordPlugin
# cloudbaseinit.plugins.common.networkconfig.NetworkConfigPlugin
# cloudbaseinit.plugins.common.sshpublickeys.SetUserSSHPublicKeysPlugin
# cloudbaseinit.plugins.windows.wintrmlistener.ConfigWinRMListenerPlugin
# cloudbaseinit.plugins.windows.licensing.WindowsLicensingPlugin
# cloudbaseinit.plugins.windows.ntpclient.NTPClientPlugin
# cloudbaseinit.plugins.common.userdata.UserDataPlugin

# Miscellaneous.
allow_reboot=false # allow the service to reboot the system
stop_service_on_exit=false

ccloudbase-init-unattend.conf

[DEFAULT]
username=Admin
groups=Administrators
inject_user_password=true
config_drive_raw_hhd=true
config_drive_cdrom=true
config_drive_vfat=true
bsdtar_path=C:\Program Files\Cloudbase Solutions\Cloudbase-Init\bin\bsdtar.exe
mtools_path=C:\Program Files\Cloudbase Solutions\Cloudbase-Init\bin\
verbose=true
debug=true
logdir=C:\Program Files\Cloudbase Solutions\Cloudbase-Init\log\
logfile=ccloudbase-init-unattend.log
default_log_levels=comtypes=INFO,suds=INFO,iso8601=WARN,requests=WARN
logging_serial_port_settings=
mtu_use_dhcp_config=true
ntp_use_dhcp_config=true
local_scripts_path=C:\Program Files\Cloudbase Solutions\Cloudbase-Init\LocalScripts\
metadata_services=ccloudbaseinit.metadata.services.configdrive.ConfigDriveService,
ccloudbaseinit.metadata.services.httpservice.HttpService,ccloudbaseinit.metadata.services.maasservice.
MaaSHttpService
plugins=ccloudbaseinit.plugins.common.mtu.MTUPlugin,ccloudbaseinit.plugins.common.
sethostname.SetHostNamePlugin,ccloudbaseinit.plugins.windows.extendvolumes.
ExtendVolumesPlugin
allow_reboot=false
stop_service_on_exit=false
check_latest_version=false

unattend.xml

<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
<settings pass="generalize">
    <SkipRearm>1</SkipRearm>
  </component>
</settings>
<PersistAllDeviceInstalls>false</PersistAllDeviceInstalls>
<DoNotCleanUpNonPresentDevices>false</DoNotCleanUpNonPresentDevices>
</component>
</settings>
<settings pass="oobeSystem">
    <component name="Microsoft-Windows-International-Core" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS" xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <InputLocale>en-US</InputLocale>
        <SystemLocale>en-US</SystemLocale>
        <UILanguage>en-US</UILanguage>
        <UserLocale>en-US</UserLocale>
    </component>
        <OOBE>
            <HideEULAPage>true</HideEULAPage>
            <ProtectYourPC>1</ProtectYourPC>
            <NetworkLocation>Home</NetworkLocation>
            <HideWirelessSetupInOOBE>true</HideWirelessSetupInOOBE>
        </OOBE>
        <TimeZone>UTC</TimeZone>
        <UserAccounts>
            <AdministratorPassword>
                <Value>administratorPassword</Value>
                <PlainText>true</PlainText>
            </AdministratorPassword>
            <LocalAccounts>
                <LocalAccount wcm:action="add">
                    <Password>
                        <Value>password</Value>
                        <PlainText>true</PlainText>
                    </Password>
                    <Group>administrators</Group>
                    <DisplayName>morpheus</DisplayName>
                    <Name>morpheus</Name>
                    <Description>Morpheus User</Description>
                </LocalAccount>
            </LocalAccounts>
        </UserAccounts>
    </component>
    <settings pass="specialize"/>
</settings>
</unattend>

4. Save and changes to cloudbase-init.conf, cloudbase-init-unattend.conf, and unattend.xml files.

Note: The Administrator password is being set in the unattend.xml file to be set upon boot after sysprep. This is not required if sysprep is not being used, and may not be preferred. Other mechanisms such as requiring the Administrator password to be reset or randomly generated can be used. Morpheus can also securely via the user_data file at provision time.

5. To run a sysprep using the cloudbase-init configuraiton, run the following in a command prompt:
cd C:\Program Files\Cloudbase Solutions\Cloudbase-Init\conf
C:\Windows\System32\sysprep\sysprep.exe /generalize /oobe /unattend:Unattend.xml

6. Sysprep will run and Windows will be powered down. The VM can now be converted to an Image/Blueprint and synced or uploaded to Morpheus and used for Provisioning.

Important: Upon upload or sync of the Virtual Image, ensure cloudbase enable is checked in the Virtual Image config, and the existing or unattend.xml credentials when using sysprep are populated.

### 10.3.27 Creating a CentOS 7 Morpheus Image

#### Overview

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 base CentOS 7 Image for use within Morpheus.

#### Creating a CentOS 7 Morpheus VMware Image

**VMWare**

When running in VMWare it is highly recommended that VMware Tools be installed. Without it, Morpheus will have difficulty assessing the host ip address and performing some additional automation tasks for the operating system.

**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 the Administration -> Provisioning -> 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
```

Gotyas

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"
```

(continues on next page)
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.

### 10.3.28 Morpheus Cloud Capability Coverage

<table>
<thead>
<tr>
<th>Cloud Integration</th>
<th>Ubuntu</th>
<th>CentOS</th>
<th>Debian</th>
<th>Linux Guest Cust</th>
<th>Cloud Init</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>Alibaba</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>Azure</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Cloud Limited</td>
</tr>
<tr>
<td>Digital Ocean</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>Google Cloud</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>Huawei</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>Hyperv</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>IBM Cloud / Softlayer</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>KVM</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>Nutanix</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>Openstack</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>Oracle Cloud</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>OVM</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>OTC</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>SCVMM</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>NO</td>
<td>Yes</td>
</tr>
<tr>
<td>Upcloud</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>VCD</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Vmware</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Xen</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### 10.4 Containers

#### 10.4.1 Docker
Overview

Morpheus can provision Docker Hosts into any cloud, convert existing Hosts to Docker Hosts, or even make itself a Docker Host.

Add a Docker Host to any cloud:

**Important:** As of Morpheus v3.6.2 Ubuntu 14.04 is no longer supported for Docker Hosts OS due to Docker version upgrade to address CVE-2019-5736. Existing Morpheus Docker Hosts running on 14.04 will not be able to upgrade to latest Morpheus Agent version. All new Morpheus Docker hosts using Ubuntu require Ubuntu 16.04 or 18.04.

1. Navigate to Infrastructure -> Hosts
2. Click the `+CONTAINER HOST` button
3. Select a container host type
4. Select a Group
5. Enter the following:
   • Name
   • Description
   • Visibility
   • Select a Cloud
   • Enter tags (optional)

Then click NEXT.
6. Configure the host options

Select a Service Plan (Volume, Memory and CPU count fields may not be shown if selected service plan does not have custom options):

- Add and set size the volumes
- Set memory size
- Set the CPU count
- Select a network

Optionally configure the following:

- OS username
- OS password
- Domain name
- Hostname (default is the name previously provided for the container host)

Then click the NEXT button.
7. Optionally add any Automation Workflows and configure for Backups.

8. Review and click Complete to save
Your new container host will begin provisioning, and soon be running and ready for containers.

**Add an existing Docker Host**

Morpheus can manage and inventory existing/brownfield Docker Hosts by using the *Manual Docker Host* option.

**Note:** Adding a Docker Host that was previously managed by another Morpheus Appliance will disable management of the host on that Appliance as the Morpheus Agent settings will be reconfigured.

**Note:** *Container Mode* on the Cloud settings where the Host is being added must be set to Morpheus for non-Kubernetes/Swarm hosts.

1. Navigate to Infrastructure -> Hosts
2. Select *+CONTAINER HOST* button
3. Select *Manual Docker Host*
4. In the *CREATE HOST* Wizard, enter the following:
   - **GROUP**
   - **GROUP** Select the Group this Host will be available for
   - *Select NEXT*
   - **NAME**
**CLOUD**  Select the Cloud the Host will be assigned to  
**NAME**  Enter name for the Docker Host in Morpheus  
**DESCRIPTION**  Enter optional description for the Docker Host  
**VISIBILITY**  Select Tenant Visibility  
**TAGS**  Add optional Morpheus tags (these are not meta-data tags)  
Select *NEXT*  
**CONFIGURE**  
**SSH HOST**  Enter IP or resolvable hostname of the target host  
**SSH USER**  Enter existing username on the target host  
**SSH PASSWORD**  Enter password for SSH User  
**PUBLIC KEY**  For key auth (recommended), copy and add the displayed Public Key to the `authorized_keys` file on the target host.  
**PLAN**  Default Manual  
**LVM ENABLED?**  Deselect if target host is not LVM enabled (required when using Morpheus provided docker images)  
**DATA VOLUME**  Enter path of the target data volume on the target host  
**SOFTWARE RAID?**  Enable for software RAID (disabled by default)  
**NET INTERFACE**  Enter network interface name of target host’s target network  
Select *NEXT*  
**AUTOMATION**  
**POST PROVISION**  Select a workflow to execute after Host is added (optional).  
Select *NEXT*  
**REVIEW**  Review settings and select *COMPLETE* to add the Manual Docker Host.  

Your new container host will begin provisioning, and soon be running and ready for containers.  

**Note:** Existing containers will be inventoried after the Hosts is successfully added.

---

### 10.4.2 Kubernetes

**Overview**

The Kubernetes Cloud type allow users to inventory and provision to existing Kubernetes clusters. New Kubernetes clusters can also be provisioning using Docker mode setting in clouds and provisioning new Docker hosts.

**Add Kubernetes Cloud**

1. Navigate to *Infrastructure* -> *Clouds*  
2. Select + *CREATE CLOUD*, select Kubernetes Cloud, and then click *Next*.  
3. Enter the following into the Create Cloud modal:
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.

API URL  Kubernetes API URL
API TOKEN  Kubernetes User API Token

Inventory Existing Instances  If enabled, existing Containers will be inventoried and appear in the Containers tab for the Kubernetes Cloud.

4. Save Changes

Create Kubernetes Cluster

Kubernetes Clusters can be provisioned into any Cloud Type by setting the CONTAINER MODE to Kubernetes in the Advanced Settings of a Cloud.

Important:  The CONTAINER MODE must be set prior to provisioning any Docker Hosts. Once Docker Hosts exist in a Cloud, the CONTAINER MODE setting cannot be changed.

Once the CONTAINER MODE is set on a Cloud, a Kubernetes Cluster can be created by selecting + CONTAINER HOST -> Kubernetes Master and then Kubernetes Worker from Infrastructure -> Hosts or Infrastructure -> Clouds -> select Cloud -> Hosts.

Important:  For the Kubernetes Cluster to be successfully created, the Kubernetes Master must finish provisioning before the worker(s) are created. Do not start provisioning a worker in the cluster until the Master is completed.

10.4.3 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.

10.5 Deployment

10.5.1 Git

Authentication

Add a private Github or Git Repository, an SSH Key pair must be added to Morpheus and the public key added to Github or the Git Repository for authentication.

To add a Key Pair to Morpheus:

1. Generate an SSH Key Pair, or use an existing SSH Key Pair.
2. Navigate to Infrastructure -> Key Pairs
3. Select + ADD # Enter both the Public and Private Private

---

10.6 DNS

10.6.1 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.
10.6.2 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.

Add Microsoft DNS Integration

Microsoft DNS 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 Microsoft DNS
   - NAME Name for the Integration in Morpheus
   - DNS SERVER IP or resolvable hostname of DNS server
   - USERNAME DNS provider username
   - PASSWORD DNS provider user password
   - ZONE (Optional) Enter a dns zone to limit scope
   - CREATE POINTERS Enabled to create A 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.

10.6.3 Power DNS

Overview

Morpheus integrates directly with Power DNS to automatically create DNS entries for Instances provisioned to a configured Cloud or Group. Morpheus also syncs in Power DNS Domains for easy selection while provisioning, or setting as the default Domain on a Cloud or Network.

Add Power DNS Integration

Power DNS 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  Power DNS
   NAME  Name for the Integration in Morpheus
   API HOST  URL of Power DNS API. Example: http://10.30.20.10:8081
   Token  Power DNS API Token
   Version  Power DNS 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, Power 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 Power 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.

10.7 Identity Management

10.7.1 Active Directory
Overview

Active Directory is Microsoft’s primary authentication service widely used in Enterprise organizations and even via Microsoft’s cloud services. While Active Directory also supports LDAP protocol support (which Morpheus can integrate with as well), the main Active Directory integration can also be utilized. It is even possible to map Active Directory groups to equivalent Roles within Morpheus.

Note: To use Active Directory, a valid / trusted SSL certificate must be in place on the Active Directory services (self signed will not work).

Adding an Active Directory Integration

1. Navigate to Administration -> Tenants
2. Select a Tenant
3. Select IDENTITY SOURCES
4. Select + IDENTITY SOURCE
5. Choose “Active Directory”
6. Populate the following:
   - **Name**: Unique name for authentication type.
   - **AD Server**: Hostname or IP address of AD Server.
   - **Domain**: Domain name of AD Domain.
   - **Binding Username**: Service account username for bind user.
   - **Binding Password**: Password for bind service account.
   - **Required Group**: The AD group users must be in to have access (optional)
   - **Default Role**: The default role a user is assigned if no group is listed under AD user that maps under Role Mappings section.
   - **Service Account Holder**: This is the admin account type in Morpheus and an AD group can be created and populated to a user that this role should be assigned. Roles are assigned dynamically based on group membership.
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: Only the username is required with password, not the username@domain.

Note: Sub-tenant Morpheus API authentication for Active Directory generated users is not currently supported.
10.7.2 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 (external login) from the TYPE field
6. Add a Name and optional Description for the SAML integration
There are 3 sections with fields that need to be populated depending on the desired configuration:

- **SAML Configuration**
  - **LOGIN REDIRECT URL** This is the SAML endpoint Morpheus will redirect to when a user signs into Morpheus via SAML.
  - **LOGOUT POST URL** The url morpheus will post to when a SAML user log out of Morpheus to log out of the SAML provider as well.
  - **SIGNING PUBLIC KEY** Add the X.509 Certificate public key from the SAML provider.

- **Role Mappings**
  - **DEFAULT ROLE** Role a saml user will be assigned by default when no role is mapped
  - **ROLE ATTRIBUTE NAME** The name of the attribute field that will map to morpheus roles, such a MemberOf

- **User Attribute Names**
REQUIRED ROLE ATTRIBUTE VALUE  Role attribute value that a user must be assigned/a member of to be authorized, such as group or role in the SAML SP.

The rest of the Role Mapping Fields will be the existing Roles in morpheus with a Role Attribute Value field.

User Attribute Names

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 NAME  SAML SP field value to map to Morpheus user email address
Identity Source

**TYPE**  SAML [Beta]

**ACTIVE**  Yes

**NAME**  SAML

**DESCRIPTION**  onelogin SAML

SAML Configuration

**LOGIN REDIRECT URL**  https://morpheusdata-dev.onelogin.com/trust/saml2/http-/...

- Do not include SAMLRequest parameter

**LOGOUT POST URL**  https://morpheusdata-dev.onelogin.com/trust/saml2/http-/...

**SIGNING PUBLIC KEY**  MIIEFzCCAv+gAwIBAgIuayYdMuoXBTGcalAARanxhrJwwtQwDQYJKoZIhvcNAQEF

Role Mappings

**DEFAULT ROLE**  System Admin

**ROLE ATTRIBUTE NAME**  MemberOf

**REQUIRED ROLE ATTRIBUTE VALUE**  dev

**LEGACY ACCOUNT ADMIN**  Role Attribute Value

User Attribute Names

**GIVEN NAME ATTRIBUTE NAME**  firstName
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.

- SP ENTITY ID
- SP ACS URL*
- IDP LOGIN REDIRECT URL
- IDP LOGOUT POST URL
- SP METADATA

Sample Metadata code output:

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<EntityDescriptor entityID="https://someip.com/saml/CDWPjmZt" 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/CDWPjmZt"/>
    </SPSSODescriptor>
</EntityDescriptor>
```

**Note:** Different SAML providers will have different field names and requirements. A onelogin SAML Test Connector (IdP w/attr) was used for the example integration this article.

### Onelogin SAML SSO

For Onelogin SAML integration, the following fields are mapped:

- LOGIN REDIRECT URL : SAML 2.0 Endpoint (HTTP)
- LOGOUT POST URL : SLO Endpoint (HTTP)
- SIGNING PUBLIC KEY : X.509 Certificate
- SP ENTITY ID: ACS (Consumer) URL Validator
- SP ACS URL: ACS (Consumer) URL
10.7.3 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 a Azure AD SAML Integration

Azure requires inputing the Identifier (Entity ID) and Reply URL (Assertion Consumer Service URL) in the Azure SSO configuration before it provides the Endpoints and Certificate neccessary 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 config, we need to create a base SAML Integration in Morpheus first.

To add a base 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 (external login) from the TYPE field
6. Add a Name, optional Description and any value in the LOGIN REDIRECT URL field. Since we do not have the LOGIN REDIRECT URL from Azure yet, type any text such as test into the LOGIN REDIRECT URL field so the Identity Source Integration can be saved and the Identifier (Entity ID) and Reply URL (Assertion Consumer Service URL) generated. We will edit the Integration with the proper LOGIN REDIRECT URL after configuring SSO in Azure.
7. Select SAVE CHANGES.

Upon save 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 config.

Configure Azure SSO

This guide assumes an Azure AD Application has already been created in Azure, with a subscription level high enough to configure SSO in the application. Please refer to Azure documentation if this has not already been configured.

1. Next, in the Azure Active Directory Application details page, select Single sign-on, then enter the following:
   - Single Sign-on Mode dropdown Select SAML-based Sign-on
   - 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.
2. Save and click the Test SAML Settings button. Azure will confirm connecton with Morpheus
3. In Azure SSO config step 3, select user.userprincipalname as the User Identifier.
4. Also in step 3, select “View and edit all other user attributes” the copy the NAMESPACE url for the following:
   - Name givenname Value: user.givenname
   - Namespace: http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname
Morpheus Documentation, Release 3.6.2

Name: surname Value: user.surname Namespace: http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname

Name: emailaddress Value: user.mail Namespace: http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress

Note: The Namespace URLs will be used in the Role Attribute Value section in the Morpheus Identity Source Integration.

5. In Azure SSO config step 4, if one has not been generated, select Create new certificate to generate a new SAML Signing Certificate.

6. Enter a valid email address to receive certificate expiration notifications at (not related to Morpheus).

7. In Azure SSO config step 5, select `Configure {AD App Name}

8. In the Configure sign-on pane, copy the following:
   - SAML Single Sign-On Service URL This will be used for the LOGIN REDIRECT URL in the Morpheus Identity Source Integration settings
   - Sign-Out URL This will be used for the LOGOUT POST URL in the Morpheus Identity Source Integration settings
   - Click on the SAML XML Metadata link, open the xml file, and copy the key between the <X509Certificate> and </X509Certificate>. This will be used for the SIGNING PUBLIC KEY in the Morpheus Identity Source Integration settings

Example Key (the key has been altered and is not valid):
MIIC8ECCAdigAwIBAgIQEEO2X1NvY9Dc60wlsKEMzANBgkqhkiG9w0BAQsFADA0MTIwMAYDVQQDEyJNaWNyb3NvZnQgQXp1cmUgRmVkZXJhdGVkIFNTTyBDZX...

Example Key (the key has been altered and is not valid):
MIIC8ECCAdigAwIBAgIQEEO2X1NvY9Dc60wlsKEMzANBgkqhkiG9w0BAQsFADA0MTIwMAYDVQQDEyJNaWNyb3NvZnQgQXp1cmUgRmVkZXJhdGVkIFNTTyBDZX...

9. Save the SSO config in Azure AD app and return to Morpheus

**Edit the existing Azure AD SAML Integration**

Now that we have the required information, we can finalize the Azure AD SAML Integration in Morpheus

1. Edit the existing Azure AD SAML Integration created above and populate the following:
   - LOGIN REDIRECT URL Add the SAML Single Sign-On Service URL copied from Azure SSO config.
   - LOGOUT POST URL Add the Sign-Out URL copied from Azure SSO config.
   - SIGNING PUBLIC KEY (uncheck “Do not validate SAMLResponse signatures” if desired) Add the SAML XML Metadata key copied from Azure SSO config.
   - GIVEN NAME ATTRIBUTE NAME Enter the givenname Namespace url from Azure SSO config: http://schemas.xmlsoap.org/ws/2005/05/identity/claims
   - EMAIL ATTRIBUTE NAME Enter the surname Namespace url from Azure SSO config: http://schemas.xmlsoap.org/ws/2005/05/identity/claims
   - SURNAME ATTRIBUTE NAME Enter the emailaddress Namespace url from Azure SSO config: http://schemas.xmlsoap.org/ws/2005/05/identity/claims
Configure Role Mappings

Role mappings will map Azure AD Groups to Morpheus Roles. Azure AD users will be assigned Roles in Morpheus upon signing 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.

**ROLE ATTRIBUTE NAME** Enter http://schemas.microsoft.com/ws/2008/06/identity/claims/groups for Azure AD SSO

**REQUIRED ROLE ATTRIBUTE VALUE** 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.

**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

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.

**Signing In to Morpheus**

When there is an active SAML/Azure AD SSO Identity Source Integration, a new button will appear on the Morpheus login page below LOGIN WITH with the name of the Identity Source Integration as the button title. Example: **AZURE AD**. Another button titled “USERNAME AND PASSWORD” is also added in place of the standard Username and Password fields.

- **SAML/Azure AD SSO users can log into Morpheus by clicking the SAML button** This will redirect the User to Azure AD app sign in url. If they are currently signed into Azure and authorized, the user will be instantly signed into Morpheus.

- **Local Morpheus users can select “USERNAME AND PASSWORD” to sign in with their local credentials as before.** If no local users other than the System Admin have been created, “USERNAME AND PASSWORD” option will not be displayed, only the SAML option.

**10.7.4 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 authenticate 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 if 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.
6. 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.

10.8  ITSM

10.8.1  ServiceNow

Add Service Now Integration

1. Navigate to Administration -> Integrations
2. Select + NEW INTEGRATION
3. Select ServiceNow from the TYPE dropdown.

4. Add the following:
   - **NAME** Name of the Integration in Morpheus.
   - **ENABLED** Leave checked to enable the Integration.
   - **HOST** Url of the ServiceNow Instance ex: https://your.instance.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** Above ServiceNow user's password

5. Save Changes

---

**Important:** When using ServiceNow version London, the following steps must also be performed. An administrator needs to modify the access permissions on the 'catalog_script_client' and 'io_set_item' tables. This is performed by ensuring the ‘Can create’, ‘Can update’, and ‘Can Delete’ are checked under Application Access for ‘All application scopes’ for these tables.

---

**ServiceNow Approval Policies**

**Add ServiceNow Provision Approval Policy to a Cloud**

---

**Note:** Any Instance provisioned into a Cloud with an Approval Policy enabled will require approval.

To add a ServiceNow Approval policy to a Cloud:

1. Navigate to *Infrastructure* -> *Clouds*
2. Select a Cloud by clicking on the Cloud Name link
3. Select the POLICIES tab
4. Select + 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 workflow in ServiceNow. Note 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 sub-tenant, type the name of the tenant(s) and select the Tenant(s) from the 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 require 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 link
3. Select the POLICIES tab
4. Select + 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 workflow in ServiceNow. Note 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 sub-tenant, type the name of the tenant(s) and select the Tenant(s) from the 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 in 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 Internal Approvals.

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 5 minutes of being approved in ServiceNow.

**ServiceNow Service Catalog Integration**

The following is a guide to installing the Morpheus ServiceNow application.

**Important:** A valid SSL Certificate is required on the Morpheus Appliance for the ServiceNow plugin to be able to communicate with the appliance.
ServiceNow Configuration

1. Install the Morpheus Application from the ServiceNow store
2. Navigate to Morpheus Catalog -> Properties
3. Set the following properties:
   - **Morpheus Appliance Endpoint** The full url to your Morpheus appliance
   - **Password** Password of the Morpheus Administrator
   - **Username** Username of the Morpheus Administrator
4. Create a new User
5. Assign the following roles to the user:
   - `x_moda_morpheus_ca.integration`
   - `catalog_admin`
   - `itil`
   - `rest_service`

Important: When using ServiceNow version London, the following steps must also be performed. An administrator needs to modify the access permissions on the ‘catalog_script_client’ and ‘io_set_item’ tables. This is performed by ensuring the ‘Can create’, ‘Can update’, and ‘Can Delete’ are checked under Application Access for ‘All application scopes’ for these tables.

Morpheus Configuration

1. Navigate to Administration -> Integrations
2. Click `+ NEW INTEGRATION`
3. Select ‘ServiceNow’ in the Type field
4. Fill in the Host, User and Password fields (using the User and Password created in the previous section)

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
### 10.8.2 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](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:

```json
{
   "Description":"Created from Morpheus by ${instance.createdByUsername} in ${zone.name}"
}
```

5. Save Changes

### 10.8.3 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.

<table>
<thead>
<tr>
<th>API Endpoint</th>
<th>Action</th>
<th>BMC Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/arsys/v1/entry/CTM:People</td>
<td>GET</td>
<td>CTM:People</td>
</tr>
<tr>
<td>/api/arsys/v1/entry/COM:Company?q=%27Status%27=%22Enabled%22&amp;fields=values(Company)</td>
<td>GET</td>
<td>COM:Company</td>
</tr>
<tr>
<td>/api/arsys/v1/entry/User</td>
<td>GET</td>
<td>User</td>
</tr>
<tr>
<td>/api/arsys/v1/entry/Group</td>
<td>GET</td>
<td>Group</td>
</tr>
<tr>
<td>/api/arsys/v1/entry/CHG:Infrastructure%20Change</td>
<td>POST</td>
<td>CHG:Infrastructure</td>
</tr>
<tr>
<td>/api/arsys/v1/entry/CHG:Infrastructure%20Change</td>
<td>PUT</td>
<td>CHG:Infrastructure</td>
</tr>
<tr>
<td>/api/arsys/v1/entry/CHG:Infrastructure%20Change</td>
<td>GET</td>
<td>CHG:Infrastructure</td>
</tr>
<tr>
<td>/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_DiskDrive</td>
<td>POST</td>
<td>BMC.CORE:BMC_DiskDrive</td>
</tr>
<tr>
<td>/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_DiskDrive</td>
<td>PATCH</td>
<td>BMC.CORE:BMC_DiskDrive</td>
</tr>
<tr>
<td>/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_DiskDrive</td>
<td>DELETE</td>
<td>BMC.CORE:BMC_DiskDrive</td>
</tr>
<tr>
<td>/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_IPEndpoint</td>
<td>POST</td>
<td>BMC.CORE:BMC_IPEndpoint</td>
</tr>
<tr>
<td>/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_IPEndpoint</td>
<td>PATCH</td>
<td>BMC.CORE:BMC_IPEndpoint</td>
</tr>
<tr>
<td>/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_IPEndpoint</td>
<td>DELETE</td>
<td>BMC.CORE:BMC_IPEndpoint</td>
</tr>
<tr>
<td>/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_Memory</td>
<td>POST</td>
<td>BMC.CORE:BMC_Memory</td>
</tr>
<tr>
<td>/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_Memory</td>
<td>PATCH</td>
<td>BMC.CORE:BMC_Memory</td>
</tr>
<tr>
<td>/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_Memory</td>
<td>DELETE</td>
<td>BMC.CORE:BMC_Memory</td>
</tr>
<tr>
<td>/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_Processor</td>
<td>POST</td>
<td>BMC.CORE:BMC_Processor</td>
</tr>
<tr>
<td>/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_Processor</td>
<td>PATCH</td>
<td>BMC.CORE:BMC_Processor</td>
</tr>
<tr>
<td>/api/cmdb/v1.0/instances/BMC.ASSET/BMC.CORE/BMC_Processor</td>
<td>DELETE</td>
<td>BMC.CORE:BMC_Processor</td>
</tr>
<tr>
<td>/api/arsys/v1/entry/AST:ComputerSystem</td>
<td>GET</td>
<td>AST:ComputerSystem</td>
</tr>
<tr>
<td>/api/arsys/v1/entry/AST:ComputerSystem</td>
<td>PUT</td>
<td>AST:ComputerSystem</td>
</tr>
<tr>
<td>/api/arsys/v1/entry/AST:ComputerSystem</td>
<td>POST</td>
<td>AST:ComputerSystem</td>
</tr>
<tr>
<td>/api/arsys/v1/entry/AST:IPEndpoint</td>
<td>GET</td>
<td>AST:IPEndpoint</td>
</tr>
<tr>
<td>/api/arsys/v1/entry/AST:IPEndpoint</td>
<td>PUT</td>
<td>AST:IPEndpoint</td>
</tr>
<tr>
<td>/api/arsys/v1/entry/AST:IPEndpoint</td>
<td>POST</td>
<td>AST:IPEndpoint</td>
</tr>
<tr>
<td>/api/arsys/v1/entry/AST:DiskDrive</td>
<td>GET</td>
<td>AST:DiskDrive</td>
</tr>
<tr>
<td>/api/arsys/v1/entry/AST:DiskDrive</td>
<td>PUT</td>
<td>AST:DiskDrive</td>
</tr>
<tr>
<td>/api/arsys/v1/entry/AST:DiskDrive</td>
<td>POST</td>
<td>AST:DiskDrive</td>
</tr>
<tr>
<td>/api/arsys/v1/entry/AST:Processor</td>
<td>GET</td>
<td>AST:Processor</td>
</tr>
<tr>
<td>/api/arsys/v1/entry/AST:Processor</td>
<td>PUT</td>
<td>AST:Processor</td>
</tr>
<tr>
<td>/api/arsys/v1/entry/AST:Memory</td>
<td>POST</td>
<td>AST:Memory</td>
</tr>
<tr>
<td>/api/arsys/v1/entry/AST:Memory</td>
<td>PUT</td>
<td>AST:Memory</td>
</tr>
<tr>
<td>/api/jwt/login</td>
<td>POST</td>
<td></td>
</tr>
</tbody>
</table>

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

### 10.9 Keys and Certificates

#### 10.9.1 Venafi

**Overview**

Morpheus integrates with Venafi to sync and request SSL certificates

**Add Venafi**

1. Navigate to [Administration > Integrations](#)
2. Select + NEW INTEGRATION
3. Enter in the following:
   - Name
   - Venafi Host
   - Username
   - Password
4. Click SAVE CHANGES

**Link Venafi To Cloud**

To add Venafi as the *Trust Provider* for a cloud

1. Navigate to [Infrastructure > Clouds](#)
2. Select Cloud
3. Select EDIT
4. Under *Advanced Options* select the Venafi integration from the *TRUST PROVIDER* dropdown
5. Select *SAVE CHANGE*

### 10.10 Load Balancers

#### 10.10.1 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

#### 10.10.2 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

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 post 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 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 as *: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.

10.11 Logs

10.11.1 LogRhythm

Adding LogRhythm Integration

1. Navigate to Administration -> Logs
2. Expand the LogRhythm section
3. Enable the integration
4. Fill in the following:
   - Enabled  Enable the LogRhythm integration
   - Host  IP or Hostname of the LogRhythm server.
   - Port  Port configured to access the LogRhythm server.
5. SAVE
10.11.2 Splunk

Overview

The Morpheus Splunk Integration allows forwarding logs from managed Linux hosts and vm’s to a target Splunk listener by changing the rsyslogd config on linux vm’s to point to Splunk forwarders. The logs will be forwarded from the clients, not from the Morpheus Appliance.

Adding Splunk Integration

1. Add a syslog listener configuration in Splunk.
2. Navigate to Administration -> Logs
3. Expand the Splunk section
4. Enable the integration
5. Fill in the following:
   - **Enabled**: Enable the Splunk integration
   - **Host**: IP or Hostname of the Splunk server.
   - **Port**: Port configured to access the Splunk server.
6. SAVE

Once added, syslogs from managed Linux hosts and vm’s will be forwards from the clients to the target Splunk listener.

10.11.3 Syslog

Adding Syslog Integration

1. Navigate to Administration -> Logs
2. Expand the Morpheus logging section
3. Add the Syslog forwarding rules
4. Select **QUICK ADD**

10.12 Monitoring

10.12.1 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

<table>
<thead>
<tr>
<th>Morpheus Severity</th>
<th>ServiceNow Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info</td>
<td>Low/Medium/High</td>
</tr>
<tr>
<td>Warning</td>
<td>Low/Medium/High</td>
</tr>
<tr>
<td>Critical</td>
<td>Low/Medium/High</td>
</tr>
</tbody>
</table>

10.12.2 AppDynamics

AppDynamics is a very powerful performance and application monitoring tool. It features advanced correlation features and profiling capabilities for a very wide range of application platforms including native Docker support. Due to the level of capabilities of AppDynamics there are more required settings to integrate it with Morpheus.

Configuring The AppDynamics Integration

1. Navigate to Administration > Monitoring
2. Expand the AppDynamics section
3. Toggle the Enable slider
4. Fill out desired fields
5. Save

Once saved, all hosts will automatically be configured to install the AppDynamics agent.

AppDynamics is capable of being run as a paid SaaS based service as well as an on premise installation and Morpheus supports both configurations. Most input fields related to connecting to AppDynamics provide helpful tips as to what information exactly needs provided and where to acquire it.

10.12.3 NewRelic

Configuring The NewRelic Integration

1. Navigate to Administration > 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.

10.13 Networking

10.13.1 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

1. Navigate to Infrastructure - Network - Services
2. Select + ADD -> IPAM -> Infoblox
3. Enter the following:
**NAME**  Name of the Integration in Morpheus

**Enabled**  Deselect to disable the Integration

**URL**  Infoblox wapi url. Example: https://x.x.x.x/wapi/v2.2.1

**USERNAME**  Infoblox user username

**PASSWORD**  Infoblox user password

**Disable SSL SNI Verification**  Leave selected to disable SSL SNI Verification

**NETWORK FILTER**  Filter which networks are synced into Morpheus. Example: Network Filter: [network_view=default&*Building=work]

**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

---

10.13. Networking

405
4. Select **SAVE IPAM INTEGRATION**

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 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

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

<table>
<thead>
<tr>
<th>HOSTNAME</th>
<th>sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP ADDRESS</td>
<td>10.30.23.88</td>
</tr>
<tr>
<td>DOMAIN</td>
<td>infoblox.den.bertramlabs.com</td>
</tr>
</tbody>
</table>

- **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

<table>
<thead>
<tr>
<th>NAME</th>
<th>sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>A</td>
</tr>
<tr>
<td>CONTENT</td>
<td>10.30.22.89</td>
</tr>
<tr>
<td>TTL</td>
<td>86400</td>
</tr>
</tbody>
</table>

**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`

### 10.13.2 phpIPAM

**Configuration**

**Configure phpIPAM API**

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 none.

**Add phpIPAM integration to Morpheus**

1. Navigate to *Infrastructure - Network - Services*
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

10.13.3 NSX

Add NSX Integration

1. Navigate to INFRASTRUCTURE -> NETWORK
2. Select the SERVICES tab
3. Select Select + ADD -> VMWare NSX
4. Enter the following:
   - NAME  Name for the NSX Integration in Morpheus
   - API HOST  URL of NSX Manager
   - USERNAME  NSX Manager Admin Username
   - PASSWORD  NSX Manager Admin password
   - VMWARE CLOUD  Select the existing VMware cloud associated with this NSX integration.
5. Select ADD NETWORK INTEGRATION

Once the NSX Integration is added Morpheus will sync in existing Transport Zones, Logical Switches, and Edge Gateways. New Transport Zones, Logical Switches, and Edge Gateways can be now be created.

Create NSX Transport Zone

1. Navigate to INFRASTRUCTURE -> NETWORK
2. Select the SERVICES tab
3. Select the name of NSX Integration
4. Select the TRANSPORT ZONES tab
5. Select + CREATE NSX TRANSPORT ZONE
   - NAME  Name of Transport Zone
   - DESCRIPTION  Description for the Transport Zone
   - CLUSTER  Select the Cluster the Transport Zone will be provisioned to
Create NSX Logical Switch and Edge Gateway

**Important:** Prior to creating a Logical Switch and Edge Gateway, associated External VMware Networks must be configured in Morpheus. Navigate to **INFRASTRUCTURE -> NETWORK** and edit any Distributed Switch Groups that will be used and populate the Gateway, DNS and CIDR.

1. Navigate to **INFRASTRUCTURE -> NETWORK**
2. Select the **SERVICES** tab
3. Select the name of NSX Integration
4. Select the **LOGICAL SWITCHES** tab
5. Select *CREATE NSX LOGICAL SWITCH*
6. Populate the following for the Logical Switch and Edge Gateway Configurations:
   
   **Logical Switch Configuration:**
   
   - **NAME** Name of the Logical Switch
   - **DESCRIPTION** d
   - **TRANSPORT ZONE** Select an existing Transport Zone
   - **CIDR** Add the CIDR for the Logical Switch. Example: 10.30.28.0/24
   - **TENANT NAME** Enter Tenant name for the Logical Switch (Optional)

   **Edge Gateway Configuration:**
   
   - **HOSTNAME** Enter Hostname of the Edge Gateway
   - **SIZE** Select Size of the Edge Gateway
   - **EXTERNAL NETWORK** Select the External Network for the Edge Gateway.
   - **IP ADDRESS** Populate IP address to be assigned to the Edge Gateway
   - **DATA STORE** Select the Datastore for the Gateway
   - **RESOURCE POOL** Select the Resource Pool for the Gateway
   - **FOLDER** Select a Folder for the Edge Gateway (optional)
   - **USERNAME** Enter a Username for the Edge Gateway
   - **PASSWORD** Enter a Password for the Edge Gateway

   **Important:** The Gateway, DNS and CIDR must be populated on an external network for it to be selectable when creating an Edge Gateway.

   **Note:** Password length must be at-least 12 characters and at-max 255 characters. It must contain mix of alphabets with both upper case and lower case, numbers and at-least one special character. Password must not contain username as substring. Character must not consecutively repeat 3 or more times.

7. Select *CREATE*
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

<table>
<thead>
<tr>
<th>▼ ACI Options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ENDPOINT GROUP</strong></td>
</tr>
<tr>
<td><strong>APP PROFILE</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>▼ ACI Security</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CONSUMES</strong></td>
</tr>
<tr>
<td><strong>PROVIDES</strong></td>
</tr>
</tbody>
</table>

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

• 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.

10.13.5 Bluecat

Overview

Morpheus integrates with Bluecat IPAM to scope pools to networks for Static IP assignment from Infoblox to your Morpheus instances.

Adding Bluecat to Morpheus

1. Navigate to Infrastructure > Network > Services
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 the id of a config, block or network, or comma separated combination of configs, blocks and/or networks.
5. Click **SAVE CHANGES**
The Bluecat Integration will be saved, IP pools will sync in and populate under Infrastructure > Network > IP Pools, and Domain 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 a 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

### 10.14 Service Discovery

#### 10.14.1 Consul

Morpheus can integrate with Consul to automatically install the Consul Agent in Client Mode on Instances and configure communication with the Consul host.

**Add Consul Integration**

1. Navigate to Administration -> Integrations and select + New Integration
2. Select Integration Type *Consul Service Registry*
3. Populate the following fields:
   - **Name** Name of the Consul Integration in Morpheus
   - **Enabled** Enabled by default
   - **Consul Host** IP or Url of the Consul Host
   - **Consul Http Port** Http port of the Consul Host
   - **Username** Consul Host User
   - **Password** Consul Host User Password
Datacenter ID  Validator key for the organization

4. Save Changes

The added Consul Integration is now available for use in Morpheus, but must be scoped to a Cloud or Group to automatically install the Consul Agent while provisioning.

**Scope Consul Integration to a Cloud**

1. Navigate to *Infrastructure -> Clouds*
2. Edit the target Cloud
3. Expand the *Advanced Options* section
4. In the *Service Registry* dropdown, select the Consul Integration.
5. Save Changes

**Scope Consul Integration to a Group**

1. Navigate to *Infrastructure -> Groups*
2. Edit the target Group
3. Expand the *Advanced Options* section
4. In the *Service Registry* dropdown, select the Consul Integration.
5. Save Changes

And that’s it. After your integration is set up, all containers deployed within the Group or Cloud integrated will provision with the Consul Agent in Client Mode, gossiping to your Consul Server!

### 10.15 Storage

#### 10.15.1 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.

### 10.15.2 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**

### 10.15.3 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 times 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

- **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 if 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.
**Morpheus Documentation, Release 3.6.2**

**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.

### 10.15.4 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 if 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.

### 10.16 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, HPE OneView, OpenTelekom Cloud, IBM Bluemix, Softlayer and UpCloud are all supported.

In addition, Morpheus is verified to work with, but not limited to:
<table>
<thead>
<tr>
<th>Technology</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Private Cloud</strong></td>
<td></td>
</tr>
<tr>
<td>Azure Stack</td>
<td>GA</td>
</tr>
<tr>
<td>Microsoft Hyper-V</td>
<td>2012R2, 2016</td>
</tr>
<tr>
<td>Nutanix Acropolis</td>
<td>5.0 - 5.10 Note: In 5.5 - 5.7 if Prism Central is enabled, no actions that create images in Prism will function due to Prism Central Image Management.</td>
</tr>
<tr>
<td>Openstack</td>
<td>Juno, Kilo, Liberty, Mitaka, Newton, Ocata, Pike, Queens</td>
</tr>
<tr>
<td>vCloud Director</td>
<td>8.20, 9.1</td>
</tr>
<tr>
<td>VMware vCenter</td>
<td>5.5, 6.0, 6.5, 6.7</td>
</tr>
<tr>
<td>VMware ESXi</td>
<td>5.5, 6.0, 6.5, 6.7</td>
</tr>
<tr>
<td>VMware Fusion</td>
<td>8, 9, 10+</td>
</tr>
<tr>
<td>XenServer</td>
<td>7.0</td>
</tr>
<tr>
<td><strong>Monitoring</strong></td>
<td></td>
</tr>
<tr>
<td>App Dynamics</td>
<td>4.5.1</td>
</tr>
<tr>
<td><strong>Networking</strong></td>
<td></td>
</tr>
<tr>
<td>Cisco ACI</td>
<td>3.10</td>
</tr>
<tr>
<td>VMware NSX</td>
<td>.V</td>
</tr>
<tr>
<td><strong>Backups</strong></td>
<td></td>
</tr>
<tr>
<td>Veeam</td>
<td>9.5</td>
</tr>
<tr>
<td>Commvault</td>
<td>v11 sp 12</td>
</tr>
<tr>
<td>Rubrik</td>
<td>4.2</td>
</tr>
<tr>
<td><strong>Containers</strong></td>
<td></td>
</tr>
<tr>
<td>Docker</td>
<td>1.12.6</td>
</tr>
<tr>
<td>Kubernetes</td>
<td>1.11</td>
</tr>
<tr>
<td>Jenkins</td>
<td>1.64</td>
</tr>
<tr>
<td><strong>ITSM</strong></td>
<td></td>
</tr>
<tr>
<td>ServiceNow</td>
<td>Istanbul, Jakarta, Kingston, London</td>
</tr>
<tr>
<td><strong>Logging</strong></td>
<td></td>
</tr>
<tr>
<td>Splunk</td>
<td>7.10</td>
</tr>
</tbody>
</table>

If you have any specific requirements please contact support@morpheusdata.com
11.1 Ansible Troubleshooting

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 but certain os’s or network conditions can prevent automated install. You can run ansible --version in the Morpheus, or in the Ansible integration details page (Administration -> Integrations -> Select Ansible Integration, or in the Ansible tab of a group or cloud scoped to Ansible) just run --version as ansible is already included in the command.

11.1.1 If Ansible is not installed

Follow these instructions to install, or use your preferred installation method

Ubuntu:

```bash
sudo apt-get install software-properties-common
sudo apt-add-repository ppa:ansible/ansible
sudo apt-get update
sudo apt-get install ansible
```

CentOS:

```bash
sudo yum install epel-release
sudo yum install ansible
```

Then create the working Ansible directory for Morpheus:

```bash
sudo mkdir /opt/morpheus/.ansible
sudo chown morpheus-app.morpheus-app /opt/morpheus/.ansible
```
11.1.2 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 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.

11.2 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 Operations -> Health -> Morpheus Logs. Please copy relevant logs and add to case as an attachment.
11.3 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

11.4 Cannot Login

11.4.1 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.

11.4.2 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**

### 11.4.3 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.

### 11.5 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
```

### 11.6 Common Ports & Requirements

The following chart is useful for troubleshooting Agent install, Static IP assignment, Remote Console connectivity, and Image transfers.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Method</th>
<th>OS</th>
<th>Source</th>
<th>Destination</th>
<th>Port</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent Communication</td>
<td>All</td>
<td>All</td>
<td>Node</td>
<td>Appliance</td>
<td>443</td>
<td>DNS Resolution from node to appliance url</td>
</tr>
<tr>
<td>Agent Install</td>
<td>All</td>
<td>Linux</td>
<td>Node</td>
<td>Appliance</td>
<td>80</td>
<td>Used for appliance yum and apt repos</td>
</tr>
<tr>
<td></td>
<td>SSH</td>
<td>Linux</td>
<td>Appliance</td>
<td>Node</td>
<td>22</td>
<td>DNS Resolution from node to appliance url Virtual Images configured</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SSH Enabled on Virtual Image</td>
</tr>
<tr>
<td>WinRM</td>
<td>Windows</td>
<td>Windows</td>
<td>Node</td>
<td>Appliance</td>
<td>5985</td>
<td>DNS Resolution from node to appliance url Virtual Images configured</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>WinRM Enabled on Virtual Image(\textit{winrm quickconfig})</td>
</tr>
<tr>
<td>Cloud-init</td>
<td>Linux</td>
<td>Linux</td>
<td></td>
<td></td>
<td></td>
<td>Cloud-init installed on template/image</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cloud-init settings populated in User Settings or in \textit{Admin -&gt; Provisioning}</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Agent install mode set to Cloud-Init in Cloud Settings</td>
</tr>
<tr>
<td>Cloudbase-init</td>
<td>Windows</td>
<td>Windows</td>
<td></td>
<td></td>
<td></td>
<td>Cloudbase-init installed on template/image</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cloud-init settings populated in User Settings or in \textit{Admin -&gt; Provisioning}</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Agent install mode set to Cloud-Init in Cloud Settings</td>
</tr>
<tr>
<td>VMTools</td>
<td>All</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
<td>VMTools installed on template</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cloud-init settings populated in Morpheus user settings or in \textit{Administration -&gt; Provisioning} when using Static IP’s</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Existing User credentials entered on Virtual Image when using DHCP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RPC mode set to VMtools in VMware cloud settings.</td>
</tr>
<tr>
<td>Static IP Assignment &amp; IP Pools</td>
<td>Cloud-Init</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
<td>Network configured in Morpheus (Gateway, Primary and Secondary DNS, CIDR populated, DHCP disabled)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cloud-init/Cloudbase-init installed on template/image</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cloud-init settings populated in Morpheus user settings or in \textit{Administration -&gt; Provisioning}</td>
</tr>
<tr>
<td>VMware Tools</td>
<td>All</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
<td>Network configured in Morpheus (Gateway, Primary and Secondary DNS, CIDR populated, DHCP disabled)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
11.7 How to un-manage an Instance/VM/Host

11.7.1 Description

A managed VM (and associated Instance) needs to be unmanaged and returned to Discovered type.

11.7.2 Solution

Delete the record from the Infrastructure – Hosts (! 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.

11.7.3 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 – Hosts – 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:
11.8 Deleting Instances

It is important to know the difference between deleting an Instance from the Provisioning section, and deleting a VM from the Infrastructure section.

Instances are managed resources that may have one or multiple Virtual Machines associated. Since the VM’s in the Instance are managed by *morpheus*, deleting an Instance a with Virtual Machines in it will always try to delete the actual Virtual Machines.

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.
There are scenarios where deleting, or attempting to delete the associated Virtual Machines is not desired:

- The Instance needs to be deleted, but the actual Virtual Machines need to remain.
- The actual Virtual Machines have already been deleted outside of Morpheus, so only the records in Morpheus need to be removed.

### 11.8.1 Deleting an Instance without deleting Infrastructure

It is not possible to delete an Instance from the Provisioning section without removing the associated Infrastructure/VM’s. However, this can be accomplished from the Infrastructure section by deselecting “Remove Infrastructure” when deleting the VM:

1. Navigate to the Virtual Machine record by clicking on the VM’s name in the Virtual Machines section in the Instances details section, or by navigating to `Infrastructure - Hosts - Virtual Machines` and selecting the VM.
2. Click “DELETE”
3. In the delete confirmation modal:
   - Uncheck “Remove Infrastructure”
   - Check “Remove Associated Instances”

**Important:** Ensure “Remove Infrastructure” is NOT checked if you do not want to delete the actual Virtual Machine.

4. Select DELETE

This will delete the Virtual Machine record as well as the Instance record, but leave the Infrastructure/VM in place. If the VM is in a Cloud that is being inventoried, it will s
11.8.2 Deleting an Instance/VM that does not exist anymore

Deleting a managed resource outside of Morpheus is not recommended as it will leave stranded record in Morpheus and cause deleting the records in Morpheus to get stuck on delete when Morpheus tries to remove infrastructure that is no longer there.

To select an Instance and/or VM record in Morpheus for a Virtual Machine that no longer exists:

1. Navigate to the Virtual Machine record by clicking on the VM’s name in the Virtual Machines section in the Instances details section, or by navigating to Infrastructure - Hosts - Virtual Machines and selecting the VM.
2. Click “DELETE”
3. In the delete confirmation modal:
   - Uncheck “Remove Infrastructure”
   - Check “Remove Associated Instances”

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.

4. Select DELETE

The key point is when deleting an Instance, or when selecting “Remove Infrastructure” when deleting a VM record, Morpheus will always try to remove the Infrastructure. If the Infrastructure/VM no longer exists, or you do not want to remove it, simply delete from the Infrastructure section and uncheck “Remove Infrastructure”.

Note: When deleting a managed VM, if that VM is the only VM inside the associated Instance, the Associated Instance must also be removed.
11.9 Morpheus Agent Install Troubleshooting

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 Admin -> Settings over 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.

In addition to the main appliance_url in Admin -> 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.

11.9.1 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 (Admin -> Settings) is both reachable and resolvable from the provisioned node.
- The appliance_url begins with to 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 Provisioning - 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 configure, 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 Provisioning -> 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 Provisioning -> 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 custom or synced image. Credentials can be entered on images in the Provisioning -> Virtual Images section.

Cloud-Init agent install mode

• Cloud-Init is configured in Admin -> Provisioning section

• Provisioned image/blueprint has Cloud-Init (linux) or Cloudbase-Init (windows) installed
11.9.2 Manually Installing a Morpheus Agent

While it should not be necessary to manually install an agent if the requirements are met, it is possible to manually install an agent on an instance. This can also be handy when troubleshooting an agent install.

**Linux**

1. In Morpheus, go to the VM’s host detail page in Infrastructure->Hosts->Virtual Machines you will see an API Key that is unique to that host.
2. As root user, run: (replacing ${} with the relevant information)
   
   ```bash
curl -k -s "${opts.applianceUrl}/api/server-script/agentInstall?apiKey=${opts.apiKey}" | bash
   ``
3. This will pull the Morpheus Agent install script from the Morpheus appliance and run it.
4. Once the agent is installed, run `morpheus-node-ctl reconfigure` to complete the manual process.

**Windows**

- The windows agent setup can be downloaded at `https://[morpheus-appliance-url]/msi/morpheus-agent/MorpheusAgentSetup.msi`
- On the Morpheus appliance package the windows agent is located at `/var/opt/morpheus/package-repos/msi/morpheus-agent`
- WinRM, VMware Tools, or Cloudbase-Init can be used to install the agent from the Morpheus appliance
- The initial windows installer is MorpheusAgentSetup.msi
- Once the Windows agent is downloaded and installed with Morpheus AgentSetup.msi the agent is located and runs from `/Program Files x86/morpheus/morpheus Windows Agent`
- Logs can be viewed in the Event Viewer under Applications and Services Logs -> Morpheus Windows Agent

1. Replace the values for `$apiKey` and `$applianceUrl` in the script below.
2. Execute this script on the Windows box in Powershell.

```powershell
$apiKey = "add VM apiKey here"
$applianceUrl = "https://your_appliance_url.com/"
$client.DownloadFile($applianceUrl + "\msi\morpheus-agent\MorpheusAgentSetup.msi", "C:\Program Files (x86)\Common Files\MorpheusAgentSetup.msi")
Start-Sleep -Seconds 10
cd ${env:commonprogramfiles(x86)}
$serviceName = "Morpheus Windows Agent"
$serviceId = (get-wmiobject Win32_Product -Filter "Name = 'Morpheus Windows Agent'" | Format-Wide -Property IdentifyingNumber | Out-String).Trim()
cmd.exe /c "msiexec /x $serviceId /q"
} [Console]::Out.Flush()
[gc]::collect()
```
try {
    Write-VolumeCache C
} catch {

} $MSIArguments= @( 
"/i"
"MorpheusAgentSetup.msi"
"/qn"
"/norestart"
"/i+v"
"morpheus_install.log"
"apiKey=$apiKey"
"host=$applianceUrl"
"username=\LocalSystem"
"vmMode=true"
"logLevel=1"
)
$installResults = Start-Process msiexec.exe -Verb runAs -Wait -ArgumentList ...
[Console]::Out.Flush()
[gc]::collect()
try {
    Write-VolumeCache C
} catch {

} start-sleep -s 10
$attempts = 0
do {
    try {
        Get-Service $serviceName -ea silentlycontinue -ErrorVariable err
        if([string]::isNullOrEmpty($err)) {
            Break
        } else {
            start-sleep -s 10
            $attempts++
        }
    } catch {
        start-sleep -s 10
        $attempts++
    }
} while ($attempts -ne 6)
Set-Service $serviceName -starttype "automatic"
$service = Get-WmiObject -Class Win32_Service -Filter "Name='$serviceName'"
if ($service -And $service.State -ne "Running") {Restart-Service -displayname ...
exit $installResults.ExitCode

3. If the agent doesn’t install, logs can be found in the morpheus_install.log file located at C:\Program Files (x86)\Common Files\
11.9.3 Restarting the Morpheus Agent

In some situations it may be 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

You can use the following to uninstall the linux agent:

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

---

11.9.4 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.

---

11.10 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

   ```bash
   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:

   ```bash
   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:

   ```bash
   mysql -u morpheus -p -h 127.0.0.1 morpheus
   ```

   At the prompt, enter the mysql password from the application.yml

   Then run:

   ```bash
   DELETE FROM DATABASECHANGELOGLOCK;
   ```

   Then restart morpheus-ui:
11.11 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.

11.11.1 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.

11.11.2 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.

### 11.11.3 Copy and Paste

**Note:** Copy and Paste for Text is supported for SSH and RDP protocols only.

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.

11.11.4 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 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 guac is properly running:

<table>
<thead>
<tr>
<th>guacd[16899]: INFO:</th>
<th>Guacamole proxy daemon (guacd) version 0.9.14 started</th>
</tr>
</thead>
<tbody>
<tr>
<td>guacd[16899]: INFO:</td>
<td>Listening on host 127.0.0.1, port 4822</td>
</tr>
</tbody>
</table>

6. Additional information in the guacd logs appears when Morpheus is making a console connection. A successful connection will resemble:
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

11.12 Restart a Morpheus Installation

If the initial reconfigure is stopped or your installation is damaged beyond reconfiguring again, it may be necessary to start over.

On the Morpheus appliance:

1. Run morpheus-ctl cleanse
2. Remove the Morpheus package
   - deb: dpkg --purge morpheus-appliance... using the appropriate package name.
   - rpm: rpm -e (morpheus-appliance...) using the appropriate package name.
3. Then Run
   - rm -rf /etc/morpheus
   - rm -rf /var/opt/morpheus
   - rm -rf /var/run/morpheus
   - rm -rf /var/log/morpheus
   - rm -rf /opt/morpheus
4. Re-install Morpheus

   If the elasticsearch cluster is unhealthy and needs purged, run:
11.13 Unable to Delete Tenant

**Problem** When trying to delete a tenant, a message stating manage 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

11.14 Unable to Provision a Custom Image

Prior to provisioning an custom image, the image must be configured in the Provisioning -> 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

11.15 Variables

The following are the map structures passed to scripts and templates during provisioning inside of a <%= %> block. Variables can also be passed in Naming Policies using ${ } block.

Important: Variables are case sensitive

PowerShell Example: $app_id = "<%= instance.metadata.app_id %>"

Bash Example: HOSTNAME="<%= container.server.hostname %>">

Python Example: hostname = container['server']['hostname']

HTTP Body Example: {"name": "<%= instance.createdByUsername %>"}

Instance Naming Policy example: ${userInitials}-${cloudCode}-${platform == 'windows' ? 'W' : 'L'}-${sequence}

Tip: Variables can be extremely useful when utilized in the environment tab, metadata, and environment variables.

Note: customOptions are user defined as Option Types or Option Lists in custom Library items.
11.15. Variables
zone.agentMode: <%=zone.agentMode%>
zone.datacenterId: <%=zone.datacenterId%>
group.code: <%=group.code%>
group.name: <%=group.name%>
group.location: <%=group.location%>
group.datacenterId: <%=group.datacenterId%>

instance {
    instanceTypeName,
    instanceTypeCode,
    provisionType,
    instanceVersion,
    tenantSubdomain,
    plan,
    name,
    displayName,
    description,
    environmentPrefix,
    hostname,
    domainName,
    firewallEnabled,
    status,
    userStatus,
    networkLevel,
    instanceLevel,
    deployGroup,
    instanceContext,
    autoScale,
    statusMessage,
    expireDate,
    tags,
    storage,
    memory,
    cores,
    configId,
    configGroup,
    configRole
    containers:[],
    metadata:[],
    evars:[]
}

container {
    containerTypeName,
    containerTypeCode,
    containerTypeShortName,
    provisionType,
    dataPath,
    logsPath,
    configPath,
    planCode,
    dateCreated,
    status,
    environmentPrefix,
    version,
    image,
}
internalHostname,
hostname,
domainName,
storage,
memory,
cores,
internalIp,
externalIp,
sshHost,
hostMountPoint,
configId,
configGroup,
configRole,
serverId,
server: {}

server {
  serverTypeName,
  serverTypeCode,
  parentServerId,
  plan,
  visibility,
  osTypeCode,
  sourceImageId,
  name,
  displayName,
  internalName,
  category,
  description
  internalId,
  externalId,
  platform,
  platformVersion,
  agentVersion,
  nodePackageVersion,
  sshHost,
  sshPort,
  sshUsername,
  consoleType,
  consoleHost,
  consolePort,
  consoleUsername,
  internalSshUsername,
  internalIp,
  externalIp,
  osDevice,
  dataDevice,
  lvmEnabled,
  apiKey,
  softwareRaid,
  status,
  powerState,
  dateCreated,
  lastAgentUpdate,
  serverType,
osType,
commType,
managed,
agentInstalled,
toolsInstalled,
hostname,
domainName,
statusMessage,
maxStorage,
maxMemory,
maxCores,
macAddress,
serverVendor,
serverModel,
serialNumber,
tags,
configId,
configGroup,
configRole
volumes {
    name
    id
    deviceName
    maxStorage
    unitNumber
    displayOrder
    rootVolume
}
}

zone {
    name,
    code,
    location,
    cloudTypeName,
    cloudTypeCode,
    domainName,
    scalePriority,
    firewallEnabled,
    regionCode,
    agentMode,
    datacenterId
}

group {
    code,
    name,
    location,
    datacenterId
}

customOptions {
    customOptions.fieldName
}
12.1 v3.6.2

Release date: 4/9/2019

**Important:** As of v3.6.2 Ubuntu 14.04 is no longer supported for Docker Hosts OS due to Docker version upgrade to address CVE-2019-5736. Existing Morpheus Docker Hosts running on 14.04 will not be able to upgrade to latest Morpheus Agent version. All new Morpheus Docker hosts using Ubuntu require Ubuntu 16.04 or 18.04.

### 12.1.1 Highlights

**Amazon: Infrastructure Expansion**

AWS VPCs can now be created and deleted directly from the Infrastructure section with the + ADD RESOURCE POOL button in the RESOURCES tab in AWS Cloud detail pages. AWS Subnets can also be created and deleted from the Infrastructure -> Networks section. In addition, VPC’s and subnets created with CloudFormation blueprints will automatically created in Morpheus upon CloudFormation App provisioning.

Existing AWS S3 Buckets, ELB’s and ALBs now sync for AWS Clouds, and Amazon Security Groups and Rules are now listed in the Network Security Group section. Full Amazon Auto Scale Group support added too, including creation, and management of Scale Groups, as well as adding new or existing Instances to Scale Groups and managing auto scale settings.

**Task Results**

Task Results bring the ability for Tasks in a Workflow to use output from other Tasks via results variables. Result types, include json, key/val pairs and single results, can be set on script tasks and the results are available using `<%=results.taskCode%>` variables. See https://docs.morpheusdata.com/en/3.6.2/provisioning/automation/automation.html#task-results for more details.
Azure Costs & Cloud Summary Tab

Azure Cloud costs now sync nightly, reflecting actual charges from Azure rather than estimates for tracked resources. Azure Cloud detail pages now contain a Summary tab with details about the cloud config, Azure resources statistics, and a Costing section including month to date costs, estimated spend, last months costs, and a Services Breakdown for costs.

Infoblox Optimizations

Infoblox can contain a ton of records, and while Morpheus can sync all of them in, we added a few more options to allow syncing in only what is needed. The Zone Filter, like Network Filter, allows control over which zones/domains are synced using Infoblox API filters, and the new “Inventory Existing” flag in the Infoblox Integration settings allows disabling the sync of host and zone record details.

These new controls as well as additional sync optimizations allow for a reduction in network and cpu overhead when Integrating large enterprise Infoblox environments with Morpheus.

More Ansible Love

3.6.2 adds another performance improvement (up to 40%) for the Morpheus Agent Command Bus mode for Ansible Integrations, Ansible Group management for existing and brownfield Instances, live std out for Ansible executions, and additional governance role permission for Ansible Integrations.

Security

3.6.2 contains multiple upgrades to address vulnerabilities and enhance security. To address CVE-2019-5736 Docker has been upgraded to 18.09.3, which results in the removal of support for Docker Hosts running Ubuntu 14.04. All new Morpheus Docker hosts require Ubuntu 16.04 or 18.04, or CentOS/RHEL 7+. Existing Morpheus Docker Hosts running on 14.04 will not be able to upgrade to latest Morpheus Agent version.

12.1.2 New Features

- Analytics: Completely revamped Analytics section that expands available metrics, filtering and flexibility.
- Ansible: Live std out of Ansible execution added to instance history
- Ansible: Ansible Group field added to Edit Instance Modal for Instances in Groups scoped to Ansible
- Ansible: Command Bus Execution times improved up to 40%
- Ansible: “Integrations: Ansible” Role Permission added. Controls User access to Ansible Tab in Groups & Clouds, Access to Ansible Integration detail page, and ability for User to edit or delete Ansible Integrations.
- API: A Master Tenant Multitenant User Role can now be used when creating Subtenant Users
- API: Additional validation added for Instance and App creation
- API: ownerId added to /api/zones/list
- API: UUID added to Clouds, Instances, Hosts, Apps, Groups, Containers for /api/billing reference
- Appliance: ntp['enable'] = false option added for /etc/morpheus/morpheus.rb
- AWS: Application Load Balancer sync added
- AWS: Elastic Load Balancer (ELB) and Application Load Balancer (ALB) sync added in Infrastructure - Load Balancers and AWS Cloud detail pages Load Balancers tab
- AWS: Existing S3 bucket sync added in Infrastructure - Storage - Buckets
- AWS: Amazon Auto Scale Groups sync and creation added. Auto Scale Groups can be created and managed in Provisioning - Automation - Scale Thresholds, and Scale Groups can be configured during Provisioning or in the Instances Scale tab for Instances that support scaling.
- AWS: Security Group Rule syncing added, AWS Security groups now listed in Infrastructure - Network - Security Groups
- AWS: Subnet creation and deletion added to Infrastructure - Network
- AWS: VPC creation and deletion added to AWS Cloud detail pages Resources tab
- AWS: Security Groups now filter by VPC selection during provisioning
- Azure: Azure Costs sync added
- Azure: Cloud Summary tab added to Azure Cloud Detail page with cloud config summary, Azure Resources summary, and Costing section including month to date, estimated spend, Services Breakdown list and chart.
- Azure: Sync optimizations improve sync times and reduce api calls to Azure
- CloudFormation: Added “Apply State” action to apply updates to a Cloud Formation stack
- CloudFormation: VPC’s and Subnets created from CloudFormation Blueprints are now created in Morpheus
- Docker: Added support for Docker Host provisioning on Ubuntu 18.04
- Docker: Removed support for Docker Host provisioning on Ubuntu 14.04
- Docker: Updated the version of docker installed on docker hosts from 1.12.6 to 18.09.3 .. IMPORTANT:: The Docker v18.09.3 upgrade addresses CVE-2019-5736. Docker v18.09.3 does not support Ubuntu 14.04. Existing Morpheus Docker Hosts running on 14.04 will not be able to upgrade to the latest Morpheus Agent version, as the host Docker version will be upgraded to v18.09.3 upon Agent upgrade. All new Morpheus Docker hosts using
- Google Cloud: Added public image syncing
- Hosts: Windows Server option added to + BARE METAL action
- Infoblox: “Inventory Existing” flag added to enabled/disable sync of host and zone records
- Infoblox: Zone filter added to filter Zones/Domains synced from Infoblox
- Instances: New Cloning Wizard adds all of the capabilities of the Provisioning Wizard to Cloning.
- Instances: Sync restore status added
- Library: Instance Types and Layouts now filter by user role permissions
- Library: Ubuntu 18.04 VMware Layout and Node Type added to system Ubuntu Instance Type
- Load Balancers: AD Authentication added for F5 Integrations
- Operations: New Budgets section
- Plans: Group access settings added
- Policies: When scaling or adding nodes to an Instance with ${sequence} naming policy, the ${sequence} is now used for additional nodes (previously -1, -2 etc was appended)
- Policies: Workflow Policy type added
- Pricing: CAD currency added
- Provisioning: ${app.name}, ${customOptions.} and ${instance.instanceContext} variables now evaluated in review pane
• Provisioning: Cloning wizard updated to match provisioning configuration options.
• Roles: “Integrations: Ansible” permission added
• Roles: “Operations: Activity” permission added
• Roles: “Operations: Budgets” permission added
• SCVMM: Hosts selection no longer required
• SCVMM: Generation selection moved from Provisioning modals to Virtual Image settings
• SCVMM: Hypervisor Console added (beta)
• Storage: Delete confirmation modal w/ validation added when deleting Storage Buckets
• Tasks: Local Shell Script task type added w/ git repo Integration option
• Tasks: Task Results added. Allow Tasks to use the output from preceding Tasks in the same Workflow via results variables. Single Value, Key/Value Pairs, and JSON results types. Output is stored in <%=results.taskCode%> or <%=results["Task Name"]%> variables.
• Tenants: CAD currency added
• UI: “Services” tabs renamed to “Integrations”
• vCloud Director: Additional error messages displayed upon provisioning failure
• vCloud Director: Discovered server usage tracking added
• vCloud Director: Plan matching added for Discovered Servers
• Virtual Images: Ubuntu 18.04 vmdk system image added
• VMware: Windows VM auto-resolution added for Hypervisor Console
• VMware: Networks now filter by cluster during provisioning when cloud is scoped to all clusters

12.1.3 Fixes

• Ansible: Fix for running playbooks from Ansible Integrations section
• Apps: Fix for ${App.name} not showing correct value in the Review App section
• Apps: Fix for Instance configurations get reset when going back from the REVIEW page
• AWS: Fix for detecting ip address changes during provisioning due to EIP assignment
• Backups: Fix for backups associated with an Instance not moving to subtenant when the Instance is assigned to a subtenant
• Backups: Fix for recent backups failures entries and deletion
• Backups: Fix for snapshot retention when “Archive Snapshots” is disabled on target Storage Provider
• Blueprints: Fix for sub-tenants not able to set blueprint access when custom access has been set at the master tenant level for blueprints.
• CLI: Fix for morpheus groups use command error when not using morpheus shell
• CLI: Fix for morpheus instances add using Nutanix Instance Types
• CLI: Fix for missing field prompts in morpheus apps add $name –blueprints
• Commvault: Fix for Backup deletion handling
• Commvault: Fix for cloning a synced job
• Commvault: Fix for Sub-Tenants using Commvault Integration shared form Master Tenant
• Console: ctrl-alt-delete restart warning dialog no longer displayed for Windows vm’s
• GCP: Fix for image selection when creating Google Node Types
• HA: Fix for mysql read-only and auto-commit locks when using external mysql failover configurations
• Health: Updates to Appliance Health: System Memory Usage accuracy
• Hosts: Fix for converting to managed using key instead of username/password
• Instances: Fix for core count updated after reconfigured Instances that were converted to managed
• KVM: Fix for cloning kvm hosts
• NSX: Fix for creating NSX Integration from Administration - Integrations
• Nutanix: Fix for cloning Instances with 1 TB+ storage
• Nutanix: Fix for cloud-init iso upload for docker host provisioning
• Nutanix: Fix for duplicate Virtual Images
• OpenStack: Fix for reconfiguring Host and VM’s from Infrastructure - Hosts
• Option Types: source_url length increased to from 255 to 1000
• OTC & Huawei: Fix for Backups: Restore to New
• OTC: Fix for Cloning instances
• Plans & Pricing: Fix for price plans assigned to sub-tenant hidden from master tenant in ui
• Plans & Pricing: Price Sets and Price tabs now hidden for sub-tenants
• SCVMM: Fix for unattend and iso cleanup
• SCVMM: Fix for Gen 2 Provisioning issue caused by hardware profile
• Storage: ECS bucket url added to bucket details
• Tenants: Fix for Tenant delete error due to missing storage_group_storage_volume
• Tenants: Fix for Users paging issue
• UI: Fix for Hosts pagination removing filters
• vCD: /api removed from new cloud wizard placeholder text
• vCD: Fix for adding network adapter during reconfigure
• vCD: Fix for Inventory Existing when 50-100 vapps exist in target vCD
• vCD: Fix for inventory issues (instances not showing storage and aren’t getting tied to correct plans)
• VCD: Fix for stop/start Instance not updating usage records
• VCD: Fix for Discovery vDC scoping.
• VIO: Fix for Backups created in qcow2 format
• Veeam: Fix for service hidden from UI when disabled
• VMware: Fix for blank hypervisor console displaying for certain Windows VM resolutions
• VMware: Fix for restoring a VMware backup after deleting associated Instance
• VMware: Fix for issue where incorrect thin/thick provision spec was applied
12.1.4 Appliance Package Version Updates

- commons-fileupload 1.3.2
- commons-httpclient 3.1
- gettext 0.19.8.1
- jsch 0.1.53
- jackson-databind 2.8.10
- lucene-queryparser 6.6.1
- myqsl-connector-java 5.1.33
- spring-security-web 4.2.3
- spring-amqp 1.7.4
- spring-websocket 4.3.7
- spring-security-oauth2 2.0.14
- spring-core 4.3.12
- spring-web 4.3.12
- spring-webmvc 4.3.12
- spring-messaging 4.3.7
- tomcat-embed-core 8.5.23
- tomcat-util-scan 8.0.36

12.2 v3.6.1

Release date: 2/21/2019

12.2.1 New Features

- Ansible: *Built-in Cypher lookup plugin* added. "{{lookup('cypher','secret=secret/name')}}" can now be used for Cypher Secrets in Ansible playbooks natively.
- Ansible: Refresh info log added morpheus-ui current log
- Ansible: Morpheus Agent Command Bus mode speed enhancements
- API/CLI: createdBy filter added for Instances, Apps and Hosts
- API/CLI: Instances: Improvements to DELETE for Instances and Apps. If an Instance or App delete is called and the instance status is still Provisioning, resizing, restarting or cloning the delete is not attempted and an error message is returned, unless force=true
- AWS: **Costing** setting added to advanced cloud config options. AWS Costing sync can now be set to “None”, “Costing”, or “Costing and Reservations”.
- Azure: Summary tab added to Azure Cloud detail pages
12.2.2 Built-in Cypher lookup plugin

A great feature with using Ansible and Morpheus together is the built-in support for utilizing some of the services that Morpheus exposes for automation. v3.6.1 adds native support for using Cypher secrets in Ansible (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.

```yaml
- 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:

```yaml
{{ 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.

12.2.3 Fixes

- Ansible: Fix for automated Ansible install when Ansible integration is added to Morpheus.
- Ansible: Fix for execution User when Morpheus agent install is skipped
- Ansible: security fixes for when Morpheus agent is not installed
- Ansible: Updates and fixes for Morpheus Agent Command Bus mode
- Apps: Fix Delete App leaving instances stuck in removing status
- Apps: Fix for syncing pool member association
- Archives: Fix for downloading large files from Archives service stopping at 1GB. Nginx restart required `sudo morpheus-ctl restart nginx`
- AWS: Cost Explorer Reservation API calls changed to Daily
- AWS: Fix for AWS sync updates for removed vpcs
- AWS: Fix for sync errors caused by missing permissionService
- Azure: Fix for “Daily Syncing” status displayed after changing sync to Full
- Backups: Fix for backup history migration `durationMillis` and `sizeInMb` issue when upgrading from 3.4 to 3.6
- F5: Fix for synced pool member association
- Instances: Fix for flashing Instance Type icon in Instance Detail pages
- Login: Fix for “Forgot Password” subtenant user email notifications
- Nutanix: Fix for provisioning Docker hosts using IP Pools
Morpheus Documentation, Release 3.6.2

- Nutanix: Fix for provisioning Docker Hosts using Ubuntu 16 images
- Operations: Nav Dropdown Icon alignment updates
- VMware: Fix for maxStorage Service Plan filter issue

12.3 v3.6.0

Morpheus v3.6 promotes v3.5 to a LTS branch. v3.6.0 contains everything in 3.5.3, please refer to all 3.5 release notes when upgrading from the 3.4 LTS branch to v3.6.

Note: If upgrading from 3.5.2 or earlier, yum and msi agent package requests are now over port 443 instead of 80. 80 is still required for apt packages.

Important: If upgrading from 3.5.2 or earlier, nginx needs to be restarted after 3.5.3 upgrade with `morpheus-ctl restart nginx` to accommodate yum and msi agent installs over 443

Important: Amazon Cost & Reservations sync uses the AWS Cost Explorer API. Please be aware of any additional costs incurred from Amazon.

Note: Update to the latest morpheus-cli with `gem update morpheus-cli`

Release date: 2/9/19

12.3.1 New Features

- Agent: ‘New Morpheus Windows Agent Install modes’._ Morpheus Windows Agent can now install via Guest Customizations or Cloudbase-Init’._
- Agent: Morpheus Windows Agent updated to v1.4.4
- Ansible: ‘Ansible execution over command bus’._ added. Morpheus has removed the need for ssh or winrm for Ansible!
- Ansible: ‘Ansible verbose logging mode’._ added
- API & CLI: ‘Mute and Unmute Monitoring Checks now available from CLI & API’._
- API & CLI: Billing Data now contains server ID & external ID
- Appliance: Check server agent TLS support for RabbitMQ connection added
- Apps: App Tiered provisioning timeouts raised to accommodate apps that require longer workload times (15+ hours)
- ARM Templates: Added support for conditional ‘if(…)’ statements
- Automation: Added Post Provision phase output to Instance history for Workflows.
- AWS: ‘Amazon Clouds can now be scoped to all VPCs’._ in a region instead of a single VPC per cloud.
• AWS: ‘Amazon Cost & Reservations sync’ : Morpheus now syncs in AWS costs and EC2 Reservations from AWS.
• AWS: New Summary Tab added to AWS Cloud detail pages containing cost and EC2 Reservation statistics.
• Azure: Resource Pool creation and removal added
• Backups: Backup archives on failures are now automatically cleaned up
• Clouds: ‘Disable Clouds now filtered‘ in Instance, App, Host, Migration and Blueprint wizards
• Clouds: ‘Huawei Cloud integration added‘
• DISA Compliance: sudoers.d via cloud-init no longer adds users with NOPASSWD flag
• Google Cloud: Windows support added
• Google: Multi-network support added for Google Instances
• Infrastructure: VMs & Hosts: Version and Layout selection added to Convert to Managed
• Instances: Run Task Instance Action added. Allows executing individual tasks on Instances
• Instances: ‘Long Instance and Host names now wrap‘ in list view
• Isilon: ‘NFS mount path for Isilon volumes‘ displayed in Isilon tab
• KVM: Import Image: Added ability to import a KVM virtual machine image from both a managed and unmanaged KVM cloud. Supports both CEPH and LVM
• Policies: ‘New User Group Creation Policy Type‘. Automatically add User Groups via Global, Tenant, Group, Cloud and User Policies
• Policies: New ‘File Share Storage Quotas‘ and ‘Object Storage Quotas‘ policy types added.
• Policies: New Policy Icons
• Remedy: Remedy Integrations now support Approval Policies for Provisioning and Lifecycle extension Approvals in Remedy.
• Remedy: Tenant and User added to main asset
• Rubrik: SLA Domains added to instance detail backup tab and backup details
• SCVMM : Do not require cloud selection for SCVMM clouds
• SCVMM : Host Agent is now used for SCVMM communication instead of winrm
• Storage Servers: Tenancy Visibility setting added for Dell EMC Isilon and ECS Storage Servers
• TerraForm: ‘Apply State Action added to Terraform Apps‘
• TerraForm: Apply App State wizard added to reapply Terraform App state
• Terraform: HCL4j parser updated to handle 100% coverage of terraform syntax formats
• Usage: Added usage tracking for changes to Discovered resource sizes for VM’s in VMware vCenter and Nutanix Cloud types
• VMware: ‘Keyboard layout selection added to VMware vCenter Clouds‘

12.3.2 Highlights

Morpheus now syncs in actual costs from AWS, including Month To Date, estimated spend, last months costs, broken down by service, including On-Demand and Reserved hours per EC2 Instance Type.
Morpheus v3.6.0 added the ability to choose all VPC’s in an AWS region when adding an Amazon cloud. VPS’s are listed in the Resources tab in an Amazon Cloud detail page and similar to other resource pools, a VPC has Group Access, Tenant Access, Service Plan Access, default and active configuration options. This allows users to add a single aws cloud per region and then carve up access across groups and tenants and limit which Service Plans/EC2 Instance Types can be access per VPC. Users with access to multiple VPC’s can choose target VPC at provision time.
Morpheus has removed the need for ssh or winrm for Ansible! The new Ansible execution over command bus feature allows running ansible over the Morpheus agent command bus, removing the need for an ssh or winrm connection for Ansible, or any open incoming ports on the target host. The Morpheus Agent (which can be installed with no open incoming ports on the target vm/host as well) will pull and execute Ansible commands locally.

To enable Ansible execution over the Morpheus agent command bus, edit you Ansible integration and select ” Use Morpheus Agent Command Bus”
When using Ansible execution over the Morpheus Agent command bus, verbose debug logging can be enabled but editing an Ansible Integration. Along with the “Use Morpheus Agent Command Bus” flag, “Enable Verbose Logging” is a new setting on Ansible integration settings that will output debug level logs for Ansible execution, visible in the Instance history section by selection the “i” bubble on an executed Ansible task.

The new File Share Storage Quotas Policy type allows, you guessed it, setting Storage Quotas for File Shares.

To add, in Administration -> Policies add a new Policy, and select File Share Storage Quota
Then set the Name, give it a good description, set the size in GB, and scope it Globally or to a User.

The new Object Storage Quotas Policy type allows, you guessed it, setting Storage Quotas for Object Stores. To add, in Administration -> Policies add a new Policy, and select Object Storage Quota.

Then set the Name, give it a good description, set the size in GB, and scope it Globally or to a User.

Previously, deselecting “Enabled” in a Clouds configuration settings only disabled sync and greyed it out in the Clouds list view. Now disabled clouds are filtered out/hidden in Instance, App, Host, Migration and Blueprint wizards.

Note: If you disable a cloud to temporarily pause sync, be sure to re-enable it for provisioning.

Instance and Host list pages now wrap long Instance and Host names, allowing for viewing the entirety of a long Instance or Host name without having to select or hover.

Huawei Cloud now available as a Cloud integration in Morpheus v3.6.0.

The path to the file share is now displayed on the Isilon tab to make it easier for users to know how to mount the share in a VM.

Usage: morpheus monitor-checks mute [name]
Usage: morpheus monitor-groups mute [name]
Usage: morpheus monitor-apps mute [name]
Mute a check, check group, or monitoring app. This prevents it from creating new incidents.
[name] is required. This is the name or id of a check.
--disable can be passed to disable mute state instead, the same as unmute

Usage: morpheus monitor-checks mute-all
Usage: morpheus monitor-groups mute-all
Usage: morpheus monitor-apps mute-all

Mute all checks. This prevents the creation new incidents.
--disable can be passed to disable mute state instead, the same as unmute

Usage: morpheus monitor-incidents mute-all

Mute all incidents.

Usage: morpheus monitor-checks unmute [name]

Unmute a check.
[name] is required. This is the name or id of a check.

3.6.0 adds the ability to execute Windows Agent install script with vmware guest customizations or cloudbase-Init utilizing unattend.xml. Now, in addition to winrm and vmtools, morpheus can inject the agent install script into the unattend.xml, reducing vmware provision times and removing winrm/5985 requirement for agent install when using images configured with cloudbase-init.

Note: Agent Install mode must be set to “Cloud-init when available” on target cloud for cloudbase-init agent install.

Automatically add User Groups via Global, Tenant, Group, Cloud and User Policies

Note: User groups can be configured in Administration- Users- User Groups tab or in the CLI with user-groups add/update

New Apply State action brings up the new Apply App State wizard to re-apply state to TerraForm Apps.

Keyboard layouts for Remote Console connections cannot be set in VMware vCenter cloud in the Cloud configuration.

12.3.3 Fixes & other updates

• Active Directory: Fix for user accounts locking during login because of perceived failed login attempts
• Administration: Fix for tenant delete issue when a master tenant service plan is assigned to the sub-tenant
• API/CLI: Fix for API processes history for appId query
• API/CLI: Fix for API/CLI network creation
• API/CLI: Fix for Create App not handling JSON parameters (description & site ID)
• API/CLI: Fix for creating a restart task on CLI
• API/CLI: Fix for creating a security group rule not persisting instanceTypeId
• API/CLI: Fix for history API query filters not working
• API/CLI: Fix for updating cloud access settings in roles
• API/CLI: Fix for updating user role multitenant flag
• API/CLI: Fix for Morpheus push API checks
• Apps: Fix for instance configuration getting reset when using “previous” in the app wizard
• Archive: Fix for Morpheus user session timeout during large file upload
• Backups: Updated backup message in Admin - Backups when backups are disabled
• Blueprint: Fix for exposed ports not being lockable
• Blueprint: Fix for tier order display in multi tier blueprint
• Blueprint: Fix for volume size field responsiveness
• Clouds: Fix for AWS security group subtenant visibility
• Clouds: Fix for Open Telekom Cloud router & network creation
• Clouds: Fix for Morpheus IP-pool ranges being deleted when a cloud with associated network is deleted
• Custom Library: Fix for instance wizard not allowing a custom library item for oracleVM
• Custom Library: Fix for Sub-Tenants role permissions for custom instance types when tenant role Instance Type Access is set to Global:Custom
• Groups: Fix for User Group deletion when user group is associated with existing instances
• Hosts: Fix for Docker & Kubernetes hosts not deployable on VIO
• Hosts: Fix for Ubuntu Docker hosts DNS name servers being removed on reboot
• Images: Fix for GPS images incorrect labeling
• Images: Fix for images being larger than selected plan
• Images: Fix for seeded CentOS 7.2 & 7.3 Docker system images
• Instances Fix for Plan/Resource change detected when altered on cloud (not creating new usage records)
• Instances: Fix for multi-select convert to managed not having group selection
• Instances: Fix for not being able to clone Windows instance
• Instances: Fix for reconfiguring VM/Host not adding new usage record
• Instances: Fix for start/stop service on instance list showing when not applicable
• Instances: Fix for sub nav bar not appearing when clicking app link in instance detail page
• Load Balancer: Fix for AVI fields missing in LB provisioning wizard
• Networking: Fix for not being able to create NSX edge gateway
• Nutanix: Fix for Windows hostnames being counted/truncated
• OCI: Fix for regions not working properly
• Openstack: Fix for wrong plan tied being to instance
• Pricing: Fix for changes in machine configuration not being recorded in billing
• Pricing: Fix for cloud not showing in pricing label
• Pricing: Fix for hourly plan inconsistency
• Provisioning: Cloudbase-init: Fix for user creation via Cloudbase-init
• Provisioning: Fix for Windows VM turning status green prematurely
• Reporting: Fix for cost reports currency conversion
• Reporting: Fix Group and Tenant Cost reports both using the Term Zone instead of Cloud
• SAML: Fix for subtenant SAML Logout redirecting to error page
• Scaling: Fix for scale cloud priority settings, additional help text added
• SCVMM: Fix for SCVMM provisioning failing during failover cluster setup
• SCVMM: Fix for SVCMM cloud sync datastore cache
• Storage: Fix for storage showing wrong datastore
• vCD: Fix for cloud-init iso file cleanup
• vCD: Fix for custom vCD instance types not available for selection on convert to managed
• vCD: Fix for discovered & converted VM’s not creating an instance type
• vCD: Fix for hostname truncated with container ID
• vCD: Fix for issue with large number of templates
• vCD: Fix for vCD guest customizations running after instance restart triggered
• vCD: Fix for Windows instances appearing as discovered when provisioned onto vCD
• vCD: Fix for Morpheus triggering a power off instead of graceful shutdown for vApps
• vCD: vmId and vappId sent to their proper homes
• Veeam: Fix for Actions -> Backup on an instance executing the entire job
• Veeam: Fix for removing existing backup job removing other jobs
• VIO: Fix for not being able to delete instance with attached load balancer
• VMware: Fix for 8th additional disk using SCSI 0:7
• VMware: Fix for reconfigure not respecting cloud thick/thin disk setting
• VMware: Fix for record removal when changing Cloud Cluster scope from ALL to a single cluster
• VMware: Reconfigure now warns about requirement to delete snapshots
• VMware: Sync now updates cloud association when a VM is migrated between clusters scoped to multiple clouds.
• VMware: Sync now updates volume info for volumes that no longer exist on discovered vms

12.4 v3.5.3

Release date: 12/13/18

Note: yum and msi agent installs are now over port 443
Important: Nginx needs to be restarted after 3.5.3 upgrade with `morpheus-ctl restart nginx`

### 12.4.1 New Features

- API: Added externalId for vm’s
- API: Added user setting endpoints
- API: Blueprint Group & Tenant Access
- API: Customizing Blueprint Access
- API: Instance history added
- API: Provision Instances/Apps using Workflow names
- API: Track usage for Discovered VM’s
- API: Update host ssh password
- Appliance: ElasticSearch: Added support for ElasticSearch TLS connections
- Appliance: Global general performance improvement
- Appliance: Increase timeout for linux tasks
- Appliance: RabbitMQ: STOMP SSL support added
- Apps: Added App execution aborting
- Apps: Added review tab to app wizard
- Automation: Ansible: Morpheus Variables support added. containerScriptConfig map now sent to an extraVars.yml for Ansible use
- Backups: Commvault: Refactored Commvault integration
- Backups: Rubrik integration added with SLA Domain sync, Backup creation and restore.
- Backups: Rubrik Integration added.
- Backups: Zerto improvements
- Blueprints: Permissions section with Group Access and Tenant visibility added
- Clouds: Nutanix: CVM’s now shown as Hosts
- Clouds: Nutanix: V2 & V3 API stats for discovered VMs, Image Sync
- Clouds: Nutanix: Windows: Domain Join added to unattend.xml
- Clouds: Nutanix: Windows: License application added to unattend.xml
- Clouds: OpenStack: Added Network type choices for OpenStack SDN
- Clouds: OpenStack: LBaaSv2 API endpoints for VIO added
- Clouds: Openstack: Specify Floating IP option added
- Clouds: OTC: Added support for availability zones
- Clouds: SCVMM: Static IP’s now injected in unattend.xml
- Clouds: SCVMM: Windows: Domain Join added to unattend.xml
- Clouds: SCVMM: Windows: License application added to unattend.xml
• Clouds: Status in Clouds list view now shown as disabled and greyed out when a Cloud is not enabled
• Clouds: VMware: Hypervisor Console: WMKS console added, replacing VNC. GDB Server port requirement on ESXi hosts is not required for WMKS.
• Load Balancers: F5: SSL profile creation added
• Infrastructure: Custom Instance Types can now be used on Convert To Managed
• Infrastructure: Renamed “Unmanaged” to “Discovered”
• Infrastructure: Service Plan can now be specified on Convert to Managed
• ITSM: New Cherwell Integration
• ITSM: New Remedy Integration
• ITSM: ServiceNow: CMDB update CI on status change
• Library: “Supports Convert To Managed” flag added to Layouts
• Operations: Activity: New History section added with active processes and process history
• Operations: Health: Added info message for single node Elasticsearch health
• Operations: Usage: Added Type Filter with Container, Host and Discovered options
• Provisioning: Auto-truncation of Windows hostnames if specified hostname is over 15 character limit. If truncated name matches existing hostname sequence added.
• Provisioning: Morpheus Agent msi and yum packages now transferred over 443. 80 still required for deb agents
• Provisioning: Support Deployments enabled on System MySQL Instance Type
• Provisioning: Windows: Agent install optimizations, speed improvements
• Storage: Dell EMC Isilon: Create exports to allow access from ip addresses
• Storage: Dell EMC Isilon: Create new NFS shares
• Storage: Dell EMC Isilon: Create storage providers from an Isilon volume
• Storage: Dell EMC Isilon: Manage existing NFS shares
• Storage: Dell EMC Isilon: Sync NFS volume shares to Morpheus
• Virtual Images: Prevent form autofill for username/password

12.4.2 Fixes

• Administration: Fix for deleting users that have history records in Archives
• Administration: Fix for external smtp settings requiring username
• Administration: Fix for saving white label settings in subtenants
• Administration: Fix for Windows passwords with ampersand
• Analytics: Fix for utilization time filters not returning data
• Ansible Tower: Fix for “Limit to Instance” flag
• Ansible Tower: Fix for Ansible Tower groups not set to mandatory
• Ansible Tower: Fix for Ansible Tower integration details search
• Ansible Tower: Fix for editing an Ansible Tower in Provisioning: Automation: Services
• Ansible: Fix for Ansible workflow execution fails as no hosts are found
• Ansible: Fix for intermittent git lock issue
• Ansible: Security Updates
• Appliance: Installer: Fix for :9200 being appended in elasticsearch.yml unicast hosts array
• Apps: Fix for App export creating extra configs when group is defined
• Apps: Fix for variables in Instance names not evaluated in App Tier view
• Automation: Fix for Automation Tasks search not working beyond first page
• Azure: Scale Sets: Fix for missed deployments when scaling multiple nodes
• Blueprints: Fix for blueprint export including name and templateName
• Blueprints: Fix for incorrect price data displayed for custom plans
• CLI: Fix for sub-tenant user invalid login counter
• Clouds: Fix for display of warning message when attempting to delete a cloud with existing managed vm’s
• Clouds: Fix for Instance usage record creation for Convert to Managed
• Dell ECS: Fix for ECS Bucket Edit unknown error on Bucket Name Update
• ESXI: Fix for ESXI Docker Host not using LVM on 2 disk Hosts
• Git: Fix for auto appending of .git to git url in Git Integrations
• Global Search: Fix for partial Instance Name search
• Groups: Fix for add clouds to group buttons hidden if no public clouds are enabled in Administraiton - Settings
• Instances: Fix for instance clone not respecting cloud selection
• Instances: Fix for instance details auto refresh resetting history pagination
• Integrations: Fix for Integration: Edit dialog clearing host, username, & password fields on authentication failure
• Integrations: Fix for issue with display of Group & Cloud Integration scoping
• Load Balancers: F5: Fix for adding F5 Load balancer with SSL configured from instance scale tab
• Load Balancers: F5: Fix for editing F5 Load balancer settings from instance scale tab
• Logs: Fix for date filters
• Logs: Fix for log availability timeframe setting not applying
• Logs: Fix for page size when using log level filter
• Networking: Security Groups: Fix for ICMP rule requiring port
• Nutanix: Fix for plan assignment on Instances and associated VM’s for convert to managed
• Nutanix: Fix for Windows unattend.xml DNS settings
• Nutanix: Windows unattend.xml forced flag corrected to force
• OTC: Fix for OTC provisioning not surfacing failures
• Pricing: Fix for OracleVM plans issue with incorrect cores
• Pricing: Fix for price comparison missing some expected cloud prices
• Pricing: Fix for tenant Plan visibility
• Provisioning: /morpheus directory perms created by Agent Install now set to drwxrwxr-x
• Provisioning: Fix for Load Balancer not retaining settings in instance wizard
• Provisioning: Fix for overzealous enforcement of min ram setting on custom Plans
• Provisioning: Fix for scaling down not cleaning up environment variables
• SCVMM: Fix for Resource Pool input styling
• SCVMM: Fix for Network Group Validation error
• SCVMM: Fix for VHD/VHDX in node type dropdown
• Storage: Fix for storage bucket/shares input validation
• Tenants: Fix for inability to delete tenants with IP pools in use
• UCS: Fix for UCS cloud credentials not working on add, only edit
• User Settings: Fix for Default Cloud selection displaying clouds in groups not accessible to user
• Users: Fix for Bearer token expiration
• VCD: Fix for provisioning when using a synced Image from non-morpheus generated vCD catalogue.
• VCD: Fix for uploading Images over 1 GB
• VCD: Network DNS settings now updated upon cloud sync
• VCD: Removed Hostname value in discovered vm’s detail view. This was set to machine name previously as Hostname is not avilable via vcd cloud sync.
• Veeam: Fix for Veeam backup restores not working for existing jobs
• VMware: Fix for VMware Docker Host creation when cloud is scoped to a Resource Pool
• VMware: Fix for VMware reconfigure listing duplicate storage controllers
• Workflows: Fix for executing workflows on Hosts

12.5 v3.4.4

Release date: 11/2018

12.5.1 New Features

• Currencies: Rand currency added
• Administration: Inactive option added for Tenants
• Load Balancers: Allow host selection when creating HAProxy LB
• Hosts: External ID added to Host, VM and Bare Metal detail views
• Nutanix: Hypervisor Hosts sync added, improved statistics for inventoried Hosts and VM’s
• API: externalId added to Hosts
• License: Applying Windows Licenses support added to Amazon, Azure, Alibaba, Nutanix and Openstack Clouds
12.5.2 Fixes

- **Hosts:** SSH Username corrected to SSH Password when adding Unmanaged Linux Vm
- **Forgot Password Email:** Email subject more indicative of password reset, not account lock
- **Remote Console:** Fix for Console issue when instance has multiple nodes
- **Global Search:** Fix for User results missing from Quick Results
- **VMware:** Fix for Resize issue with add/remove volumes when disk type selection disabled
- **Hosts:** Resolved toggle for “Manage Internal Firewall” on Edit Host to reset all acls on save.
- **Provisioning:** Issue evaluating variables in instance name while using copies
- **Provisioning:** Fix for not showing error icon for all invalid fields
- **Provisioning:** Fix for Service Plans not filtered by Min Ram setting for “VMware” Instance Type
- **Docker:** Fix for intermittent LVM volume creation issue
- **User Details:** Fix for updating User Settings changing password on save when passwords were not updated.
- **Instance List:** Fix for Instance Category icons not filtering the Instances List
- **Search:** Search bar displayed now after 250ms hover
- **User Impersonation:** Fix for impersonating when duplicate usernames exist across Tenants
- **Load Balancers:** F5: Fix for issues with edit pool
- **Load Balancers:** F5: Monitor - parent monitor not working with custom monitors
- **Load Balancers:** F5: Fix for error when saving with no port number defined when >=2 members
- **Load Balancers:** F5: Fix for description not being set for pool
- **Load Balancers:** F5: Fix for editing F5 Load balancer settings from instance scale tab
- **Load Balancers:** F5: Fix for adding F5 Load balancer with SSL configured from instance scale tab
- **Infoblox:** Fix for issues with pool and domain records
- **Nutanix:** Windows Computer Name now set in sysprep unattend.xml
- **Instances:** /morpheus directory perms created by Agent Install now set to drwxrwxr-x
- **Users:** Fix for deleting a user with Archives history records
- **Instance details:** Fix for Price data refreshing to inaccurate value
- **Provisioning:** Fix for overzealous enforcement of min ram setting on custom Plans
- **Whitelabeling:** Fix for issue saving sub-tenant Whitelabel setting
- **Tenants:** Fix for sub-tenant created plans names visible in other Tenants when creating Docker Hosts
- **Blueprints:** Fix for incorrect price data displayed for custom plans
- **OracleVM:** Fix for provisioning using deprecated CPU value rather than Cores value
- **Instances:** Clone: Fix for Cloud selection when cloning
- **VCD:** Fix for provisioning when using a synced Image from non-morpheus generated vCD catalogue.
- **Config Management:** Fix for Config Management scoping sometimes not visible when editing a Group or Cloud
- **UCS Cloud:** Fix for credentials not saving and add cloud
- **VMware:** Fix for duplicate storage controllers on reconfigure
• Nutanix: Fix for Windows static IP assignment via sysprep unattend.xml
• Nutanix: Fix for Windows DNS settings via sysprep unattend.xml
• Provisioning: Price Comparison: Fix for missing Azure price comparisons
• Scaling: Fix for scaling down not cleaning up environment variables
• Nutanix: Fix for sysprep unattend.xml typo in powershell cmdlets
• VCD: Fix for VCD IP Pools Network DNS Servers changes updating on cloud sync
• Administration: Settings: Fix for external SMTP server username requirement (now not required)
• Automation: Tasks and Workflows: Fix for search scope limited to 1st page
• Git Integration: Fix for .git being appending to git urls
• Provisioning: Apps: Fix for extra definitions in App export json/yaml
• Groups: Fix for Add Cloud to Group options hidden if no Public Clouds are enabled
• Usage: Fix for usage records not being generated upon “Convert to Managed”
• VCD: Fix for uploading images over 1 GB
• VCD: Removed Hostname value in Unmanaged Host detail view. This was set to machine name previously as Hostname not available via sync.
• Administration: Fix for deleting Tenants with existing IP Pools
• ESXi: Docker Host Provisioning: Fix for 2 disk hosts not using LVM
• User settings: Default cloud selection now only lists user accessible clouds
• Reports: Fix for Reports permissions
• Installer: Fix for :9200 being appended in elasticsearch.yml unicast hosts array

12.6 v3.5.2

Release date: 10/22/2018

12.6.1 New Features

• Administration: Inactive option added for Tenants
• Ansible: Run output now always displayed in Instance history
• API/CLI: Storage Providers renamed to Storage Buckets
• API/CLI: Update instance metadata and power schedule added
• Appliance: Tomcat upgraded to v9.0.12, addresses CVE-2018-11784
• Apps: App Loading performance improvements
• Automation: Provisioning Workflow and Script output always added to Instance history
• Automation: Execute Options added to Tasks w/ Retryable Flag, Retry Count and Retry Delay settings
• Automation: Tasks sets set on “Post Provision” phase now run when executing workflows post-provision
• Automation: Workflows will now stop running on a task failure
• Blueprints: AWS CloudFormation support added
• Clouds: Virtustream integration added
• Commvault: Integration expanded including sync, instance support, job management and restores.
• Currencies: Rand currency added
• Hosts: External ID added to Host, VM and Bare Metal detail views
• Infoblox: Sync optimizations
• Instances: Owner assignment added to Instances
• Load Balancers: Allow host selection when creating HAProxy LB
• Logs: Enhanced logs display and formatting
• License: Applying Windows Licenses support added to Amazon, Azure, Alibaba, Nutanix and Openstack Clouds
• Network: External ID added to Network detail view
• Nutanix: Hypervisor Console added
• Openstack: Hypervisor Console added
• Operations: Morpheus Health section added with Health, Alerts and Appliance Logs
• Provisioning: Add timezone option to Provisioning wizard
• Roles: Operations: Health, Provisioning: Blueprints - CloudFormation Feature Access permissions added
• SCVMM: Multi host and datastore support added
• Storage: Dell EMC ECS Integration added
• Storage: Dell EMC Isilon Integration added
• Storage: NFS volume share sync added
• Tenants: Message added to Users section when no multi-tenant User Roles exist
• VCD: CentOS Image and Instance type added
• VCD: Debian image and Instance type added
• VCD: IP Pool allocated IPs sync added
• VCD: Ubuntu 16.04 layout
• Xen: Hypervisor Console added

12.6.2 Fixes

• API: Fix for /api/check-types call fails w/ unable to find API endpoint error
• API: Fix for /api/key-pairs call triggering 403 error
• API: Fix for /api/certificates call triggering 403 error
• API: Fix for /api/instances/:id call resulting in http 500 error if the Instance has any backups
• App Wizard: Fix for app validation issue with service plan ranges
• App Wizard: Fix for blueprint search when no config exists in Blueprint
• Clouds: Fix for Cloud Timezone setting not being respected for IBM Cloud, UpCloud and Softlayer
• Docker: Fix for intermittent LVM volume creation issue
• Forgot Password Email: Email subject more indicative of password reset, not account lock
• Global Search: Fix for User results missing from Quick Results
• Hosts: Resolved toggle for “Manage Internal Firewall” on Edit Host to reset all acls on save.
• Hosts: SSH Username corrected to SSH Password when adding Unmanaged Linux Vm
• Infoblox: Fix for issues with pool and domain records
• Instance List: Fix for Instance Category icons not filtering the Instances List
• Load Balancers: F5: Fix for description not being set for pool
• Load Balancers: F5: Fix for error when saving with no port number defined when >=2 members
• Load Balancers: F5: Fix for issues with edit pool
• Load Balancers: F5: Monitor - parent monitor not working with custom monitors
• Load Balancers: Fix for LB’s not getting created when provisioned via app wizard
• Nutanix: Fix for applying Static IP’s via unattend.xml
• OTC: Local Disk provision option removed (not supported)
• Price Policy: Provisioning Instances now counted towards policy
• Pricing: CPU Only price type removed
• Provisioning: Fix for not showing error icon for all invalid fields
• Provisioning: Fix for Service Plans not filtered for “VMware” Instance Type
• Remote Console: Fix for Console issue when instance has multiple nodes
• Search: Search bar displayed now after 250ms hover
• Tenants: Tenant Limits removed, replaced by Tenant Policies
• User Details: Fix for updating User Settings changing password on save when passwords were not updated.
• User Impersonation: not properly handling duplicate usernames
• VCD: Fix for Cloud sync not syncing all VMs
• Veeam: Fix for Veeam Service legacy data (pre 3.5.0) causing backups servers not appearing
• VMware: Fix for Resize issue with add/remove volumes when disk type selection disabled

12.7 v3.4.3

Release date: 9/21/2018

12.7.1 New Features

• Appliance: Reduced memory database query overhead for agent comms
• VMware: API session management improvements
• VMware: Massive cloud sync speed improvements
12.7.2 Fixes

- AWS: Fix for io volume type selection
- Azure: Fix for price plans syncing
- Docker Hosts: Total Storage value fix
- Docker Hosts: Total Storage value fix
- ESXi: Fix for file cleanup on failed provisions
- Guidance: Improved core count recommendations
- Identity Sources: Fix for form resetting back to LDAP on unsuccessful save
- KVM: Fix for Backup Restore for multi-disk VMs
- KVM: Fix for Ubuntu 14 backups
- Load Balancers: AVI: Fix for edit instance lb setup
- Nutanix: Fix for automated Domain joins
- OTC: Fix for Reconfigures
- Policies: Fix for Max VMs Policy enforcement
- Policies: Unable to edit policy assigned to tenant
- Reports: Fix for 500 error when report generated by deleted user exists
- Scaling: Fix for NGINX autoscaling
- Scaling: Fix for Scaling Schedules
- User Settings: Fix for unable to set user settings when connected to AD identity source.
- Venafi: Fix for deleting integration
- Venafi: Fix for deleting integration

12.8 v3.5.1

Release date: 9/21/2018

12.8.1 New Features

- API: Services - Cypher added
- API: User creation password Confirmation requirement removed
- Appliance: Reduced memory database query overhead for agent comms
- Backups: Avamar Integration added
- Bluemix renamed to IBM Cloud
- Infoblox: Variable support for ext attributes added
- Instances: Metadata fields added to Edit Instance modal
- Instances: Post-Provision Metadata and Tags editing
- Morpheus Agent: Ubuntu 17.10 support added
• Openstack: Additional support for versioned endpoints added
• Openstack: Octavia Load Balancer Support added
• Openstack: Queens support added
• Option Lists: Authentication added for remote data providers
• Policies - Policies section added to Administration for managing all Policies.
• Policies: Budget Policy type Added
• Policies: Global and User policy levels added
• PXE: webHost in answer files now forced to http
• Reports: Workload Summary Report added
• Softlayer: Washington 6 region added
• Storage: Storage Browser for NFS, Azure & Alibaba provider types added
• Tenants: Disable option added for sub-tenants
• VCD: support for vcd v9
• VCD: Virtual Machine sync now includes undeployed VMs
• vCloud Director: Support for v9.x added
• Veeam: Service detail page added
• VMware: API session management improvements
• VMware: Massive cloud sync speed improvements

12.8.2 Fixes

• Ansible Tower: Fix for Edit Integration
• Ansible Tower: Fix for removing Ansible Tower Integration from Group or Cloud
• Approvals: Fix for VM state updating on Instances list page after request is rejected or cancelled
• Azure: Fix for price plans syncing
• Backup: AWS backup shows in progress when snapshot limit exceeded
• Backups: Default Backup Schedule not applied
• Docker Hosts: Total Storage value fix
• ESXi: Fix for file cleanup on failed provisions
• Guidance: Fix for resize action
• Guidance: Improved core count recommendations
• Identity Sources: Fix for form resetting back to LDAP on unsuccessful save
• Identity Sources: Fix for Required Group still applying after removal
• Instances: Clone: Fix for Cloud selection not being respected
• Instances: Fix for creating Instance with numbers-only names
• KVM: Fix for Backup Restore for multi-disk VMs
• KVM: Fix for Ubuntu 14 backups
• LB: AVI - SSL shows ‘Select’ instead of ‘No SSL’
• Library: Layouts: “Min Ram” not enforced when using custom plans
• Library: Option Lists: Fix for option list with failed validation still creating records
• Library: Scripts: Fix for WindowsFeature "$Feature$Number" causing scripts to be un-editable
• Load Balancers: AVI: Fix for edit instance lb setup
• Load Balancers: AVI: Fix for manually adding Virtual Server
• Localizations: added message for Venafi delete
• Networks: Fix for deleting an ACI Integration
• Nutanix: Fix for automated Domain joins
• OTC: Docker Host setup fails to complete
• OTC: Fix for Reconfigures
• Policies: Fix for issue with Power Schedule Tenant permissions
• Policies: Fix for Max VMs Policy enforcement
• Policies: Unable to edit policy assigned to tenant
• Pricing: Fix for Azure price discrepancy
• Provisioning: Fix for Cores per socket resetting back to 1 when going back in provisioning form
• Provisioning: Fix for Price comparison modal for custom images
• PXE: Fix for deleting kickstart files for subtenants
• Reports: Fix for 500 error when report generated by deleted user exists
• Scaling: Fix for NGINX autoscaling
• Scaling: Fix for Scaling Schedules
• Subtenants: Removed invalid link to Shared Clouds on Instance Details page
• User Settings: Fix for unable to set user settings when connected to AD identity source.
• User Settings: Username uniqueness now scoped to Tenants
• vCloud Director: Fix for sync error when a vm has no disks
• Veeam: Fix for Legacy referenceData causes backupServer dropdown to not load
• Venafi: Fix for deleting integration
• Venafi: Fix for incorrect subject set when applying cert during provisioning
• Virtual Images: Removed “Convert” option for synced images

12.9 v3.4.2

Release date: 8/23/2018
12.9.1 New Features

- Amazon: Additional service plans available
- API & CLI: Instances metadata added
- AWS: China cn-north-1/Beijing and cn-northwest-1/Ningxia regions added
- Infoblox: Extended attributes for A and PTR records added
- Infoblox: Script variable support added for extended attributes
- Instances: Custom exposed ports are now globally available
- KVM: VM Actions, MultiDisk, Reconfigure, Apps & Blueprints. Backups/Restore added
- Morpheus Agent: Ubuntu 17.10 support added
- PXE: Now force webHost to http for answer files
- Xen: CUSTOM PORT option added to Xen Cloud Config

12.9.2 Fixes

- ACI: Fix for deleting an ACI integration
- Active Directory: Fix for removing Required Group after one is added
- Approvals: VM status now moves from pending to cancelled/rejected on cancelled/rejected action from approval screen
- AVI: multiple fixes and improvements
- Azure: Fix for discrepancy in the prices shown in Morpheus for Azure prices
- Evars: Fix for environment variables/deployment option values longer than 255 result in 500 error
- Guidance: Fix for resize action on recommendations
- Image builder: fix for config issue with plans when custom cpu is enabled
- Instances: Cloning: Fix for cloud selection
- LB: AVI: Create Pool - fix for blank monitors field
- LB: AVI: Nodes: Fix for empty Monitor column
- LB: Edit AVI - avi cloud field not populating
- Load Balancers: Fix for Load Balancer for failing to save when multiple ports are available but only on 1 port is configured
- Monitoring: Fix for updating existing App Checks causing a 500 error
- Oracle Cloud: Fix for Virtual Image Icon missing
- Provisioning: Fix for creating instances with names containing only numbers
- Provisioning: Price comparisons: Fix for custom images not displaying Price comparisons
- PXE: Fix for subtenant kickstart file deletion
- Scale Schedules: Fix for schedule execution
- Seed: Fix for hasNodeMonitors seed error
- Subtenants: Removed hyperlink on Cloud Name for inaccessible clouds that would result in 500 error
• Unmanaged VMs: Fix for cores not updated on refresh
• User Settings: User name uniqueness now scoped to Tenant
• VCD: Fix for Guest Customization for Windows failing to run.
• Venafi: Fix for Add Integration from Administration section
• Venafi: Fix for Details page error
• Venafi: Fix for subject on certs added during provisioning
• Venafi: Fix for viewing Certificates
• Venafi: Instance Settings: Fix for edit cert
• Virtual Images: Convert option removed for synced images
• Virtual Images: Fix for deleting failed conversions

12.10 v3.5.0

Release date: 7/31/2018

12.10.1 New Features

• Administration: Added support link white label
• Analytics: Added Cost by Tenant report
• Analytics: Added Cost by Group report
• Analytics: Added Cost by Application report
• Analytics: Added Cost by Instance report
• Analytics: Added Cost by Cloud report
• Analytics: Added Tenant Instance Usage report
• Analytics: Added metadata filter to reports
• Ansible: Added Ansible Tower integration
• API/CLI: Workflows now support custom option types
• API/CLI: DataStores added
• API/CLI: Subtenant Groups added
• API/CLI: Added “taskPhase” for workflow tasks
• API/CLI: Added CLI login via token
• API/CLI: Added optimization for virtual image upload
• API/CLI: Added metadata for instances
• Backups: Direct Stream to Storage Provider added for VMware, Xen and KVM backups and Image imports.
• Backups refactored with Schedules, Job, Types, and Services
• Backups: Zerto Integration added
• Multicloud object storage backup and archival
• Blueprints: Added ACI fields to Blueprint and App wizards
• Cloud Foundry: Added inventory of applications during cloud sync
• Cloud Foundry: Added support scaling
• Cloud Foundry: Added sync for build packs
• Cloud Foundry: Added custom route
• Plans & Pricing: Prices: Datastore Price Type added
• KVM: Added KVM migration workflow
• KVM: Added CEPH storage support
• KVM: Backup/restore is now possible for local datastores and LVM
• Library: Debian 9.4 qcow2 image added
• Instances: Backup type, provider, job, latest, next, added to instance detail page
• Network: VMware NSX Integration added
• Option Lists: Tenant visibility added to Option Lists
• Option Lists: Added realtime sync option
• Oracle: Added support for different storage types
• Oracle: Added Oracle Cloud library nodes
• Oracle: Added ability to upload a local image and provision to Oracle Cloud
• phpIPAM: Added additional details (description, hostname, owner)
• Variables: Added tenantSubdomain variable
• Policies: Backup Creation policy type added
• Policies: User Creation policy type added
• Infrastructure: Hosts: Added host OS, Type and Status filters
• Plans & Pricing: Added Custom Ranges to Service Plans for min/max storage, memory and cores
• Provisioning: Added ability to toggle between MB and GB for memory settings in provisioning wizard
• Roles: Added Global Access: Custom option to roles
• SAML: Added response validation
• ServiceNow: Added credential validation
• Storage: Renamed “Object Stores” to “Providers”
• Storage Providers: Added details page that allows you to browse, upload and delete files
• Storage Providers: Verify permissions to path on save added
• Storage Providers: Removed option for Tenants to create local storage providers
• Storage Providers: Retention policy added
• Storage Providers: Amazon S3: Added Create Bucket option
• Users: Subtenant users will no longer be able to login from the main login page without specifying their subdomain
• Users: Added API token generation for users via UI
- Virtual Images: Default “force guest customizations” to “on”
- VMware: Added VMware “Clone to Template” Action
- VMware: Direct stream Backups added
- XenServer: Backups don’t offload backup

### 12.10.2 Fixes

- Fix for Chef detail section not displaying in app wizard
- Fix for deployment failure with ARM template copies
- Fix for Blueprints: broken image for library items with no custom icon
- Fix for Analytics: Instance Count by Type and Cloud report: Cloud value
- Fix for Instance Count by Cloud report
- Fix for ServiceNow Plugin and API issue
- Fix for web instance types
- Fix for ServiceNow not listing datastores
- Fix for issue deleting ServiceNow integration
- Fix for OpenStack space issue
- Fix for OpenStack floating IP issue
- Fix for F5 destination port issue
- Fix for VMware ServiceNow Plugin
- Fix for manually deleting F5 pools and nodes
- Fix for Nutanix VM taking multiple IP slots
- Fix for PXE answer file
- Fix for ENI records not being cleaned up in Amazon
- Fix for tenant delete issues
- Fix for Nutanix image endpoint
- Fix for Install Agent flag for Nutanix
- Fix for OTC missing icons
- Fix for HCL parsing not saving
- Fix for CentOS VM conversion issue
- Fix for Blueprints resource pool
- Fix for instance provisioning hanging at configure stage
- Fix for Blueprints not connecting to Azure LB
- Fix for Option List ignoring SSL errors
- Fix for OTC duplicate images
- Fix for changing roles via AD groups
- Fix for user data in Windows Cloud
• Fix for provisioning Windows into vCD
• Fix for Custom Checks
• Fix for local firewall not working for VM’s
• Fix for Cloud Foundry and inventoried apps
• Fix for Cloud Foundry API check type missing logo
• Fix for Cloud Foundry cloud details
• Fix for Cloud Foundry orphaned instances and hosts
• Fix for Instance Inventory Summary
• Fix for deployment versions not sorting
• Fix for Archive Snapshots not saving
• Fix for Alibaba CentOS 6.9 bad layouts
• Fix for security groups
• Fix for Hadoop Ports map error
• Fix for modals not loading
• Fix for price editing
• Fix for multidisk component not using virtual image locations
• Fix for instance name validation
• Fix for instance type search limited to selected category
• Fix for adding disks requiring logs
• Fix for volume addition requiring infrastructure
• Fix for private hosts visible to tenants
• Fix for AD login when user OU changes
• Fix for changing plan not updating volume
• Fix for order of logs display
• Fix for cloud not saving user data
• Fix for network CRUD issues
• Fix for VMware ‘Clone to Image’ not showing failures
• Fix for API task ‘deleting’ failing
• Fix for log parse error
• Fix for creating app from existing template with no blueprint
• Fix for Azure ARM parsing error
• Fix for VMware ‘Clone to Image’ not working for stopped VM’s
• Fix for billing API
• Fix for App wizard naming issue
• Fix for Oracle Cloud Virtual Image icon missing
• Fix for Venafi integration details page
Fix for AVI LB monitors field being blank
Fix for Image Builder configuration issue
Fix for Veeam integration not being checked by default
Fix for Blueprints missing backup locks
Fix for error deleting Cloud Foundry service
Fix for i18n missing label

12.11 v3.4.1

Release date: 7/20/2018

12.11.1 New Features

- API: Run workflow now allows arbitrary customOptions to be passed
- API/CLI: Added CLI login via token. login -T <token> can be used instead of requiring a username and password.
- API/CLI: Added optimizations for Virtual Image uploads
- API/CLI: Added “taskPhase” for workflow tasks
- Backups: Direct Stream to Storage Provider added for VMware, Xen and KVM backups and Image imports.
- Blueprints: Added ACI fields to Blueprints and Apps wizards
- phpIPAM: Added additional information including description, hostname and owner
- Policies: Added variable <%= tenantSubdomain %> for Tenant SubDomain value in scripts and naming policies. The SubDomain is used for creating a direct login url in Identity Sources or as a login prefix to identify the tenant i.e. ‘subdomainusername’
- Puppet: Expanded OS support for Puppet Agent install.
- Users: User Settings: API Access section added for API & CLI user token generation
- VMware: Added Clone to Image Instance Action to create new VMware Templates from Instances with corresponding Morpheus Virtual Image record.
- VMware: Virtual Images: Switched “Force Guest Customizations” to on by default for Windows Virtual Images

12.11.2 Fixes

- Fix for VMware ovf export timeouts for large images. Impacted offloaded backups, Import as Image actions.
- Fix for Deployment versions not sorting
- Fix for disabling ‘Archive Snapshots’ flag on Storage Providers not saving. When executing backups, the ‘Archive Snapshots’ flag enables ovf export of VMware snapshots to default or specified backup Storage Provider and removes the snapshot in VMware. If disabled, backup jobs will only trigger and leave snapshots in VMware.
- Fix for Active Directory user login when users OU has been changed
- Fix for duplicate Alibaba CentOS 6.9 layouts
• Fix for Azure: ARM Templates copyIndex parsing errors
• Fix for Security Groups with Source Type set to All
• Fix for Actions: Reconfigure modal not loading in Infrastructure -> Hosts Section
• Fix for Instance Name uniqueness validation in provisioning Wizard
• Reverted Provision Wizard: Instance Type search being limited to selected category
• Fix for adding disks requiring Logs: Read or Full Role permissions
• Fix for volume addition requiring Infrastructure - Storage: Read or Full Role permissions
• Fix for Master Tenant private KVM Host Subtenant visibility
• Fix for changing plan not updating volume size for Nutanix Instance Type until image is selected
• Fix for adding AVI load balancer integration.
• Fix for API: /groups/:id/update-zones removal of zone not passed in the zones parameter
• Fix for Apps wizard not loading when User Role permission set to Provisioning: Blueprints: None
• Fix for Logs: Date parsing error when Windows Event logs are localized

12.12 v3.4.0

Release Date: 6/19/2018

**Important:** Morpheus v3.4.0 adds support for subtenant users to login via the main tenant url using subtenant id or subdomain prefix, ie subtenantId\username or subdomain\username. Subtenant local users will no longer be able to login from main login url without using subtenant id or subdomain prefix. Tenant subdomain can be defined by editing the Tenant settings and updating the SUBDOMAIN field.

12.12.1 New Features

• API/CLI: DataStores added
• API/CLI: Salt command options added
• API/CLI: Subtenant Groups added
• API/CLI: Workflows now support custom option types
• Blueprints: ARM template Git integration added
• Currency: Conversion settings added to Administration -> Settings.
• Datastores: Storage > Datastores capacity column now hidden when “Hide Datastore Stats On Selection” is enabled
• Identity Sources: Subtenant users can now login from the main login url using subtenant name or id prefix, including users created from Identity Source Integrations.
• KVM: Added OnApp migration support
• Library: CentOS 7.5 qcow2 image added
• Library: Ubuntu 16.04, 18.04 qcow2 images added
• Policies: User Creation Policy added
• Provisioning: Added default datastore for additional disks to match first disk
• Provisioning: Additional Network Interface types now default to match first Interface type
• SAML: Validation configuration options added
• Storage Providers: Verify permissions to path on save added

Note: In 3.4.0+ currency conversion data users will need to provide an open exchange or fixer.io api key under Administration -> Settings.

12.12.2 Fixes

• Fix for Chef detail section hidden in UI
• Fix for Cost by Cloud error
• Fix for incorrect Instance provisioning status when using copies with ARM Templates
• Fix for broken image issue for library items with no custom icon
• Fix for Instance Type Count - By Clouds Value in Analytics
• Fix for Instance Count by Cloud Group Count Value in Analytics
• Fix for CLI Bad Request issue when creating a cloud
• Fix for Web Instance Types to be scalable
• Fix for Cost Month to Date report
• Fix for ServiceNow listed Datastores
• Fix for Cloud Init enabled in Images
• Fix for KVM provisioning issue
• Fix for i18n missing message
• Fix for App Wizard Amazon validation issue
• Fix for space issue with Openstack
• Fix for postgresQL issue with BluemixCF
• Fix for Clouds stuck in deleting when removing
• Fix for issue with adding/editing local storage objects
• Fix for OpenStack Floating IP’s being required
• Fix for F5 destination port issue
• Fix for Storage Object issue with NFS
• Fix for instance Wizard Issues
• Fix for ServiceNow Plugin unable to provision VMware instance
• Fix for ServiceNow Plugin Datastores duplicate options
• Fix for F5 error deleting Pools and Nodes
• Fix for missing scale tab for VCD Instances
• Fix for KVM SSH fail after stop/start
• Fix for App Wizard form update when changing cloud
• Fix for ESXI Debian 8.8 failing to provision
• Fix for KVM provision failing on self managed host
• Fix for OEL build fail if more than 3 disks are requested during provisioning
• Fix for Open Telekom Cloud provisioning issue
• Fix for Cost by Cloud projected values
• Fix for Nutanix duplicate IP’s when using IP Pools
• Fix for PXE answer file not getting generated
• Fix for ENI records not being cleaned up (Amazon)
• Fix for Tenant delete issues
• Fix for Nutanix sysprep unattend.xml passwords
• Fix for “Install Agent” Flag for Nutanix Images
• Fix for missing image icons
• Fix Terraform HCL parsing
• Fix for some converted-to-managed VM’s having docker check type
• Fix for unnecessary resource pool request
• Fix for \texttt{vmToolsInstalled}: no such property

Morpheus v3.4.0 is available in the Downloads section of .
The morpheus cli is a ruby based cli that provides a lot of functionality out of the box and is rapidly growing in coverage to be able to perform every task that can be performed in the morpheus UI. It is also a great way to get started in exploring the morpheus API and understanding some of the data model aspects.

### 13.1 Installation

A Prerequisite to running the CLI is to have ruby 2.0.0+ installed (2.5 or greater recommended). Once the ruby runtime is installed simply use rubygems to install the cli gem `gem install morpheus-cli`. Once the gem is installed all cli commands can be run on the shell via `morpheus`.

#### 13.1.1 Linux

A Prerequisite to running the CLI is to have ruby 2.0.0+ installed (2.5 or greater recommended). Once the ruby runtime is installed simply use rubygems to install the cli gem `gem install morpheus-cli`. Once the gem is installed all cli commands can be run on the shell via `morpheus`.

Add this line to your application’s Gemfile:

```ruby
gem 'morpheus-cli'
```

And then execute:

```bash
$ bundle
```

Or install it yourself as:

```bash
$ gem install morpheus-cli
```
13.1.2 Windows

The morpheus cli is capable of running on many platforms due to its ruby runtime. This includes windows based platforms. To get started, we must first ensure ruby is running on the windows machine in question. To do this please visit http://rubyinstaller.org/downloads and download at least Ruby version 2.0.0 (2.3.3 recommended).

**NOTE:** When installingruby on windows, make sure the options are selected for adding the ruby binaries to your PATH.

Now that ruby is installed, simply open a PowerShell window and run

```
gem install morpheus-cli --no-ri --no-rdoc
```

A list of installed dependencies should start sliding by the screen. Once this has completed the CLI setup is complete. Now all that must be done is configuring the cli to point to an appliance for use.

```
morpheus remote add myapp https://applianceUrl
morpheus remote use myapp
morpheus login
```

Credentials are used to acquire an access token which is then stored in the users home directory in a folder called .morpheus. Now all commands provided by the CLI are available for use just as if running in a nix based environment.

13.2 Setup Appliance

The first thing that needs to be done after installing the cli is pointing the cli to the appliance. The CLI can be pointed at many appliances and uses the RESTful OAUTH public developer apis to perform tasks. To set this up simply add a remote appliance with the morpheus remote add command

```
morpheus remote add myappliance https://applianceUrl
morpheus remote use myappliance
morpheus login
```

There are several commands available when dealing with configuration of remote appliances. To see what commands are available just type morpheus remote.

13.3 Get Started

To get started with the morpheus cli its helpful to use morpheus shell. The shell provides a handy shell with history and some autocomplete features for learning to use it. All commands mentioned prefixed with morpheus can be omitted since we are in shell mode.

To confirm that we are hooked into the appliance properly lets check our authentication information:

```
morpheus> whoami
Current User
==================
ID: 1
Account: Labs (Master Account)
First Name: Demo
Last Name: Environment
Username: david
```

(continues on next page)
Fantastic! We are now ready to start our adventure in the Morpheus CLI. If this command fails please be sure to verify the appliance url entered previously is correct, and also verify the provided credentials are correctly entered.

While the CLI is relatively young there are a ton of features provided with it that can make it very convenient for working with morpheus. There are several base commands with subcommands within for example. Let's look at what happens when we simply type `morpheus` on the command line:

```
Usage: morpheus [command] [options]

Commands:
    remote
    login
    logout
    whoami
    groups
    clouds
    hosts
    load-balancers
    shell
    tasks
    workflows
    deployments
    instances
    apps
    app-templates
    deploy
    license
    instance-types
    security-groups
    security-group-rules
    accounts
    users
    roles
    key-pairs
    virtual-images
    library
    version
```

As you can see the CLI is split into sections. Each of these sections has subcommands available for performing certain actions. For example let's look at `morpheus instances`:

```
morpheus> instances

Usage: morpheus instances [list, add, remove, stop, start, restart, backup, run-workflow, stop-service, start-service, restart-service, resize, upgrade, clone, envs, setenv, delenv] [name]
```

13.3. Get Started
13.4 Provisioning

To get started provisioning instances from the CLI a few prerequisite commands must be setup in the CLI. First we must decide what Group we want to provision into. We can first get a list of available groups to use by running `morpheus groups list`.

```
morpheus> groups list
Morpheus Groups
==================
= Automation - denver
=> Demo - Multi
= Morpheus AWS - US-West
= Morpheus Azure - US West
= Morpheus Google - Auto
= morpheus-approvals -
= NICK-Demo - Chicago
= San Mateo Hyper-V - San Mateo, CA
= San Mateo Nutanix - San Mateo, CA
= San Mateo Openstack - San Mateo, CA
= San Mateo Servers - San Mateo, CA
= San Mateo UCS - San Mateo, CA
= San Mateo Vmware - San Mateo, CA
= San Mateo Xen - San Mateo, CA
= snow-approvals -
= SoftLayer - Dallas-9
```

In the above example the currently active group is Demo as can be seen by the => symbol to the left of the group name. To switch groups simply run:

```
morpheus groups use "San Mateo Xen"
```

This now becomes the active group we would like to provision into. Another thing to know before provisioning is we do have to also specify the cloud we want to provision into. This does require the cloud be in the group that is currently active. To see a list of clouds in the relevant group simply run:

```
morpheus clouds list -g [groupName]
```

This will scope the clouds command to list only clouds in the group specified.

Morpheus makes it very easy to get started provisioning via the CLI. It provides a list of instance-types that can be provisioned via the `instance-types list` command. Let's get started by provisioning an ubuntu virtual machine.

```
morpheus> instances add
Usage: morpheus instances add TYPE NAME
 -g, --group GROUP    Group
 -c, --cloud CLOUD    Cloud
```

(continues on next page)
-O, --option OPTION Option
-N, --no-prompt Skip prompts. Use default values for all optional fields.
-j, --json JSON Output
-d, --dry-run Dry Run, print json without making the actual request.
-r, --remote REMOTE Remote Appliance
-U, --url REMOTE API Url
-u, --username USERNAME Username
-p, --password PASSWORD Password
-T, --token ACCESS_TOKEN Access Token
-C, --nocolor ANSI
-V, --debug Print extra output for debugging.
-h, --help Prints this help

morpheus> instances add ubuntu MyInstanceName -c "San Mateo Vmware"
morpheus> instances add ubuntu -c "San Mateo Vmware" dre-test
Layout ['?' for options]: ?
  * Layout [-O layout=] - Select which configuration of the instance type to be provisioned.

Options
=======
  * Docker Ubuntu Container [104]
  * VMware VM [105]
  * Existing Ubuntu [497]

Layout ['?' for options]: VMware VM
Plan ['?' for options]: ?
  * Plan [-O servicePlan=] - Choose the appropriately sized plan for this instance

Options
=======
  * Memory: 512MB Storage: 10GB [10]
  * Memory: 2GB Storage: 20GB [12]
  * Memory: 4GB Storage: 40GB [13]
  * Memory: 8GB Storage: 80GB [14]
  * Memory: 24GB Storage: 240GB [16]
  * Memory: 32GB Storage: 320GB [17]

Plan ['?' for options]: 10
Root Volume Label [root]:
Root Volume Size (GB) [10]:
Root Datastore ['?' for options]: ?
  * Root Datastore [-O rootVolume.datastoreId=] - Choose a datastore.

Options
=======
  * Auto - Cluster [autoCluster]
  * Auto - Datastore [auto]
  * cluster: labs-ds-cluster - 2.9TB Free [19]
  * store: ds-130-root - 178.5GB Free [5]
* store: ds-131-root - 191.3GB Free [1]
* store: ds-131-vm - 798.9GB Free [9]
* store: ds-177-root - 399.4GB Free [3]
* store: labs-vm - 2.9TB Free [18]
* store: VeeamBackup_WIN-0JNJSO32KI4 - 5.1GB Free [8]
* store: VeeamBackup_WIN-QGARB6FA1GQ - 2.7GB Free [17]

Root Datastore ['?' for options]: Auto - Cluster
Add data volume? (yes/no): no
Network ['?' for options]: VM Network
Network Interface Type ['?' for options]: E1000
IP Address: Using DHCP
Add another network interface? (yes/no): no
Public Key (optional) ['?' for options]:
  * Resource Pool ['?' for options]: ?
    * Resource Pool [-O config.vmwareResourcePoolId=] -

Options
=======
* Resources [resgroup-56]
* Resources / Brian [resgroup-2301]
* Resources / Brian / Macbook [resgroup-2302]
* Resources / David [resgroup-2158]
* Resources / David / Macbook [resgroup-2160]

Resource Pool ['?' for options]: resgroup-2160

As can be seen in the example above, the CLI nicely prompts the user for input on required options for provisioning this particular instance type within this particular cloud. It provides capabilities of adding multiple disks and multiple networks in this scenario. It is also possible to skip these prompts and provision everything via one command line syntax by using the 
- O optionName=value syntax:

```
morpheus> instances add ubuntu MyInstanceName -c "San Mateo Vmware" -O layout=105 -O servicePlan=10 -O rootVolume.datastoreId=autoCluster
```

This will cause morpheus cli to skip prompting for input on these prompts. All inputs have an equivalent -O option that can be passed. To see what that option argument is simply enter ? on the input prompt to get specifics.

Now your VM should be provisioning and status can be checked by simply typing `morpheus instances list`.

### 13.5 List Arguments

Most of the list command types can be queried or paged via the cli. To do this simply look at the help information for the relevant list command:

```
morpheus> instances list -h
Usage: morpheus [options]
  -g, --group GROUP Group Name
  -m, --max MAX Max Results
  -o, --offset OFFSET Offset Results
  -s, --search PHRASE Search Phrase
  -S, --sort ORDER Sort Order
```

(continues on next page)
-D, --desc
-j, --json
-r, --remote REMOTE
-U, --url REMOTE
-u, --username USERNAME
-p, --password PASSWORD
-T, --token ACCESS_TOKEN
-C, --nocolor
-V, --debug
-h, --help

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-D, --desc</td>
<td>Reverse Sort Order</td>
</tr>
<tr>
<td>-j, --json</td>
<td>JSON Output</td>
</tr>
<tr>
<td>-r, --remote</td>
<td>Remote Appliance</td>
</tr>
<tr>
<td>-U, --url REMOTE</td>
<td>API Url</td>
</tr>
<tr>
<td>-u, --username</td>
<td>Username</td>
</tr>
<tr>
<td>-p, --password</td>
<td>Password</td>
</tr>
<tr>
<td>-T, --token ACCESS_TOKEN</td>
<td>Access Token</td>
</tr>
<tr>
<td>-C, --nocolor</td>
<td>ANSI</td>
</tr>
<tr>
<td>-V, --debug</td>
<td>Print extra output for debugging.</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Prints this help</td>
</tr>
</tbody>
</table>
The Morpheus CLI can be used to setup a freshly installed Morpheus Appliance.

First, add your remote appliance with `remote add`

```
morpheus remote add myappliance https://myappliance.mysite.com --use
```

Note that the `--use` option is not necessary if this is the first and only appliance in your CLI config

Now, use the `remote setup` command to walk you through setting up your appliance.

```
morpheus remote setup
```

This will prompt you for all the settings required to initialize the appliance, and then log you as the new master System Admin user.

---

Morpheus Appliance Setup
========================

It looks like you're the first one here.
Let's initialize your remote appliance at https://myappliance.mysite.com

Create Master account
=====================

Master Account Name: root

Create Master User
==================

First Name (optional):
Last Name (optional):
Username: james
Email: james@morpheusdata.com
Password:
Confirm Password:

(continues on next page)
Initial Setup
==================
Appliance Name: myappliance
Appliance URL [http://10.0.2.2:8080/]: https://myappliance.mysite.com
Enable Backups (yes/no) [no]:
Enable Monitoring (yes/no) [yes]:
Enable Logs (yes/no) [yes]:
Initializing the appliance...

You have successfully setup the appliance.
You are now logged in as the System Admin james.

Would you like to apply your License Key now? (yes/no) [yes]:
License Key: <your key>

Do you want to create the first group now? (yes/no) [yes]:
Name: g1
Code (optional): Location (optional):
Added group g1

Morpheus Groups
==================

| ID | NAME | LOCATION | CLOUDS | HOSTS |
---|----|------|----------|--------|------|
=> | 1 | g1 | | 0 | 0 |

Viewing 1-1 of 1

# => Currently using g1

Do you want to create the first cloud now? (yes/no) [yes]:
Cloud Type ['?' for options]: Morpheus
Name: c1
Code (optional): Location (optional):
Visibility (optional) [private] ['?' for options]:

Cloud Details
==================

ID: 1
Name: c1
Type: Morpheus
Code: standard
Location:
Visibility: Private
Groups: g1
Status: OK

Cloud Servers (0)
==================

Container Hosts: 0  Hypervisors: 0  Bare Metal: 0  Virtual Machines: 0  Unmanaged: 0

(continues on next page)
That’s it, your appliance is ready for use now, and you’ve already created your first Group and Cloud.
This command can only be done once.

morpheus remote setup
Appliance has already been setup
CHAPTER 15

Managing Instances

The CLI provides several convenience commands for managing your instances. This subsection of the CLI is still being built out but some useful ones are already up and ready to go.

15.1 Listing Instances

```
morpheus instances list

Morpheus Instances
==================

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>GROUP</th>
<th>CLOUD</th>
<th>TYPE</th>
<th>ENVIRONMENT</th>
<th>NODES</th>
<th>CONNECTION</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>73</td>
<td>mysql500</td>
<td>thegroup</td>
<td>bertramlabs-aws</td>
<td>MySQL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(continues on next page)
```
Simply calling the list command will output a list of instances your account has access to. You can see it not only displays the name of the instance but also the instance type as well as running state (aka status).

## Working with a specific Instance

There are several commands pertaining to specific instances. For example it is very easy to stop, start, and restart a running instance. To do so you can execute one of the following commands:

```
```

morpheus instances stop "mysql500"
morpheus instances start "mysql500"
morpheus instances restart "mysql500"
```

You may pass the instance ID in place of NAME

morpheus instances restart 73

You may view the current usage statistics for an instance

morpheus instances stats 23

Instance Stats: testredis1 (Redis)
===============

Status: RUNNING

| Memory:  | [%] | 6.20 MiB / 256.00 MiB | 2.42% |
| Storage: | [%] | 254.08 MiB / 2.00 GiB | 12.41% |
| CPU:     | [%] | 0.12%               | 0.12% |

To get the output as JSON instead, use --json

morpheus instances stats "V1 - Redis" --json

This command outputs the following:
15.1.1 Environment Variables

The CLI provides several useful commands for managing the environment variables applied to the running instance. To list the known environment variables simply execute:

```
morpheus instances envs "Spud Marketing"
```

You can assign environment variables as well with the `setenv` command:

```
morpheus instances setenv INSTANCE NAME VALUE [-e]
```

The `-e` argument allows you to mark an environment variable as exportable. Exportable means that if this instance were to be added to an App, all other instances in that app would automatic get this environment variable.

To delete an environment variable simply use the `delenv` command:

```
morpheus instances delenv INSTANCE NAME
```

**NOTE**: Containers must be restarted for new environment variables to be applied. Be sure to run a restart of the instance after you are done manipulating the environment.

15.2 Creating Instances

The CLI makes it very easy to add new instances into the environment. There are still features being added here for container specific configuration but most of this is operational. The first step is to ensure an active Server group is selected within your cli. To do so simply execute:

```
morpheus groups list
```

then select a group using:

```
morpheus groups use "My group name"
```

Now we are ready to create an instance. There are several different instance types available to choose from. We add a shorthand code to make it easy to provision these without having to type the full formal name of the instance type. You can get a list of instance types from the catalog by executing:

```
morpheus instance-types list
```
This will list all items in the catalog as well as their known configuration options. The shorthand name will be in parenthesis.

Let's start by creating a node.js instance.

```
morpheus instances add "My Test Instance" node

Configurations:
  1) Single Node (node-4.0.0-single)
Selection: 1

Select a Plan:
  1) Memory: 128MB Storage: 1GB
  2) Memory: 256MB Storage: 3GB
  3) Memory: 512MB Storage: 5GB
  4) Memory: 1GB Storage: 10GB
  5) Memory: 2GB Storage: 20GB
  6) Memory: 4GB Storage: 40GB
  7) Memory: 8GB Storage: 80GB
  8) Memory: 16GB Storage: 160GB
Selection: 1
```

That's it, now we have created a new node.js app. If you did MySQL, you would get prompted for some additional configuration information. We are actively working to make these selections all from the main command line as well and some of them are already.

You may also have the CLI prompt you for all the available options.

```
morpheus instances add
Cloud [? for options]: vcenter
Type [? for options]: Node
Instance Name: My Test Instance
Description (optional):
```

### 15.3 Removing Instances

Removing Morpheus instances is also fairly simple.

```
morpheus instances remove "My Test Instance"
```

**NOTE:** As of appliance version 2.1.9 a delete issue has been discovered with the CLI and API. We should have this resolved in the next release cycle 2.2.x
The CLI provides a command called shell that allows you enter an interactive shell. This is a convenient way to test things out quickly, and to avoid prefixing your commands with morpheus.

16.1 Open a shell

```
morpheus shell
```

Now you may run all the available commands like so:

```
morpheus> remote use qa
morpheus> login
morpheus> clouds list
```

To see the list of all available commands, use help

```
morpheus> help
```

TAB can be used to auto-complete the name of a command or any aliases you’ve defined.

CTRL + R can be used to search for available commands and previously executed commands.

The ↑ and ↓ arrow keys can be used to navigate through the shell history.

To see a list of all the commands you’ve executed in a shell, use history

```
morpheus> history
Last 4 commands
  1 flush-history
  2 groups list
  3 clouds list
  4 instances list
```

Historical commands can be executed by prefixing the command number with !
And again, just like bash, you can re-execute the last command with `!!`

The `flush-history` command can be used to delete the shell history.

It is possible to execute multiple commands sequentially inside a shell, like this:

You may define an Alias within a shell, which will persist through subsequent shells.

To exit a shell, use `exit`
We divide morpheus into commands. Every morpheus command may have 0-N sub-commands that it supports. Commands generally map to the functionality provided in the Morpheus UI.

You can get help for any morpheus command by using the -h option.

The available commands and their options are also documented below.

Usage: morpheus [command] [options]

Commands:
  access-token
  alias
  apps
  archives
  benchmark
  blueprints
  clouds
  containers
  cypher
  datastores
  deploy
  deployments
  edit-profile
  edit-rc
  execute-schedules
  execution-request
  file-copy-request
  groups
  hosts
  image-builder
  instance-types
  instances
  key-pairs
  library-file-templates
library-instance-types
library-layouts
library-node-types
library-option-lists
library-option-types
library-scripts
library-upgrades
license
load-balancers
login
logout
monitor-apps
monitor-checks
monitor-contacts
monitor-groups
monitor-incidents
network-domains
network-groups
network-pool-servers
network-pools
network-proxies
network-services
networks
passwd
policies
power-schedules
process
recent-activity
remote
roles
security-group-rules
security-groups
shell
storage-buckets
tasks
tenants
user-groups
user-settings
user-sources
users
version
virtual-images
whoami
workflows

Options:
-\-e, --exec EXPRESSION  Execute the command(s) expression. This is an alternative to passing [command] [options]
-\-noprofile  Do not read and execute the personal initialization script .morpheus_profile
-\-C, --nocolor  Disable ANSI coloring
-\-B, --benchmark  Print benchmark time after the command is finished.
-\-V, --debug  Print extra output for debugging.
-\-v, --version  Print the version.
-\-h, --help  Print this help

For more information, see https://github.com/gomorpheus/morpheus-cli/wiki
17.1 morpheus access-token

Usage: morpheus access-token [command] [options]

Commands:
- details
- get
- refresh

17.1.1 morpheus access-token details

Usage: morpheus access-token details
- -r, --remote REMOTE Remote name. The current remote is used by default.
- default.
- --remote-url URL Remote url. The current remote url is used by default.
- default.
- -T, --token TOKEN Access token for authentication with --remote.
- Saved credentials are used by default.
- -U, --username USERNAME Username for authentication.
- -P, --password PASSWORD Password for authentication.
- -I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- -H, --header HEADER Additional HTTP header to include with requests.
- --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- -q, --quiet No Output, do not print to stdout
- -c, --nocolor Disable ANSI coloring
- -b, --benchmark Print benchmark time after the command is finished.
- -v, --debug Print extra output for debugging.
- -h, --help Print this help

Print your current authentication credentials. This contains tokens that should be kept secret, be careful.

17.1.2 morpheus access-token get

Usage: morpheus access-token get
- -r, --remote REMOTE Remote name. The current remote is used by default.
- default.
- --remote-url URL Remote url. The current remote url is used by default.
- default.
- -T, --token TOKEN Access token for authentication with --remote.
- Saved credentials are used by default.
- -U, --username USERNAME Username for authentication.
- -P, --password PASSWORD Password for authentication.
- -I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- -H, --header HEADER Additional HTTP header to include with requests.
- --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- -q, --quiet No Output, do not print to stdout
- -c, --nocolor Disable ANSI coloring
- -b, --benchmark Print benchmark time after the command is finished.
17.1.3 morpheus access-token refresh

Usage: morpheus access-token refresh

- y, --yes
- r, --remote REMOTE
- default.
- remote-url URL
- default.
- t, --token TOKEN
- default.
- u, --username USERNAME
- p, --password PASSWORD
- i, --insecure
- s, --ssl certificate.
- h, --header HEADER
- timeout SECONDS
- 30 seconds.
- d, --dry-run
- --cURL
- --cURL
- --scrub
- Authorization header. For use with --cURL and --dry-run.
- j, --json
- q, --quiet
- c, --nocolor
- b, --benchmark
- finished.
- v, --debug
- h, --help

Use your refresh token.
This will replace your current access and refresh tokens with a new values.
Your current access token will be invalidated
All other users or applications with access to your token will need to update to the
new token.

17.2 morpheus alias

Usage: morpheus alias [command] [options]

Commands:
  add
  export
  list
  remove
17.2.1 morpheus alias add

Usage: morpheus alias add [name]='[command]'
   -e, --export Export this alias to your .morpheus_profile for future use
   -h, --help Print this help

Define a new alias.
[name] is required. This is the alias name. It should be one word.
[command] is required. This is the full command wrapped in quotes.
Aliases can be exported for future use with the -e option.
The `alias add` command can be invoked with `alias [name]=[command]`

Examples:
   alias cloud=clouds
   alias ij='instances get -j'
   alias new-hosts='hosts list -S id -D'
For more information, see https://github.com/gomorpheus/morpheus-cli/wiki/Alias

17.2.2 morpheus alias export

Usage: morpheus alias export [alias] [alias2] [alias3]
   -C, --nocolor Disable ANSI coloring
   -B, --benchmark Print benchmark time after the command is finished.
   -V, --debug Print extra output for debugging.
   -h, --help Print this help

Export an alias, saving it to your .morpheus_profile for future use

17.2.3 morpheus alias list

Usage: morpheus alias list
   -f, --format FORMAT The format for the output: export, json, list, table (default).
   -e, --export Include the '-e' switch after each alias in the output. This implies --format export.
   -m, --max MAX Max Results
   -o, --offset OFFSET Offset Results
   -s, --search PHRASE Search Phrase
   -S, --sort ORDER Sort Order
   -D, --desc Reverse Sort Order
   -j, --json JSON Output
   -C, --nocolor Disable ANSI coloring
   -B, --benchmark Print benchmark time after the command is finished.
   -V, --debug Print extra output for debugging.
   -h, --help Print this help

Print list of defined aliases.
Use the --format option to vary output.
The `alias list` command can be abbreviated as just `alias`.
For more information, see https://github.com/gomorpheus/morpheus-cli/wiki/Alias
17.2.4 morpheus alias remove

Usage: morpheus alias remove [alias1] [alias2]
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

This is how you remove alias definitions from your .morpheus_profile

17.3 morpheus apps

Usage: morpheus apps [command] [options]
Commands:
- add
--add-instance
--apply-security-groups
- count
--firewall-disable
--firewall-enable
- get
--history
--list
--logs
--remove
--remove-instance
--restart
--security-groups
--start
--stop
--update

17.3.1 morpheus apps add

Usage: morpheus apps add [name] [options]
-b, --blueprint BLUEPRINT Blueprint Name or ID. The default value is 'existing' which means no blueprint, for creating a blank app and adding existing instances.
-g, --group GROUP Group Name or ID
-c, --cloud CLOUD Default Cloud Name or ID.
--name VALUE Name
--description VALUE Description
-e, --environment VALUE Environment Name
--validate Validate Only. Validates the configuration and skips creating it.
--refresh [SECONDS] Refresh until status is running, failed. Default interval is 5 seconds.
-O, --option OPTION Option in the format -O field="value"
-P, --prompt Always prompts. Use passed options as the default value.
-N, --no-prompt Skip prompts. Use default values for all optional fields.
17.3.2 morpheus apps add-instance

Usage: morpheus apps add-instance [app] [instance] [tier]

-0, --option OPTION Option in the format -O field="value"
-P, --prompt Always prompts. Use passed options as the default value.
-N, --no-prompt Skip prompts. Use default values for all optional fields.
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it.
-curl Dry Run to output API request as a curl command.
-scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-g, --quiet No Output, do not print to stdout
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Add an existing instance to an app.

[app] is required. This is the name or id of an app.
[instance] is required. This is the name or id of an instance.
[tier] is required. This is the name of the tier.

17.3.3 morpheus apps apply-security-groups

Usage: morpheus apps apply-security-groups [app] [--clear] [-s]

c, --clear Clear all security groups

17.3. morpheus apps
-s, --secgroups SECGROUPS  Apply the specified comma separated security groups
-j, --json  JSON Output
-d, --dry-run  Dry Run, print the API request instead of executing it
--curl  Dry Run to output API request as a curl command.
--scrub  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-C, --nocolor  Disable ANSI coloring
-B, --benchmark  Print benchmark time after the command is finished.
-V, --debug  Print extra output for debugging.
-h, --help  Print this help

17.3.4 morpheus apps count

Usage: morpheus apps count [options]
  --created-by USER  Created By User Username or ID
  -s, --search PHRASE  Search Phrase
  -Q, --query PARAMS  Query parameters. PARAMS format is 'phrase=foobar&category=web'
  -r, --remote REMOTE  Remote name. The current remote is used by default.
  --remote-url URL  Remote url. The current remote url is used by default.
  -T, --token TOKEN  Access token for authentication with --remote.
  -U, --username USERNAME  Username for authentication.
  -P, --password PASSWORD  Password for authentication.
  -I, --insecure  Allow insecure HTTPS communication. i.e. bad SSL certificate.
  -H, --header HEADER  Additional HTTP header to include with requests.
  --timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
  -d, --dry-run  Dry Run, print the API request instead of executing it
  --curl  Dry Run to output API request as a curl command.
  --scrub  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -C, --nocolor  Disable ANSI coloring
  -B, --benchmark  Print benchmark time after the command is finished.
  -V, --debug  Print extra output for debugging.
  -h, --help  Print this help

Get the number of apps.

17.3.5 morpheus apps firewall-disable

Usage: morpheus apps firewall-disable [app]
  -j, --json  JSON Output
  -d, --dry-run  Dry Run, print the API request instead of executing it

(continues on next page)
17.3.6 morpheus apps firewall-enable

Usage: morpheus apps firewall-enable [app]

- j, --json
- d, --dry-run
- exec
  --curl
  --scrub
- Authorization header. For use with --curl and --dry-run.
- C, --nocolor
- B, --benchmark
- finished.
- V, --debug
- h, --help

17.3.7 morpheus apps get

Usage: morpheus apps get [app]

- refresh [SECONDS]
- interval is 5 seconds.
- refresh-until STATUS
- j, --json
- yam
- csv
- csv-delim CHAR
- csv-newline [CHAR]
- csv-quotes
- csv-no-header
- F, --fields x,y,z
- out FILE
- terminal.
- d, --dry-run
- exec
  --curl
  --scrub
- Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE
- default.
- remote-url URL
- default.
- T, --token TOKEN
- saved credentials are used by default.
- U, --username USERNAME

(continues on next page)
Get details about an app.

17.3.8 morphesus apps history

Usage: morphesus apps history [app]

- --events Display sub processes (events).
- --output Display process output.
- --details Display more details. Shows everything.
- --untruncated.
- -m, --max MAX Max Results
- -o, --offset OFFSET Offset Results
- -s, --search PHRASE Search Phrase
- -S, --sort ORDER Sort Order
- -D, --desc Reverse Sort Order
- -Q, --query PARAMS Query parameters. PARAMS format is
- 'phrase=foobart&category=web'
- -j, --json JSON Output
- --yaml YAML Output
- --csv CSV Output
- --csv-delim CHAR Delimiter for CSV Output values. Default: ','
- --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
- --csv-quotes Wrap CSV values with "". Default: false
- --csv-no-header Exclude header for CSV Output.
- -F, --fields x,y,z Filter CSV to a limited set of fields. Default is all fields.
- -d, --dry-run Dry Run, print the API request instead of executing it.
- --curl Dry Run to output API request as a curl command.
- --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- -r, --remote REMOTE Remote name. The current remote is used by default.
- --remote-url URL Remote url. The current remote url is used by default.
- -T, --token TOKEN Access token for authentication with --remote.
- --saved-credentials are used by default.
- -U, --username USERNAME Username for authentication.
- -P, --password PASSWORD Password for authentication.
- -I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
17.3.9 morpheus apps list

Usage: morpheus apps list

-created-by USER Created By User Username or ID
--details Display more details: memory and storage usage
-used / max values.
-m, --max Max Results
-o, --offset OFFSET Offset Results
-s, --search PHRASE Search Phrase
-S, --sort ORDER Sort Order
-D, --desc Reverse Sort Order
-Q, --query PARAMS Query parameters. PARAMS format is 'phrase=foobar&category=web'
-j, --json JSON Output
--yaml YAML Output
--csv CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
--csv-quotes Wrap CSV values with ''. Default: false
--csv-no-header Exclude header for CSV Output.
-F, --fields x,y,z Filter Output to a limited set of fields. Default is all fields.
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-t, --token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

List historical processes for a specific app.
[app] is required. This is the name or id of an app.
List apps.

### 17.3.10 morpheus apps logs

**Usage:** morpheus apps logs [app]

- `-m`, `--max MAX` Max Results
- `-o`, `--offset OFFSET` Offset Results
- `-s`, `--search PHRASE` Search Phrase
- `-S`, `--sort ORDER` Sort Order
- `-D`, `--desc` Reverse Sort Order
- `-j`, `--json` JSON Output
- `-d`, `--dry-run` Dry Run, print the API request instead of executing it
  - `--curl` Dry Run to output API request as a curl command.
  - `--scrub` Mask secrets in output, such as the Authorization header. For use with `--curl` and `--dry-run`.
- `-C`, `--nocolor` Disable ANSI coloring
- `-B`, `--benchmark` Print benchmark time after the command is finished.
- `-V`, `--debug` Print extra output for debugging.
- `-h`, `--help` Print this help

List logs for an app.

[app] is required. This is the name or id of an app.

### 17.3.11 morpheus apps remove

**Usage:** morpheus apps remove [app]

- `--remove-instances [on|off]` Remove instances. Default is off.
- `--preserve-volumes [on|off]` Preserve Volumes. Default is off. Applies to certain types only.
- `--keep-backups` Preserve copy of backups
- `--releaseEIPS [on|off]` Release EIPs. Default is on. Applies to Amazon only.
- `-f`, `--force` Force Delete
- `-j`, `--json` JSON Output
- `-d`, `--dry-run` Dry Run, print the API request instead of executing it
  - `--curl` Dry Run to output API request as a curl command.
  - `--scrub` Mask secrets in output, such as the Authorization header. For use with `--curl` and `--dry-run`.
- `-q`, `--quiet` No Output, do not print to stdout
- `-y`, `--yes` Auto Confirm
- `-C`, `--nocolor` Disable ANSI coloring
- `-B`, `--benchmark` Print benchmark time after the command is finished.
- `-V`, `--debug` Print extra output for debugging.
- `-h`, `--help` Print this help

Delete an app.

[app] is required. This is the name or id of an app.
### 17.3.12 morpheus apps remove-instance

Usage: morpheus apps remove-instance [app] [instance]

- `-O, --option OPTION` Option in the format `-O field="value"
- `-P, --prompt` Always prompts. Use passed options as the
  default value.
- `-N, --no-prompt` Skip prompts. Use default values for all
  optional fields.
- `-j, --json` JSON Output
- `-d, --dry-run` Dry Run, print the API request instead of
- executing it
  `-c, --curl` Dry Run to output API request as a curl command.
  `-s, --scrub` Mask secrets in output, such as the
- Authorization header. For use with `-c, --curl and --dry-run.
- `-C, --nocolor` Disable ANSI coloring
- `-B, --benchmark` Print benchmark time after the command is
  finished.
- `-v, --debug` Print extra output for debugging.
- `-h, --help` Print this help

Remove an instance from an app.

[app] is required. This is the name or id of an app.

[instance] is required. This is the name or id of an instance.

### 17.3.13 morpheus apps restart

Usage: morpheus apps restart [app]

- `-y, --yes` Auto Confirm
- `-j, --json` JSON Output
- `-d, --dry-run` Dry Run, print the API request instead of
  executing it
  `-c, --curl` Dry Run to output API request as a curl command.
  `-s, --scrub` Mask secrets in output, such as the
- Authorization header. For use with `-c, --curl and --dry-run.
- `-q, --quiet` No Output, do not print to stdout
- `-r, --remote REMOTE` Remote name. The current remote is used by
  default.
- `-u, --remote-url URL` Remote url. The current remote url is used by
  default.
- `-T, --token TOKEN` Access token for authentication with --remote.
- Saved credentials are used by default.
- `-U, --username USERNAME` Username for authentication.
- `-P, --password PASSWORD` Password for authentication.
- `-I, --insecure` Allow insecure HTTPS communication. i.e. bad
  SSL certificate.
- `-H, --header HEADER` Additional HTTP header to include with requests.
- `-t, --timeout SECONDS` Timeout for api requests. Default is typically
  30 seconds.
- `-C, --nocolor` Disable ANSI coloring
- `-B, --benchmark` Print benchmark time after the command is
  finished.
- `-v, --debug` Print extra output for debugging.
- `-h, --help` Print this help

Restart an app.

[app] is required. This is the name or id of an app. Supports 1-N [app] arguments.
17.3.14 morpheus apps security-groups

Usage: morpheus apps security-groups [app]
  -j, --json                JSON Output
  -d, --dry-run             Dry Run, print the API request instead of executing it
  --executing it
    --curl                  Dry Run to output API request as a curl command.
    --scrub                 Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -C, --nocolor             Disable ANSI coloring
  -B, --benchmark           Print benchmark time after the command is finished.
  -v, --debug               Print extra output for debugging.
  -h, --help                Print this help

17.3.15 morpheus apps start

Usage: morpheus apps start [app]
  -y, --yes                 Auto Confirm
  -j, --json                JSON Output
  -d, --dry-run             Dry Run, print the API request instead of executing it
  --executing it
    --curl                  Dry Run to output API request as a curl command.
    --scrub                 Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -q, --quiet               No Output, do not print to stdout
  -r, --remote REMOTE      Remote name. The current remote is used by default.
  --remote-url URL         Remote url. The current remote url is used by default.
  -t, --token TOKEN        Access token for authentication with --remote.
  -U, --username USERNAME   Username for authentication.
  -P, --password PASSWORD  Password for authentication.
  -I, --insecure           Allow insecure HTTPS communication. i.e. bad SSL certificate.
  -H, --header HEADER      Additional HTTP header to include with requests.
  --timeout SECONDS        Timeout for api requests. Default is typically 30 seconds.
  -C, --nocolor            Disable ANSI coloring
  -B, --benchmark          Print benchmark time after the command is finished.
  -v, --debug              Print extra output for debugging.
  -h, --help               Print this help

Start an app.
[app] is required. This is the name or id of an app. Supports 1-N [app] arguments.

17.3.16 morpheus apps stop

Usage: morpheus apps stop [app]
  -y, --yes                 Auto Confirm
  -j, --json                JSON Output

(continues on next page)
Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.

Stop an app.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Update an app.

[app] is required. This is the name or id of an app.

17.4 morpheus archives

Usage: morpheus archives [command] [options]
Commands:

add
add-file-link
download
download-bucket
download-link
file
file-history
file-links
get
list
list-files
ls
read
remove
remove-file
remove-file-link
rm
update
upload

17.4.1 morpheus archives add

Usage: morpheus archives add [options]

--name VALUE Name
--description VALUE Description
--storageProvider VALUE Storage Provider ID
--visibility [private|public]
Visibility determines if read access is restricted to the specified Tenants (Private) or all tenants (Public).
--accounts LIST Tenant Accounts (comma separated ids)
--isPublic [on|off] Enabling Public URL allows files to be downloaded without any authentication.
-O, --option OPTION Option in the format -O field="value"
-P, --prompt Always prompts. Use passed options as the default value.
-N, --no-prompt Skip prompts. Use default values for all optional fields.
--payload FILE Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
Create a new archive bucket.

17.4.2 morpheus archives add-file-link

Usage: morpheus archives add-file-link [bucket:/path]
   -e, --expire SECONDS The time to live for this link. The default is 1200 (20 minutes). A value less than 1 means never expire.
   -j, --json JSON Output
   -d, --dry-run Dry Run, print the API request instead of executing it
       --curl Dry Run to output API request as a curl command.
       --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -q, --quiet No Output, do not print to stdout
   -C, --nocolor Disable ANSI coloring
   -B, --benchmark Print benchmark time after the command is finished.
   -V, --debug Print extra output for debugging.
   -h, --help Print this help

Create a public link to a file.
[bucket:/path] is required. This is the name of the bucket and /path the file or folder to be fetched.

17.4.3 morpheus archives download

Usage: morpheus archives download [bucket:/path] [local-file]
   -f, --force Overwrite existing [local-file] if it exists.
   --mkdir Create missing directories for [local-file] if they do not exist.
   -p, --public Use Public Download URL instead of Private. The file must be in a public archives.
   -d, --dry-run Dry Run, print the API request instead of executing it
       --curl Dry Run to output API request as a curl command.
       --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -q, --quiet No Output, do not print to stdout

(continues on next page)
### 17.4.4 morpheus archives download-bucket

**Usage:** morpheus archives download-bucket [bucket] [local-file]

- **-f, --force** Overwrite existing [local-file] if it exists.
- **-p, --mkdir** Create missing directories for [local-file] if they do not exist.
- **-d, --dry-run** Dry Run, print the API request instead of executing it.
  - **--curl** Dry Run to output API request as a curl command.
  - **--scrub** Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- **-q, --quiet** No Output, do not print to stdout.
- **-C, --nocolor** Disable ANSI coloring.
- **-B, --benchmark** Print benchmark time after the command is finished.
- **-V, --debug** Print extra output for debugging.
- **-h, --help** Print this help

Download an archive file or directory.
[bucket:/path] is required. This is the name of the bucket and /path the file or folder to be downloaded.
[local-file] is required. This is the full local filepath for the downloaded file.
Directories will be downloaded as a .zip file, so you'll want to specify a [local-file] with a .zip extension.

### 17.4.5 morpheus archives download-link

**Usage:** morpheus archives download-link [link-key] [local-file]

- **-f, --force** Overwrite existing [local-file] if it exists.
- **-p, --mkdir** Create missing directories for [local-file] if they do not exist.
- **-d, --dry-run** Dry Run, print the API request instead of executing it.
  - **--curl** Dry Run to output API request as a curl command.
  - **--scrub** Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- **-q, --quiet** No Output, do not print to stdout.
- **-C, --nocolor** Disable ANSI coloring.
- **-B, --benchmark** Print benchmark time after the command is finished.
- **-V, --debug** Print extra output for debugging.
- **-h, --help** Print this help

Download an entire archive bucket as a .zip file.
[bucket] is required. This is the name of the bucket.
[local-file] is required. This is the full local filepath for the downloaded file.
Buckets are be downloaded as a .zip file, so you'll want to specify a [local-file] with a .zip extension.
17.4.6 morpheus archives file

Usage: morpheus archives file [bucket:/path]

- L, --all-links         Display all links instead of only 10.
- H, --all-history       Display all history instead of only 10.
- j, --json              JSON Output
- d, --dry-run           Dry Run, print the API request instead of executing it.
- --curl                 Dry Run to output API request as a curl command.
- --scrub                Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- C, --nocolor           Disable ANSI coloring
- B, --benchmark         Print benchmark time after the command is finished.
- V, --debug             Print extra output for debugging.
- h, --help              Print this help

Get details about an archive file.
[bucket:/path] is required. This is the name of the bucket and /path the file or folder to be fetched.
[id] can be passed instead of [bucket:/path]. This is the numeric File ID.

17.4.7 morpheus archives file-history

Usage: morpheus archives file-history [bucket:/path]

- m, --max MAX           Max Results
- o, --offset OFFSET     Offset Results
- s, --search PHRASE     Search Phrase
- S, --sort ORDER        Sort Order
- D, --desc              Reverse Sort Order
- j, --json              JSON Output
- d, --dry-run           Dry Run, print the API request instead of executing it.
- --curl                 Dry Run to output API request as a curl command.
- --scrub                Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- C, --nocolor           Disable ANSI coloring
- B, --benchmark         Print benchmark time after the command is finished.
- V, --debug             Print extra output for debugging.
- h, --help              Print this help

List history log events for an archive file.
17.4.8 morpheus archives file-links

Usage: morpheus archives file-links [bucket:/path]

- \( m \), --max MAX
  Max Results

- \( o \), --offset OFFSET
  Offset Results

- \( s \), --search PHRASE
  Search Phrase

- \( S \), --sort ORDER
  Sort Order

- \( D \), --desc
  Reverse Sort Order

- \( j \), --json
  JSON Output

- \( d \), --dry-run
  Dry Run, print the API request instead of executing it

- \(--\text{curl}\)
  Dry Run to output API request as a curl command.

- \(--\text{scrub}\)
  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.

- \( \text{-C} \), --nocolor
  Disable ANSI coloring

- \( \text{-B} \), --benchmark
  Print benchmark time after the command is finished.

- \( \text{-V} \), --debug
  Print extra output for debugging.

- \( \text{-h} \), --help
  Print this help

List links for an archive file.

17.4.9 morpheus archives get

Usage: morpheus archives get [bucket:/path]

- \( j \), --json
  JSON Output

- \( d \), --dry-run
  Dry Run, print the API request instead of executing it

- \(--\text{curl}\)
  Dry Run to output API request as a curl command.

- \(--\text{scrub}\)
  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.

- \( \text{-C} \), --nocolor
  Disable ANSI coloring

- \( \text{-B} \), --benchmark
  Print benchmark time after the command is finished.

- \( \text{-V} \), --debug
  Print extra output for debugging.

- \( \text{-h} \), --help
  Print this help

Display archive bucket details and files.
The [bucket] component of the argument is the name or id of an archive bucket.
The [:/path] component is optional and can be used to display files under a sub-directory.

17.4.10 morpheus archives list

Usage: morpheus archives list

- \( m \), --max MAX
  Max Results

- \( o \), --offset OFFSET
  Offset Results

- \( s \), --search PHRASE
  Search Phrase

- \( S \), --sort ORDER
  Sort Order

- \( D \), --desc
  Reverse Sort Order

- \( j \), --json
  JSON Output

- \( d \), --dry-run
  Dry Run, print the API request instead of executing it

(continues on next page)
17.4.11 morpheus archives list-files

Usage: morpheus archives list-files [bucket:/path]
   -a, --all Show all files, including subdirectories under 
   --the /path.
   -m, --max MAX Max Results
   -o, --offset OFFSET Offset Results
   -s, --search PHRASE Search Phrase
   -S, --sort ORDER Sort Order
   -D, --desc Reverse Sort Order
   -j, --json JSON Output
   -d, --dry-run Dry Run, print the API request instead of 
   --executing it
   --curl Dry Run to output API request as a curl command.
   --scrub Mask secrets in output, such as the
   --Authorization header. For use with --curl and --dry-run.
   -C, --nocolor Disable ANSI coloring
   -B, --benchmark Print benchmark time after the command is
   --finished.
   -V, --debug Print extra output for debugging.
   -h, --help Print this help

List files in an archive bucket. Include [/path] to show files under a directory.

17.4.12 morpheus archives ls

Usage: morpheus archives ls [bucket/path]
   -a, --all Show all files, including subdirectories under 
   --the /path.
   -l, --long Lists files in the long format, which contains
   lots of useful information, e.g. the exact size of the file, the file type, and
   when it was last modified.
   -H, --human Humanized file sizes. The default is just the
   --number of bytes.
   -i, --oneline One file per line. The default delimiter is a
   --single space.
   -m, --max MAX Max Results
   -o, --offset OFFSET Offset Results
   -s, --search PHRASE Search Phrase
   -S, --sort ORDER Sort Order

(continues on next page)
Print filenames for a given archive location.
Pass archive location in the format bucket/path.

17.4.13 morpheus archives read

Usage: morpheus archives read [bucket:/path]

-y, --yes Auto Confirm
-d, --dry-run Dry Run, print the API request instead of executing it
--executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Print the contents of an archive file.
[bucket:/path] is required. This is the name of the bucket and /path the file or folder to be downloaded.
Confirmation is needed if the specified file is more than 1KB.
This confirmation can be skipped with the -y option.

17.4.14 morpheus archives remove

Usage: morpheus archives remove [bucket]

-y, --yes Auto Confirm
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help
17.4.15 morpheus archives remove-file

Usage: morpheus archives remove-file [bucket:/path]
   -R, --recursive Delete a directory and all of its files. This must be passed if specifying a directory.
   -y, --yes Auto Confirm
   -j, --json JSON Output
   -d, --dry-run Dry Run, print the API request instead of executing it
     --curl Dry Run to output API request as a curl command.
     --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -C, --nocolor Disable ANSI coloring
   -B, --benchmark Print benchmark time after the command is finished.
   -V, --debug Print extra output for debugging.
   -h, --help Print this help

Delete an archive file or directory.

17.4.16 morpheus archives remove-file-link

Usage: morpheus archives remove-file-link [bucket:/path] [token]
   -y, --yes Auto Confirm
   -d, --dry-run Dry Run, print the API request instead of executing it
     --curl Dry Run to output API request as a curl command.
     --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -q, --quiet No Output, do not print to stdout
   -C, --nocolor Disable ANSI coloring
   -B, --benchmark Print benchmark time after the command is finished.
   -V, --debug Print extra output for debugging.
   -h, --help Print this help

Delete a public link to a file.
[bucket:/path] is required. This is the name of the bucket and /path the file or folder to be fetched.[token] is required. This is the secret access key that identifies the link.

17.4.17 morpheus archives rm

Usage: morpheus archives remove-file [bucket:/path]
   -R, --recursive Delete a directory and all of its files. This must be passed if specifying a directory.
   -y, --yes Auto Confirm
   -j, --json JSON Output
   -d, --dry-run Dry Run, print the API request instead of executing it
     --curl Dry Run to output API request as a curl command.
     --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.

(continues on next page)
Delete an archive file or directory.

17.4.18 morpheus archives update

Usage: morpheus archives update [bucket] [options]
    --name VALUE          Name
    --description VALUE   Description
    --payload-file FILE   JSON Payload from a local file
    --visibility [private|public] Visibility determines if read access is restricted to the specified Tenants (Private) or all tenants (Public).
    --accounts LIST       Tenant Accounts (comma separated ids)
    --isPublic [on|off]   Enabling Public URL allows files to be downloaded without any authentication.
    -O, --option OPTION   Option in the format -O field="value"
    -P, --prompt          Always prompts. Use passed options as the default values for all optional fields.
    -N, --no-prompt       Skip prompts. Use default values for all optional fields.
    --payload FILE        Payload from a local JSON or YAML file, skip all prompting
    --payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
    --payload-json JSON   Payload JSON, skip all prompting
    --payload-yaml YAML   Payload YAML, skip all prompting
    -j, --json            JSON Output
    -d, --dry-run         Dry Run, print the API request instead of executing it
    --curl                Dry Run to output API request as a curl command.
    --scrub               Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
    -q, --quiet           No Output, do not print to stdout
    -C, --nocolor         Disable ANSI coloring
    -B, --benchmark       Print benchmark time after the command is finished.
    -V, --debug           Print extra output for debugging.
    -h, --help            Print this help

Update an existing archive bucket.

17.4.19 morpheus archives upload

Usage: morpheus archives upload [local-file] [bucket:/path]
    -R, --recursive       Upload a directory and all of its files. This must be passed if [local-file] is a directory.
    -j, --json            JSON Output
    --ignore-files PATTERN Pattern of files to be ignored when uploading a directory.
    -R, --recursive       Upload a directory and all of its files. This must be passed if [local-file] is a directory.
    -j, --json            JSON Output
    --ignore-files PATTERN Pattern of files to be ignored when uploading a directory.
-y, --yes Auto Confirm
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Upload a local file or folder to an archive bucket. The first argument [local-file] should be the path of a local file or directory. The second argument [bucket:/path] should contain the bucket name. The [:/path] component is optional and can be used to specify the destination of the uploaded file or folder. The default destination is the same name as the [local-file], under the root bucket directory '/'. This will overwrite any existing remote files that match the destination /path.

17.5 morpheus benchmark

Usage: morpheus benchmark [command] [options]
Commands:
  exec
  off
  off?
  on
  on?
  start
  status
  stop

17.5.1 morpheus benchmark exec

Usage: morpheus benchmark exec [command...]
-n, --iterations NUMBER Number of iterations to run. The default is 1.
--name NAME Name for the benchmark. Default is the command itself.
-q, --quiet No Output, do not print to stdout
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Benchmark a specified command. [command] is required. This is the command to execute.
17.5.2 morpheus benchmark off

Usage: morpheus benchmark off
- q, --quiet No Output, do not print to stdout
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

Disable global benchmarking.
The default state for this setting is off.

17.5.3 morpheus benchmark off?

Usage: morpheus benchmark off?
- q, --quiet No Output, do not print to stdout
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

Print the value of the global benchmark setting.
Exit 0 if off.

17.5.4 morpheus benchmark on

Usage: morpheus benchmark on
- q, --quiet No Output, do not print to stdout
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

Enable global benchmarking.
This behaves the same as if you were to add the -B switch to every command.

17.5.5 morpheus benchmark on?

Usage: morpheus benchmark on?
- q, --quiet No Output, do not print to stdout
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

Print the value of the global benchmark setting.
Exit 0 if on.
17.5.6 morpheus benchmark start

Usage: morpheus benchmark start [name]

-quiet
-nocolor
--benchmark
--quiet
--nocolor
--benchmark

Start recording a benchmark.

[name] is required. This is just a name for the routine.
This allows you to record how long it takes to run a series of commands.
Just run 'benchmark stop' when you are finished.

17.5.7 morpheus benchmark status

Usage: morpheus benchmark status [name]

-quiet
-nocolor
--benchmark
--quiet
--nocolor
--benchmark

Print status of benchmark.

[name] is optional. This is the name of the benchmark to inspect.
The last benchmark is used by default.

17.5.8 morpheus benchmark stop

Usage: morpheus benchmark stop [name]

--exit CODE
--error ERROR
-quiet
-nocolor
--benchmark
--exit CODE
--error ERROR
-quiet
-nocolor
--benchmark

Stop recording a benchmark.

[name] is optional. This is the name of the benchmark to stop.
The last benchmark is used by default.

17.6 morpheus blueprints
## 17.6.1 morpheus blueprints add

Usage: morpheus blueprints add [name] [options]

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--name VALUE</td>
<td>Name - Enter a name for this app</td>
</tr>
<tr>
<td>--description VALUE</td>
<td>Description (optional)</td>
</tr>
<tr>
<td>--category VALUE</td>
<td>Category (optional)</td>
</tr>
<tr>
<td>-t, --type TYPE</td>
<td>Blueprint Type. Default is morpheus.</td>
</tr>
<tr>
<td>-O, --option OPTION</td>
<td>Option in the format -O field=&quot;value&quot;</td>
</tr>
<tr>
<td>--prompt</td>
<td>Always prompts. Use passed options as the default value.</td>
</tr>
<tr>
<td>-N, --no-prompt</td>
<td>Skip prompts. Use default values for all optional fields.</td>
</tr>
<tr>
<td>--payload FILE</td>
<td>Payload from a local JSON or YAML file, skip all prompting.</td>
</tr>
<tr>
<td>--payload-dir DIRECTORY</td>
<td>Payload from a local directory containing 1-N JSON or YAML files, skip all prompting</td>
</tr>
<tr>
<td>--payload-json JSON</td>
<td>Payload JSON, skip all prompting</td>
</tr>
<tr>
<td>--payload-yaml YAML</td>
<td>Payload YAML, skip all prompting</td>
</tr>
<tr>
<td>-j, --json</td>
<td>JSON Output</td>
</tr>
<tr>
<td>-d, --dry-run</td>
<td>Dry Run, print the API request instead of executing it</td>
</tr>
<tr>
<td>--executing it</td>
<td>Dry Run to output API request as a curl command.</td>
</tr>
<tr>
<td>--Authorization header. For use with --curl and --dry-run.</td>
<td>Mask secrets in output, such as the Authorization header.</td>
</tr>
<tr>
<td>-q, --quiet</td>
<td>No Output, do not print to stdout</td>
</tr>
<tr>
<td>-r, --remote REMOTE</td>
<td>Remote name. The current remote is used by default.</td>
</tr>
<tr>
<td>--default.</td>
<td>Remote url. The current remote url is used by default.</td>
</tr>
<tr>
<td>-T, --token TOKEN</td>
<td>Access token for authentication with --remote.</td>
</tr>
<tr>
<td>-U, --username USERNAME</td>
<td>Username for authentication.</td>
</tr>
<tr>
<td>-P, --password PASSWORD</td>
<td>Password for authentication.</td>
</tr>
</tbody>
</table>

(continues on next page)
Create a new blueprint.

[name] is required. This is the name of the new blueprint.

17.6.2 morpheus blueprints add-instance

Usage: morpheus blueprints add-instance [id] [tier] [instance-type]

- **--name VALUE** Instance Name
- **-O, --option OPTION** Option in the format `-O field="value"`
- **--prompt** Always prompts. Use passed options as the default value.
- **-N, --no-prompt** Skip prompts. Use default values for all optional fields.
- **-j, --json** JSON Output
- **-d, --dry-run** Dry Run, print the API request instead of executing it.
- **--curl** Dry Run to output API request as a curl command.
- **--scrub** Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- **-r, --remote REMOTE** Remote name. The current remote is used by default.
- **--remote-url URL** Remote url. The current remote url is used by default.
- **-T, --token TOKEN** Access token for authentication with --remote.
- Saved credentials are used by default.
- **-U, --username USERNAME** Username for authentication.
- **-P, --password PASSWORD** Password for authentication.
- **-I, --insecure** Allow insecure HTTPS communication. i.e. bad SSL certificate.
- **-H, --header HEADER** Additional HTTP header to include with requests.
- **--timeout SECONDS** Timeout for api requests. Default is typically 30 seconds.
- **-C, --nocolor** Disable ANSI coloring
- **-B, --benchmark** Print benchmark time after the command is finished.
- **-V, --debug** Print extra output for debugging.
- **-h, --help** Print this help

Update a blueprint, adding an instance.

[id] is required. This is the name or id of a blueprint.
[tier] is required and will be prompted for. This is the name of the tier.
[instance-type] is required and will be prompted for. This is the type of instance.
17.6.3 morpheus blueprints add-instance-config

Usage: morpheus blueprints add-instance-config [id] [tier] [instance]
   -g, --group GROUP Group
   -c, --cloud CLOUD Cloud
   -e, --env ENVIRONMENT Environment
   --name VALUE Instance Name
   -O, --option OPTION Option in the format -O field="value"
   --prompt Always prompts. Use passed options as the default value.
   -N, --no-prompt Skip prompts. Use default values for all optional fields.
   -j, --json JSON Output
   -d, --dry-run Dry Run, print the API request instead of executing it
   --curl Dry Run to output API request as a curl command.
   --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE Remote name. The current remote is used by default.
   --remote-url URL Remote url. The current remote url is used by default.
   -T, --token TOKEN Access token for authentication with --remote.
   Saved credentials are used by default.
   -U, --username USERNAME Username for authentication.
   -P, --password PASSWORD Password for authentication.
   -I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
   --header HEADER Additional HTTP header to include with requests.
   --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
   -C, --nocolor Disable ANSI coloring
   -B, --benchmark Print benchmark time after the command is finished.
   -V, --debug Print extra output for debugging.
   -h, --help Print this help

Update a blueprint, adding an instance config. [id] is required. This is the name or id of a blueprint. [tier] is required. This is the name of the tier. [instance] is required. This is the type of instance.

17.6.4 morpheus blueprints add-tier

Usage: morpheus blueprints add-tier [id] [tier]
   --name VALUE Tier Name
   --bootOrder NUMBER Boot Order
   --linkedTiers x,y,z Connected Tiers.
   --tierIndex NUMBER Tier Index. Used for Display Order
   -O, --option OPTION Option in the format -O field="value"
   --prompt Always prompts. Use passed options as the default value.
   -N, --no-prompt Skip prompts. Use default values for all optional fields.
   -j, --json JSON Output

(continues on next page)
17.6.5 morpheus blueprints available-tiers

Usage: morpheus blueprints available-tiers
   -j, --json                JSON Output
   -d, --dry-run             Dry Run, print the API request instead of executing it
   --curl                    Dry Run to output API request as a curl command.
   --scrub                   Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE      Remote name. The current remote is used by default.
   --remote-url URL         Remote url. The current remote url is used by default.
   -T, --token TOKEN        Access token for authentication with --remote.
   -U, --username USERNAME  Username for authentication.
   -P, --password PASSWORD  Password for authentication.
   -I, --insecure           Allow insecure HTTPS communication. i.e. bad SSL certificate.
   -H, --header HEADER      Additional HTTP header to include with requests.
   --timeout SECONDS        Timeout for api requests. Default is typically 30 seconds.
   -c, --nocolor            Disable ANSI coloring
   -B, --benchmark          Print benchmark time after the command is finished.
   -V, --debug              Print extra output for debugging.
   -h, --help               Print this help

17.6. morpheus blueprints

537
17.6.6 morpheus blueprints connect-tiers

Usage: morpheus blueprints connect-tiers [id] [Tier1] [Tier2]
-\(j\), --json JSON Output
-\(d\), --dry-run Dry Run, print the API request instead of
executing it
-\(c\), --curl Dry Run to output API request as a curl command.
-\(s\), --scrub Mask secrets in output, such as the
Authorization header. For use with --curl and --dry-run.
-\(r\), --remote REMOTE Remote name. The current remote is used by
default.
-\(u\), --remote-url URL Remote url. The current remote url is used by
default.
-\(t\), --token TOKEN Access token for authentication with --remote.
Saved credentials are used by default.
-\(u\), --username USERNAME Username for authentication.
-\(p\), --password PASSWORD Password for authentication.
-\(i\), --insecure Allow insecure HTTPS communication. i.e. bad
SSL certificate.
-\(h\), --header HEADER Additional HTTP header to include with requests.
-\(t\), --timeout SECONDS Timeout for api requests. Default is typically
30 seconds.
-\(c\), --nocolor Disable ANSI coloring
-\(b\), --benchmark Print benchmark time after the command is
finished.
-\(v\), --debug Print extra output for debugging.
-\(h\), --help Print this help

17.6.7 morpheus blueprints disconnect-tiers

Usage: morpheus blueprints disconnect-tiers [id] [Tier1] [Tier2]
-\(j\), --json JSON Output
-\(d\), --dry-run Dry Run, print the API request instead of
executing it
-\(c\), --curl Dry Run to output API request as a curl command.
-\(s\), --scrub Mask secrets in output, such as the
Authorization header. For use with --curl and --dry-run.
-\(r\), --remote REMOTE Remote name. The current remote is used by
default.
-\(u\), --remote-url URL Remote url. The current remote url is used by
default.
-\(t\), --token TOKEN Access token for authentication with --remote.
Saved credentials are used by default.
-\(u\), --username USERNAME Username for authentication.
-\(p\), --password PASSWORD Password for authentication.
-\(i\), --insecure Allow insecure HTTPS communication. i.e. bad
SSL certificate.
-\(h\), --header HEADER Additional HTTP header to include with requests.
-\(t\), --timeout SECONDS Timeout for api requests. Default is typically
30 seconds.
-\(c\), --nocolor Disable ANSI coloring
-\(b\), --benchmark Print benchmark time after the command is
finished.
-\(v\), --debug Print extra output for debugging.
-\(h\), --help Print this help
### 17.6.8 morpheus blueprints duplicate

<table>
<thead>
<tr>
<th>Usage: morpheus blueprints duplicate [id] [new name]</th>
</tr>
</thead>
<tbody>
<tr>
<td>-y, --yes</td>
</tr>
<tr>
<td>-j, --json</td>
</tr>
<tr>
<td>-d, --dry-run</td>
</tr>
<tr>
<td>-c, --curl</td>
</tr>
<tr>
<td>-s, --scrub</td>
</tr>
<tr>
<td>-r, --remote REMOTE</td>
</tr>
<tr>
<td>-u, --remote-url URL</td>
</tr>
<tr>
<td>-T, --token TOKEN</td>
</tr>
<tr>
<td>-U, --username USERNAME</td>
</tr>
<tr>
<td>-P, --password PASSWORD</td>
</tr>
<tr>
<td>-i, --insecure</td>
</tr>
<tr>
<td>-h, --header HEADER</td>
</tr>
<tr>
<td>-t, --timeout SECONDS</td>
</tr>
<tr>
<td>-C, --nocolor</td>
</tr>
<tr>
<td>-B, --benchmark</td>
</tr>
<tr>
<td>-V, --debug</td>
</tr>
<tr>
<td>-h, --help</td>
</tr>
</tbody>
</table>

Duplicate a blueprint.

[id] is required. This is the name or id of a blueprint.

[new name] is required. This is the name for the clone.

### 17.6.9 morpheus blueprints get

<table>
<thead>
<tr>
<th>Usage: morpheus blueprints get [id]</th>
</tr>
</thead>
<tbody>
<tr>
<td>-c, --config</td>
</tr>
<tr>
<td>-j, --json</td>
</tr>
<tr>
<td>-y, --yaml</td>
</tr>
<tr>
<td>-s, --csv</td>
</tr>
<tr>
<td>-c, --csv-delim CHAR</td>
</tr>
<tr>
<td>-n, --csv-newline [CHAR]</td>
</tr>
<tr>
<td>-q, --csv-quotes</td>
</tr>
<tr>
<td>-n, --csv-no-header</td>
</tr>
<tr>
<td>-F, --fields x,y,z</td>
</tr>
<tr>
<td>-o, --out FILE</td>
</tr>
<tr>
<td>-d, --dry-run</td>
</tr>
<tr>
<td>-c, --curl</td>
</tr>
<tr>
<td>-s, --scrub</td>
</tr>
<tr>
<td>-r, --remote REMOTE</td>
</tr>
</tbody>
</table>

(continues on next page)
Get details about a blueprint.

[id] is required. This is the name or id of a blueprint.

## 17.6.10 morpheus blueprints list

Usage: morpheus blueprints list

```
-m, --max MAX
-o, --offset OFFSET
-s, --search PHRASE
-S, --sort ORDER
-D, --desc
-Q, --query PARAMS
-j, --json
-Y, --yaml
-CSV
--csv-delim CHAR
--csv-newline [CHAR]
--csv-quotes
--csv-no-header
-F, --fields x,y,z
-d, --dry-run
--executing it
--curl
--scrub
-r, --remote REMOTE
-U, --username USERNAME
-P, --password PASSWORD
-I, --insecure
-H, --header HEADER
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max MAX</code></td>
<td>Max Results</td>
</tr>
<tr>
<td><code>--offset OFFSET</code></td>
<td>Offset Results</td>
</tr>
<tr>
<td><code>--search PHRASE</code></td>
<td>Search Phrase</td>
</tr>
<tr>
<td><code>--sort ORDER</code></td>
<td>Sort Order</td>
</tr>
<tr>
<td><code>--desc</code></td>
<td>Reverse Sort Order</td>
</tr>
<tr>
<td><code>--query PARAMS</code></td>
<td>Query parameters. PARAMS format is <code>phrase=foobaz&amp;category=web</code></td>
</tr>
<tr>
<td><code>--json</code></td>
<td>JSON Output</td>
</tr>
<tr>
<td><code>--yaml</code></td>
<td>YAML Output</td>
</tr>
<tr>
<td><code>--csv</code></td>
<td>CSV Output</td>
</tr>
<tr>
<td><code>--csv-delim CHAR</code></td>
<td>Delimiter for CSV Output values. Default: ','</td>
</tr>
</tbody>
</table>
| `--csv-newline [CHAR]` | Delimiter for CSV Output rows. Default: '
'                                |
| `--csv-quotes`  | Wrap CSV values with ". Default: false                                       |
| `--csv-no-header` | Exclude header for CSV Output.                                               |
| `--fields x,y,z` | Filter Output to a limited set of fields.                                   |
| `--dry-run`     | Dry Run, print the API request instead of executing it                       |
| `--executing it` | Dry Run to output API request as a curl command.                            |
| `--scrub`       | Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run. |
| `--remote REMOTE` | Remote name. The current remote is used by default.                          |
| `--username USERNAME` | Username for authentication.                                                |
| `--password PASSWORD` | Password for authentication.                                                |
| `--insecure`   | Allow insecure HTTPS communication. i.e. bad SSL certificate.                |
| `--header HEADER` | Additional HTTP header to include with requests.                            |

(continues on next page)
Morpheus Documentation, Release 3.6.2

```
--timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor  Disable ANSI coloring
-B, --benchmark  Print benchmark time after the command is finished.
-V, --debug  Print extra output for debugging.
-h, --help  Print this help

List blueprints.
```

### 17.6.11 morpheus blueprints remove

**Usage:** morpheus blueprints remove [id]

- **-y, --yes** Auto Confirm
- **-j, --json** JSON Output
- **-d, --dry-run** Dry Run, print the API request instead of executing it
  - **--curl** Dry Run to output API request as a curl command.
  - **--scrub** Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- **-r, --remote REMOTE** Remote name. The current remote is used by default.
- **--remote-url URL** Remote url. The current remote url is used by default.
- **-T, --token TOKEN** Access token for authentication with --remote.
- **-U, --username USERNAME** Username for authentication.
- **-P, --password PASSWORD** Password for authentication.
- **-I, --insecure** Allow insecure HTTPS communication. i.e. bad SSL certificate.
- **-H, --header HEADER** Additional HTTP header to include with requests.
  - **--timeout SECONDS** Timeout for api requests. Default is typically 30 seconds.
- **-C, --nocolor** Disable ANSI coloring
- **-B, --benchmark** Print benchmark time after the command is finished.
- **-V, --debug** Print extra output for debugging.
- **-h, --help** Print this help

Delete a blueprint. [id] is required. This is the name or id of a blueprint.

### 17.6.12 morpheus blueprints remove-instance

**Usage:** morpheus blueprints remove-instance [id] [tier] [instance]

- **-y, --yes** Auto Confirm
- **-j, --json** JSON Output
- **-d, --dry-run** Dry Run, print the API request instead of executing it
  - **--curl** Dry Run to output API request as a curl command.
  - **--scrub** Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- **-r, --remote REMOTE** Remote name. The current remote is used by default.
- **--remote-url URL** Remote url. The current remote url is used by default.
- **-T, --token TOKEN** Access token for authentication with --remote.
- **-U, --username USERNAME** Username for authentication.
- **-P, --password PASSWORD** Password for authentication.
- **-I, --insecure** Allow insecure HTTPS communication. i.e. bad SSL certificate.
- **-H, --header HEADER** Additional HTTP header to include with requests.
  - **--timeout SECONDS** Timeout for api requests. Default is typically 30 seconds.
- **-C, --nocolor** Disable ANSI coloring
- **-B, --benchmark** Print benchmark time after the command is finished.
- **-V, --debug** Print extra output for debugging.
- **-h, --help** Print this help

(continues on next page)
Update a blueprint, removing a specified instance config. [id] is required. This is the name or id of a blueprint. [tier] is required. This is the name of the tier. [instance] is required. This is the type of instance. The config scope is specified with the -g GROUP, -c CLOUD and -e ENV. The -g and -c options are required.

(continues on next page)
17.6.14 morpheus blueprints remove-tier

Usage: morpheus blueprints remove-tier [id] [tier]
   -y, --yes           Auto Confirm
   -j, --json          JSON Output
   -d, --dry-run       Dry Run, print the API request instead of
   --executing it
       --curl          Dry Run to output API request as a curl command.
       --scrub         Mask secrets in output, such as the
   --Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE Remote name. The current remote is used by
   --default.
       --remote-url URL Remote url. The current remote url is used by
   --default.
   -T, --token TOKEN   Access token for authentication with --remote.
   --Saved credentials are used by default.
   -U, --username USERNAME Username for authentication.
   -P, --password PASSWORD Password for authentication.
   -I, --insecure      Allow insecure HTTPS communication. i.e. bad
   --SSL certificate.
   -H, --header HEADER Additional HTTP header to include with requests.
   --timeout SECONDS  Timeout for api requests. Default is typically
   30 seconds.
   -C, --nocolor       Disable ANSI coloring
   -B, --benchmark     Print benchmark time after the command is
   --finished.
   -V, --debug         Print extra output for debugging.
   -h, --help          Print this help

17.6.15 morpheus blueprints types

Usage: morpheus blueprints types
   -j, --json          JSON Output
   -yml               YAML Output
   -csv               CSV Output
   --csv-delim CHAR   Delimiter for CSV Output values. Default: ','
   --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '
'
   --csv-quotes       Wrap CSV values with ''. Default: false
   --csv-no-header    Exclude header for CSV Output.
   --Default is all fields.
   --fields x,y,z     Filter Output to a limited set of fields.
   -d, --dry-run      Dry Run, print the API request instead of
   --executing it
       --curl          Dry Run to output API request as a curl command.
       --scrub         Mask secrets in output, such as the
   --Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE Remote name. The current remote is used by
   --default.
       --remote-url URL Remote url. The current remote url is used by
   --default.
   -T, --token TOKEN   Access token for authentication with --remote.
   --Saved credentials are used by default.
-U, --username USERNAME  Username for authentication.
-P, --password PASSWORD  Password for authentication.
-I, --insecure  Allow insecure HTTPS communication. i.e. bad
-SSL certificate.
-H, --header HEADER  Additional HTTP header to include with requests.
   --timeout SECONDS  Timeout for api requests. Default is typically
   30 seconds.
-C, --nocolor  Disable ANSI coloring
-B, --benchmark  Print benchmark time after the command is
   finished.
-V, --debug  Print extra output for debugging.
-h, --help  Print this help

List blueprint types.

17.6.16 morpheus blueprints update

Usage: morpheus blueprints update [id] [options]
   --name VALUE  Name (optional) - Enter a name for this app
   --category VALUE  Category (optional)
   -O, --option OPTION  Option in the format -O field="value"
   --prompt  Always prompts. Use passed options as the
   default value.
   -N, --no-prompt  Skip prompts. Use default values for all
   optional fields.
   --payload FILE  Payload from a local JSON or YAML file, skip all
   prompting
   --payload-dir DIRECTORY  Payload from a local directory containing 1-N
   JSON or YAML files, skip all prompting
   --payload-json JSON  Payload JSON, skip all prompting
   --payload-yaml YAML  Payload YAML, skip all prompting
   -j, --json  JSON Output
   -d, --dry-run  Dry Run, print the API request instead of
   executing it
   --curl  Dry Run to output API request as a curl command.
   --scrub  Mask secrets in output, such as the
   Authorization header. For use with --curl and --dry-run.
   -q, --quiet  No Output, do not print to stdout
   -r, --remote REMOTE  Remote name. The current remote is used by
   default.
   --remote-url URL  Remote url. The current remote url is used by
   default.
   --token TOKEN  Access token for authentication with --remote.
   Saved credentials are used by default.
   -U, --username USERNAME  Username for authentication.
   -P, --password PASSWORD  Password for authentication.
   -I, --insecure  Allow insecure HTTPS communication. i.e. bad
-SSL certificate.
-H, --header HEADER  Additional HTTP header to include with requests.
   --timeout SECONDS  Timeout for api requests. Default is typically
   30 seconds.
-C, --nocolor  Disable ANSI coloring
-B, --benchmark  Print benchmark time after the command is
   finished.
-V, --debug  Print extra output for debugging.
Update a blueprint.  
**[id]** is required. This is the name or id of a blueprint.  
**[options]** Available options include --name and --description. This will update only the specified values.  
[--config] or [--config-file] can be used to replace the entire blueprint.

### 17.6.17 morpheus blueprints update-permissions

**Usage:** `morpheus blueprints update-permissions [id] [options]`

- **--group-access-all** [on|off] Toggle Access for all groups.  
- **--group-access** LIST Group Access, comma separated list of group IDs.  
- **--visibility** [private|public] Visibility  
- **--name** VALUE Name (optional) - Enter a name for this app  
- **--category** VALUE Category (optional)  
- **-O, --option** OPTION Option in the format -O field="value"  
- **--prompt** Always prompts. Use passed options as the default value.  
- **--no-prompt** Skip prompts. Use default values for all optional fields.  
- **--payload** FILE Payload from a local JSON or YAML file, skip all prompting  
- **--payload-dir** DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting  
- **--payload-json** JSON Payload JSON, skip all prompting  
- **--payload-yaml** YAML Payload YAML, skip all prompting  
- **--json** JSON Output  
- **--dry-run** Dry Run, print the API request instead of executing it  
- **--curl** Dry Run to output API request as a curl command.  
- **--scrub** Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.  
- **-q, --quiet** No Output, do not print to stdout  
- **-r, --remote** REMOTE Remote name. The current remote is used by default.  
- **--remote-url** URL Remote url. The current remote url is used by default.  
- **--token** TOKEN Access token for authentication with --remote.  
- **-U, --username** USERNAME Username for authentication.  
- **-P, --password** PASSWORD Password for authentication.  
- **-I, --insecure** Allow insecure HTTPS communication. i.e. bad SSL certificate.  
- **-H, --header** HEADER Additional HTTP header to include with requests.  
- **--timeout** SECONDS Timeout for api requests. Default is typically 30 seconds.  
- **-C, --nocolor** Disable ANSI coloring  
- **-B, --benchmark** Print benchmark time after the command is finished.  
- **-V, --debug** Print extra output for debugging.  
- **-h, --help** Print this help

Update a blueprint permissions.
[id] is required. This is the name or id of a blueprint.

### 17.6.18 morpheus blueprints update-tier

Usage: morpheus blueprints update-tier [id] [tier]
- `--name VALUE` Tier Name
- `--bootOrder NUMBER` Boot Order
- `--linkedTiers x,y,z` Connected Tiers
- `--tierIndex NUMBER` Tier Index. Used for Display Order
- `--option OPTION` Option in the format -O field="value"
- `--prompt` Always prompts. Use passed options as the default value.
- `-N, --no-prompt` Skip prompts. Use default values for all optional fields.
- `--option OPTION` Option in the format -O field="value"
- `--remote REMOTE` Remote name. The current remote is used by default.
- `--remote-url URL` Remote url. The current remote url is used by default.
- `--token TOKEN` Access token for authentication with --remote.
- `--username USERNAME` Username for authentication.
- `--password PASSWORD` Password for authentication.
- `--insecure` Allow insecure HTTPS communication. i.e. bad SSL certificate.
- `--header HEADER` Additional HTTP header to include with requests.
- `--timeout SECONDS` Timeout for api requests. Default is typically 30 seconds.
- `-C, --nocolor` Disable ANSI coloring
- `-B, --benchmark` Print benchmark time after the command is finished.
- `-v, --debug` Print extra output for debugging.
- `-h, --help` Print this help

### 17.6.19 morpheus blueprints upload-image

Usage: morpheus blueprints upload-image [id] [file]
- `--json` JSON Output
- `--dry-run` Dry Run, print the API request instead of executing it
- `--curl` Dry Run to output API request as a curl command.
- `--scrub` Mask secrets in output, such as the Authorization header.
- `--quiet` No Output, do not print to stdout
- `--remote REMOTE` Remote name. The current remote is used by default.
- `--remote-url URL` Remote url. The current remote url is used by default.
- `--token TOKEN` Access token for authentication with --remote.
Usage: morpheus clouds [command] [options]

Commands:

- add
  - apply-security-groups
  - count
  - firewall-disable
  - firewall-enable
  - get
  - list
  - remove
  - security-groups
  - types
  - update

17.7.1 morpheus clouds add

Usage: morpheus clouds add [name] --group GROUP --type TYPE
  -g, --group GROUP
  -t, --type TYPE
  --description DESCRIPTION
  --certificate-provider CODE
  -O, --option OPTION
  --prompt Always prompts. Use passed options as the default value.
  -N, --no-prompt
  --payload FILE
  --payload-dir DIRECTORY
  --payload-json JSON
  --payload-yaml YAML
  --optional-fields.
  --prompting
  --prompting-field FILE
  --prompting-fields DIRECTORY
  --prompting-json JSON
  --prompting-yaml YAML

Upload an image file to be used as the icon for a blueprint.
[id] is required. This is the name or id of a blueprint.
[file] is required. This is the local path of a file to upload [png|jpg|svg].
17.7.2 morpheus clouds apply-security-groups

Usage: morpheus clouds apply-security-groups [name] [-s] [--clear]
-c, --clear Clear all security groups
-s, --secgroupps SECGROUPS Apply the specified comma separated security groups
-agroup ids
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header.
--remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

(continues on next page)
17.7.3 morpheus clouds count

Usage: morpheus clouds count [options]
- Q, --query PARAMS Query parameters. PARAMS format is
- 'phrase=foobar&category=web'
- r, --remote REMOTE Remote name. The current remote is used by default.
- default.
- remote-url URL Remote url. The current remote url is used by default.
- T, --token TOKEN Access token for authentication with --remote.
- Saved credentials are used by default.
- u, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
- timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- d, --dry-run Dry Run, print the API request instead of executing it.
- curl Dry Run to output API request as a curl command.
- scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

Get the number of clouds.

17.7.4 morpheus clouds firewall-disable

Usage: morpheus clouds firewall-disable [name]
- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of executing it.
- curl Dry Run to output API request as a curl command.
- scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by default.
- remote-url URL Remote url. The current remote url is used by default.
- T, --token TOKEN Access token for authentication with --remote.
- Saved credentials are used by default.
- u, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.

(continues on next page)
17.7.5 morpheus clouds firewall-enable

Usage: morpheus clouds firewall-enable [name]

- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by default.
- T, --token TOKEN Access token for authentication with --remote.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
- curl Dry Run to output API request as a curl command.
- nocolor Disable ANSI coloring
- benchmark Print benchmark time after the command is finished.
- v, --debug Print extra output for debugging.
- h, --help Print this help

17.7.6 morpheus clouds get

Usage: morpheus clouds get [name]

- j, --json JSON Output
- y, --yaml YAML Output
- csv CSV Output
  --csv-delim CHAR Delimiter for CSV Output values. Default: ','
  --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
  --csv-quotes Wrap CSV values with ".
  --csv-no-header Exclude header for CSV Output.
  -F, --fields x,y,z Filter Output to a limited set of fields.
  -d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- h, --help Print this help
17.7.7 morpheus clouds list

Usage: morpheus clouds list
- g, --group GROUP Group Name
- t, --type TYPE Cloud Type
- m, --max MAX Max Results
- o, --offset OFFSET Offset Results
- s, --search PHRASE Search Phrase
- S, --sort ORDER Sort Order
- D, --desc Reverse Sort Order
- Q, --query PARAMS Query parameters. PARAMS format is 'phrase=foobar&category=web'
- j, --json JSON Output
- y, --yaml YAML Output
- csv CSV Output
- csv-delim CHAR Delimiter for CSV Output values. Default: ','
- csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
- csv-quotes Wrap CSV values with "." Default: false
- csv-no-header Exclude header for CSV Output.
- F, --fields x,y,z Filter Output to a limited set of fields. Default is all fields.
- d, --dry-run Dry Run, print the API request instead of executing it
- curl Dry Run to output API request as a curl command.
- scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by default.
- default Remote url. The current remote url is used by default.
- T, --token TOKEN Access token for authentication with --remote.
- default Saved credentials are used by default.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
- timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

Get details about a cloud. [name] is required. This is the name or id of a cloud.
Morpheus Documentation, Release 3.6.2

(continued from previous page)

-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
   --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

List clouds.

17.7.8 morpheus clouds remove

Usage: morpheus clouds remove [name]
-f, --force Force Remove
-y, --yes Auto Confirm
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
   --curl Dry Run to output API request as a curl command.
   --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
   --remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
   --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.7.9 morpheus clouds security-groups

Usage: morpheus clouds security-groups [name]
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
   --curl Dry Run to output API request as a curl command.
   --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
17.7.10 morpheus clouds types

Usage: morpheus clouds types
- m, --max MAX Max Results
- o, --offset OFFSET Offset Results
- s, --search PHRASE Search Phrase
- S, --sort ORDER Sort Order
- D, --desc Reverse Sort Order
- Q, --query PARAMS Query parameters. PARAMS format is
  'phrase=foobar&category=web'
- j, --json JSON Output
- y, --yaml YAML Output
- c, --csv CSV Output
- csv-delim CHAR Delimiter for CSV Output values. Default: ','
- csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
- csv-quotes Wrap CSV values with "". Default: false
- csv-no-header Exclude header for CSV Output.
- F, --fields x,y,z Filter CSV Output to a limited set of fields.
- d, --dry-run Dry Run, print the API request instead of executing it
- d, --dry-run Dry Run to output API request as a curl command.
- Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by default.
- r, --remote REMOTE Remote name. The current remote is used by default.
- remote-url URL Remote url. The current remote url is used by default.
- T, --token TOKEN Access token for authentication with --remote.
- Saved credentials are used by default.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
- timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

17.7. morpheus clouds
17.7.11 morpheus clouds update

```
Usage: morpheus clouds update [name] [options]

-O, --option OPTION Option in the format -O field="value"
--prompt Always prompts. Use passed options as the default value.
-N, --no-prompt Skip prompts. Use default values for all optional fields.
--payload FILE Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help
```

17.8 morpheus containers

```
Usage: morpheus containers [command] [options]

Commands:
  action
  actions
  eject
```

(continues on next page)
17.8.1 morpheus containers action

Usage: morpheus containers action [id list] -a CODE

- a, --action CODE Container Action CODE to execute
- y, --yes Auto Confirm
- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of executing it
- Authorization header. For use with --curl and --dry-run.
- q, --quiet No Output, do not print to stdout
- r, --remote REMOTE Remote name. The current remote is used by default.
- --remote-url URL Remote url. The current remote url is used by default.
- T, --token TOKEN Access token for authentication with --remote.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
- C, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

Execute an action for a container or containers

17.8.2 morpheus containers actions

Usage: morpheus containers actions [id list]

- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of executing it
- Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by default.
- --remote-url URL Remote url. The current remote url is used by default.

(continues on next page)
-T, --token TOKEN  | Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME  | Username for authentication.
-P, --password PASSWORD  | Password for authentication.
-I, --insecure  | Allow insecure HTTPS communication. i.e. bad
-SSL certificate.
-H, --header HEADER  | Additional HTTP header to include with requests.
--timeout SECONDS  | Timeout for api requests. Default is typically
30 seconds.
-C, --nocolor  | Disable ANSI coloring
-B, --benchmark  | Print benchmark time after the command is finished.
-V, --debug  | Print extra output for debugging.
-h, --help  | Print this help

This outputs the list of the actions available to specified container(s).

### 17.8.3 morpheus containers eject

**Usage:** morpheus containers eject [id list]

- y, --yes  | Auto Confirm
- j, --json  | JSON Output
- d, --dry-run  | Dry Run, print the API request instead of executing it
--curl  | Dry Run to output API request as a curl command.
--scrub  | Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-q, --quiet  | No Output, do not print to stdout
-r, --remote REMOTE  | Remote name. The current remote is used by default.
--remote-url URL  | Remote url. The current remote url is used by default.
-T, --token TOKEN  | Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME  | Username for authentication.
-P, --password PASSWORD  | Password for authentication.
-I, --insecure  | Allow insecure HTTPS communication. i.e. bad
-SSL certificate.
-H, --header HEADER  | Additional HTTP header to include with requests.
--timeout SECONDS  | Timeout for api requests. Default is typically
30 seconds.
-C, --nocolor  | Disable ANSI coloring
-B, --benchmark  | Print benchmark time after the command is finished.
-V, --debug  | Print extra output for debugging.
-h, --help  | Print this help

### 17.8.4 morpheus containers exec

**Usage:** morpheus containers exec [id] [options]

--script SCRIPT  | Script to be executed
--file FILE  | File containing the script. This can be used instead of --script

(continues on next page)
17.8.5 morpheus containers get

Usage: morpheus containers get [name]

--actions Display Available Actions
--refresh [SECONDS] Refresh until status is running,failed. Default
--refresh-until STATUS Refresh until a specified status is reached.
-j, --json JSON Output
-yaml YAML Output
--csv CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '
'
--csv-quotes Wrap CSV values with ''. Default: false

Executed an arbitrary command or script on a container.
{id} is required. This is the id of a container.
[script] is required. This is the script that is to be executed.
17.8.6 morpheus containers restart

Usage: morpheus containers restart [id list]

-y, --yes          Auto Confirm
-j, --json         JSON Output
-d, --dry-run      Dry Run, print the API request instead of
-executing it      Dry Run to output API request as a curl command.
--curl             Mask secrets in output, such as the
--scrub            Authorization header. For use with --curl and --dry-run.

-r, --remote REMOTE Remote name. The current remote is used by
-default.          Remote url. The current remote url is used by
--remote-url URL   --default.

-T, --token TOKEN  Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure     Allow insecure HTTPS communication. i.e. bad
--ssl certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS  Timeout for api requests. Default is typically
-30 seconds.
-C, --nocolor      Disable ANSI coloring
-B, --benchmark    Print benchmark time after the command is
-finished.
-V, --debug        Print extra output for debugging.
-h, --help         Print this help
17.8.7 morpheus containers start

Usage: morpheus containers start [id list]

-foo, --yes Auto Confirm
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
  --remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-q, --quiet No Output, do not print to stdout
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.8.8 morpheus containers stop

Usage: morpheus containers stop [id list]

-foo, --yes Auto Confirm
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
  --remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-q, --quiet No Output, do not print to stdout
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.8.9 morpheus containers suspend

Usage: morpheus containers suspend [id list]
- y, --yes Auto Confirm
- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of executing it
   --curl Dry Run to output API request as a curl command.
   --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- q, --quiet No Output, do not print to stdout
- r, --remote REMOTE Remote name. The current remote is used by default.
   --remote-url URL Remote url. The current remote url is used by default.
   - T, --token TOKEN Access token for authentication with --remote.
   Saved credentials are used by default.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.9 morpheus cypher

Usage: morpheus cypher [command] [options]
Commands:
  get
  list
  put
  remove

560 Chapter 17. Morpheus Commands
### 17.9.1 morpheus cypher get

**Usage:** morpheus cypher get [key]

- `-v, --value` Print only the decrypted value.
- `-t, --ttl SECONDS` Time to live, the lease duration before this key expires. Use if creating new key.
- `-e` Use if creating new key.
- `-j, --json` JSON Output
- `-y, --yaml` YAML Output
- `-c, --csv` CSV Output
- `-d, --csv-delim CHAR` Delimiter for CSV Output values. Default: ','
- `-n, --csv-newline [CHAR]` Delimiter for CSV Output rows. Default: '\n'
- `-q, --csv-quotes` Wrap CSV values with ''. Default: false
- `-n, --csv-no-header` Exclude header for CSV Output.
- `-F, --fields x,y,z` Filter Output to a limited set of fields.
- `-d, --dry-run` Dry Run, print the API request instead of executing it
- `-c, --curl` Dry Run to output API request as a curl command.
- `-s, --scrub` Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- `-q, --quiet` No Output, do not print to stdout
- `-r, --remote REMOTE` Remote name. The current remote is used by default.
- `-u, --remote-url URL` Remote url. The current remote url is used by default.
- `-t, --token TOKEN` Access token for authentication with --remote.
- `-U, --username USERNAME` Username for authentication.
- `-P, --password PASSWORD` Password for authentication.
- `-i, --insecure` Additional HTTP header to include with requests. Use --insecure to allow insecure HTTPS communication. i.e. bad SSL certificate.
- `-H, --header HEADER` Additional HTTP header to include with requests.
- `-T, --timeout SECONDS` Timeout for api requests. Default is typically 30 seconds.
- `-c, --nocolor` Disable ANSI coloring
- `-B, --benchmark` Print benchmark time after the command is finished.
- `-V, --debug` Print extra output for debugging.
- `-h, --help` Print this help

Read a cypher item and display the decrypted value.

[key] is required. This is the cypher key to read.

Use --ttl to specify a ttl if expecting cypher engine to automatically create the key.

### 17.9.2 morpheus cypher list

**Usage:** morpheus cypher list [key]

- `-m, --max MAX` Max Results
- `-o, --offset OFFSET` Offset Results
- `-s, --search PHRASE` Search Phrase
- `-S, --sort ORDER` Sort Order
- `-D, --desc` Reverse Sort Order
- `-Q, --query PARAMS` Query parameters. PARAMS format is `phrase=foobar&category=web`
- `-y, --yaml` YAML Output

(continues on next page)
```
--csv
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
--csv-quotes Wrap CSV values with "". Default: false
--csv-no-header Exclude header for CSV Output.
-F, --fields x,y,z Filter Output to a limited set of fields.
  Default is all fields.
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote. Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
-T, --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --help Print this help

List cypher keys.
[key] is optional. This is the cypher key or path to search for.

17.9.3 morpheus cypher put

Usage: morpheus cypher put [key] [value]

-v, --value VALUE Secret value
-t, --ttl SECONDS Time to live, the lease duration before this key expires.

-y, --yes Auto Confirm
-O, --option OPTION Option in the format -O field="value"

-prompt Always prompts. Use passed options as the default value.
-N, --no-prompt Skip prompts. Use default values for all optional fields.

-F, --fields x,y,z Filter Output to a limited set of fields.

--json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
```
Create or update a cypher key.
[key] is required. This is the key of the cypher being created or updated.
[value] is required. This is the new value or value pairs being stored. Supports
--format foo=bar, 1-N arguments.
The --payload option can be used instead of passing [value] argument.

17.9.4 morpheus cypher remove

Usage: morpheus cypher remove [key]
  -y, --yes Auto Confirm
  -j, --json JSON Output
  -d, --dry-run Dry Run, print the API request instead of
  --executing it
    --curl Dry Run to output API request as a curl command.
    --scrub Mask secrets in output, such as the
  --Authorization header. For use with --curl and --dry-run.
    -q, --quiet No Output, do not print to stdout
    -r, --remote REMOTE Remote name. The current remote is used by
  --default.
    --remote-url URL Remote url. The current remote url is used by
  --default.
  -T, --token TOKEN Access token for authentication with --remote.
  --Saved credentials are used by default.
    -U, --username USERNAME Username for authentication.
    -P, --password PASSWORD Password for authentication.
    -I, --insecure Allow insecure HTTPS communication. i.e. bad
  --SSL certificate.
    -H, --header HEADER Additional HTTP header to include with requests.
    --timeout SECONDS Timeout for api requests. Default is typically
  --30 seconds.
    -C, --nocolor Disable ANSI coloring
    -B, --benchmark Print benchmark time after the command is
  --finished.
    -V, --debug Print extra output for debugging.
    -h, --help Print this help
(continues on next page)
-V, --debug Print extra output for debugging.
-h, --help Print this help

Delete a cypher.
[key] is required. This is the key of a cypher.

## 17.10 morpheus datastores

Usage: morpheus datastores [command] [options]

Commands:
  get
  list
  update

## 17.10.1 morpheus datastores get

Usage: morpheus datastores get [datastore]

- c, --cloud CLOUD Cloud Name or ID
- j, --json JSON Output
  --yaml YAML Output
  --csv CSV Output
  --csv-delim CHAR Delimiter for CSV Output values. Default: ','
  --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
  --csv-quotes Wrap CSV values with '. Default: false
  --csv-no-header Exclude header for CSV Output.
  -F, --fields x,y,z Filter Output to a limited set of fields.
  Default is all fields.
- d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by default.
  --remote-url URL Remote url. The current remote url is used by default.
  --token TOKEN Access token for authentication with --remote.
  Saved credentials are used by default.
- u, --username USERNAME Username for authentication.
- p, --password PASSWORD Password for authentication.
- i, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
  --header HEADER Additional HTTP header to include with requests.
  --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- n, --nocolor Disable ANSI coloring
- b, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

Get details about a datastore.
[datastore] is required. This is the name or id of a datastore.
-c [cloud] is required. This is the name or id of the cloud.

17.10.2 morpheus datastores list

Usage: morpheus datastores list -c [cloud]
   -c, --cloud CLOUD  Cloud Name or ID
   -m, --max MAX      Max Results
   -o, --offset OFFSET Offset Results
   -s, --search PHRASE Search Phrase
   -S, --sort ORDER   Sort Order
   -D, --desc         Reverse Sort Order
       --yaml         YAML Output
       --csv          CSV Output
       --csv-delim CHAR Delimiter for CSV Output values. Default: ','
       --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '
       --csv-quotes Wrap CSV values with '. Default: false
       --csv-no-header Exclude header for CSV Output.
       -F, --fields x,y,z Filter Output to a limited set of fields. Default is all fields.
       -j, --json        JSON Output
       -d, --dry-run     Dry Run, print the API request instead of executing it
       --curl           Dry Run to output API request as a curl command.
       --scrub          Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
       -r, --remote REMOTE Remote name. The current remote is used by default.
       --remote-url URL Remote url. The current remote url is used by default.
       -T, --token TOKEN Access token for authentication with --remote.
       --ssl certificate Additional HTTPS communication. i.e. bad SSL certificate.
       -H, --header HEADER Additional HTTP header to include with requests.
       --timeout SECONDS Timeout for API requests. Default is typically 30 seconds.
       -C, --nocolor     Disable ANSI coloring
       -B, --benchmark   Print benchmark time after the command is finished.
       -V, --debug       Print extra output for debugging.
       -h, --help        Print this help

List datastores for a cloud.
-c [cloud] is required. This is the name or id of the cloud.

17.10.3 morpheus datastores update

Usage: morpheus datastores update [datastore] -c [cloud] [options]
   -c, --cloud CLOUD  Cloud Name or ID
   -m, --max MAX      Max Results
   -o, --offset OFFSET Offset Results
   -s, --search PHRASE Search Phrase
   -S, --sort ORDER   Sort Order
   -D, --desc         Reverse Sort Order
       --yaml         YAML Output
       --csv          CSV Output
       --csv-delim CHAR Delimiter for CSV Output values. Default: ','
       --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '
       --csv-quotes Wrap CSV values with '. Default: false
       --csv-no-header Exclude header for CSV Output.
       -F, --fields x,y,z Filter Output to a limited set of fields. Default is all fields.
       -j, --json        JSON Output
       -d, --dry-run     Dry Run, print the API request instead of executing it
       --curl           Dry Run to output API request as a curl command.
       --scrub          Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
       -r, --remote REMOTE Remote name. The current remote is used by default.
       --remote-url URL Remote url. The current remote url is used by default.
       -T, --token TOKEN Access token for authentication with --remote.
       --ssl certificate Additional HTTPS communication. i.e. bad SSL certificate.
       -H, --header HEADER Additional HTTP header to include with requests.
       --timeout SECONDS Timeout for API requests. Default is typically 30 seconds.
       -C, --nocolor     Disable ANSI coloring
       -B, --benchmark   Print benchmark time after the command is finished.
       -V, --debug       Print extra output for debugging.
       -h, --help        Print this help
--group-access-all [on|off] Toggle Access for all groups.
--group-access LIST Group Access, comma separated list of group IDs.
--tenants LIST Tenant Access, comma separated list of account IDs.
--visibility [private|public] Visibility

-O, --option OPTION Option in the format -O field="value"
--prompt Always prompts. Use passed options as the default value.
-N, --no-prompt Skip prompts. Use default values for all optional fields.
--payload FILE Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting
--executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
--remote-url REMOTE Remote name. The current remote is used by default.
--token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
--username USERNAME Username for authentication.
--password PASSWORD Password for authentication.
--insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
--nocolor Disable ANSI coloring
--benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Update a datastore.
[datastore] is required. This is the id of a datstore.

17.11 morpheus deploy

Usage: morpheus deploy [environment]
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
--token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Deploy to an environment using the morpheus.yml file, located in the working directory.

17.12 morpheus deployments

Usage: morpheus deployments [command] [options]

Commands:
add
list
remove
update
versions

17.12.1 morpheus deployments add

Usage: morpheus deployments add [name]

--description DESCRIPTION Description
-O, --option OPTION Option in the format -O field="value"
--prompt Always prompts. Use passed options as the default value.
-N, --no-prompt Skip prompts. Use default values for all optional fields.
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote. Saved credentials are used by default.

(continues on next page)
17.12.2 morpheus deployments list

Usage: morpheus deployments list

- m, --max MAX Max Results
- o, --offset OFFSET Offset Results
- s, --search PHRASE Search Phrase
- S, --sort ORDER Sort Order
- D, --desc Reverse Sort Order
- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of 

executing it
- curl Dry Run to output API request as a curl command.
- scrub Mask secrets in output, such as the

Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by 

default.
- T, --token TOKEN Access token for authentication with --remote.

Saved credentials are used by default.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. i.e. bad

SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
- timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is 

finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

17.12.3 morpheus deployments remove

Usage: morpheus deployments remove [name]

- y, --yes Auto Confirm
- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of

executing it
17.12.4 morpheus deployments update

Usage: morpheus deployments update [name] [options]
-0, --option OPTION Option in the format -O field="value"
--prompt Always prompts. Use passed options as the value.
--default value.
-N, --no-prompt Skip prompts. Use default values for all optional fields.
--executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
--saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help
17.12.5 morpheus deployments versions

Usage: morpheus deployments versions [name] versions

-\m, --max MAX Max Results
-\o, --offset OFFSET Offset Results
-\s, --search PHRASE Search Phrase
-\S, --sort ORDER Sort Order
-\D, --desc Reverse Sort Order
-\j, --json JSON Output
-\d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-\r, --remote REMOTE Remote name. The current remote is used by default.
  --remote-url URL Remote url. The current remote url is used by default.
-\T, --token TOKEN Access token for authentication with --remote.
  Saved credentials are used by default.
-\U, --username USERNAME Username for authentication.
-\P, --password PASSWORD Password for authentication.
-\I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-\H, --header HEADER Additional HTTP header to include with requests.
  --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-\C, --nocolor Disable ANSI coloring
-\B, --benchmark Print benchmark time after the command is finished.
-\V, --debug Print extra output for debugging.
-\h, --help Print this help

17.13 morpheus edit-profile

Usage: morpheus edit-profile

-\e, --editor PROGRAM Editor program to use. The default is $EDITOR.
-\C, --nocolor Disable ANSI coloring
-\B, --benchmark Print benchmark time after the command is finished.
-\V, --debug Print extra output for debugging.
-\h, --help Print this help

Edit your interactive shell script at /Users/jwheeler/.morpheus/.morpheus_profile

17.14 morpheus edit-rc

Usage: morpheus edit-rc

-\e, --editor PROGRAM Editor program to use. The default is $EDITOR.
-\C, --nocolor Disable ANSI coloring
-\B, --benchmark Print benchmark time after the command is finished.
-V, --debug                      Print extra output for debugging.
-h, --help                       Print this help

Edit your morpheus initialization script at /Users/jwheeler/.morpheus/.morpheusrc

17.15  morpheus execute-schedules

Usage: morpheus execute-schedules [command] [options]

Commands:
  add
  add-hosts
  add-instances
  get
  list
  remove
  remove-hosts
  remove-instances
  update

17.15.1  morpheus execute-schedules add

Usage: morpheus execute-schedules add [name]
  --name VALUE                  Name
  --description VALUE           Description
  --type [execute]              Type of Schedule. Default is 'execute'
  --timezone CODE              The timezone. Default is UTC.
  --cron EXPRESSION            Cron Expression. Default is daily at midnight '0 0 * * *'
  --enabled [on|off]           Can be used to disable it
  -O, --option OPTION          Option in the format -O field="value"
  --prompt                     Always prompts. Use passed options as the default value.
  -N, --no-prompt              Skip prompts. Use default values for all optional fields.
  --payload FILE               Payload from a local JSON or YAML file, skip all prompting
  --payload-dir DIRECTORY      Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
  --payload-json JSON          Payload JSON, skip all prompting
  --payload-yaml YAML          Payload YAML, skip all prompting
  -j, --json                    JSON Output
  -d, --dry-run                 Dry Run, print the API request instead of executing it
  --curl                       Dry Run to output API request as a curl command.
  --scrub                      Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -r, --remote REMOTE          Remote name. The current remote is used by default.
  --remote-url URL             Remote url. The current remote url is used by default.
  -T, --token TOKEN            Access token for authentication with --remote.
  --remote-cred PATH           Path to credentials file.
Create a new execute schedule.
[name] is required and can be passed as --name instead.

17.15.2 morpheus execute-schedules add-hosts

Usage: morpheus execute-schedules add-hosts [name] [host]

Payload from a local JSON or YAML file, skip all prompting
--payload FILE
--payload-dir DIRECTORY
--payload-json JSON
--payload-yaml YAML

JSON or YAML files, skip all prompting
--json
--dry-run

Dry Run, print the API request instead of executing it
--curl
--scrub

Dry Run to output API request as a curl command.

Authorization header. For use with --curl and --dry-run.
--remote REMOTE
--remote-url URL

Remote name. The current remote is used by default.

Access token for authentication with --remote.
--token TOKEN

Saved credentials are used by default.

Username for authentication.
--username USERNAME

Password for authentication.
--password PASSWORD

Allow insecure HTTPS communication. i.e. bad SSL certificate.
--insecure

Additional HTTP header to include with requests.
--header HEADER

Timeout for API requests. Default is typically 30 seconds.
--timeout SECONDS

No Output, do not print to stdout
--quiet

Disable ANSI coloring
--nocolor

Print benchmark time after the command is finished.
--benchmark

Print extra output for debugging.
--debug

Print this help
--help

Assign hosts to a execute schedule.
[name] is required. This is the name or id of a execute schedule.
[host] is required. This is the name or id of a host. More than one can be passed.
17.15.3 morpheus execute-schedules add-instances

Usage: morpheus execute-schedules add-instances [name] [instance]

  --payload FILE Payload from a local JSON or YAML file, skip all prompting
  --payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
  --payload-json JSON Payload JSON, skip all prompting
  --payload-yaml YAML Payload YAML, skip all prompting
  -j, --json JSON Output
  -d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -r, --remote REMOTE Remote name. The current remote is used by default.
  --remote-url URL Remote url. The current remote url is used by default.
  -T, --token TOKEN Access token for authentication with --remote.
- Saved credentials are used by default.
  -U, --username USERNAME Username for authentication.
  -P, --password PASSWORD Password for authentication.
  -I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
  -H, --header HEADER Additional HTTP header to include with requests.
  --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
  -q, --quiet No Output, do not print to stdout
  -C, --nocolor Disable ANSI coloring
  -B, --benchmark Print benchmark time after the command is finished.
  -V, --debug Print extra output for debugging.
  -h, --help Print this help

Assign instances to a execute schedule. [name] is required. This is the name or id of a execute schedule. [instance] is required. This is the name or id of an instance. More than one can be passed.

17.15.4 morpheus execute-schedules get

Usage: morpheus execute-schedules get [name]

  --max-instances VALUE Display a limited number of instances in schedule. Default is 25
  --max-hosts VALUE Display a limited number of hosts in schedule. Default is 25
  -j, --json JSON Output
  --yaml YAML Output
  --csv CSV Output
  --csv-delim CHAR Delimiter for CSV Output values. Default: ','
  --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
  --csv-quotes Wrap CSV values with "'. Default: false
  --csv-no-header Exclude header for CSV Output.
  -F, --fields x,y,z Filter Output to a limited set of fields. Default is all fields.
17.15.5 morpheus execute-schedules list

Usage: morpheus execute-schedules list

- m, --max MAX Max Results
- o, --offset OFFSET Offset Results
- s, --search PHRASE Search Phrase
- S, --sort ORDER Sort Order
- D, --desc Reverse Sort Order
- j, --json JSON Output
- yam1 --yaml YAML Output
- csv --csv CSV Output
  --csv-delim CHAR Delimiter for CSV Output values. Default: ','
  --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
  --csv-quotes Wrap CSV values with '. Default: false
  --csv-no-header Exclude header for CSV Output.
- F, --fields x,y,z Filter Output to a limited set of fields.
  Default is all fields.
- d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by default.
  --remote-url URL Remote url. The current remote url is used by default.
- T, --token TOKEN Access token for authentication with --remote.
  Saved credentials are used by default.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
  --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

(continues on next page)
-H, --header HEADER  Additional HTTP header to include with requests.
-30 seconds.
-C, --nocolor  Disable ANSI coloring
-B, --benchmark  Print benchmark time after the command is finished.
-V, --debug  Print extra output for debugging.
-h, --help  Print this help

17.15.6 morpheus execute-schedules remove

Usage: morpheus execute-schedules remove [name]
    -j, --json  JSON Output
    -d, --dry-run  Dry Run, print the API request instead of executing it
    --curl  Dry Run to output API request as a curl command.
    --scrub  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
    -q, --quiet  No Output, do not print to stdout
    -y, --yes  Auto Confirm
    -C, --nocolor  Disable ANSI coloring
    -B, --benchmark  Print benchmark time after the command is finished.
    -V, --debug  Print extra output for debugging.
    -h, --help  Print this help

17.15.7 morpheus execute-schedules remove-hosts

Usage: morpheus execute-schedules remove-hosts [name] [host]
    --payload FILE  Payload from a local JSON or YAML file, skip all prompting
    --payload-dir DIRECTORY  Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
    --payload-json JSON  Payload JSON, skip all prompting
    --payload-yaml YAML  Payload YAML, skip all prompting
    -j, --json  JSON Output
    -d, --dry-run  Dry Run, print the API request instead of executing it
    --curl  Dry Run to output API request as a curl command.
    --scrub  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
    -r, --remote REMOTE  Remote name. The current remote is used by default.
    --remote-url URL  Remote url. The current remote url is used by default.
    -T, --token TOKEN  Access token for authentication with --remote.
    -U, --username USERNAME  Username for authentication.
    -P, --password PASSWORD  Password for authentication.
    -I, --insecure  Allow insecure HTTPS communication. i.e. bad SSL certificate.
    -H, --header HEADER  Additional HTTP header to include with requests.
Remove hosts from an execute schedule.

[name] is required. This is the name or id of a execute schedule.
[host] is required. This is the name or id of a host. More than one can be passed.

17.15.8 morpheus execute-schedules remove-instances

Usage: morpheus execute-schedules remove-instances [name] [instance]

- --payload FILE Payload from a local JSON or YAML file, skip all prompting
- --payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
- --payload-json JSON Payload JSON, skip all prompting
- --payload-yaml YAML Payload YAML, skip all prompting
- -j, --json JSON Output
- -d, --dry-run Dry Run, print the API request instead of executing it
- --curl Dry Run to output API request as a curl command.
- --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- -r, --remote REMOTE Remote name. The current remote is used by default.
- --remote-url URL Remote url. The current remote url is used by default.
- -t, --token TOKEN Access token for authentication with --remote.
- -U, --username USERNAME Username for authentication.
- -P, --password PASSWORD Password for authentication.
- -I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- -H, --header HEADER Additional HTTP header to include with requests.
- --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- -q, --quiet No Output, do not print to stdout
- -C, --nocolor Disable ANSI coloring
- -B, --benchmark Print benchmark time after the command is finished.
- -V, --debug Print extra output for debugging.
- -h, --help Print this help

Remove instances from an execute schedule.

[name] is required. This is the name or id of a execute schedule.
[instance] is required. This is the name or id of an instance. More than one can be passed.
17.15.9 morpheus execute-schedules update

Usage: morpheus execute-schedules update [name]

--name VALUE Name
--description VALUE Description
--type [execute] Type of Schedule. Default is 'execute'
--timezone CODE The timezone. Default is UTC.
--cron EXPRESSION Cron Expression
--enabled [on|off] Can be used to disable it
-O, --option OPTION Option in the format -O field="value"
--prompt Always prompts. Use passed options as the default value.
-N, --no-prompt Skip prompts. Use default values for all optional fields.
--payload FILE Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
--remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote. Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-q, --quiet No Output, do not print to stdout
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Update a execute schedule.
[name] is required. This is the name or id of a execute schedule.

17.16 morpheus execution-request

Usage: morpheus execution-request [command] [options]
Commands:
    execute
    get
17.16.1 `morpheus execution-request execute`

```
Usage: morpheus execution-request execute [options]

  --server ID  Server ID
  --instance ID Instance ID
  --container ID Container ID
  --request ID  Execution Request ID
  --script SCRIPT Script to be executed
  --file FILE  File containing the script. This can be used instead of
               --script
  --no-refresh  Do not refresh until finished
  -O, --option OPTION  Option in the format -O field="value"
  --prompt  Always prompts. Use passed options as the default value.
  -N, --no-prompt  Skip prompts. Use default values for all optional fields.
  --payload FILE  Payload from a local JSON or YAML file, skip all prompting
  --payload-dir DIRECTORY  Payload from a local directory containing 1-N JSON or
                           YAML files, skip all prompting
  --payload-json JSON  Payload JSON, skip all prompting
  --payload-yaml YAML  Payload YAML, skip all prompting
  -j, --json  JSON Output
  -d, --dry-run  Dry Run, print the API request instead of executing it
  --curl  Dry Run to output API request as a curl command.
  --scrub  Mask secrets in output, such as the Authorization header. For use with
           --curl and --dry-run.
  -q, --quiet  No Output, do not print to stdout
  -r, --remote REMOTE  Remote name. The current remote is used by default.
  --remote-url URL  Remote url. The current remote url is used by default.
  -T, --token TOKEN  Access token for authentication with --remote.
  --saved-credentials are used by default.
  -U, --username USERNAME  Username for authentication.
  -p, --password PASSWORD  Password for authentication.
  -I, --insecure  Allow insecure HTTPS communication. i.e. bad SSL certificate.
  -H, --header HEADER  Additional HTTP header to include with requests.
  --timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
  -C, --nocolor  Disable ANSI coloring
  -b, --benchmark  Print benchmark time after the command is finished.
  -V, --debug  Print extra output for debugging.
  -h, --help  Print this help

Execute an arbitrary script. [server] or [instance] or [container] is required. This is the id of a server, instance or container. [script] is required. This is the script that is to be executed.
```
17.16.2 morpheus execution-request get

Usage: morpheus execution-request get [uid]

- `-Q, --query PARAMS` Query parameters. PARAMS format is
  `'phrase=foobar&category=web'`
- `-j, --json` JSON Output
- `--yaml` YAML Output
- `--csv` CSV Output
- `--csv-delim CHAR` Delimiter for CSV Output values. Default: ','
'
- `--csv-quotes` Wrap CSV values with '"'. Default: false
- `--csv-no-header` Exclude header for CSV Output.
- `-F, --fields x,y,z` Filter Output to a limited set of fields.
  Default is all fields.
- `-d, --dry-run` Dry Run, print the API request instead of executing it
  - `--curl` Dry Run to output API request as a curl command.
  - `--scrub` Mask secrets in output, such as the Authorization header. For use with `--curl` and `--dry-run`.
- `-r, --remote REMOTE` Remote name. The current remote is used by default.
- `-d, --remote-url URL` Remote url. The current remote url is used by default.
- `--token TOKEN` Access token for authentication with `--remote`.
  Saved credentials are used by default.
- `-U, --username USERNAME` Username for authentication.
- `-P, --password PASSWORD` Password for authentication.
- `--insecure` Allow insecure HTTPS communication. i.e. bad SSL certificate.
- `-H, --header HEADER` Additional HTTP header to include with requests.
- `-C, --nocolor` Disable ANSI coloring
- `--benchmark` Print benchmark time after the command is finished.
- `-V, --debug` Print extra output for debugging.
- `-h, --help` Print this help
  - `--refresh [SECONDS]` Refresh until execution is finished. Default interval is 5 seconds.

Get details about an execution request.

[uid] is required. This is the unique id of an execution request.

17.17 morpheus file-copy-request

Usage: morpheus file-copy-request [command] [options]

Commands:
  download
  execute
  get
17.17.1 morpheus file-copy-request download

Usage: morpheus file-copy-request download [uid] [file]

- -file FILE
  Local file destination for the downloaded file.
- -f, --force
  Overwrite existing [file] if it exists.
- -p, --mkdir
  Create missing directories for [file] if they do not exist.
- -O, --option OPTION
  Option in the format -O field="value"
- -prompt
  Always prompts. Use passed options as the default value.
- -N, --no-prompt
  Skip prompts. Use default values for all optional fields.
- -j, --json
  JSON Output
- -d, --dry-run
  Dry Run, print the API request instead of executing it
- -c, --curl
  Dry Run to output API request as a curl command.
- -s, --scrub
  Mask secrets in output, such as the Authorization header.
  For use with --curl and --dry-run.
- -q, --quiet
  No Output, do not print to stdout
- -r, --remote REMOTE
  Remote name. The current remote is used by default.
- -u, --remote-url URL
  Remote url. The current remote url is used by default.
- -O, --option OPTION
  Option in the format -O field="value"
- -prompt
  Always prompts. Use passed options as the default value.
- -N, --no-prompt
  Skip prompts. Use default values for all optional fields.
- -j, --json
  JSON Output
- -d, --dry-run
  Dry Run, print the API request instead of executing it
- -c, --curl
  Dry Run to output API request as a curl command.
- -s, --scrub
  Mask secrets in output, such as the Authorization header.
  For use with --curl and --dry-run.
- -q, --quiet
  No Output, do not print to stdout
- -r, --remote REMOTE
  Remote name. The current remote is used by default.
- -u, --remote-url URL
  Remote url. The current remote url is used by default.
- -T, --token TOKEN
  Access token for authentication with --remote.
- -U, --username USERNAME
  Username for authentication.
- -P, --password PASSWORD
  Password for authentication.
- -I, --insecure
  Allow insecure HTTPS communication. i.e. bad SSL certificate.
- -H, --header HEADER
  Additional HTTP header to include with requests.
- -timeout SECONDS
  Timeout for api requests. Default is typically 30 seconds.
- -C, --nocolor
  Disable ANSI coloring
- -B, --benchmark
  Print benchmark time after the command is finished.
- -V, --debug
  Print extra output for debugging.
- -h, --help
  Print this help

Download a file associated with a file copy request.
[uid] is required. This is the unique id of a file copy request.
[file] is required. This is the full local filepath for the downloaded file.

17.17.2 morpheus file-copy-request execute

Usage: morpheus file-copy-request execute [options]

- --server ID
  Server ID
- --instance ID
  Instance ID
- --container ID
  Container ID
- --request ID
  File Copy Request ID
- --file FILE
  Local file to be copied.
- --target-path PATH
  Target path for file on destination host.
- --no-refresh
  Do not refresh until finished
- -O, --option OPTION
  Option in the format -O field="value"
- -prompt
  Always prompts. Use passed options as the default value.
-N, --no-prompt
  Skip prompts. Use default values for all optional fields.

-j, --json
  JSON Output

-d, --dry-run
  Dry Run, print the API request instead of executing it

--curl
  Dry Run to output API request as a curl command.

--scrub
  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.

-g, --quiet
  No Output, do not print to stdout

-r, --remote REMOTE
  Remote name. The current remote is used by default.

--remote-url URL
  Remote url. The current remote url is used by default.

-T, --token TOKEN
  Access token for authentication with --remote.

-Saved credentials are used by default.

-U, --username USERNAME
  Username for authentication.

-P, --password PASSWORD
  Password for authentication.

-i, --insecure
  Allow insecure HTTPS communication. i.e. bad SSL certificate.

-H, --header HEADER
  Additional HTTP header to include with requests.

--timeout SECONDS
  Timeout for api requests. Default is typically 30 seconds.

-c, --nocolor
  Disable ANSI coloring

-B, --benchmark
  Print benchmark time after the command is finished.

-V, --debug
  Print extra output for debugging.

-h, --help
  Print this help

Copy a file to a remote host(s).

[server] or [instance] or [container] is required. This is the id of a server, instance or container.

[file] is required. This is the local filename that is to be copied.

[target-path] is required. This is the target path for the file on the destination host.

17.17.3 morpheus file-copy-request get

Usage: morpheus file-copy-request get [uid]

-Q, --query PARAMS
  Query parameters. PARAMS format is 'phrase=foobar&category=web'

-j, --json
  JSON Output

--yaml
  YAML Output

--csv
  CSV Output

--csv-delim CHAR
  Delimiter for CSV Output values. Default: ','

--csv-newline [CHAR]
  Delimiter for CSV Output rows. Default: '\n'

--csv-quotes [CHAR]
  Wrap CSV values with '. Default: false

--csv-no-header
  Exclude header for CSV Output.

-F, --fields x,y,z
  Filter Output to a limited set of fields.

-D, --dry-run
  Dry Run, print the API request instead of executing it

--curl
  Dry Run to output API request as a curl command.

--scrub
  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.

-r, --remote REMOTE
  Remote name. The current remote is used by default.
Get details about a file copy request. [uid] is required. This is the unique id of a file copy request.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--remote-url URL</td>
<td>Remote url. The current remote url is used by default.</td>
</tr>
<tr>
<td>-T, --token TOKEN</td>
<td>Access token for authentication with --remote. Saved credentials are used by default.</td>
</tr>
<tr>
<td>-U, --username USERNAME</td>
<td>Username for authentication.</td>
</tr>
<tr>
<td>-P, --password PASSWORD</td>
<td>Password for authentication.</td>
</tr>
<tr>
<td>-I, --insecure</td>
<td>Allow insecure HTTPS communication. i.e. bad SSL certificate.</td>
</tr>
<tr>
<td>-H, --header HEADER</td>
<td>Additional HTTP header to include with requests.</td>
</tr>
<tr>
<td>--timeout SECONDS</td>
<td>Timeout for api requests. Default is typically 30 seconds.</td>
</tr>
<tr>
<td>-C, --nocolor</td>
<td>Disable ANSI coloring</td>
</tr>
<tr>
<td>-B, --benchmark</td>
<td>Print benchmark time after the command is finished.</td>
</tr>
<tr>
<td>-V, --debug</td>
<td>Print extra output for debugging.</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Print this help</td>
</tr>
<tr>
<td>--refresh [SECONDS]</td>
<td>Refresh until execution is finished. Default interval is 5 seconds.</td>
</tr>
</tbody>
</table>

## 17.18 morpheus groups

Usage: morpheus groups [command] [options]

Commands:
- add
- add-cloud
- current
- get
- list
- remove
- remove-cloud
- unuse
- update
- use

## 17.18.1 morpheus groups add

Usage: morpheus groups add [name]

- --name VALUE | Name |
- --code VALUE | Code (optional) |
- --location VALUE | Location (optional) |
- --use | Make this the current active group |
- -O, --option OPTION | Option in the format -O field="value" |
- --prompt | Always prompts. Use passed options as the default value. |
- -N, --no-prompt | Skip prompts. Use default values for all optional fields. |
- -j, --json | JSON Output |
- -d, --dry-run | Dry Run, print the API request instead of executing it |
Create a new group.

### 17.18.2 morpheus groups add-cloud

Usage: morpheus groups add-cloud [name] CLOUD

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-j, --json</td>
<td>JSON Output</td>
</tr>
<tr>
<td>-d, --dry-run</td>
<td>Dry Run, print the API request instead of executing it</td>
</tr>
<tr>
<td>-c, --curl</td>
<td>Dry Run to output API request as a curl command.</td>
</tr>
<tr>
<td>-s, --scrub</td>
<td>Mask secrets in output, such as the Authorization header.</td>
</tr>
<tr>
<td>-r, --remote REMOTE</td>
<td>Remote name. The current remote is used by default.</td>
</tr>
<tr>
<td>-u, --username USERNAME</td>
<td>Username for authentication.</td>
</tr>
<tr>
<td>-p, --password PASSWORD</td>
<td>Password for authentication.</td>
</tr>
<tr>
<td>-i, --insecure</td>
<td>Allow insecure HTTPS communication. i.e. bad SSL certificate.</td>
</tr>
<tr>
<td>-H, --header HEADER</td>
<td>Additional HTTP header to include with requests.</td>
</tr>
<tr>
<td>-t, --token TOKEN</td>
<td>Access token for authentication with --remote.</td>
</tr>
<tr>
<td>-u, --username USERNAME</td>
<td>Username for authentication.</td>
</tr>
<tr>
<td>-p, --password PASSWORD</td>
<td>Password for authentication.</td>
</tr>
<tr>
<td>-i, --insecure</td>
<td>Allow insecure HTTPS communication. i.e. bad SSL certificate.</td>
</tr>
<tr>
<td>-H, --header HEADER</td>
<td>Additional HTTP header to include with requests.</td>
</tr>
<tr>
<td>-t, --token TOKEN</td>
<td>Access token for authentication with --remote.</td>
</tr>
<tr>
<td>SSL certificate.</td>
<td>Additional HTTP header to include with requests.</td>
</tr>
<tr>
<td>-C, --insecure</td>
<td>Allow insecure HTTPS communication. i.e. bad SSL certificate.</td>
</tr>
<tr>
<td>-H, --header HEADER</td>
<td>Additional HTTP header to include with requests.</td>
</tr>
<tr>
<td>-t, --token TOKEN</td>
<td>Access token for authentication with --remote.</td>
</tr>
<tr>
<td>SSL certificate.</td>
<td>Additional HTTP header to include with requests.</td>
</tr>
<tr>
<td>30 seconds.</td>
<td>Timeout for api requests. Default is typically 30 seconds.</td>
</tr>
<tr>
<td>-c, --nocolor</td>
<td>Disable ANSI coloring</td>
</tr>
<tr>
<td>-b, --benchmark</td>
<td>Print benchmark time after the command is finished.</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Print extra output for debugging.</td>
</tr>
</tbody>
</table>

Add a cloud to a group.
17.18.3 morpheus groups current

Usage: morpheus groups current

- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

Print the name of the current active group

17.18.4 morpheus groups get

Usage: morpheus groups get [name]

- Q, --query PARAMS Query parameters. PARAMS format is 'phrase=foobar&category=web'
- j, --json JSON Output
- y, --yaml YAML Output
- csv CSV Output
- csv-delim CHAR Delimiter for CSV Output values. Default: ','
- csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
- csv-quotes Wrap CSV values with "". Default: false
- csv-no-header Exclude header for CSV Output.
- F, --fields x,y,z Filter Output to a limited set of fields.
- d, --dry-run Dry Run, print the API request instead of executing it
- curl Dry Run to output API request as a curl command.
- scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by default.
- remote-url URL Remote url. The current remote url is used by default.
- T, --token TOKEN Access token for authentication with --remote.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
- timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

Get details about a group.

[name] is required. This is the name or id of a group. Supports 1-N arguments.
### 17.18.5 morpheus groups list

<table>
<thead>
<tr>
<th>Usage: <code>morpheus groups list</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>-m, --max MAX</td>
</tr>
<tr>
<td>-o, --offset OFFSET</td>
</tr>
<tr>
<td>-s, --search PHRASE</td>
</tr>
<tr>
<td>-S, --sort ORDER</td>
</tr>
<tr>
<td>-D, --desc</td>
</tr>
</tbody>
</table>
| -q, --query PHRASE | Query parameters. PARAMS format is '

| 'phrase=foobar&category=web' |
| -j, --json | JSON Output |
| -y, --yaml | YAML Output |
| -c, --csv | CSV Output |
| --csv-delim CHAR | Delimiter for CSV Output values. Default: ',' |
| --csv-newline [CHAR] | Delimiter for CSV Output rows. Default: '\n' |
| --csv-quotes | Wrap CSV values with "". Default: false |
| --csv-no-header | Exclude header for CSV Output. |
| -F, --fields x,y,z | Filter Output to a limited set of fields. |
| Default is all fields. |
| -d, --dry-run | Dry Run, print the API request instead of executing it |
| --c-request | Dry Run to output API request as a curl command. |
| --scrub | Mask secrets in output, such as the Authorization header. For use with --c-request and --dry-run. |
| -r, --remote REMOTE | Remote name. The current remote is used by default. |
| --remote-url URL | Remote url. The current remote url is used by default. |
| -t, --token TOKEN | Access token for authentication with --remote. Saved credentials are used by default. |
| -U, --username USERNAME | Username for authentication. |
| -p, --password PASSWORD | Password for authentication. |
| -I, --insecure | Allow insecure HTTPS communication. i.e. bad SSL certificate. |
| -H, --header HEADER | Additional HTTP header to include with requests. |
| --timeout SECONDS | Timeout for api requests. Default is typically 30 seconds. |
| -C, --nocolor | Disable ANSI coloring |
| -b, --benchmark | Print benchmark time after the command is finished. |
| -v, --debug | Print extra output for debugging. |
| -h, --help | Print this help |

List groups (sites).

### 17.18.6 morpheus groups remove

<table>
<thead>
<tr>
<th>Usage: <code>morpheus groups remove [name]</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>-j, --json</td>
</tr>
<tr>
<td>-d, --dry-run</td>
</tr>
<tr>
<td>--c-request</td>
</tr>
<tr>
<td>--scrub</td>
</tr>
<tr>
<td>-y, --yes</td>
</tr>
</tbody>
</table>

(continues on next page)
Delete a group.

17.18.7 morpheus groups remove-cloud

Usage: morpheus groups remove-cloud [name] CLOUD
   -j, --json                JSON Output
   -d, --dry-run            Dry Run, print the API request instead of executing it
   --executing             Dry Run to output API request as a curl command.
   --curl                  Dry Run to output API request as a curl command.
   --scrub                 Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE     Remote name. The current remote is used by default.
   --remote-url URL        Remote url. The current remote url is used by default.
   -T, --token TOKEN       Access token for authentication with --remote.
   Saved credentials are used by default.
   -U, --username USERNAME Username for authentication.
   -P, --password PASSWORD Password for authentication.
   -I, --insecure          Allow insecure HTTPS communication. i.e. bad SSL certificate.
   --timeout SECONDS      Timeout for api requests. Default is typically 30 seconds.
   -H, --header HEADER     Additional HTTP header to include with requests.
   -C, --nocolor           Disable ANSI coloring
   -B, --benchmark         Print benchmark time after the command is finished.
   -V, --debug             Print extra output for debugging.
   -h, --help              Print this help

Remove a cloud from a group.
17.18.8 morpheus groups unused

Usage: morpheus groups unused

- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

This will clear the current active group.
You will be prompted for a Group during provisioning.

17.18.9 morpheus groups update

Usage: morpheus groups update [name] [options]

- name VALUE Name (optional)
- code VALUE Code (optional)
- location VALUE Location (optional)
- O, --option OPTION Option in the format -O field="value"
- prompt Always prompts. Use passed options as the default value.
- N, --no-prompt Skip prompts. Use default values for all optional fields.
- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by default.
- remote-url URL Remote url. The current remote url is used by default.
- T, --token TOKEN Access token for authentication with --remote.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
- timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

Update an existing group.

17.18.10 morpheus groups use
Usage: morpheus groups use [name]
  -q, --quiet No Output, do not print to stdout
  -r, --remote REMOTE Remote name. The current remote is used by
  default.
  --remote-url URL Remote url. The current remote url is used by
  default.
  -T, --token TOKEN Access token for authentication with --remote.
  Saved credentials are used by default.
  -U, --username USERNAME Username for authentication.
  -P, --password PASSWORD Password for authentication.
  --insecure Allow insecure HTTPS communication. i.e. bad
  SSL certificate.
  --header HEADER Additional HTTP header to include with requests.
  --timeout SECONDS Timeout for api requests. Default is typically
  30 seconds.
  -C, --nocolor Disable ANSI coloring
  -B, --benchmark Print benchmark time after the command is
  finished.
  -V, --debug Print extra output for debugging.
  -h, --help Print this help

This sets the active group.
The active group will be auto-selected for use during provisioning.
You can still use the --group option to override this.

17.19 morpheus hosts

Usage: morpheus hosts [command] [options]
Commands:
  add
  count
  exec
  get
  list
  logs
  make-managed
  remove
  resize
  run-workflow
  start
  stats
  stop
  types
  update
  upgrade-agent

17.19.1 morpheus hosts add

Usage: morpheus hosts add [cloud] [name]
  -g, --group GROUP Group Name or ID
  -c, --cloud CLOUD Cloud Name or ID
  -t, --type TYPE Server Type Code

(continues on next page)
## 17.19.2 morpheus hosts count

Usage: morpheus hosts count [options]

- **-a**, **--account** ACCOUNT  
  Account Name or ID

- **-g**, **--group** GROUP  
  Group Name or ID

- **-c**, **--cloud** CLOUD  
  Cloud Name or ID

- **-M**, **--managed**  
  Show only Managed Servers

- **-U**, **--unmanaged**  
  Show only Unmanaged Servers

- **-t**, **--type** TYPE  
  Show only Certain Server Types

- **-p**, **--power** STATE  
  Filter by Power Status

- **-i**, **--ip** IPADDRESS  
  Filter by IP Address

- **--vm**  
  Show only virtual machines

- **--hypervisor**  
  Show only VM Hypervisors

- **--container**  
  Show only Container Hypervisors

- **--baremetal**  
  Show only baremetal servers
--status STATUS  Show only Baremetal Servers
--agent  Filter by Status
--noagent  Show only Servers with the agent installed
--created-by USER  Show only Servers with No agent
--details  Created By User Username or ID
--used / max values.  Display more details: memory and storage usage
-s, --search PHRASE  Search Phrase
-Q, --query PARAMS  Query parameters. PARAMS format is
'phrase=foo&category=web'
-r, --remote REMOTE  Remote name. The current remote is used by
--default.
--remote-url URL  Remote url. The current remote url is used by
--default.
-T, --token TOKEN  Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME  Username for authentication.
-P, --password PASSWORD  Password for authentication.
-I, --insecure  Allow insecure HTTPS communication. i.e. bad
SSL certificate.
-H, --header HEADER  Additional HTTP header to include with requests.
--timeout SECONDS  Timeout for api requests. Default is typically
30 seconds.
-d, --dry-run  Dry Run, print the API request instead of
-executing it
--curl  Dry Run to output API request as a curl command.
--scrub  Mask secrets in output, such as the
Authorization header. For use with --curl and --dry-run.
-C, --nocolor  Disable ANSI coloring
-B, --benchmark  Print benchmark time after the command is
finished.
-V, --debug  Print extra output for debugging.
-h, --help  Print this help

Get the number of hosts.

17.19.3 morpheus hosts exec

Usage: morpheus hosts exec [id] [options]
   --script SCRIPT  Script to be executed
   --file FILE  File containing the script. This can be used
   --instead of --script
   --no-refresh  Do not refresh until finished
   -O, --option OPTION  Option in the format -O field="value"
   --prompt  Always prompts. Use passed options as the
   default value.
   -N, --no-prompt  Skip prompts. Use default values for all
   --optional fields.
   --payload FILE  Payload from a local JSON or YAML file, skip all
   --prompting
   --payload-dir DIRECTORY  Payload from a local directory containing 1-N
   --JSON or YAML files, skip all prompting

(continues on next page)
## 17.19.4 morpheus hosts get

Usage: morpheus hosts get [name]

- **--refresh [SECONDS]** Refresh until status is running, failed. Default is 5 seconds.
- **--refresh-until STATUS** Refresh until a specified status is reached.
- **-j, --json** JSON Output
- **--csv** CSV Output
- **--csv-delim CHAR** Delimiter for CSV Output values. Default: ','
- **--csv-newline [CHAR]** Delimiter for CSV Output rows. Default: '\n'
- **--csv-quotes** Wrap CSV values with "". Default: false
- **--csv-no-header** Exclude header for CSV Output.
- **-y, --yaml** YAML Output
- **-F, --fields x,y,z** Filter Output to a limited set of fields. Default is all fields.
- **-d, --dry-run** Dry Run, print the API request instead of executing it.
- **--curl** Dry Run to output API request as a curl command.
- **--scrub** Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- **-r, --remote REMOTE** Remote name. The current remote is used by default.
- **--remote-url URL** Remote url. The current remote url is used by default.
- **-T, --token TOKEN** Access token for authentication with --remote.
- **-U, --username USERNAME** Username for authentication.
- **-P, --password PASSWORD** Password for authentication.
- **-I, --insecure** Allow insecure HTTPS communication. i.e. bad SSL certificate.
- **-H, --header HEADER** Additional HTTP header to include with requests.
- **--timeout SECONDS** Timeout for api requests. Default is typically 30 seconds.
- **-C, --nocolor** Disable ANSI coloring
- **-B, --benchmark** Print benchmark time after the command is finished.
- **-V, --debug** Print extra output for debugging.
- **-h, --help** Print this help

Execute an arbitrary command or script on a host. [id] is required. This is the id a host. [script] is required. This is the script that is to be executed.
### 17.19.5 `morpheus hosts list`

Usage: morpheus hosts list

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--account ACCOUNT</code></td>
<td>Account Name or ID</td>
</tr>
<tr>
<td><code>--group GROUP</code></td>
<td>Group Name or ID</td>
</tr>
<tr>
<td><code>--cloud CLOUD</code></td>
<td>Cloud Name or ID</td>
</tr>
<tr>
<td><code>--managed</code></td>
<td>Show only Managed Servers</td>
</tr>
<tr>
<td><code>--unmanaged</code></td>
<td>Show only Unmanaged Servers</td>
</tr>
<tr>
<td><code>--type TYPE</code></td>
<td>Show only Certain Server Types</td>
</tr>
<tr>
<td><code>--power STATE</code></td>
<td>Filter by Power Status</td>
</tr>
<tr>
<td><code>--ip IPADDRESS</code></td>
<td>Filter by IP Address</td>
</tr>
<tr>
<td><code>--vm</code></td>
<td>Show only virtual machines</td>
</tr>
<tr>
<td><code>--hypervisor</code></td>
<td>Show only VM Hypervisors</td>
</tr>
<tr>
<td><code>--container</code></td>
<td>Show only Container Hypervisors</td>
</tr>
<tr>
<td><code>--baremetal</code></td>
<td>Show only Baremetal Servers</td>
</tr>
<tr>
<td><code>--status STATUS</code></td>
<td>Filter by Status</td>
</tr>
<tr>
<td><code>--agent</code></td>
<td>Show only Servers with the agent installed</td>
</tr>
<tr>
<td><code>--noagent</code></td>
<td>Show only Servers with No agent</td>
</tr>
<tr>
<td><code>--created-by USER</code></td>
<td>Created By User Username or ID</td>
</tr>
<tr>
<td><code>--details</code></td>
<td>Display more details: memory and storage usage</td>
</tr>
</tbody>
</table>

Use `--max MAX` for max results, `--offset OFFSET` for offset results, `--search PHRASE` for search phrase, `--sort ORDER` for sort order, `--desc` for reverse sort order, `--query PARAMS` for query parameters, `--json` for JSON output, `--yaml` for YAML output, `--csv` for CSV output, `--csv-delim CHAR` for delimiter for CSV output values, and `--csv-newline [CHAR]` for delimiter for CSV output rows.
List hosts.

17.19.6 morpheus hosts logs

Usage: morpheus hosts logs [name]

-\n  --max MAX Max Results
-\n  --offset OFFSET Offset Results
-\n  --search PHRASE Search Phrase
-\n  --sort ORDER Sort Order
-\n  --desc Reverse Sort Order
-\n  --json JSON Output
-\n  --dry-run Dry Run, print the API request instead of
-\n  --curl Dry Run to output API request as a curl command.
-\n  --scrub Mask secrets in output, such as the
-\n  Authorization header. For use with --curl and --dry-run.
-\n  --remote REMOTE Remote name. The current remote is used by
-\n  --default.
-\n  --remote-url URL Remote url. The current remote url is used by
-\n  --default.
-\n  --token TOKEN Access token for authentication with --remote.
-\n  --username USERNAME Username for authentication.
-\n  --password PASSWORD Password for authentication.
-\n  --insecure Allow insecure HTTPS communication. i.e. bad
-\n  SSL certificate.
-\n  --header HEADER Additional HTTP header to include with requests.
-\n  --timeout SECONDS Timeout for api requests. Default is typically
  30 seconds.
-\n  --nocolor Disable ANSI coloring
-\n  --benchmark Print benchmark time after the command is
  finished.
-\n  --debug Print extra output for debugging.
-\n  --help Print this help
17.19.7 morphes hosts make-managed

Usage: morpheus hosts make-managed [name]

-sshUsername VALUE
-sshPassword VALUE
-serverOs VALUE
--json
--dry-run
--curl
--scrub
-Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE
--remote-url URL
--token TOKEN
--username USERNAME
--password PASSWORD
--insecure
--header HEADER
--timeout SECONDS
--remove-resources [on|off]
--preserve-volumes [on|off]
--remove-instances [on|off]
--release-eips [on|off]
-f, --force
-y, --yes
-j, --json
-d, --dry-run

17.19.8 morphes hosts remove

Usage: morpheus hosts remove [name]

--remove-resources [on|off]
--preserve-volumes [on|off]
--remove-instances [on|off]
--release-eips [on|off]
-f, --force
-y, --yes
-j, --json
-d, --dry-run
17.19.9 morpheus hosts resize

Usage: morpheus hosts resize [name]
-0, --option OPTION Option in the format -O field="value"
--prompt Always prompts. Use passed options as the
--default value.
-n, --no-prompt Skip prompts. Use default values for all
optional fields.
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of
executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the
Authorization header. For use with --curl and --dry-run.
-q, --quiet No Output, do not print to stdout
-r, --remote REMOTE Remote name. The current remote is used by
default.
--remote-url URL Remote url. The current remote url is used by
default.
-T, --token TOKEN Access token for authentication with --remote.
Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad
SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically
30 seconds.
-c, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is
finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help
17.19.10 `morpheus hosts run-workflow`

Usage: `morpheus hosts run-workflow [name] [workflow] [options]`

- **-O, --option OPTION** Option in the format `-O field="value"`
- **--prompt** Always prompts. Use passed options as the default value.
- **-N, --no-prompt** Skip prompts. Use default values for all optional fields.
- **--payload FILE** Payload from a local JSON or YAML file, skip all prompting.
- **--payload-dir DIRECTORY** Payload from a local directory containing 1-N JSON or YAML files, skip all prompting.
- **--payload-json JSON** Payload JSON, skip all prompting.
- **--payload-yaml YAML** Payload YAML, skip all prompting.
- **-j, --json JSON** Output.
- **-d, --dry-run** Dry Run, print the API request instead of executing it.
- **--curl** Dry Run to output API request as a curl command.
- **--scrub** Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- **-y, --yes** Auto Confirm.
- **-q, --quiet** No Output, do not print to stdout.
- **-j, --json** JSON Output.
- **-d, --dry-run** Dry Run, print the API request instead of executing it.
- **--curl** Dry Run to output API request as a curl command.
- **--scrub** Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- **--timeout SECONDS** Timeout for api requests. Default is typically 30 seconds.
- **--nocolor** Disable ANSI coloring.
- **-B, --benchmark** Print benchmark time after the command is finished.
- **-V, --debug** Print extra output for debugging.
- **-h, --help** Print this help

17.19.11 `morpheus hosts start`

Usage: `morpheus hosts start [name]`

- **-y, --yes** Auto Confirm.
- **-q, --quiet** No Output, do not print to stdout.
- **-j, --json JSON Output.
- **-d, --dry-run** Dry Run, print the API request instead of executing it.
- **--curl** Dry Run to output API request as a curl command.
- **--scrub** Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE
  Remote name. The current remote is used by default.
--remote-url URL
  Remote url. The current remote url is used by default.
-T, --token TOKEN
  Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME
  Username for authentication.
P, --password PASSWORD
  Password for authentication.
-I, --insecure
  Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER
  Additional HTTP header to include with requests.
--timeout SECONDS
  Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor
  Disable ANSI coloring
-B, --benchmark
  Print benchmark time after the command is finished.
-V, --debug
  Print extra output for debugging.
-h, --help
  Print this help

Start a host.
[name] is required. This is the name or id of a host. Supports 1-N [name] arguments.

17.19.12 morpheus hosts stats

Usage: morpheus hosts stats [name]
  -j, --json
  JSON Output
  -y, --yaml
  YAML Output
  -csv
  CSV Output
  --csv-delim CHAR
  Delimiter for CSV Output values. Default: ','
  --csv-newline [CHAR]
  Delimiter for CSV Output rows. Default: '\n'
  --csv-quotes
  Wrap CSV values with ''. Default: false
  --csv-no-header
  Exclude header for CSV Output.
  -F, --fields x,y,z
  Filter Output to a limited set of fields. Default is all fields.
  -d, --dry-run
  Dry Run, print the API request instead of executing it
  --curl
  Dry Run to output API request as a curl command.
  --scrub
  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -r, --remote REMOTE
  Remote name. The current remote is used by default.
  --remote-url URL
  Remote url. The current remote url is used by default.
  -T, --token TOKEN
  Access token for authentication with --remote.
  Saved credentials are used by default.
  -U, --username USERNAME
  Username for authentication.
P, --password PASSWORD
  Password for authentication.
-I, --insecure
  Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER
  Additional HTTP header to include with requests.
  --timeout SECONDS
  Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor
  Disable ANSI coloring
-B, --benchmark
  Print benchmark time after the command is finished.
17.19.13 morpheus hosts stop

Usage: morpheus hosts stop [name]

- y, --yes
- q, --quiet
- j, --json
- d, --dry-run
- --executing
- --curl
- --scrub
- --remote
- r, --remote-url
- --default
- T, --token
- U, --username
- P, --password
- I, --insecure
- H, --header
- --timeout
- C, --nocolor
- B, --benchmark
- V, --debug
- h, --help

Auto Confirm
No Output, do not print to stdout
JSON Output
Dry Run, print the API request instead of executing it
Dry Run to output API request as a curl command.
Mask secrets in output, such as the Authorization header for use with --curl and --dry-run.
Remote name. The current remote is used by default.
Remote url. The current remote url is used by default.
Access token for authentication with --remote.
Saved credentials are used by default.
Username for authentication.
Password for authentication.
Allow insecure HTTPS communication. i.e. bad SSL certificate.
Additional HTTP header to include with requests.
Timeout for api requests. Default is typically 30 seconds.
Disable ANSI coloring
Print benchmark time after the command is finished.
Print extra output for debugging.
Print this help

Stop a host.
[name] is required. This is the name or id of a host. Supports 1-N [name] arguments.

17.19.14 morpheus hosts types

Usage: morpheus hosts types

- c, --cloud
- m, --max
- o, --offset
- s, --search
- S, --sort
- D, --desc
- Q, --query
- j, --json
- yml
- csv
- csv-delim
- csv-newline

Cloud Name or ID
Max Results
Offset Results
Search Phrase
Sort Order
Reverse Sort Order
Query parameters. PARAMS format is phrase=foobar&category=web'
JSON Output
YAML Output
CSV Output
Delimiter for CSV Output values. Default: ','
Delimiter for CSV Output rows. Default: '\n'
17.19.5 morpheus hosts update

Usage: morpheus hosts update [name]

|--name VALUE  Name
|--description VALUE  Description
|--ssh-username VALUE  SSH Username
|--ssh-password VALUE  SSH Password
|--power-schedule-type ID  Power Schedule Type ID
-0, --option OPTION  Option in the format -O field="value"
-0, --prompt Always prompts. Use passed options as the
--default value.
-N, --no-prompt  Skip prompts. Use default values for all
--optional fields.
-0, --payload FILE  Payload from a local JSON or YAML file, skip all
--prompting
-0, --payload-dir DIRECTORY  Payload from a local directory containing 1-N
-0, --payload-json JSON  Payload JSON, skip all prompting
-0, --payload-yaml YAML  Payload YAML, skip all prompting
-0, --json JSON Output
-0, --dry-run  Dry Run, print the API request instead of
--executing it
-0, --curl  Dry Run to output API request as a curl command.
-0, --scrub Mask secrets in output, such as the
--Authorization header. For use with --curl and --dry-run.
-T, --token TOKEN  Access token for authentication with --remote.
-U, --username USERNAME  Username for authentication.
-P, --password PASSWORD  Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad
-H, --header HEADER  Additional HTTP header to include with requests.
-t, --timeout SECONDS  Timeout for api requests. Default is typically
30 seconds.
-C, --nocolor  Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is
-finished.
-V, --debug Print extra output for debugging.
-h, --help  Print this help

List host types.
17.19.16 morpheus hosts upgrade-agent

Usage: morpheus hosts upgrade-agent [name]
   -j, --json  JSON Output
   -d, --dry-run  Dry Run, print the API request instead of executing it
     --curl  Dry Run to output API request as a curl command.
     --scrub  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -q, --quiet  No Output, do not print to stdout
   -r, --remote REMOTE  Remote name. The current remote is used by default.
   --remote-url URL  Remote url. The current remote url is used by default.
   -T, --token TOKEN  Access token for authentication with --remote.
   -U, --username USERNAME  Username for authentication.
   -P, --password PASSWORD  Password for authentication.
   -I, --insecure  Allow insecure HTTPS communication. i.e. bad SSL certificate.
   -H, --header HEADER  Additional HTTP header to include with requests.
   --timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
   -C, --nocolor  Disable ANSI coloring
   -B, --benchmark  Print benchmark time after the command is finished.
   -V, --debug  Print extra output for debugging.
   -h, --help  Print this help

17.20 morpheus image-builder

Usage: morpheus image-builder [command] [options]
Commands:

(continues on next page)
17.20.1 morpheus image-builder add

Usage: morpheus image-builder add [options]

-t, --type TYPE      Image Build Type
-\t--name VALUE      Name
-\t--description VALUE  Description
-\t--group GROUP      Group Name or ID
-\t--cloud CLOUD      Cloud Name or ID
-\t--config JSON      Instance Config JSON
-\t--config-yaml YAML  Instance Config YAML
-\t--config-file FILE Instance Config from a local JSON or YAML file
-\t--bootScript VALUE Boot Script ID
-\t--bootCommand VALUE Boot Command. This can be used in place of a
-\t--preseedScript VALUE  Preseed Script ID
-\t--scripts LIST Additional Scripts (comma separated names or ids)
-\t--sshUsername VALUE    SSH Username
-\t--sshPassword VALUE    SSH Password
-\t--storageProvider VALUE Storage Provider ID
-\t--isCloudInit [on|off] Cloud Init?
-\t--buildOutputName VALUE Build Output Name
-\t--conversionFormats VALUE Conversion Formats ie. ovf, qcow2, vhd
-\t--keepResults VALUE Keep only the most recent builds. Older
-\t  executions will be deleted along with their associated Virtual Images. The value 0
-\t  disables this functionality.
-\t-O, --option OPTION  Option in the format -O field="value"
-\t--prompt Always prompts. Use passed options as the
-\t  default value.
-\t-N, --no-prompt Skip prompts. Use default values for all
-\t  optional fields.
-\t--payload FILE Payload from a local JSON or YAML file, skip all
-\t  prompting
-\t--payload-dir DIRECTORY Payload from a local directory containing 1-N
-\t  or YAML files, skip all prompting
-\t--payload-json JSON Payload JSON, skip all prompting
-\t--payload-yaml YAML Payload YAML, skip all prompting
-\t-j, --json JSON Output
-\t-d, --dry-run Dry Run, print the API request instead of
-\t  executing it
-\t--curl Dry Run to output API request as a curl command.
-\t--scrub Mask secrets in output, such as the
-\t  Authorization header. For use with --curl and --dry-run.
-\t-r, --remote REMOTE  Remote name. The current remote is used by
-\t  default.
-\t--remote-url URL Remote url. The current remote url is used by
-T, --token TOKEN  Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-g, --quiet No Output, do not print to stdout
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.20.2 morpheus image-builder boot-scripts

Usage: morpheus image-builder boot-scripts [command] [options]
Commands:
  add
  get
  list
  remove
  update

17.20.3 morpheus image-builder get

Usage: morpheus image-builder get [image-build]
  -j, --json JSON Output
  -d, --dry-run Dry Run, print the API request instead of executing it
    --curl Dry Run to output API request as a curl command.
    --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -r, --remote REMOTE Remote name. The current remote is used by default.
  -remote-url URL Remote url. The current remote url is used by default.
  -T, --token TOKEN  Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help
17.20.4 morpheus image-builder list

Usage: morpheus image-builder list

- m, --max MAX Max Results
- o, --offset OFFSET Offset Results
- s, --search PHRASE Search Phrase
- S, --sort ORDER Sort Order
- D, --desc Reverse Sort Order
- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of executing it

- curl Dry Run to output API request as a curl command.
- scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by default.
- default.
- remote-url URL Remote url. The current remote url is used by default.
- default.
- T, --token TOKEN Access token for authentication with --remote.
- default.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
- timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

17.20.5 morpheus image-builder list-runs

Usage: morpheus image-builder list-runs [image-build]

- m, --max MAX Max Results
- o, --offset OFFSET Offset Results
- s, --search PHRASE Search Phrase
- S, --sort ORDER Sort Order
- D, --desc Reverse Sort Order
- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of executing it

- curl Dry Run to output API request as a curl command.
- scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by default.
- default.
- remote-url URL Remote url. The current remote url is used by default.
- default.
- T, --token TOKEN Access token for authentication with --remote.
- default.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.

(continues on next page)
-I, --insecure  Allow insecure HTTPS communication. i.e. bad SSL certificate.

-H, --header HEADER  Additional HTTP header to include with requests.

-30 seconds.  Timeout for api requests. Default is typically 30 seconds.

-C, --nocolor  Disable ANSI coloring

-B, --benchmark  Print benchmark time after the command is finished.

-V, --debug  Print extra output for debugging.

-h, --help  Print this help

Display a list of executions for an image build.

17.20.6 morpheus image-builder preseed-scripts

Usage: morpheus image-builder preseed-scripts [command] [options]

Commands:
  add
  get
  list
  remove
  update

17.20.7 morpheus image-builder remove

Usage: morpheus image-builder remove [image-build]

-K, --keep-virtual-images  Preserve associated virtual images

-a, --account ACCOUNT  Account Name

-A, --account-id ID  Account ID

-y, --yes  Auto Confirm

-j, --json  JSON Output

-d, --dry-run  Dry Run, print the API request instead of executing it

--curl  Dry Run to output API request as a curl command.

--scrub  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.

-r, --remote REMOTE  Remote name. The current remote is used by default.

--remote-url URL  Remote url. The current remote url is used by default.

-t, --token TOKEN  Access token for authentication with --remote.

Saved credentials are used by default.

-U, --username USERNAME  Username for authentication.

-P, --password PASSWORD  Password for authentication.

-I, --insecure  Allow insecure HTTPS communication. i.e. bad SSL certificate.

-H, --header HEADER  Additional HTTP header to include with requests.

-30 seconds.  Timeout for api requests. Default is typically 30 seconds.

-C, --nocolor  Disable ANSI coloring

-B, --benchmark  Print benchmark time after the command is finished.

-V, --debug  Print extra output for debugging.
17.20.8 morpheus image-builder run

Usage: morpheus image-builder run [image-build]
    -a, --account ACCOUNT  Account Name
    -A, --account-id ID    Account ID
    -y, --yes              Auto Confirm
    -j, --json             JSON Output
    -d, --dry-run          Dry Run, print the API request instead of executing it
                           --curl                               Dry Run to output API request as a curl command.
                           --scrub                               Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
                           -r, --remote REMOTE                   Remote name. The current remote is used by default.
                           --default.                            Remote url. The current remote url is used by default.
    -T, --token TOKEN      Access token for authentication with --remote.
    -U, --username USERNAME Username for authentication.
    -P, --password PASSWORD Password for authentication.
    -I, --insecure         Allow insecure HTTPS communication. i.e. bad SSL certificate.
    -H, --header HEADER    Additional HTTP header to include with requests.
                           --timeout SECONDS                    Timeout for api requests. Default is typically 30 seconds.
    -C, --nocolor          Disable ANSI coloring
    -B, --benchmark        Print benchmark time after the command is finished.
    -V, --debug            Print extra output for debugging.
    -h, --help             Print this help

17.20.9 morpheus image-builder update

Usage: morpheus image-builder update [image-build] [options]
    -t, --type TYPE        Image Build Type
    --name VALUE           New Name
    --description VALUE    Description
    -g, --group GROUP      Group Name or ID
    -c, --cloud CLOUD      Cloud Name or ID
    --config JSON          Instance Config JSON
    --config-yaml YAML     Instance Config YAML
    --config-file FILE     Instance Config from a local JSON or YAML file
    --bootScript VALUE     Boot Script ID
    --bootCommand VALUE    Boot Command. This can be used in place of a bootScript.
    --preseedScript VALUE  Preseed Script ID
    --scripts LIST         Additional Scripts (comma separated names or ids)
    --sshUsername VALUE    SSH Username
    --sshPassword VALUE    SSH Password
    --storageProvider VALUE Storage Provider ID

(continues on next page)
--isCloudInit [on|off] Cloud Init?
--buildOutputName VALUE Build Output Name
--conversionFormats VALUE Conversion Formats ie. ovf, qcow2, vhd
--keepResults VALUE Keep only the most recent builds. Older
executions will be deleted along with their associated Virtual Images. The value 0
--disables this functionality.
-O, --option OPTION Option in the format -O field="value"
--prompt Always prompts. Use passed options as the
default value.
-N, --no-prompt Skip prompts. Use default values for all
--optional fields.
--payload FILE Payload from a local JSON or YAML file, skip all
prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of
--executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the
Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by
--default.
--remote-url URL Remote url. The current remote url is used by
--default.
-t, --token TOKEN Access token for authentication with --remote.
Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad
SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically
30 seconds.
-q, --quiet No Output, do not print to stdout
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is
finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.21 morpheus instance-types

Usage: morpheus instance-types [command] [options]
Commands:
  get
  list
## 17.21.1 morpheus instance-types get

<table>
<thead>
<tr>
<th>Usage: morpheus instance-types get [name]</th>
</tr>
</thead>
<tbody>
<tr>
<td>-j, --json output</td>
</tr>
<tr>
<td>--yaml output</td>
</tr>
<tr>
<td>--csv output</td>
</tr>
<tr>
<td>--csv-delim CHAR</td>
</tr>
<tr>
<td>--csv-newline [CHAR]</td>
</tr>
<tr>
<td>--csv-quotes</td>
</tr>
<tr>
<td>--csv-no-header</td>
</tr>
<tr>
<td>-F, --fields x,y,z</td>
</tr>
<tr>
<td>Default is all fields.</td>
</tr>
<tr>
<td>-d, --dry-run</td>
</tr>
<tr>
<td>Default is all fields.</td>
</tr>
<tr>
<td>-r, --remote REMOTE</td>
</tr>
<tr>
<td>Default.</td>
</tr>
<tr>
<td>-r, --remote-url URL</td>
</tr>
<tr>
<td>Default.</td>
</tr>
<tr>
<td>-T, --token TOKEN</td>
</tr>
<tr>
<td>Default.</td>
</tr>
<tr>
<td>-U, --username USERNAME</td>
</tr>
<tr>
<td>-P, --password PASSWORD</td>
</tr>
<tr>
<td>-I, --insecure</td>
</tr>
<tr>
<td>SSL certificate.</td>
</tr>
<tr>
<td>-H, --header HEADER</td>
</tr>
<tr>
<td>--timeout SECONDS</td>
</tr>
<tr>
<td>30 seconds.</td>
</tr>
<tr>
<td>-C, --nocolor</td>
</tr>
<tr>
<td>-B, --benchmark</td>
</tr>
<tr>
<td>Default.</td>
</tr>
<tr>
<td>-V, --debug</td>
</tr>
<tr>
<td>-h, --help</td>
</tr>
</tbody>
</table>

Get instance type details. [name] is required. This is the name or id of an instance type.

## 17.21.2 morpheus instance-types list

<table>
<thead>
<tr>
<th>Usage: morpheus instance-types list</th>
</tr>
</thead>
<tbody>
<tr>
<td>-m, --max MAX</td>
</tr>
<tr>
<td>-o, --offset OFFSET</td>
</tr>
<tr>
<td>-s, --search PHRASE</td>
</tr>
<tr>
<td>-S, --sort ORDER</td>
</tr>
<tr>
<td>-d, --desc</td>
</tr>
<tr>
<td>-Q, --query PARAMS</td>
</tr>
<tr>
<td>-j, --json output</td>
</tr>
<tr>
<td>--yaml output</td>
</tr>
<tr>
<td>--csv output</td>
</tr>
<tr>
<td>--csv-delim CHAR</td>
</tr>
<tr>
<td>--csv-newline [CHAR]</td>
</tr>
<tr>
<td>--csv-quotes</td>
</tr>
</tbody>
</table>

(continues on next page)
List instance types.

17.22 morpheus instances

Usage: morpheus instances [command] [options]

Commands:
- action
- actions
- add
- apply-security-groups
- backup
- backups
- clone
- console
- containers
- count
- delenv
- eject
- envs
- exec
- firewall-disable
- firewall-enable
- get
- history

(continues on next page)
17.22.1 morpheus instances action

Usage: morpheus instances action [id list] -a CODE
-a, --action CODE Instance Action CODE to execute
-y, --yes Auto Confirm
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-q, --quiet No Output, do not print to stdout
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help
17.22.2 morpheus instances actions

Usage: morpheus instances actions [id or name list]
  -j, --json                JSON Output
  -d, --dry-run             Dry Run, print the API request instead of
  executing it
  -c, --curl                Dry Run to output API request as a curl command.
  --scrub
  -r, --remote REMOTE      Remote name. The current remote is used by
  --default.
  --remote-url URL        Remote url. The current remote url is used by
  --default.
  -T, --token TOKEN       Access token for authentication with --remote.
  --default.
  -u, --username USERNAME Username for authentication.
  -p, --password PASSWORD Password for authentication.
  -i, --insecure           Allow insecure HTTPS communication. i.e. bad
  SSL certificate.
  -H, --header HEADER     Additional HTTP header to include with requests.
  --timeout SECONDS       Timeout for api requests. Default is typically
  30 seconds.
  -c, --nocolor            Disable ANSI coloring
  -b, --benchmark          Print benchmark time after the command is
  finished.
  -v, --debug              Print extra output for debugging.
  -h, --help               Print this help

This outputs the list of the actions available to specified instance(s).

17.22.3 morpheus instances add

Usage: morpheus instances add [name] -c CLOUD -t TYPE
  -g, --group GROUP       Group Name or ID
  -c, --cloud CLOUD      Cloud Name or ID
  -t, --type CODE        Instance Type
  --name NAME            Instance Name
  --description [TEXT]   Description
  --copies NUMBER        Number of copies to provision
  --layout-size NUMBER   Apply a multiply factor of containers/vms within
  --workflow ID          Automation: Workflow ID
  --create-user on|off   User Config: Create Your User. Default is on
  --user-group USERGROUP User Config: User Group
  --shutdown-days NUMBER Automation: Shutdown Days
  --expire-days NUMBER   Automation: Expiration Days
  --create-backup [on|off] Automation: Create Backups.
  -O, --option OPTION    Option in the format -O field="value"
  --prompt
  -N, --no-prompt        Skip prompts. Use default values for all
  --optional fields.

(continues on next page)
17.22.4 morpheus instances apply-security-groups

Usage: morpheus instances apply-security-groups [name] [-S] [-c]
  -S, --secgrouops SECGROUPS    Apply the specified comma separated security group ids
  -c, --clear                   Clear all security groups
  -j, --json                    JSON Output
  -d, --dry-run                 Dry Run, print the API request instead of executing it
  --curl                        Dry Run to output API request as a curl command.
  --scrub                       Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -r, --remote REMOTE          Remote name. The current remote is used by default.
  --remote-url URL             Remote url. The current remote url is used by default.
  -T, --token TOKEN            Access token for authentication with --remote.
  --timeout SECONDS            Timeout for api requests. Default is typically 30 seconds.
  -g, --quiet                  No Output, do not print to stdout
  -C, --nocolor                Disable ANSI coloring
  -B, --benchmark              Print benchmark time after the command is finished.
  -V, --debug                  Print extra output for debugging.
  -h, --help                   Print this help

Create a new instance.
[name] is required. This is the new instance name.
The available options vary by --type.
17.22.5 morpheus instances backup

Usage: morpheus instances backup [name]

    -y, --yes          Auto Confirm
    -j, --json         JSON Output
    -d, --dry-run      Dry Run, print the API request instead of executing it
    --executing        Dry Run to output API request as a curl command.
                      --curl
                      --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
    -r, --remote REMOTE Remote name. The current remote is used by default.
    --remote-url URL   Remote url. The current remote url is used by default.
    -T, --token TOKEN  Access token for authentication with --remote.
    -U, --username USERNAME Username for authentication.
    -P, --password PASSWORD Password for authentication.
    -I, --insecure     Allow insecure HTTPS communication. i.e. bad SSL certificate.
    -H, --header HEADER Additional HTTP header to include with requests.
    --timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
    -C, --nocolor      Disable ANSI coloring
    -b, --benchmark    Print benchmark time after the command is finished.
    -V, --debug        Print extra output for debugging.
    -h, --help         Print this help

17.22.6 morpheus instances backups

Usage: morpheus instances backups [name]

    -j, --json         JSON Output
    -d, --dry-run      Dry Run, print the API request instead of executing it
    --executing        Dry Run to output API request as a curl command.
                      --curl
                      --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
    -r, --remote REMOTE Remote name. The current remote is used by default.
17.22.7 morpheus instances clone

Usage: morpheus instances clone [name] -g GROUP

--name VALUE Name - Enter a name for the new instance
-g, --group GROUP Group Name or ID for the new instance
-c, --cloud CLOUD Cloud Name or ID for the new instance
-y, --yes Auto Confirm
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.

--remote-url URL Remote name. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
 Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.22.8 morpheus instances console

Usage: morpheus instances console [name] -n NODE_ID

-n, --node NODE_ID Scope console to specific Container or VM

(continues on next page)
### 17.22.9 morpheus instances containers

Usage: morpheus instances containers [name]

- **-j, --json** JSON Output
- **--yaml** YAML Output
- **--csv** CSV Output
- **--csv-delim** CHAR Delimiter for CSV Output values. Default: ','
- **--csv-newline** [CHAR] Delimiter for CSV Output rows. Default: '
'
- **--csv-quotes** Wrap CSV values with '. Default: false
- **--csv-no-header** Exclude header for CSV Output.
- **-F, --fields** x,y,z Filter Output to a limited set of fields. Default is all fields.
- **-d, --dry-run** Dry Run, print the API request instead of executing it.
- **--curl** Dry Run to output API request as a curl command.
- **--scrub** Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- **-r, --remote** REMOTE Remote name. The current remote is used by default.
- **--remote-url** URL Remote url. The current remote url is used by default.
- **-T, --token** TOKEN Access token for authentication with --remote.
- **-U, --username** USERNAME Username for authentication.
- **-P, --password** PASSWORD Password for authentication.
- **-I, --insecure** Allow insecure HTTPS communication. i.e. bad SSL certificate.
- **-H, --header** HEADER Additional HTTP header to include with requests.
- **--timeout** SECONDS Timeout for api requests. Default is typically 30 seconds.
- **-c, --nocolor** Disable ANSI coloring
- **-B, --benchmark** Print benchmark time after the command is finished.
- **-V, --debug** Print extra output for debugging.
- **-h, --help** Print this help
17.22.10 morpheus instances count

Usage: morpheus instances count [options]

- g, --group GROUP Group Name or ID
- c, --cloud CLOUD Cloud Name or ID
  --host HOST Host Name or ID
  --created-by USER Created By User Username or ID
- s, --search PHRASE Search Phrase
- Q, --query PARAMS Query parameters. PARAMS format is
  'phrase=foobar&category=web'
- r, --remote REMOTE Remote name. The current remote is used by
  --default.
  --remote-url URL Remote url. The current remote url is used by
  --default.
- T, --token TOKEN Access token for authentication with --remote.
  Saved credentials are used by default.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. i.e. bad
- S, --ssl certificate.
- H, --header HEADER Additional HTTP header to include with requests.
  --timeout SECONDS Timeout for api requests. Default is typically
  30 seconds.
- d, --dry-run Dry Run, print the API request instead of
  executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the
  Authorization header. For use with --curl and --dry-run.
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is
  finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

Get the number of instances.

17.22.11 morpheus instances delenv

Usage: morpheus instances delenv [name] VAR

- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of
  executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the
  Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by
  --default.
  --remote-url URL Remote url. The current remote url is used by
  --default.
- T, --token TOKEN Access token for authentication with --remote.
  Saved credentials are used by default.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. i.e. bad
- S, --ssl certificate.

(continues on next page)
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.22.12 morpheus instances eject

Usage: morpheus instances eject [name]
   -y, --yes Auto Confirm
   -q, --quiet No Output, do not print to stdout
   -j, --json JSON Output
   -d, --dry-run Dry Run, print the API request instead of executing it
   --curl Dry Run to output API request as a curl command.
   --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE Remote name. The current remote is used by default.
   --remote-url URL Remote url. The current remote url is used by default.
   -T, --token TOKEN Access token for authentication with --remote.
   -U, --username USERNAME Username for authentication.
   -P, --password PASSWORD Password for authentication.
   -I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
   -H, --header HEADER Additional HTTP header to include with requests.
   --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
   -C, --nocolor Disable ANSI coloring
   -B, --benchmark Print benchmark time after the command is finished.
   -V, --debug Print extra output for debugging.
   -h, --help Print this help

Eject an instance.
[name] is required. This is the name or id of an instance. Supports 1-N [name] arguments.

17.22.13 morpheus instances envs

Usage: morpheus instances envs [name]
   -j, --json JSON Output
   -d, --dry-run Dry Run, print the API request instead of executing it
   --curl Dry Run to output API request as a curl command.
   --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE Remote name. The current remote is used by default.
   --default.
17.22.14 morpheus instances exec

Usage: morpheus instances exec [id] [options]
--script SCRIPT     Script to be executed
--file FILE         File containing the script. This can be used
--instead-of --script
--no-refresh        Do not refresh until finished
-O, --option OPTION Option in the format -O field="value"
--prompt            Always prompts. Use passed options as the
--default value.
-N, --no-prompt      Skip prompts. Use default values for all
--optional fields.
--payload FILE      Payload from a local JSON or YAML file, skip all
--payload-dir DIRECTORY Payload from a local directory containing 1-N
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting
-j, --json           JSON Output
-d, --dry-run        Dry Run, print the API request instead of
--executing it
--curl               Dry Run to output API request as a curl command.
--scrub              Mask secrets in output, such as the
--Authorization header. For use with --curl and --dry-run.
-q, --quiet          No Output, do not print to stdout
-r, --remote REMOTE Remote name. The current remote is used by
--default.
--remote-url URL     Remote url. The current remote url is used by
--default.
-T, --token TOKEN    Access token for authentication with --remote.
--Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure       Allow insecure HTTPS communication. i.e. bad
--SSL certificate.
-H, --header HEADER  Additional HTTP header to include with requests.
--timeout SECONDS    Timeout for api requests. Default is typically
--30 seconds.
-C, --nocolor        Disable ANSI coloring
-B, --benchmark      Print benchmark time after the command is
--finshed.
-V, --debug          Print extra output for debugging.
-h, --help           Print this help

(continues on next page)
Morpheus Documentation, Release 3.6.2

17.22.15 morpheus instances firewall-disable

Usage: morpheus instances firewall-disable [name]
- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of executing it.
- --curl Dry Run to output API request as a curl command.
- --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- q, --quiet No Output, do not print to stdout
- r, --remote REMOTE Remote name. The current remote is used by default.
- --remote-url URL Remote url. The current remote url is used by default.
- T, --token TOKEN Access token for authentication with --remote.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
- --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

17.22.16 morpheus instances firewall-enable

Usage: morpheus instances firewall-enable [name]
- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of executing it.
- --curl Dry Run to output API request as a curl command.
- --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- q, --quiet No Output, do not print to stdout
- r, --remote REMOTE Remote name. The current remote is used by default.
- --remote-url URL Remote url. The current remote url is used by default.
- T, --token TOKEN Access token for authentication with --remote.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
- --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help
-T, --token TOKEN  Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME  Username for authentication.
-P, --password PASSWORD  Password for authentication.
-I, --insecure  Allow insecure HTTPS communication. i.e. bad

-SSL certificate.
-H, --header HEADER  Additional HTTP header to include with requests.
- --timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor  Disable ANSI coloring
-B, --benchmark  Print benchmark time after the command is finished.
-V, --debug  Print extra output for debugging.
-h, --help  Print this help

17.22.17 morpheus instances get

Usage: morpheus instances get [instance]
--containers  Display Instance Containers
--nodes  Alias for --containers
--vms  Alias for --containers
--scaling  Display Instance Scaling Settings
--refresh [SECONDS]  Refresh until status is running, failed. Default interval is 5 seconds.
--refresh-until STATUS  Refresh until a specified status is reached.
-Q, --query PARAMS  Query parameters. PARAMS format is 'phrase=foobar&category=web'
-J, --json  JSON Output
--yaml  YAML Output
--csv  CSV Output
--csv-delim CHAR  Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR]  Delimiter for CSV Output rows. Default: '\n'
--csv-quotes  Wrap CSV values with '. Default: false
--csv-no-header  Exclude header for CSV Output.
-F, --fields x,y,z  Filter Output to a limited set of fields. Default is all fields.
-d, --dry-run  Dry Run, print the API request instead of executing it
--curl  Dry Run to output API request as a curl command.
--scrub  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE  Remote name. The current remote is used by default.
--default.
--remote-url URL  Remote url. The current remote url is used by default.
-T, --token TOKEN  Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME  Username for authentication.
-P, --password PASSWORD  Password for authentication.
-I, --insecure  Allow insecure HTTPS communication. i.e. bad

-SSL certificate.
-H, --header HEADER  Additional HTTP header to include with requests.
- --timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor  Disable ANSI coloring
Get details about an instance. [instance] is required. This is the name or id of an instance. Supports 1-N [instance] arguments.

17.22.18 morpheus instances history

Usage: morpheus instances history [instance]
--events Display sub processes (events).
--output Display process output.
--details Display more details: memory and storage usage
--used / max values.
--process-id ID Display details about a specific process only.
--event-id ID Display details about a specific process event
--only.
--max MAX Max Results
--offset OFFSET Offset Results
--search PHRASE Search Phrase
--sort ORDER Sort Order
--desc Reverse Sort Order
--event-id ID Display details about a specific process event
--query PARAMS Query parameters. PARAMS format is
'phrase=foobar&category=web'
--json JSON Output
--yaml YAML Output
--csv CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
--csv-quotes Wrap CSV values with '. Default: false
--csv-no-header Exclude header for CSV Output.
--fields x,y,z Filter Output to a limited set of fields.
--dry-run Dry Run, print the API request instead of executing it
--executing it Dry Run to output API request as a curl command.
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
--remote-remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
--token TOKEN Access token for authentication with --remote.
Saved credentials are used by default.
--username USERNAME Username for authentication.
--password PASSWORD Password for authentication.
--insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
--header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
--benchmark Print benchmark time after the command is finished.
--nocolor Disable ANSI coloring
-V, --debug  Print extra output for debugging.
-h, --help    Print this help

List historical processes for a specific instance. [instance] is required. This is the name or id of an instance.

17.22.19 morpheus instances history-details

Usage: morpheus instances history-details [instance] [process-id]
-Q, --query PARAMS  Query parameters. PARAMS format is 'phrase=foobar&category=web'
-j, --json  JSON Output
--yaml  YAML Output
--csv  CSV Output
--csv-delim CHAR  Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR]  Delimiter for CSV Output rows. Default: '\n'
--csv-quotes  Wrap CSV values with '. Default: false
--csv-no-header  Exclude header for CSV Output.
-F, --fields x,y,z  Filter Output to a limited set of fields.
-d, --dry-run  Dry Run, print the API request instead of executing it
--executing it
--curl  Dry Run to output API request as a curl command.
--scrub  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE  Remote name. The current remote is used by default.
--default.
--remote-url URL  Remote url. The current remote url is used by default.
--default.
-T, --token TOKEN  Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME  Username for authentication.
-P, --password PASSWORD  Password for authentication.
-I, --insecure  Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER  Additional HTTP header to include with requests.
--timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor  Disable ANSI coloring
-B, --benchmark  Print benchmark time after the command is finished.
-V, --debug  Print extra output for debugging.
-h, --help    Print this help

Display history details for a specific process. [instance] is required. This is the name or id of an instance. [process-id] is required. This is the id of the process.

17.22.20 morpheus instances history-event

Usage: morpheus instances history-event [instance] [event-id]
-Q, --query PARAMS  Query parameters. PARAMS format is 'phrase=foobar&category=web'

(continues on next page)
17.22.21 morpheus instances import-snapshot

Usage: morpheus instances import-snapshot [name]
        --storage-provider ID Optional storage provider
        -y, --yes Auto Confirm
        -j, --json JSON Output
        -d, --dry-run Dry Run, print the API request instead of executing it
        --curl Dry Run to output API request as a curl command.
        --scrub Mask secrets in output, such as the Authorization header.
        -r, --remote REMOTE Remote name. The current remote is used by default.
        --remote-url URL Remote url. The current remote url is used by default.
        -T, --token TOKEN Access token for authentication with --remote.
        -U, --username USERNAME Username for authentication.
        -P, --password PASSWORD Password for authentication.
        -I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
        -H, --header HEADER Additional HTTP header to include with requests.
        --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
        -C, --nocolor Disable ANSI coloring
        -B, --benchmark Print benchmark time after the command is finished.
        -V, --debug Print extra output for debugging.
        -h, --help Print this help

Display history details for a specific process event.
[instance] is required. This is the name or id of an instance.
[event-id] is required. This is the id of the process event.
### 17.22.22 morpheus instances list

**Usage:** morpheus instances list

- **-g,** **--group** GROUP  
  Group Name or ID
- **-c,** **--cloud** CLOUD  
  Cloud Name or ID
- **-h,** **--host** HOST  
  Host Name or ID
- **--created-by** USER  
  Created By User Username or ID
- **--details**  
  Display more details: memory and storage usage
- **-m,** **--max** MAX  
  Max Results
- **-o,** **--offset** OFFSET  
  Offset Results
- **-s,** **--search** PHRASE  
  Search Phrase
- **-S,** **--sort** ORDER  
  Sort Order
- **-D,** **--desc**  
  Reverse Sort Order
- **-Q,** **--query** PARAMS  
  Query parameters. PARAMS format is `phrase=foobar&category=web`
- **-j,** **--json**  
  JSON Output
- **-y,** **--yaml**  
  YAML Output
- **-c,** **--csv**  
  CSV Output
- **--csv-delim** CHAR  
  Delimiter for CSV Output values. Default: ','
- **--csv-newline** CHAR  
  Delimiter for CSV Output rows. Default: '\n'
- **--csv-quotes**  
  Wrap CSV values with ''. Default: false
- **--csv-no-header**  
  Exclude header for CSV Output.
- **-F,** **--fields** x,y,z  
  Filter Output to a limited set of fields. Default is all fields.
- **-d,** **--dry-run**  
  Dry Run, print the API request instead of executing it
- **--curl**  
  Dry Run to output API request as a curl command. Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- **--remote** REMOTE  
  Remote name. The current remote is used by default.
- **--remote-url** URL  
  Remote url. The current remote url is used by default.
- **-T,** **--token** TOKEN  
  Access token for authentication with --remote. Saved credentials are used by default.
- **-U,** **--username** USERNAME  
  Username for authentication.
- **-P,** **--password** PASSWORD  
  Password for authentication.
- **-I,** **--insecure**  
  Allow insecure HTTPS communication. i.e. bad SSL certificate.
- **-H,** **--header** HEADER  
  Additional HTTP header to include with requests.
- **--timeout** SECONDS  
  Timeout for api requests. Default is typically 30 seconds.
### 17.22.23 morpheus instances logs

Usage: morpheus instances logs [name]

- **-n, --node NODE_ID**
  Scope logs to specific Container or VM
- **-m, --max MAX**
  Max Results
- **-o, --offset OFFSET**
  Offset Results
- **-s, --search PHRASE**
  Search Phrase
- **-s, --sort ORDER**
  Sort Order
- **-d, --desc**
  Reverse Sort Order
- **-q, --query PARAMS**
  Query parameters. PARAMS format is
  - **phrase=foobara&category=web**
  - **-j, --json**
    JSON Output
  - **-csv**
    CSV Output
  - **-csv-delim CHAR**
    Delimiter for CSV Output value. Default: ','
  - **-csv-newline [CHAR]**
    Delimiter for CSV Output rows. Default: '\n'
  - **-csv-quotes**
    Wrap CSV values with '. Default: false
  - **-csv-no-header**
    Exclude header for CSV Output.
  - **-d, --dry-run**
    Dry Run, print the API request instead of executing it
  - **--curl**
    Dry Run to output API request as a curl command.
  - **--scrub**
    Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  - **-r, --remote REMOTE**
    Remote name. The current remote is used by default.
  - **--remote-url URL**
    Remote url. The current remote url is used by default.
  - **-T, --token TOKEN**
    Access token for authentication with --remote.
  - **--insecure**
    Allow insecure HTTPS communication. i.e. bad SSL certificate.
  - **-H, --header HEADER**
    Additional HTTP header to include with requests.
  - **--timeout SECONDS**
    Timeout for api requests. Default is typically 30 seconds.
  - **-C, --nocolor**
    Disable ANSI coloring
  - **-B, --benchmark**
    Print benchmark time after the command is finished.
  - **-V, --debug**
    Print extra output for debugging.
  - **-h, --help**
    Print this help

### 17.22.24 morpheus instances remove

Usage: morpheus instances remove [name]

- **--keep-backups**
  Preserve copy of backups

(continues on next page)
17.22.25 morpheus instances resize

Usage: morpheus instances resize [name]

-0, --option OPTION Option in the format -O field="value"
--prompt Always prompts. Use passed options as the
--default value.
-N, --no-prompt Skip prompts. Use default values for all
--optional fields.
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of
-executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the
--Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by
--default.
--remote-url URL Remote url. The current remote url is used by
--default.
-T, --token TOKEN Access token for authentication with --remote.
--remote-url URL Remote url. The current remote url is used by
--default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad
--H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically
--30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is
--finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

---

--preserve-volumes [on|off] Preserve Volumes. Default is off. Applies to
--certain types only.
--releaseEIPs [on|off] Release EIPS. Default is on. Applies to Amazon
--only.
-f, --force Force Delete
-y, --yes Auto Confirm
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of
-executing it
--curl
--scrub
--Authorization header. For use with --curl and --dry-run.
-q, --quiet No Output, do not print to stdout
-r, --remote REMOTE Remote name. The current remote is used by
--default.
--remote-url URL Remote url. The current remote url is used by
--default.
-T, --token TOKEN Access token for authentication with --remote.
--remote-url URL Remote url. The current remote url is used by
--default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad
--H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically
--30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is
--finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

---
### 17.22.26 morpheus instances restart

Usage: morpheus instances restart [name]

- `--yes` Auto Confirm
- `--quiet` No Output, do not print to stdout
- `--json` JSON Output
- `--dry-run` Dry Run, print the API request instead of executing it
- `--curl` Dry Run to output API request as a curl command.
- `--scrub` Mask secrets in output, such as the Authorization header. For use with `--curl` and `--dry-run`.
- `--remote REMOTE` Remote name. The current remote is used by default.
- `--remote-url URL` Remote url. The current remote url is used by default.
- `--token TOKEN` Access token for authentication with `--remote`.
- `--username USERNAME` Username for authentication.
- `--password PASSWORD` Password for authentication.
- `--insecure` Allow insecure HTTPS communication. i.e. bad SSL certificate.
- `--header HEADER` Additional HTTP header to include with requests.
- `--timeout SECONDS` Timeout for api requests. Default is typically 30 seconds.
- `--nocolor` Disable ANSI coloring
- `--benchmark` Print benchmark time after the command is finished.
- `--debug` Print extra output for debugging.
- `--help` Print this help

Restart an instance. [name] is required. This is the name or id of an instance. Supports 1–N [name] arguments.

### 17.22.27 morpheus instances restart-service

Usage: morpheus instances restart-service [name]

- `--yes` Auto Confirm
- `--quiet` No Output, do not print to stdout
- `--json` JSON Output
- `--dry-run` Dry Run, print the API request instead of executing it
- `--curl` Dry Run to output API request as a curl command.
- `--scrub` Mask secrets in output, such as the Authorization header. For use with `--curl` and `--dry-run`.
- `--remote REMOTE` Remote name. The current remote is used by default.
- `--remote-url URL` Remote url. The current remote url is used by default.
- `--token TOKEN` Access token for authentication with `--remote`.
- `--username USERNAME` Username for authentication.
- `--password PASSWORD` Password for authentication.
- `--insecure` Allow insecure HTTPS communication. i.e. bad SSL certificate.
- `--header HEADER` Additional HTTP header to include with requests.
- `--timeout SECONDS` Timeout for api requests. Default is typically 30 seconds.
- `--nocolor` Disable ANSI coloring
- `--benchmark` Print benchmark time after the command is finished.
- `--debug` Print extra output for debugging.
- `--help` Print this help
Restart service on an instance.

Usage: morpheus instances run-workflow [name] [workflow] [options]
-O, --option OPTION Option in the format -O field="value"
--prompt Always prompts. Use passed options as the default value.
-N, --no-prompt Skip prompts. Use default values for all optional fields.
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
--benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.22. morpheus instances run-workflow
17.22.29 morpheus instances scaling

Usage: morpheus instances scaling [name]

- j, --json JSON Output
--yaml YAML Output
--csv CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
--csv-quotes Wrap CSV values with '". Default: false
--csv-no-header Exclude header for CSV Output.
-F, --fields x,y,z Filter Output to a limited set of fields.
-d, --dry-run Dry Run, print the API request instead of executing it
--crl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header.
--remoteREMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
--token TOKEN Access token for authentication with --remote.
-saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
--insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
--header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
--nocolor Disable ANSI coloring
--benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Show scaling threshold information for an instance.

17.22.30 morpheus instances scaling-update

Usage: morpheus instances scaling-update [name]

- autoUp on|off Auto Upscale - Enable auto upscaling
--autoDown on|off Auto Downscale - Enable auto downscaling
--zoneId ID Cloud (optional) - Choose a cloud to scale into.
--minCount NUMBER Min Count (optional) - Minimum number of nodes
--maxCount NUMBER Max Count (optional) - Maximum number of nodes
--memoryEnabled on|off Enable Memory Threshold - Scale when memory thresholds are met.
--memoryEnabled on|off Enable Memory Threshold - Scale when memory thresholds are met.
--minMemory PERCENT Min Memory (optional) - Minimum memory percent (0-100)}
(continued from previous page)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--maxMemory PERCENT</td>
<td>Max Memory (optional) - Maximum memory percent.</td>
</tr>
<tr>
<td>(0-100)</td>
<td></td>
</tr>
<tr>
<td>--diskEnabled on</td>
<td>off</td>
</tr>
<tr>
<td>--minDisk PERCENT</td>
<td>Min Disk (optional) - Minimum storage percent (0-100).</td>
</tr>
<tr>
<td>--maxDisk PERCENT</td>
<td>Max Disk (optional) - Maximum storage percent (0-100).</td>
</tr>
<tr>
<td>--cpuEnabled on</td>
<td>off</td>
</tr>
<tr>
<td>--minCpu PERCENT</td>
<td>Min CPU (optional) - Minimum CPU percent (0-100).</td>
</tr>
<tr>
<td>--maxCpu PERCENT</td>
<td>Max CPU (optional) - Maximum CPU percent (0-100).</td>
</tr>
<tr>
<td>-O, --option OPTION</td>
<td>Option in the format -O field=&quot;value&quot;. Always prompts. Use passed options as the default value.</td>
</tr>
<tr>
<td>-N, --no-prompt</td>
<td>Skip prompts. Use default values for all optional fields.</td>
</tr>
<tr>
<td>-j, --json</td>
<td>JSON Output.</td>
</tr>
<tr>
<td>-d, --dry-run</td>
<td>Dry Run, print the API request instead of executing it.</td>
</tr>
<tr>
<td>--curl</td>
<td>Dry Run to output API request as a curl command.</td>
</tr>
<tr>
<td>--scrub</td>
<td>Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.</td>
</tr>
<tr>
<td>-r, --remote REMOTE</td>
<td>Remote name. The current remote is used by default.</td>
</tr>
<tr>
<td>--remote-url URL</td>
<td>Remote url. The current remote url is used by default.</td>
</tr>
<tr>
<td>-T, --token TOKEN</td>
<td>Access token for authentication with --remote. Saved credentials are used by default.</td>
</tr>
<tr>
<td>-U, --username USERNAME</td>
<td>Username for authentication.</td>
</tr>
<tr>
<td>-P, --password PASSWORD</td>
<td>Password for authentication.</td>
</tr>
<tr>
<td>-I, --insecure</td>
<td>Allow insecure HTTPS communication. i.e. bad SSL certificate.</td>
</tr>
<tr>
<td>-H, --header HEADER</td>
<td>Additional HTTP header to include with requests.</td>
</tr>
<tr>
<td>--timeout SECONDS</td>
<td>Timeout for api requests. Default is typically 30 seconds.</td>
</tr>
<tr>
<td>-C, --nocolor</td>
<td>Disable ANSI coloring</td>
</tr>
<tr>
<td>-b, --benchmark</td>
<td>Print benchmark time after the command is finished.</td>
</tr>
<tr>
<td>-V, --debug</td>
<td>Print extra output for debugging.</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Print this help</td>
</tr>
</tbody>
</table>

Update scaling threshold information for an instance.

### 17.22.31 morpheus instances security-groups

#### Usage: morpheus instances security-groups [name]

- `--maxMemory PERCENT`: Max Memory (optional) - Maximum memory percent.
- `--diskEnabled on|off`: Enable Disk Threshold - Scale when disk thresholds are met.
- `--minDisk PERCENT`: Min Disk (optional) - Minimum storage percent (0-100).
- `--maxDisk PERCENT`: Max Disk (optional) - Maximum storage percent (0-100).
- `--cpuEnabled on|off`: Enable CPU Threshold - Scale when cpu thresholds are met.
- `--minCpu PERCENT`: Min CPU (optional) - Minimum CPU percent (0-100).
- `--maxCpu PERCENT`: Max CPU (optional) - Maximum CPU percent (0-100).

(continues on next page)
--remote-url URL  Remote url. The current remote url is used by default.
-T, --token TOKEN  Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME  Username for authentication.
-P, --password PASSWORD  Password for authentication.
-I, --insecure  Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER  Additional HTTP header to include with requests.
--timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
--no-color  Disable ANSI coloring
-B, --benchmark  Print benchmark time after the command is finished.
-V, --debug  Print extra output for debugging.
-h, --help  Print this help

### 17.22.32 morpheus instances setenv

Usage: morpheus instances setenv [name] VAR VALUE [-e]
-e  Exportable
-M  Masked
-j, --json  JSON Output
-d, --dry-run  Dry Run, print the API request instead of executing it
--curl  Dry Run to output API request as a curl command.
--scrub  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE  Remote name. The current remote is used by default.
--remote-url URL  Remote name. The current remote url is used by default.
-T, --token TOKEN  Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME  Username for authentication.
-P, --password PASSWORD  Password for authentication.
-I, --insecure  Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER  Additional HTTP header to include with requests.
--timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
-q, --quiet  No Output, do not print to stdout
-C, --no-color  Disable ANSI coloring
-B, --benchmark  Print benchmark time after the command is finished.
-V, --debug  Print extra output for debugging.
-h, --help  Print this help

### 17.22.33 morpheus instances start

Usage: morpheus instances start [name]
-y, --yes  Auto Confirm

(continues on next page)
-q, --quiet No Output, do not print to stdout
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote. Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Start an instance.
[name] is required. This is the name or id of an instance. Supports 1-N [name] arguments.

17.22.34 morpheus instances start-service

Usage: morpheus instances start-service [name]

-yes Auto Confirm
-q, --quiet No Output, do not print to stdout
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote. Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
Start service on an instance. [name] is required. This is the name or id of an instance. Supports 1-N [name] arguments.

### 17.22.35 morpheus instances stats

Usage: morpheus instances stats [name]

- `--json` JSON Output
- `--yaml` YAML Output
- `--csv` CSV Output
- `--csv-delim` CHAR Delimiter for CSV Output values. Default: ','
- `--csv-newline` CHAR Delimiter for CSV Output rows. Default: '
'
- `--csv-quotes` Wrap CSV values with ''. Default: false
- `--csv-no-header` Exclude header for CSV Output.
- `--fields` x,y,z Filter Output to a limited set of fields.
- `--json` JSON Output
- `--yaml` YAML Output
- `--csv` CSV Output
- `--csv-delim` CHAR Delimiter for CSV Output values. Default: ','
- `--csv-newline` CHAR Delimiter for CSV Output rows. Default: '
'
- `--csv-quotes` Wrap CSV values with ''. Default: false
- `--csv-no-header` Exclude header for CSV Output.
- `--fields` x,y,z Filter Output to a limited set of fields.
- `--dry-run` Dry Run, print the API request instead of executing it
- `--curl` Dry Run to output API request as a curl command.
- `--scrub` Mask secrets in output, such as the Authorization header. For use with `--curl` and `--dry-run`.
- `--remote` REMOTE Remote name. The current remote is used by default.
- `--remote-url` URL Remote url. The current remote url is used by default.
- `--token` TOKEN Access token for authentication with `--remote`.
- `--insecure` Allow insecure HTTPS communication. i.e. bad SSL certificate.
- `--header` HEADER Additional HTTP header to include with requests.
- `--timeout` SECONDS Timeout for api requests. Default is typically 30 seconds.
- `--nocolor` Disable ANSI coloring
- `--benchmark` Print benchmark time after the command is finished.
- `--debug` Print extra output for debugging.
- `--help` Print this help

### 17.22.36 morpheus instances status-check

Usage: morpheus instances status-check [name]

- `--quiet` No Output, do not print to stdout
- `--json` JSON Output
- `--yaml` YAML Output
- `--csv` CSV Output
- `--csv-delim` CHAR Delimiter for CSV Output values. Default: ','
- `--csv-newline` CHAR Delimiter for CSV Output rows. Default: '
'
- `--csv-quotes` Wrap CSV values with ''. Default: false
- `--csv-no-header` Exclude header for CSV Output.
- `--fields` x,y,z Filter Output to a limited set of fields.
- `--remote` REMOTE Remote name. The current remote is used by default.
- `--remote-url` URL Remote url. The current remote url is used by default.
- `--token` TOKEN Access token for authentication with `--remote`.
- `--insecure` Allow insecure HTTPS communication. i.e. bad SSL certificate.
- `--header` HEADER Additional HTTP header to include with requests.
- `--timeout` SECONDS Timeout for api requests. Default is typically 30 seconds.
- `--nocolor` Disable ANSI coloring
- `--benchmark` Print benchmark time after the command is finished.
- `--debug` Print extra output for debugging.
- `--help` Print this help
-T, --token TOKEN
Saved credentials are used by default.
-U, --username USERNAME
Username for authentication.
-P, --password PASSWORD
Password for authentication.
-I, --insecure
Allow insecure HTTPS communication. i.e. bad
SSL certificate.
-H, --header HEADER
Additional HTTP header to include with requests.
--timeout SECONDS
Timeout for api requests. Default is typically 30 seconds.
--nocolor
Disable ANSI coloring
-B, --benchmark
Print benchmark time after the command is finished.
-V, --debug
Print extra output for debugging.
-h, --help
Print this help

17.22.37 morpheus instances stop

Usage: morpheus instances stop [name]
--mute-monitoring [on|off] Mute monitoring. Default is off.
-y, --yes Auto Confirm
-q, --quiet No Output, do not print to stdout
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN
Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
--nocolor
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug
Print extra output for debugging.
-h, --help
Print this help

Stop an instance.
[instance] is required. This is the name or id of an instance. Supports 1-N[instance] arguments.
17.22.38 morpheus instances stop-service

Usage: morpheus instances stop-service [name]

--mute-monitoring [on|off] Mute monitoring. Default is off.
-y, --yes Auto Confirm
-g, --quiet No Output, do not print to stdout
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--executing it
-curl Dry Run to output API request as a curl command.
--curl
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
--scrub
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote
--default.
--remote-url URL Remote url. The current remote url is used by default.
--remote-url
--default.
-T, --token TOKEN Access token for authentication with --remote.
--token
--default.
-U, --username USERNAME Username for authentication.
--username
-P, --password PASSWORD Password for authentication.
--password
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
--insecure
-H, --header HEADER Additional HTTP header to include with requests.
--header
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
--timeout
-C, --nocolor Disable ANSI coloring
--nocolor
-B, --benchmark Print benchmark time after the command is finished.
--benchmark
-V, --debug Print extra output for debugging.
--debug
-h, --help Print this help

Stop service on an instance.
[name] is required. This is the name or id of an instance. Supports 1-N [name] arguments.

17.22.39 morpheus instances suspend

Usage: morpheus instances suspend [name]

--mute-monitoring [on|off] Mute monitoring. Default is off.
-y, --yes Auto Confirm
-g, --quiet No Output, do not print to stdout
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--executing it
-curl Dry Run to output API request as a curl command.
--curl
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
--scrub
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote
--default.
--remote-url URL Remote url. The current remote url is used by default.
--remote-url
--default.
-T, --token TOKEN Access token for authentication with --remote.
--token
--default.
-U, --username USERNAME Username for authentication.
--username
-P, --password PASSWORD Password for authentication.
--password

(continues on next page)
-I, --insecure  
Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER  
Additional HTTP header to include with requests.
--timeout SECONDS  
Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor  
Disable ANSI coloring
-B, --benchmark  
Print benchmark time after the command is finished.
-V, --debug  
Print extra output for debugging.
-h, --help  
Print this help

Suspend an instance.
[name] is required. This is the name or id of an instance. Supports 1-N [name] arguments.

17.22.40 morpheus instances update

Usage: morpheus instances update [name]

--name VALUE  Name
--description VALUE  Description
--environment VALUE  Environment
--group GROUP  Group Name or ID
--metadata LIST  Metadata in the format 'name:value, name:value'
--tags LIST  Tags
--power-schedule-type ID  Power Schedule Type ID
--created-by ID  Created By User ID
-O, --option OPTION  Option in the format -O field="value"
--prompt  Always prompts. Use passed options as the default value.
-N, --no-prompt  Skip prompts. Use default values for all optional fields.
--payload FILE  Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY  Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-yaml YAML  Payload YAML, skip all prompting
--payload-json JSON  Payload JSON, skip all prompting
-j, --json  JSON Output
-d, --dry-run  Dry Run, print the API request instead of executing it
--curl  Dry Run to output API request as a curl command.
--scrub  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE  Remote name. The current remote is used by default.
--remote-url URL  Remote url. The current remote url is used by default.
-T, --token TOKEN  Access token for authentication with --remote.
--saved-credentials are used by default.
-U, --username USERNAME  Username for authentication.
-P, --password PASSWORD  Password for authentication.
-I, --insecure  Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER  Additional HTTP header to include with requests.
--timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
### 17.22.41 morpheus instances update-notes

Usage: morpheus instances update-notes [name]

- `--notes VALUE` Notes content (Markdown)
- `--file FILE` File containing the notes content. This can be used instead of `--notes`
- `--clear` Clear current notes
- `--option OPTION` Option in the format `-O field="value"
- `--prompt` Always prompts. Use passed options as the default value.
- `--no-prompt` Skip prompts. Use default values for all optional fields.
- `--payload FILE` Payload from a local JSON or YAML file, skip all prompting
- `--payload-dir DIRECTORY` Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
- `--payload-json JSON` Payload JSON, skip all prompting
- `--payload-yaml YAML` Payload YAML, skip all prompting
- `--json JSON` Output
- `--dry-run` Dry Run, print the API request instead of executing it
- `--curl` Dry Run to output API request as a curl command.
- `--scrub` Mask secrets in output, such as the Authorization header. For use with `--curl` and `--dry-run`.
- `--remote REMOTE` Remote name. The current remote is used by default.
- `--remote-url URL` Remote url. The current remote url is used by default.
- `--token TOKEN` Access token for authentication with `--remote`
- `--insecure` Allow insecure HTTPS communication. i.e. bad SSL certificate.
- `--header HEADER` Additional HTTP header to include with requests.
- `--timeout SECONDS` Timeout for api requests. Default is typically 30 seconds.
- `--nocolor` Disable ANSI coloring
- `--benchmark` Print benchmark time after the command is finished.
- `--debug` Print extra output for debugging.
- `-h, --help` Print this help
17.23 morpheus key-pairs

Usage: morpheus key-pairs [command] [options]
Commands:
  add
  get
  list
  remove
  update

17.23.1 morpheus key-pairs add

Usage: morpheus key-pairs add [name] [options]
  --public-key-file FILENAME Public Key File
  --public-key TEXT Public Key Text
  --private-key-file FILENAME Private Key File
  --private-key TEXT Private Key Text
  -a, --account ACCOUNT Account Name
  -A, --account-id ID Account ID
  -O, --option OPTION Option in the format -O field="value"
  --prompt Always prompts. Use passed options as the default value.
  -N, --no-prompt Skip prompts. Use default values for all optional fields.
  -j, --json JSON Output
  -d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -r, --remote REMOTE Remote name. The current remote is used by default.
  -T, --token TOKEN Access token for authentication with --remote.
  -U, --username USERNAME Username for authentication.
  -P, --password PASSWORD Password for authentication.
  -I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
  -H, --header HEADER Additional HTTP header to include with requests.
  --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
  -C, --nocolor Disable ANSI coloring
  -B, --benchmark Print benchmark time after the command is finished.
  -V, --debug Print extra output for debugging.
  -h, --help Print this help
17.23.2 morpheus key-pairs get

Usage: morpheus key-pairs get [name]

-\(a\), --account ACCOUNT
  Account Name
-\(A\), --account-id ID
  Account ID
-\(Q\), --query PARAMS
  Query parameters. PARAMS format is
  \texttt{phrase=foobar\&category=web}'
-\(j\), --json
  JSON Output
-\(y\), --yaml
  YAML Output
-\(c\), --csv
  CSV Output
  --csv-delim CHAR
  Delimiter for CSV Output values. Default: ','
  --csv-newline [CHAR]
  Delimiter for CSV Output rows. Default: '
'
  --csv-quotes
  Wrap CSV values with '. Default: false
  --csv-no-header
  Exclude header for CSV Output.
  --fields x,y,z
  Filter Output to a limited set of fields.
-\(d\), --dry-run
  Dry Run, print the API request instead of executing it
  --curl
  Dry Run to output API request as a curl command.
  --scrub
  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-\(r\), --remote REMOTE
  Remote name. The current remote is used by default.
  --remote-url URL
  Remote url. The current remote url is used by default.
-\(T\), --token TOKEN
  Access token for authentication with --remote. Saved credentials are used by default.
  --username USERNAME
  Username for authentication.
  --password PASSWORD
  Password for authentication.
  --insecure
  Allow insecure HTTPS communication. i.e. bad SSL certificate.
  --timeout SECONDS
  Timeout for api requests. Default is typically 30 seconds.
  --nocolor
  Disable ANSI coloring
  --benchmark
  Print benchmark time after the command is finished.
-\(V\), --debug
  Print extra output for debugging.
-\(h\), --help

17.23.3 morpheus key-pairs list

Usage: morpheus key-pairs list

-\(a\), --account ACCOUNT
  Account Name
-\(A\), --account-id ID
  Account ID
-\(m\), --max MAX
  Max Results
-\(o\), --offset OFFSET
  Offset Results
-\(s\), --search PHRASE
  Search Phrase
-\(S\), --sort ORDER
  Sort Order
  --desc
  Reverse Sort Order
-\(Q\), --query PARAMS
  Query parameters. PARAMS format is
  \texttt{phrase=foobar\&category=web}'
-\(j\), --json
  JSON Output
-\(y\), --yaml
  YAML Output
  --csv
  CSV Output
17.23.4 morpheus key-pairs remove

Usage: morpheus key-pairs remove [name]
- a, --account ACCOUNT       Account Name
- A, --account-id ID         Account ID
- y, --yes                   Auto Confirm
- j, --json                   JSON Output
- d, --dry-run               Dry Run, print the API request instead of
  executing it
    -c, --curl                     Dry Run to output API request as a curl command.
    -s, --scrub                   Mask secrets in output, such as the
  Authorization header. For use with --dry-run.
- r, --remote REMOTE          Remote name. The current remote is used by
- u, --remote-url URL         Remote url. The current remote url is used by
  default.
- t, --token TOKEN            Access token for authentication with --remote.
- U, --username USERNAME      Username for authentication.
- P, --password PASSWORD      Password for authentication.
- I, --insecure               Allow insecure HTTPS communication. i.e. bad
  SSL certificate.
- H, --header HEADER          Additional HTTP header to include with requests.
- 30 seconds.
- C, --nocolor                Disable ANSI coloring
- B, --benchmark             Print benchmark time after the command is
  finished.
- V, --debug                  Print extra output for debugging.
- h, --help                   Print this help

(continues on next page)
### 17.23.5 morpheus key-pairs update

Usage: morpheus key-pairs update [name] [options]

- **--name** VALUE Name
- **-a, --account** ACCOUNT Account Name
- **-A, --account-id** ID Account ID
- **-O, --option** OPTION Option in the format -O field="value"
  - **--prompt** Always prompts. Use passed options as the default value.
  - **-N, --no-prompt** Skip prompts. Use default values for all optional fields.
- **-j, --json** JSON Output
- **-d, --dry-run** Dry Run, print the API request instead of executing it
  - **--curl** Dry Run to output API request as a curl command.
  - **--scrub** Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- **-r, --remote** REMOTE Remote name. The current remote is used by default.
  - **--remote-url** URL Remote url. The current remote url is used by default.
- **-T, --token** TOKEN Access token for authentication with --remote.
- **-U, --username** USERNAME Username for authentication.
- **-P, --password** PASSWORD Password for authentication.
- **-I, --insecure** Allow insecure HTTPS communication. i.e. bad SSL certificate.
- **-H, --header** HEADER Additional HTTP header to include with requests.
- **--timeout** SECONDS Timeout for api requests. Default is typically 30 seconds.
- **-C, --nocolor** Disable ANSI coloring
- **-B, --benchmark** Print benchmark time after the command is finished.
- **-V, --debug** Print extra output for debugging.
- **-h, --help** Print this help

### 17.24 morpheus library-file-templates

Usage: morpheus library-file-templates [command] [options]

Commands:
- add
- get
- list
- remove
- update
17.24.1 morpheus library-file-templates add

Usage: morpheus library-file-templates add [name]

--name VALUE Name
--fileName VALUE File Name
--filePath VALUE File Path
--phase [start|stop|postProvision] Template Phase. Default is 'provision'
--category VALUE Category
--template TEXT Contents of the template. This can be used instead of --template
-O, --option OPTION Option in the format -O field="value"
--prompt Always prompts. Use passed options as the default value.
-N, --no-prompt Skip prompts. Use default values for all optional fields.
--payload FILE Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-q, --quiet No Output, do not print to stdout
-C, --nocolor Disable ANSI coloring
-b, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Create a new file template. [name] is required and can be passed as --name instead.

17.24.2 morpheus library-file-templates get

Usage: morpheus library-file-templates get [name]

(continues on next page)
17.24.3 morpheus library-file-templates list

Usage: morpheus library-file-templates list

-m, --max MAX Max Results
-o, --offset OFFSET Offset Results
-s, --search PHRASE Search Phrase
-S, --sort ORDER Sort Order
-D, --desc Reverse Sort Order
-j, --json JSON Output
--yaml YAML Output
--csv CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '
'
--csv-quotes Wrap CSV values with '"'. Default: false
--csv-no-header Exclude header for CSV Output.
-F, --fields x,y,z Filter Output to a limited set of fields.
-Default is all fields.
-d, --dry-run Dry Run, print the API request instead of executing it
--curl
--scrub

Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
--token TOKEN Access token for authentication with --remote.

Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.24.4 morpheus library-file-templates remove

Usage: morpheus library-file-templates remove [name]
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-q, --quiet No Output, do not print to stdout
-y, --yes Auto Confirm
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.24.5 morpheus library-file-templates update

Usage: morpheus library-file-templates update [name]
--name VALUE Name
--fileName VALUE File Name
--filePath VALUE File Path
--phase [start|stop] Template Phase
--template TEXT Contents of the template.
--file FILE File containing the template. This can be used instead of --template
--option OPTION Option in the format -O field="value"
--prompt Always prompts. Use passed options as the default value.
-N, --no-prompt Skip prompts. Use default values for all optional fields.
--payload FILE Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY  Payload from a local directory containing 1-N
--json or YAML files, skip all prompting
--payload-json JSON  Payload JSON, skip all prompting
--payload-yaml YAML  Payload YAML, skip all prompting
-j, --json  JSON Output
-d, --dry-run  Dry Run, print the API request instead of
--executing it
--curl  Dry Run to output API request as a curl command.
--scrub  Mask secrets in output, such as the
Authorization header. For use with --curl and --dry-run.
r, --remote REMOTE  Remote name. The current remote is used by
--default.
--remote-url URL  Remote url. The current remote url is used by
--default.
-t, --token TOKEN  Access token for authentication with --remote.
Saved credentials are used by default.
-U, --username USERNAME  Username for authentication.
P, --password PASSWORD  Password for authentication.
-i, --insecure  Allow insecure HTTPS communication. i.e. bad
SSL certificate.
-H, --header HEADER  Additional HTTP header to include with requests.
--timeout SECONDS  Timeout for api requests. Default is typically
30 seconds.
-q, --quiet  No Output, do not print to stdout
-C, --nocolor  Disable ANSI coloring
-B, --benchmark  Print benchmark time after the command is
finished.
-V, --debug  Print extra output for debugging.
h, --help  Print this help

Update a file template.
[name] is required. This is the name or id of a file template.

17.25 morpheus library-instance-types

Usage: morpheus library-instance-types [command] [options]
Commands:
  add
  get
  list
  remove
  update
  update-logo

17.25.1 morpheus library-instance-types add

Usage: morpheus library-instance-types add
  --name VALUE  Name
  --code VALUE  Code - Useful shortcode for provisioning naming
  --schemes and export reference.
  --description VALUE  Description (optional)
  --category VALUE  Category

(continues on next page)
Create a new instance type.

17.25.2 morpheus library-instance-types get

Usage: morpheus library-instance-types get [name]

- j, --json JSON Output
- y, --yaml YAML Output
- csv CSV Output
- csv-delim CHAR Delimiter for CSV Output values. Default: ','
- csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
17.25.3 morpheus library-instance-types list

Usage: morpheus library-instance-types list

--category VALUE Filter by category
--code VALUE Filter by code
--technology VALUE Filter by technology
-m, --max MAX Max Results
-o, --offset OFFSET Offset Results
-s, --search PHRASE Search Phrase
-S, --sort ORDER Sort Order
-D, --desc Reverse Sort Order
-Q, --query PARAMS Query parameters. PARAMS format is
-phrase=foo\&category=web'
-j, --json JSON Output
--yaml YAML Output
--csv CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
--csv-quotes Wrap CSV values with '. Default: false
--csv-no-header Exclude header for CSV Output.
-F, --fields x,y,z Filter Output to a limited set of fields.
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
--token TOKEN Access token for authentication with --remote.
-U, --username USERNAME Username for authentication.
-T, --token PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help
-r, --remote REMOTE Remote name. The current remote is used by default.
-r, --remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

List instance types.

17.25.4 morpheus library-instance-types remove

Usage: morpheus library-instance-types remove [name]
   -y, --yes Auto Confirm
   -j, --json JSON Output
   -d, --dry-run Dry Run, print the API request instead of executing it
     -c, --curl Dry Run to output API request as a curl command.
     -s, --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE Remote name. The current remote is used by default.
   -t, --remote-url URL Remote url. The current remote url is used by default.
   -T, --token TOKEN Access token for authentication with --remote.
   -U, --username USERNAME Username for authentication.
   -P, --password PASSWORD Password for authentication.
   -I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
   -H, --header HEADER Additional HTTP header to include with requests.
   --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
   -C, --nocolor Disable ANSI coloring
   -B, --benchmark Print benchmark time after the command is finished.
   -V, --debug Print extra output for debugging.
   -h, --help Print this help

Delete an instance type.
17.25.5 morpheus library-instance-types update

Usage: morpheus library-instance-types update [name] [options]
- name VALUE Name
- code VALUE Code - Useful shortcode for provisioning naming
- schemes and export reference.
  - description VALUE Description (optional)
  - category VALUE Category
  - visibility VALUE Visibility (optional) Default: private
  - environmentPrefix VALUE Environment Prefix (optional) - Used for
    exportable environment variables when tying instance types together in app contexts.
  - If not specified a name will be generated.
    - hasSettings on|off Enable Settings (optional)
    - hasAutoScale on|off Enable Scaling (Horizontal) (optional)
    - hasDeployment on|off Supports Deployments (optional) - Requires a
      data volume be configured on each version. Files will be copied into this location.
    -O, --option OPTION Option in the format -O field="value"
    -prompt Always prompts. Use passed options as the
    -N, --no-prompt Skip prompts. Use default values for all
- optional fields.
  -j, --json JSON Output
  -d, --dry-run Dry Run, print the API request instead of
  -executing it
    -c, --curl Dry Run to output API request as a curl command.
    -s, --scrub Mask secrets in output, such as the
  -r, --remote REMOTE Remote name. The current remote is used by
  -default.
    -remote-url URL Remote url. The current remote url is used by
  -default.
  -T, --token TOKEN Access token for authentication with --remote.
- Saved credentials are used by default.
  -U, --username USERNAME Username for authentication.
  -P, --password PASSWORD Password for authentication.
  -I, --insecure Allow insecure HTTPS communication. i.e. bad
  -SSL certificate.
  -H, --header HEADER Additional HTTP header to include with requests.
    -timeout SECONDS Timeout for api requests. Default is typically
  -30 seconds.
  -C, --nocolor Disable ANSI coloring
  -B, --benchmark Print benchmark time after the command is
  -finished.
  -V, --debug Print extra output for debugging.
  -h, --help Print this help

Update an instance type.
[name] is required. This is the name or id of a instance type.

17.25.6 morpheus library-instance-types update-logo

Usage: morpheus library-instance-types update-logo [name] [file]
- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of
- executing it
--curl
--scrub
--Authorization header. For use with --curl and --dry-run.
--remote REMOTE
--remote-url URL
--remote
--default.
--token TOKEN
--username USERNAME
--password PASSWORD
--insecure
--SSL certificate.
--header HEADER
--timeout SECONDS
--30 seconds.
--nocolor
--benchmark
--finished.
--debug
--help

Update the logo for an instance type.
[name] is required. This is the name or id of a instance type.
[file] is required. This is the path of the logo file

17.26 morpheus library-layouts

Usage: morpheus library-layouts [command] [options]
Commands:
  add
  get
  list
  remove
  update

17.26.1 morpheus library-layouts add

Usage: morpheus library-layouts add [instance-type]
--instance-type ID
--name VALUE
--version VALUE
--description VALUE
--technology CODE
--min-memory VALUE
--workflow ID
--option-types x,y,z
--node-types x,y,z
-O, --option OPTION
--prompt
--default value.
-N, --no-prompt

(continues on next page)
Create a new layout.

[instance-type] is required and can be passed as --instance-type instead.

### 17.26.2 morpheus library-layouts get

Usage: morpheus library-layouts get [name]

- j, --json JSON Output
- yam, --yaml YAML Output
- csv CSV Output
- csv-delim CHAR Delimiter for CSV Output values. Default: ','
- csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
- csv-quotes Wrap CSV values with ". Default: false
- csv-no-header Exclude header for CSV Output.
- f, --fields x,y,z Filter CSV Output to a limited set of fields.
- d, --dry-run Dry Run, print the API request instead of executing it
- curl Dry Run to output API request as a curl command. Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by default.
- remote-url URL Remote url. The current remote url is used by default.
- t, --token TOKEN Access token for authentication with --remote.
- u, --username USERNAME Username for authentication.
- p, --password PASSWORD Password for authentication.
- insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- header HEADER Additional HTTP header to include with requests.
- timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- nocolor Disable ANSI coloring
- benchmark Print benchmark time after the command is finished.
- debug Print extra output for debugging.
- help Print this help

Saved credentials are used by default.

(continues on next page)
17.26.3 morpheus library-layouts list

Usage: morpheus library-layouts list

--instance-type ID Filter by Instance Type
--category VALUE Filter by category
--code VALUE Filter by code
--technology VALUE Filter by technology
--max MAX Max Results
--offset OFFSET Offset Results
--search PHRASE Search Phrase
--sort ORDER Sort Order
--desc Reverse Sort Order
--query PARAMS Query parameters. PARAMS format is
'phrase=foobar&category=web'
--json JSON Output
--yaml YAML Output
--csv CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
--csv-quotes Wrap CSV values with '. Default: false
--csv-no-header Exclude header for CSV Output.
--fields x,y,z Filter CSV Output to a limited set of fields.
--dry-run Dry Run, print the API request instead of
executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the
Authorization header. For use with --curl and --dry-run.
--remote REMOTE Remote name. The current remote is used by
--default.
--remote-url URL Remote url. The current remote url is used by
--default.
--token TOKEN Access token for authentication with --remote.
Saved credentials are used by default.
--username USERNAME Username for authentication.
--password PASSWORD Password for authentication.
--insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
--header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
--nocolor Disable ANSI coloring
--benchmark Print benchmark time after the command is finished.
--help Print this help
-C, --nocolor  Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug  Print extra output for debugging.
-h, --help  Print this help

List layouts.

17.26.4 morpheus library-layouts remove

Usage: morpheus library-layouts remove [name]

- y, --yes  Auto Confirm
- j, --json  JSON Output
- d, --dry-run  Dry Run, print the API request instead of executing it
--curl  Dry Run to output API request as a curl command.
--scrub  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE  Remote name. The current remote is used by default.
--remote-url URL  Remote url. The current remote url is used by default.
-T, --token TOKEN  Access token for authentication with --remote.
--header HEADER  Additional HTTP header to include with requests.
--timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor  Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug  Print extra output for debugging.
-h, --help  Print this help

Delete a layout.

17.26.5 morpheus library-layouts update

Usage: morpheus library-layouts update [name] [options]

--name VALUE  Name for this layout
--version VALUE  Version
--description VALUE  Description
--min-memory VALUE  Minimum Memory (MB)
-workflow ID  Workflow
--option-types x,y,z  List of Option Type IDs
--node-types x,y,z  List of Node Type IDs
-O, --option OPTION  Option in the format -O field="value"
--prompt  Always prompts. Use passed options as the default value.
-N, --no-prompt  Skip prompts. Use default values for all optional fields.
Update a layout.

17.27  morpheus library-node-types

Usage: morpheus library-node-types [command] [options]
Commands:

add
get
list
remove
update

17.27.1  morpheus library-node-types add

Usage: morpheus library-node-types add

--name VALUE Name for this node type
--shortName VALUE Short Name
--version VALUE Version
--technology CODE Technology. This is the provision type code.
--scripts x,y,z List of Script IDs
--file-templates x,y,z List of File Template IDs
--option OPTION Option in the format -O field="value"
--prompt Always prompts. Use passed options as the default value.
--no-prompt Skip prompts. Use default values for all optional fields.
--payload FILE Payload from a local JSON or YAML file, skip all prompting
Create a node type.
[name] is required and can be passed as --name instead.

17.27.2 morpheus library-node-types get

Usage: morpheus library-node-types get [name]
   -j, --json          JSON Output
   -y, --yaml         YAML Output
   -c, --csv          CSV Output
   -d, --csv-delim CHAR Delimiter for CSV Output values. Default: ','
   -n, --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
   -q, --csv-quotes   Wrap CSV values with "". Default: false
   -h, --csv-no-header Exclude header for CSV Output.
   -f, --fields x,y,z Filter CSV Output to a limited set of fields.
   -d, --dry-run      Dry Run, print the API request instead of executing it.
   --curl             Dry Run to output API request as a curl command.
   --scrub            Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE Remote name. The current remote is used by default.
   -u, --remote-url URL Remote url. The current remote url is used by default.
   -t, --token TOKEN  Access token for authentication with --remote.
   -u, --username USERNAME Username for authentication.
   -p, --password PASSWORD Password for authentication.
   -i, --insecure     Allow insecure HTTPS communication. i.e. bad SSL certificate.
   -h, --header HEADER Additional HTTP header to include with requests.
   -t, --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
   -C, -- nocolor     Disable ANSI coloring
   -B, --benchmark    Print benchmark time after the command is finished.
   -v, --debug        Print extra output for debugging.
   -h, --help         Print this help

Saved credentials are used by default.
-u, --username USERNAME Username for authentication.
-p, --password PASSWORD Password for authentication.
-i, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
-30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Display node type details. [name] is required. This is the name or id of a node type.

17.27.3 morpheus library-node-types list

Usage: morpheus library-node-types list
   --layout ID Filter by Layout
   --technology VALUE Filter by technology
   --category VALUE Filter by category
   -m, --max MAX Max Results
   -o, --offset OFFSET Offset Results
   -s, --search PHRASE Search Phrase
   -S, --sort ORDER Sort Order
   -D, --desc Reverse Sort Order
   -Q, --query PARAMS Query parameters. PARAMS format is 'phrase=foobar&category=web'
   -j, --json JSON Output
   --yaml YAML Output
   --csv CSV Output
   --csv-delim CHAR Delimiter for CSV Output values. Default: ','
   --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
   --csv-quotes Wrap CSV values with '. Default: false
   --csv-no-header Exclude header for CSV Output.
   -F, --fields x,y,z Filter Output to a limited set of fields. Default is all fields.
   -d, --dry-run Dry Run, print the API request instead of executing it
   --curly Dry Run to output API request as a curl command.
   --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE Remote name. The current remote is used by default.
   --remote-url URL Remote url. The current remote url is used by default.
   -T, --token TOKEN Access token for authentication with --remote.
   Saved credentials are used by default.
   -u, --username USERNAME Username for authentication.
   -p, --password PASSWORD Password for authentication.
   -i, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
   -H, --header HEADER Additional HTTP header to include with requests.
   -30 seconds.
List node types.

17.27.4 morpheus library-node-types remove

Usage: morpheus library-node-types remove [name]

- y, --yes Auto Confirm
- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of executing it
  -c, --curl Dry Run to output API request as a curl command.
  -s, --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by default.
  -u, --remote-url URL Remote url. The current remote url is used by default.
- d, --default.
- t, --token TOKEN Access token for authentication with --remote.
  Saved credentials are used by default.
  - u, --username USERNAME Username for authentication.
  - p, --password PASSWORD Password for authentication.
  - i, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
  - h, --header HEADER Additional HTTP header to include with requests.
  - t, --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- c, --nocolor Disable ANSI coloring
- b, --benchmark Print benchmark time after the command is finished.
- v, --debug Print extra output for debugging.
- h, --help Print this help
Delete a node type.

17.27.5 morpheus library-node-types update

Usage: morpheus library-node-types update [name] [options]

  - n, --name VALUE Name for this layout
  - s, --shortName VALUE Short Name
  - v, --version VALUE Version
  - x, --scripts x,y,z List of Script IDs
  - f, --file-templates x,y,z List of File Template IDs
  - o, --option OPTION Option in the format -O field="value"
  - p, --prompt Always prompts. Use passed options as the default value.
  - n, --no-prompt Skip prompts. Use default values for all optional fields.
- j, --json JSON Output

(continues on next page)
-d, --dry-run               Dry Run, print the API request instead of executing it
--curl                      Dry Run to output API request as a curl command.
--scrub                     Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE        Remote name. The current remote is used by default.
--remote-url URL           Remote url. The current remote url is used by default.
-T, --token TOKEN          Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME    Username for authentication.
-P, --password PASSWORD    Password for authentication.
-I, --insecure             Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER        Additional HTTP header to include with requests.
--timeout SECONDS         Timeout for api requests. Default is typically 30 seconds.
-30 seconds.
-c, --nocolor              Disable ANSI coloring
-B, --benchmark            Print benchmark time after the command is finished.
-V, --debug                Print extra output for debugging.
-h, --help                 Print this help

Update a node type.

17.28 morpheus library-option-lists

Usage: morpheus library-option-lists [command] [options]
Commands:
    add
    get
    list
    remove
    update

17.28.1 morpheus library-option-lists add

Usage: morpheus library-option-lists add [type] [options]
    -t, --type TYPE              Option List Type. (rest, manual)
    --name VALUE                 Name
    --description VALUE          Description (optional)
    --sourceUrl VALUE           Source Url - A REST URL can be used to fetch list data and is cached in the appliance database.
    --ignoreSSLerrors on|off     Ignore SSL Errors (optional) Default: off
    --realTime on|off            Real Time (optional) Default: off
    --sourceMethod VALUE         Source Method Default: GET
    --initialDataset VALUE       Initial Dataset (optional) - Create an initial json dataset to be used as the collection for this option list. It should be a list containing objects with properties 'name', and 'value', However, if there is a translation script, that will also be passed through.
    --translationScript VALUE    Translation Script (optional) - 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.
-O, --option OPTION Option in the format -O field="value"
-prompt
-default value.
-N, --no-prompt Skip prompts. Use default values for all
-optional fields.
--payload FILE Payload from a local JSON or YAML file, skip all
-prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N
-or YAML files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of
-executing it
--curl
--scrub
--Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by
-default.
--remote-url URL Remote url. The current remote url is used by
-default.
-T, --token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad
-SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically
-30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is
-finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.28.2 morpheus library-option-lists get

Usage: morpheus library-option-lists get [name]
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of
-executing it
--curl
--scrub
--Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by
-default.
--remote-url URL Remote url. The current remote url is used by
-default.
-T, --token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad
-SSL certificate.
### 17.28.3 morpheus library-option-lists list

**Usage:** morpheus library-option-lists list

- `-m`, `--max MAX` Max Results
- `-o`, `--offset OFFSET` Offset Results
- `-s`, `--search PHRASE` Search Phrase
- `-S`, `--sort ORDER` Sort Order
- `-D`, `--desc` Reverse Sort Order
- `-Q`, `--query PARAMS` Query parameters. PARAMS format is `phrase=foobar&category=web`
- `-d`, `--dry-run` Dry Run, print the API request instead of executing it
- `--curl` Dry Run to output API request as a curl command.
- `--scrub` Mask secrets in output, such as the Authorization header. For use with `--curl` and `--dry-run`.
- `-j`, `--json` JSON Output
- `-r`, `--remote REMOTE` Remote name. The current remote is used by default.
- `--remote-url URL` Remote url. The current remote url is used by default.
- `-T`, `--token TOKEN` Access token for authentication with `--remote`.
- `-U`, `--username USERNAME` Username for authentication.
- `-P`, `--password PASSWORD` Password for authentication.
- `-I`, `--insecure` Allow insecure HTTPS communication. i.e. bad SSL certificate.
- `-H`, `--header HEADER` Additional HTTP header to include with requests.
- `--timeout SECONDS` Timeout for api requests. Default is typically 30 seconds.
- `-C`, `--nocolor` Disable ANSI coloring
- `-B`, `--benchmark` Print benchmark time after the command is finished.
- `-V`, `--debug` Print extra output for debugging.
- `-h`, `--help` Print this help

This outputs a list of custom Option List records.

### 17.28.4 morpheus library-option-lists remove

**Usage:** morpheus library-option-lists remove [name]

- `-y`, `--yes` Auto Confirm

(continues on next page)
### 17.28.5 morpheus library-option-lists update

Usage: morpheus library-option-lists update [name] [options]

- **--name VALUE** Name (optional)
- **--description VALUE** Description (optional)
- **--sourceUrl VALUE** Source Url (optional) - A REST URL can be used to fetch list data and is cached in the appliance database.
- **--ignoreSSLErrors on|off** Ignore SSL Errors (optional)
- **--realTime on|off** Real Time (optional)
- **--sourceMethod VALUE** Source Method (optional)
- **--initialDataset VALUE** Initial Dataset (optional) - Create an initial json dataset to be used as the collection for this option list. It should be a list containing objects with properties 'name', and 'value'. However, if there is a translation script, that will also be passed through.
- **--translationScript VALUE** Translation Script (optional) - 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.
- **-O, --option OPTION** Option in the format -O field="value" -default value.
- **-N, --no-prompt** Skip prompts. Use default values for all optional fields.
- **--payload FILE** Payload from a local JSON or YAML file, skip all prompting.
- **--payload-dir DIRECTORY** Payload from a local directory containing 1-N JSON or YAML files, skip all prompting.
- **-j, --json** JSON Output
- **-d, --dry-run** Dry Run, print the API request instead of executing it.
- **--curl** Dry Run to output API request as a curl command.
- **--scrub** Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- **-r, --remote REMOTE** Remote name. The current remote is used by default.
- **--remote-url URL** Remote url. The current remote url is used by default.
- **-T, --token TOKEN** Access token for authentication with --remote.
- **--benchmark** Print benchmark time after the command is finished.
- **-V, --debug** Print extra output for debugging.
- **-h, --help** Print this help

(continues on next page)
-d, --dry-run  Dry Run, print the API request instead of executing it
  --curl  Dry Run to output API request as a curl command.
  --scrub  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE  Remote name. The current remote is used by default.
  --remote-url URL  Remote url. The current remote url is used by default.
  --default.
  --default.
-T, --token TOKEN  Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME  Username for authentication.
-P, --password PASSWORD  Password for authentication.
-I, --insecure  Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER  Additional HTTP header to include with requests.
-timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor  Disable ANSI coloring
-B, --benchmark  Print benchmark time after the command is finished.
-V, --debug  Print extra output for debugging.
-h, --help  Print this help

17.29 morpheus library-option-types

Usage: morpheus library-option-types [command] [options]
Commands:
  add
  get
  list
  remove
  update

17.29.1 morpheus library-option-types add

Usage: morpheus library-option-types add [options]
  --name VALUE  Name
  --description VALUE  Description (optional)
  --fieldName VALUE  Field Name - This is the input fieldName property that the value gets assigned to.
  --type VALUE  Type Default: text
  --fieldLabel VALUE  Field Label - This is the input label that shows typically to the left of a custom option.
  --placeholder VALUE  Placeholder (optional)
  --defaultValue VALUE  Default Value (optional)
  --required on|off  Required (optional) Default: off
  -O, --option OPTION  Option in the format -O field="value"
  --prompt  Always prompts. Use passed options as the default value.
  -N, --no-prompt  Skip prompts. Use default values for all optional fields.
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote. Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.29.2 morpheus library-option-types get

Usage: morpheus library-option-types get [name]
-j, --json JSON Output
--yaml YAML Output
--csv CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '
'
--csv-quotes Wrap CSV values with '. Default: false
--csv-no-header Exclude header for CSV Output.
-F, --fields x,y,z Filter Output to a limited set of fields. Default is all fields.
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote. Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
17.29.3 morpheus library-option-types list

Usage: morpheus library-option-types list
  -m, --max MAX  Max Results
  -o, --offset OFFSET Offset Results
  -s, --search PHRASE Search Phrase
  -S, --sort ORDER Sort Order
  -d, --desc Reverse Sort Order
  -q, --query PARAMS Query parameters. PARAMS format is
    'phrase=foobar&category=web'
  -d, --dry-run Dry Run, print the API request instead of _executing it_
    --curl Dry Run to output API request as a curl command.
    --scrub Mask secrets in output, such as the
    Authorization header. For use with --curl and --dry-run.
  -j, --json JSON Output
  -r, --remote REMOTE Remote name. The current remote is used by
  --remote-url URL Remote url. The current remote url is used by
--default.
  -t, --token TOKEN Access token for authentication with --remote.
--Saved credentials are used by default.
  -u, --username USERNAME Username for authentication.
  -p, --password PASSWORD Password for authentication.
  -i, --insecure Allow insecure HTTPS communication. i.e. bad
--SSL certificate.
  -H, --header HEADER Additional HTTP header to include with requests.
    --timeout SECONDS Timeout for api requests. Default is typically
    30 seconds.
  -C, --nocolor Disable ANSI coloring
  -B, --benchmark Print benchmark time after the command is
  --finished.
  -V, --debug Print extra output for debugging.
  -h, --help Print this help

List option types.

17.29.4 morpheus library-option-types remove

Usage: morpheus library-option-types remove [name]
  -y, --yes Auto Confirm
  -j, --json JSON Output
  -d, --dry-run Dry Run, print the API request instead of _executing it_
    --curl Dry Run to output API request as a curl command.
    --scrub Mask secrets in output, such as the
    Authorization header. For use with --curl and --dry-run.
### 17.29.5 morpheus library-option-types update

Usage: morpheus library-option-types update [name] [options]

- **--name VALUE** Name (optional)
- **--description VALUE** Description (optional)
- **--fieldName VALUE** Field Name (optional) - This is the inputfieldName property that the value gets assigned to.
- **--type VALUE** Type (optional)
- **--fieldLabel VALUE** Field Label (optional) - This is the input label that shows typically to the left of a custom option.
- **--defaultValue VALUE** Default Value (optional)
- **--required on|off** Required (optional)
- **--prompt OPTION** Option in the format -O field="value"
- **--default value.**
- **--no-prompt** Skip prompts. Use default values for all optional fields.
- **-j, --json** JSON Output
- **-d, --dry-run** Dry Run, print the API request instead of executing it
- **--curl** Dry Run to output API request as a curl command.
- **--scrub** Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- **-r, --remote REMOTE** Remote name. The current remote is used by default.
- **--remote-url URL** Remote url. The current remote url is used by default.
- **-T, --token TOKEN** Access token for authentication with --remote.
- **-U, --username USERNAME** Username for authentication.
- **-P, --password PASSWORD** Password for authentication.
- **-I, --insecure** Allow insecure HTTPS communication. i.e. bad SSL certificate.
- **-H, --header HEADER** Additional HTTP header to include with requests.
- **--timeout SECONDS** Timeout for api requests. Default is typically 30 seconds.
17.30 morpheus library-scripts

Usage: morpheus library-scripts [command] [options]
Commands:
  add
  get
  list
  remove
  update

17.30.1 morpheus library-scripts add

Usage: morpheus library-scripts add [name]
  --name VALUE Name
  --type [bash|powershell] Script Type. Default is 'bash'
  --phase [provision|start|stop] Script Phase. Default is 'provision'
  --category VALUE Category
  --script TEXT Contents of the script.
  --file FILE File containing the script. This can be used instead of --script
  --option OPTION Option in the format -O field="value"
  --prompt Always prompts. Use passed options as the default value.
  --no-prompt Skip prompts. Use default values for all optional fields.
  --payload FILE Payload from a local JSON or YAML file, skip all prompting
  --payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
  --payload-json JSON Payload JSON, skip all prompting
  --payload-yaml YAML Payload YAML, skip all prompting
  -O, --option OPTION Option in the format -O field="value"
  --prompt Always prompts. Use passed options as the default value.
  --no-prompt Skip prompts. Use default values for all optional fields.
  --payload FILE Payload from a local JSON or YAML file, skip all prompting
  --payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
  --payload-json JSON Payload JSON, skip all prompting
  --payload-yaml YAML Payload YAML, skip all prompting
  -j, --json JSON Output
  -d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -r, --remote REMOTE Remote name. The current remote is used by default.
  --remote-url URL Remote url. The current remote url is used by default.
  -T, --token TOKEN Access token for authentication with --remote.
  -U, --username USERNAME Username for authentication.
  -P, --password PASSWORD Password for authentication.
17.30.2 morpheus library-scripts get

Usage: morpheus library-scripts get [name]

Usage: morpheus library-scripts get [name]
- j, --json        JSON Output
- yml             YAML Output
- csv             CSV Output
- csv-delim CHAR  Delimiter for CSV Output values. Default: ','
- csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
- csv-quotes      Wrap CSV values with ''. Default: false
- csv-no-header   Exclude header for CSV Output.
- F, --fields x,y,z Filter Output to a limited set of fields. Default is all fields.
- d, --dry-run    Dry Run, print the API request instead of executing it.
- curl            Dry Run to output API request as a curl command.
- scrub           Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by default.
- remote-url URL  Remote url. The current remote url is used by default.
- T, --token TOKEN Access token for authentication with --remote.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure   Allow insecure HTTPS communication. i.e. bad SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
- timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- C, --nocolor    Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug      Print extra output for debugging.
- h, --help       Print this help
17.30.3 morpheus library-scripts list

Usage: morpheus library-scripts list

- m, --max MAX Max Results
- o, --offset OFFSET Offset Results
- s, --search PHRASE Search Phrase
- S, --sort ORDER Sort Order
- D, --desc Reverse Sort Order
- Q, --query PARAMS Query parameters. PARAMS format is
  'phrase=foobar&category=web'
- j, --json JSON Output
- yml YAML Output
- csv CSV Output
  --csv-delim CHAR Delimiter for CSV Output values. Default: ','
  --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
  --csv-quotes Wrap CSV values with ''. Default: false
  --csv-no-header Exclude header for CSV Output.
- F, --fields x,y,z Filter Output to a limited set of fields.
  Default is all fields.
- d, --dry-run Dry Run, print the API request instead of
  executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the
  Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by
  default.
  --remote-url URL Remote url. The current remote url is used by
  default.
  --token TOKEN Access token for authentication with --remote.
  Saved credentials are used by default.
  --username USERNAME Username for authentication.
  --password PASSWORD Password for authentication.
  --insecure Allow insecure HTTPS communication. i.e. bad
  SSL certificate.
  --header HEADER Additional HTTP header to include with requests.
  --timeout SECONDS Timeout for api requests. Default is typically
  30 seconds.
- c, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is
  finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

List container scripts.

17.30.4 morpheus library-scripts remove

Usage: morpheus library-scripts remove [name]

- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of
  executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the
  Authorization header. For use with --curl and --dry-run.
  --quiet No Output, do not print to stdout
-y, --yes Auto Confirm
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.30.5 morpheus library-scripts update

Usage: morpheus library-scripts update [name]
--name VALUE Name
--type {bash|powershell} Script Type
--phase {start|stop} Script Phase
--category VALUE Category
--script TEXT Contents of the script. This can be used instead of --file FILE
--file FILE File containing the script. This can be used instead of --script

Option in the format -O field="value"
--prompt Always prompts. Use passed options as the default value.
--no-prompt Skip prompts. Use default values for all optional fields.
--payload FILE Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting

Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
--remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
--token TOKEN Access token for authentication with --remote.

Saved credentials are used by default.
--username USERNAME Username for authentication.
--password PASSWORD Password for authentication.
--insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
--header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
--quiet No Output, do not print to stdout
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Update a container script.
[name] is required. This is the name or id of a container script.

## 17.31 morphes library-upgrades

**Usage:** morpheus library-upgrades [command] [options]

**Commands:**
- add
- get
- list
- remove
- update

### 17.31.1 morphes library-upgrades add

**Usage:** morpheus library-upgrades add [instance-type] [name]

- **--instance-type ID** Instance Type this upgrade belongs to
- **--name VALUE** Name for this upgrade
- **--code CODE** Code
- **--description VALUE** Description
- **--source-layout ID** Source Layout ID to upgrade from
- **--target-layout ID** Target Layout ID to upgrade to
- **--upgradeCommand TEXT** Upgrade Command
- **-O, --option OPTION** Option in the format -O field="value"
- **--prompt** Always prompts. Use passed options as the default value.
- **-N, --no-prompt** Skip prompts. Use default values for all optional fields.
- **--payload FILE** Payload from a local JSON or YAML file, skip all prompting
- **--payload-dir DIRECTORY** Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
- **--payload-json JSON** Payload JSON, skip all prompting
- **--payload-yaml YAML** Payload YAML, skip all prompting
- **-j, --json JSON** Output
- **-d, --dry-run Dry Run** Dry Run, print the API request instead of executing it
- **--curl Dry Run to output API request as a curl command.** Dry Run to output API request as a curl command. Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- **-r, --remote REMOTE** Remote name. The current remote is used by default.
- **--remote-url URL** Remote url. The current remote url is used by default.
- **-T, --token TOKEN** Access token for authentication with --remote.
- **Saved credentials are used by default.**
- **-U, --username USERNAME** Username for authentication.
- **-P, --password PASSWORD** Password for authentication.
- **--insecure** Allow insecure HTTPS communication. i.e. bad SSL certificate.
- **-H, --header HEADER** Additional HTTP header to include with requests.
- **--timeout SECONDS** Timeout for api requests. Default is typically 30 seconds.
Create a new upgrade.

[instance-type] is required.

17.31.2 morpheus library-upgrades get

Usage: morpheus library-upgrades get [name]

- j, --json JSON Output
- y, --yaml YAML Output
- c, --csv CSV Output
- csv-delim CHAR Delimiter for CSV Output values. Default: ','
- csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
- csv-quotes Wrap CSV values with ''. Default: false
- csv-no-header Exclude header for CSV Output.
- d, --fields x,y,z Filter Output to a limited set of fields.
- f, --fields-delim CHAR Delimiter for CSV Output values. Default: ','
- f, --fields-newline [CHAR] Delimiter for CSV Output rows. Default: '
'
- f, --fields-quotes Wrap CSV values with ''. Default: false
- f, --fields-no-header Exclude header for CSV Output.
- d, --dry-run Dry Run, print the API request instead of executing it
- c, --curl Dry Run to output API request as a curl command.
- s, --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by default.
- u, --remote-url URL Remote url. The current remote url is used by default.
- t, --token TOKEN Access token for authentication with --remote.
- s, --ssl certificate Additional insecure HTTPS communication. i.e. bad
- t, --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- c, --nocolor Disable ANSI coloring
- b, --benchmark Print benchmark time after the command is finished.
- v, --debug Print extra output for debugging.
- h, --help Print this help

Get details about an upgrade.

[instance-type] is required.

[instance-type] is required. This is the name or id of an upgrade.

17.31.3 morpheus library-upgrades list

Usage: morpheus library-upgrades list [instance-type]

- code VALUE Filter by code

(continues on next page)
List upgrades. [instance-type] is required.

### 17.31.4 morpheus library-upgrades remove

Usage: morpheus library-upgrades remove [name]

- `-y`, `--yes` Auto Confirm
- `-j`, `--json` JSON Output
- `-d`, `--dry-run` Dry Run, print the API request instead of executing it
- `-c`, `--curl` Dry Run to output API request as a curl command.
- `-s`, `--scrub` Mask secrets in output, such as the Authorization header.
- `-r`, `--remote REMOTE` Remote name. The current remote is used by default.
- `-u`, `--username USERNAME` Username for authentication.
- `-p`, `--password PASSWORD` Password for authentication.
- `-I`, `--insecure` Allow insecure HTTPS communication. i.e. bad SSL certificate.
- `-H`, `--header HEADER` Additional HTTP header to include with requests.
- `-t`, `--timeout SECONDS` Timeout for api requests. Default is typically 30 seconds.
- `-C`, `--nocolor` Disable ANSI coloring
- `-b`, `--benchmark` Print benchmark time after the command is finished.
- `-V`, `--debug` Print extra output for debugging.
- `-h`, `--help` Print this help
Delete a upgrade.

### 17.31.5 morpheus library-upgrades update

**Usage:** morpheus library-upgrades update [name] [options]

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--name VALUE</td>
<td>Name for this upgrade</td>
</tr>
<tr>
<td>--version VALUE</td>
<td>Version</td>
</tr>
<tr>
<td>--description VALUE</td>
<td>Description</td>
</tr>
<tr>
<td>--min-memory VALUE</td>
<td>Minimum Memory (MB)</td>
</tr>
<tr>
<td>--workflow ID</td>
<td>Workflow</td>
</tr>
<tr>
<td>--option-types x,y,z</td>
<td>List of Option Type IDs</td>
</tr>
<tr>
<td>--node-types x,y,z</td>
<td>List of Node Type IDs</td>
</tr>
<tr>
<td>-O, --option OPTION</td>
<td>Option in the format -O field=&quot;value&quot;</td>
</tr>
<tr>
<td>-N, --no-prompt</td>
<td>Skip prompts. Use default values for all optional fields.</td>
</tr>
<tr>
<td>-j, --json</td>
<td>JSON Output</td>
</tr>
<tr>
<td>-d, --dry-run</td>
<td>Dry Run, print the API request instead of executing it</td>
</tr>
<tr>
<td>--executing it</td>
<td>Dry Run to output API request as a curl command.</td>
</tr>
<tr>
<td>--curl</td>
<td>Mask secrets in output, such as the Authorization header. For use with -N or --no-prompt</td>
</tr>
<tr>
<td>--scrub</td>
<td>Remote name. The current remote is used by default.</td>
</tr>
<tr>
<td>-r, --remote REMOTE</td>
<td>Remote name. The current remote is used by default.</td>
</tr>
<tr>
<td>--remote-url URL</td>
<td>Remote url. The current remote url is used by default.</td>
</tr>
<tr>
<td>-T, --token TOKEN</td>
<td>Access token for authentication with --remote.</td>
</tr>
<tr>
<td>-U, --username USERNAME</td>
<td>Username for authentication.</td>
</tr>
<tr>
<td>-P, --password PASSWORD</td>
<td>Password for authentication.</td>
</tr>
<tr>
<td>-I, --insecure</td>
<td>Allow insecure HTTPS communication. i.e. bad SSL certificate.</td>
</tr>
<tr>
<td>-H, --header HEADER</td>
<td>Additional HTTP header to include with requests.</td>
</tr>
<tr>
<td>--timeout SECONDS</td>
<td>Timeout for api requests. Default is typically 30 seconds.</td>
</tr>
<tr>
<td>-C, --nocolor</td>
<td>Disable ANSI coloring</td>
</tr>
<tr>
<td>-B, --benchmark</td>
<td>Print benchmark time after the command is finished.</td>
</tr>
<tr>
<td>-v, --debug</td>
<td>Print extra output for debugging.</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Print this help</td>
</tr>
</tbody>
</table>
Morpheus Documentation, Release 3.6.2

(continued from previous page)

-V, --debug  Print extra output for debugging.
-h, --help    Print this help

Update a upgrade.

17.32  morpheus license

Usage: morpheus license [command] [options]
Commands:
   apply
   decode
   get

17.32.1  morpheus license apply

Usage: morpheus license apply [key]
   -j, --json  JSON Output
   -d, --dry-run  Dry Run, print the API request instead of execute it
   --curl
   --scrub    Dry Run to output API request as a curl command.
   --Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE  Remote name. The current remote is used by default.
   --remote-url URL  Remote url. The current remote url is used by default.
   -T, --token TOKEN  Access token for authentication with --remote.
   -U, --username USERNAME  Username for authentication.
   -P, --password PASSWORD  Password for authentication.
   -I, --insecure  Allow insecure HTTPS communication. i.e. bad SSL certificate.
   --header HEADER  Additional HTTP header to include with requests.
   --timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
   -C, --nocolor  Disable ANSI coloring
   -B, --benchmark  Print benchmark time after the command is finished.
   -V, --debug  Print extra output for debugging.
   -h, --help    Print this help

17.32.2  morpheus license decode

Usage: morpheus license decode [key]
   -j, --json  JSON Output
   -d, --dry-run  Dry Run, print the API request instead of execute it
   --curl
   --scrub    Dry Run to output API request as a curl command.
   --Authorization header. For use with --curl and --dry-run.
Morpheus Documentation, Release 3.6.2

(continued from previous page)

```
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote. Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help
```

Decode a license key.

### 17.32.3 morpheus license get

Usage: morpheus license get
```
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote. Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help
```
17.33 morpheus load-balancers

Usage: morpheus load-balancers [command] [options]
Commands:
  add
  get
  list
  remove
  types
  update

17.33.1 morpheus load-balancers add

Usage: morpheus load-balancers add [name] -t LB_TYPE
-t, --type LB_TYPE Load Balancer Type
-O, --option OPTION Option in the format -O field="value"
--prompt Always prompts. Use passed options as the default value.
-N, --no-prompt Skip prompts. Use default values for all optional fields.
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command. Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote. Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.33.2 morpheus load-balancers get

Usage: morpheus load-balancers get [name]
-j, --json JSON Output
-csv CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
--csv-quotes Wrap CSV values with '. Default: false

(continues on next page)
--csv-no-header Exclude header for CSV Output.
--yaml YAML Output
-F, --fields x,y,z Filter Output to a limited set of fields.
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-3, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.33.3 morpheus load-balancers list

Usage: morpheus load-balancers list
     -m, --max MAX Max Results
     -o, --offset OFFSET Offset Results
     -s, --search PHRASE Search Phrase
     -S, --sort ORDER Sort Order
     -D, --desc Reverse Sort Order
     -j, --json JSON Output
     --csv CSV Output
     --csv-delim CHAR Delimiter for CSV Output values. Default: ','
     --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
     --csv-quotes Wrap CSV values with '. Default: false
     --csv-no-header Exclude header for CSV Output.
     --yaml YAML Output
     -F, --fields x,y,z Filter Output to a limited set of fields.
     --default Input fields.
     --remote-url URL Remote url. The current remote url is used by default.
     --remote REMOTE Remote name. The current remote is used by default.
     --token TOKEN Access token for authentication with --remote.
     --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
     --header HEADER Additional HTTP header to include with requests.
     --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
     -3, --nocolor Disable ANSI coloring
     -B, --benchmark Print benchmark time after the command is finished.
     -V, --debug Print extra output for debugging.
     -h, --help Print this help
## 17.33.4 morpheus load-balancers remove

Usage: morpheus load-balancers remove [name]

- `y`, `--yes`: Auto Confirm
- `j`, `--json`: JSON Output
- `d`, `--dry-run`: Dry Run, print the API request instead of executing it
  - `--curl`: Dry Run to output API request as a curl command.
  - `--scrub`: Mask secrets in output, such as the Authorization header. For use with `--curl` and `--dry-run`.
- `r`, `--remote REMOTE`: Remote name. The current remote is used by default.
  - `--remote-url URL`: Remote url. The current remote url is used by default.
  - `--token TOKEN`: Access token for authentication with `--remote`.
- Saved credentials are used by default.
- `U`, `--username USERNAME`: Username for authentication.
- `P`, `--password PASSWORD`: Password for authentication.
- `I`, `--insecure`: Allow insecure HTTPS communication. i.e. bad SSL certificate.
- `H`, `--header HEADER`: Additional HTTP header to include with requests.
  - `--timeout SECONDS`: Timeout for api requests. Default is typically 30 seconds.
- `C`, `--nocolor`: Disable ANSI coloring
- `B`, `--benchmark`: Print benchmark time after the command is finished.
- `V`, `--debug`: Print extra output for debugging.
- `-h`, `--help`: Print this help

## 17.33.5 morpheus load-balancers types

Usage: morpheus load-balancers types

- `j`, `--json`: JSON Output
- `d`, `--dry-run`: Dry Run, print the API request instead of executing it
  - `--curl`: Dry Run to output API request as a curl command.
  - `--scrub`: Mask secrets in output, such as the Authorization header. For use with `--curl` and `--dry-run`.
- `r`, `--remote REMOTE`: Remote name. The current remote is used by default.
  - `--remote-url URL`: Remote url. The current remote url is used by default.
- Saved credentials are used by default.
- `U`, `--username USERNAME`: Username for authentication.
- `P`, `--password PASSWORD`: Password for authentication.
- `I`, `--insecure`: Allow insecure HTTPS communication. i.e. bad SSL certificate.
- `H`, `--header HEADER`: Additional HTTP header to include with requests.
  - `--timeout SECONDS`: Timeout for api requests. Default is typically 30 seconds.
- `C`, `--nocolor`: Disable ANSI coloring
- `B`, `--benchmark`: Print benchmark time after the command is finished.
- `V`, `--debug`: Print extra output for debugging.
- `-h`, `--help`: Print this help

(continues on next page)
17.33.6 morpheus load-balancers update

Usage: morpheus load-balancers update [name] [options]

-0, --option OPTION  Option in the format -O field="value"
--prompt  Always prompts. Use passed options as the default value.
-N, --no-prompt  Skip prompts. Use default values for all optional fields.
-j, --json  JSON Output
-d, --dry-run  Dry Run, print the API request instead of executing it
--curl  Dry Run to output API request as a curl command.
--scrub  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE  Remote name. The current remote is used by default.
--remote-url URL  Remote url. The current remote url is used by default.
--token TOKEN  Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME  Username for authentication.
-P, --password PASSWORD  Password for authentication.
-I, --insecure  Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER  Additional HTTP header to include with requests.
--timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor  Disable ANSI coloring
-B, --benchmark  Print benchmark time after the command is finished.
-V, --debug  Print extra output for debugging.
-h, --help  Print this help
17.34 morpheus login

Usage: morpheus login [username] [password]
-ua, --username USERNAME Username
-tp, --password PASSWORD Password
-t, --test Test credentials only, does not update stored credentials for the appliance.
-It, --token ACCESS_TOKEN Use an existing access token to login instead of authenticating with a username and password.
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-q, --quiet No Output, do not print to stdout
-C, --nocolor Disable ANSI coloring
--benchmark Print benchmark time after the command is finished.
-v, --debug Print extra output for debugging.
-h, --help Print this help

Login to a remote appliance with a username and password or an access token. [username] is required. [password] is required. Logging in with username and password will make an authentication api request to obtain an access token. The --token option can be used to login with an existing token instead of username and password. Using --token makes a whoami api request to validate the token. If successful, the access token will be saved with the active session for the remote appliance. This command will first logout any active session before attempting to login. The --test option can be used for testing credentials without updating your active session.

17.35 morpheus logout

Usage: morpheus logout
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-It, --token TOKEN Access token for authentication with --remote. Saved credentials are used by default.

(continues on next page)
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-q, --quiet No Output, do not print to stdout
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.36 morpheus monitor-apps

Usage: morpheus monitor-apps [command] [options]
Commands:
  add
  get
  list
  mute
  mute-all
  remove
  unmute
  unmute-all
  update

17.36.1 morpheus monitor-apps add

Usage: morpheus monitor-apps add [name]
  --name VALUE Name
  --description VALUE Description
  --minHappy VALUE Min Checks. This specifies the minimum number of
  --severity VALUE Max Severity. Determines the maximum severity level this app can incur on an incident when failing. Default is critical
  --inUptime [on|off] Affects Availability. Default is on.
  --checks LIST Checks to include in this app, comma separated
  --checkGroups LIST Check Groups to include in this app, comma separated
  --list of IDs
  --checkGroups LIST Check Groups to include in this app, comma separated
  --separated list of IDs
  -O, --option OPTION Option in the format -O field="value"
  --prompt Always prompts. Use passed options as the default value.
  -N, --no-prompt Skip prompts. Use default values for all optional fields.
  --payload FILE Payload from a local JSON or YAML file, skip all prompting
  --payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
  --payload-json JSON Payload JSON, skip all prompting
Create a new app of monitoring checks.

[Name] is required and can be passed as --name instead.

Usage:
```
morpheus monitor-apps get [id list]
```

Options:
- `-h`, `--help` Print this help
- `-V`, `--debug` Print extra output for debugging.
- `-C`, `--nocolor` Disable ANSI coloring.
- `-B`, `--benchmark` Print benchmark time after the command is finished.
- `-q`, `--quiet` No Output, do not print to stdout
- `-I`, `--insecure` Allow insecure HTTPS communication. i.e. bad SSL certificate.
- `-H`, `--header HEADER` Additional HTTP header to include with requests.
- `-T`, `--token TOKEN` Access token for authentication with --remote.
- `-r`, `--remote REMOTE` Remote name. The current remote is used by default.
- `--remote-url URL` Remote url. The current remote url is used by default.
- `-T`, `--token TOKEN` Access token for authentication with --remote.
- `-U`, `--username USERNAME` Username for authentication.
- `-P`, `--password PASSWORD` Password for authentication.
- `-T`, `--token TOKEN` Access token for authentication with --remote.
- `-F`, `--fields x,y,z` Filter Output to a limited set of fields. Default is all fields.
- `-d`, `--dry-run` Dry Run, print the API request instead of executing it.
- `--curl` Dry Run to output API request as a curl command.
- `--scrub` Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- `-r`, `--remote REMOTE` Remote name. The current remote is used by default.
- `--remote-url URL` Remote url. The current remote url is used by default.
- `-T`, `--token TOKEN` Access token for authentication with --remote.
- `--payload-yaml YAML`, `--json JSON` Payload YAML, skip all prompting
- `--dry-run` Dry Run, print the API request instead of executing it
- `--curl` Dry Run to output API request as a curl command.
- `--scrub` Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- `--default` Remote name. The current remote is used by default.
- `--remote-url URL` Remote url. The current remote url is used by default.
- `--default` Remote name. The current remote is used by default.
- `--raw` No Output, do not print to stdout
- `--csv-delim CHAR` Delimiter for CSV Output values. Default: ','
- `--csv-newline [CHAR]` Delimiter for CSV Output rows. Default: '\n'
- `--csv-quotes Wrap CSV values with "". Default: false
- `--csv-no-header` Exclude header for CSV Output.
- `--csv-delim CHAR` Delimiter for CSV Output values. Default: ','
- `--csv-newline [CHAR]` Delimiter for CSV Output rows. Default: '\n'
- `--csv-quotes Wrap CSV values with "". Default: false
- `--csv-no-header` Exclude header for CSV Output.
- `--csv-delim CHAR` Delimiter for CSV Output values. Default: ','
- `--csv-newline [CHAR]` Delimiter for CSV Output rows. Default: '\n'
- `--csv-quotes Wrap CSV values with "". Default: false
- `--csv-no-header` Exclude header for CSV Output.
17.36.3 morpheus monitor-apps list

Usage: morpheus monitor-apps list

   --status LIST Filter by status. error,healthy,warning,muted
   -m, --max MAX Max Results
   -o, --offset OFFSET Offset Results
   --search PHRASE Search Phrase
   --sort ORDER Sort Order
   --desc Reverse Sort Order
   --query PARAMS Query parameters. PARAMS format is
     'phrase=foobar&category=web'
   --last-updated TIME Filter by Last Updated (gte)
   -j, --json JSON Output
   --yaml YAML Output
   --csv CSV Output
   --csv-delim CHAR Delimiter for CSV Output values. Default: ','
   --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '
'
   --csv-quotes Wrap CSV values with "." Default: false
   --csv-no-header Exclude header for CSV Output.
   -F, --fields x,y,z Filter Output to a limited set of fields.
     Default is all fields.
   -d, --dry-run Dry Run, print the API request instead of executing it
   --curl Dry Run to output API request as a curl command.
   --scrub Mask secrets in output, such as the Authorization header.
   -r, --remote REMOTE Remote name. The current remote is used by default.
   --remote-url URL Remote url. The current remote url is used by default.
   -T, --token TOKEN Access token for authentication with --remote.
   Saved credentials are used by default.
   -U, --username USERNAME Username for authentication.
   -P, --password PASSWORD Password for authentication.
   -I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
   Additional HTTP header to include with requests.
   --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
  -C, --nocolor Disable ANSI coloring
  -B, --benchmark Print benchmark time after the command is finished.
  -V, --debug Print extra output for debugging.
17.36.4 `morpheus monitor-apps mute`

**Usage:** morpheus monitor-apps mute [name]

- `--disable` Unmute instead, the same as the unmute command
- `-O, --option OPTION` Option in the format -O field="value"
  
  - `--prompt` Always prompts. Use passed options as the default value.
- `-N, --no-prompt` Skip prompts. Use default values for all optional fields.
- `--payload FILE` Payload from a local JSON or YAML file, skip all prompting
- `--payload-dir DIRECTORY` Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
- `--payload-json JSON` Payload JSON, skip all prompting
- `--payload-yaml YAML` Payload YAML, skip all prompting
- `-j, --json` JSON Output
- `-d, --dry-run` Dry Run, print the API request instead of executing it
- `--curl` Dry Run to output API request as a curl command.
- `--scrub` Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- `-r, --remote REMOTE` Remote name. The current remote is used by default.
- `--remote-url URL` Remote url. The current remote url is used by default.
- `--token TOKEN` Access token for authentication with --remote.
- `--username USERNAME` Username for authentication.
- `--password PASSWORD` Password for authentication.
- `--insecure` Allow insecure HTTPS communication. i.e. bad SSL certificate.
- `--header HEADER` Additional HTTP header to include with requests.
- `--timeout SECONDS` Timeout for api requests. Default is typically 30 seconds.
- `-q, --quiet` No Output, do not print to stdout
- `--nocolor` Disable ANSI coloring
- `--benchmark` Print benchmark time after the command is finished.
- `-V, --debug` Print extra output for debugging.
- `-h, --help` Print this help

Mute a monitoring app. This prevents it from creating new incidents. [name] is required. This is the name or id of a monitoring app.

17.36.5 `morpheus monitor-apps mute-all`

**Usage:** morpheus monitor-apps mute-all

- `--disable` Unmute instead, the same as the unmute-all command
- `--command` Optionally enter a command to run after unmute
- `-O, --option OPTION` Option in the format -O field="value"

(continues on next page)
Morpheus Documentation, Release 3.6.2

(continued from previous page)

```plaintext
--prompt Always prompts. Use passed options as the default value.
-N, --no-prompt Skip prompts. Use default values for all optional fields.
--prompting Payload from a local JSON or YAML file, skip all prompting
--payload FILE Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
-curl Dry Run to output API request as a curl command. Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
--scrub Remote name. The current remote is used by default.
--remote REMOTE Remote url. The current remote url is used by default.
--remote-url URL Access token for authentication with --remote.
--token TOKEN Saved credentials are used by default. Username for authentication.
-p, --password PASSWORD Allow insecure HTTPS communication. i.e. bad SSL certificate.
-i, --insecure Additional HTTP header to include with requests. Timeout for api requests. Default is typically 30 seconds.
-T, --benchmark No Output, do not print to stdout
-C, --nocolor Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help
```

Mute all monitoring apps. This prevents the creation of new incidents.

**17.36.6 morpheus monitor-apps remove**

```plaintext
Usage: morpheus monitor-apps remove [name]
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
-curl Dry Run to output API request as a curl command. Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-q, --quiet No Output, do not print to stdout
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help
```
17.36.7 morpheus monitor-apps unmute

Usage: morpheus monitor-apps unmute [name]
    --payload FILE               Payload from a local JSON or YAML file, skip all prompting
    --payload-dir DIRECTORY      Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
    --payload-json JSON          Payload JSON, skip all prompting
    --payload-yaml YAML          Payload YAML, skip all prompting
    -j, --json                   JSON Output
    -d, --dry-run                Dry Run, print the API request instead of executing it
    --curl                       Dry Run to output API request as a curl command.
    --scrub                      Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
    -r, --remote REMOTE          Remote name. The current remote is used by default.
    --remote-url URL             Remote url. The current remote url is used by default.
    -t, --token TOKEN            Access token for authentication with --remote.
    -U, --username USERNAME      Username for authentication.
    -P, --password PASSWORD      Password for authentication.
    -I, --insecure               Allow insecure HTTPS communication. i.e. bad SSL certificate.
    -H, --header HEADER          Additional HTTP header to include with requests.
    --timeout SECONDS            Timeout for api requests. Default is typically 30 seconds.
    -q, --quiet                  No Output, do not print to stdout
    -C, --nocolor                Disable ANSI coloring
    -B, --benchmark              Print benchmark time after the command is finished.
    -V, --debug                  Print extra output for debugging.
    -h, --help                   Print this help

Unmute a monitoring app. [name] is required. This is the name or id of a monitoring app.

17.36.8 morpheus monitor-apps unmute-all

Usage: morpheus monitor-apps unmute-all
    --payload FILE               Payload from a local JSON or YAML file, skip all prompting
    --payload-dir DIRECTORY      Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
    --payload-json JSON          Payload JSON, skip all prompting
    --payload-yaml YAML          Payload YAML, skip all prompting
    -j, --json                   JSON Output
    -d, --dry-run                Dry Run, print the API request instead of executing it
    --curl                       Dry Run to output API request as a curl command.
    --scrub                      Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
    -r, --remote REMOTE          Remote name. The current remote is used by default.
### 17.36.9 morpheus monitor-apps update

**Usage:** morpheus monitor-apps update [name]

```
--name VALUE       Name
--description VALUE Description
--minHappy VALUE   Min Checks. This specifies the minimum number of
                   checks within the app that must be happy to keep the app from becoming unhealthy.
--severity VALUE   Max Severity. Determines the maximum severity level this app can incur on an incident when failing. Default is critical
--inUptime [on|off] Affects Availability. Default is on.
--checks LIST      Checks to include in this app, comma separated
--checkGroups LIST Check Groups to include in this app, comma separated
--option OPTION    Option in the format -O field="value"
--prompt           Always prompts. Use passed options as the default value.
--no-prompt         Skip prompts. Use default values for all optional fields.
--payload FILE     Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting
-j, --json          JSON Output
-d, --dry-run       Dry Run, print the API request instead of executing it
--authorization-header For use with --curl and --dry-run.
--remote REMOTE    Remote name. The current remote is used by default.
--remote-url URL   Remote url. The current remote url is used by default.
-h, --help         Print this help
```

Unmute all monitoring apps.
-T, --token TOKEN  Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME  Username for authentication.
-P, --password PASSWORD  Password for authentication.
-I, --insecure  Allow insecure HTTPS communication. i.e. bad
-SSL certificate.
-H, --header HEADER  Additional HTTP header to include with requests.
--timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
-g, --quiet  No Output, do not print to stdout
-C, --nocolor  Disable ANSI coloring
-B, --benchmark  Print benchmark time after the command is finished.
-V, --debug  Print extra output for debugging.
-h, --help  Print this help

Update a monitoring app.
[name] is required. This is the name or id of a monitoring app.

17.37  morpheus monitor-checks

Usage: morpheus monitor-checks [command] [options]
Commands:
  add
  get
  history
  list
  list-types
  mute
  mute-all
  remove
  unmute
  unmute-all
  update

17.37.1  morpheus monitor-checks add

Usage: morpheus monitor-checks add [name] -t CODE
  -t, --type CODE  Check Type Code or ID
  --name VALUE  Name
  --description VALUE  Description
  --checkInterval MILLIS  Check Interval. Value is in milliseconds.
  --severity VALUE  Max Severity. Determines the maximum severity
  level this check can incur on an incident when failing. Default is critical
  --inUptime [on/off]  Affects Availability. Default is on.
  -c, --config JSON  Config settings as JSON
  -O, --option OPTION  Option in the format -O field="value"
  -p, --prompt  Always prompts. Use passed options as the
  default value.
  -N, --no-prompt  Skip prompts. Use default values for all
  optional fields.
--payload FILE Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-q, --quiet No Output, do not print to stdout
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
--token TOKEN Access token for authentication with --remote. Saved credentials are used by default.
-u, --username USERNAME Username for authentication.
-p, --password PASSWORD Password for authentication.
-i, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

List monitoring checks.

17.37.2 morpheus monitor-checks get

Usage: morpheus monitor-checks get [id list]
--history Display Check History
-j, --json JSON Output
-csv CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
--csv-quotes Wrap CSV values with "". Default: false
--csv-no-header Exclude header for CSV Output.
-F, --fields x,y,z Filter Output to a limited set of fields. Default is all fields.
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.

(continues on next page)
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-T, --token TOKEN</td>
<td>Access token for authentication with --remote. Saved credentials are used by default.</td>
</tr>
<tr>
<td>-U, --username USERNAME</td>
<td>Username for authentication.</td>
</tr>
<tr>
<td>-P, --password PASSWORD</td>
<td>Password for authentication.</td>
</tr>
<tr>
<td>-I, --insecure</td>
<td>Allow insecure HTTPS communication. i.e. bad SSL certificate.</td>
</tr>
<tr>
<td>-H, --header HEADER</td>
<td>Additional HTTP header to include with requests.</td>
</tr>
<tr>
<td>--timeout SECONDS</td>
<td>Timeout for api requests. Default is typically 30 seconds.</td>
</tr>
<tr>
<td>-C, --nocolor</td>
<td>Disable ANSI coloring</td>
</tr>
<tr>
<td>-B, --benchmark</td>
<td>Print benchmark time after the command is finished.</td>
</tr>
<tr>
<td>-V, --debug</td>
<td>Print extra output for debugging.</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Print this help</td>
</tr>
</tbody>
</table>

### 17.37.3 morpheus monitor-checks history

**Usage:** `morpheus monitor-checks history [name] [options]`

- **--severity LIST** Filter by severity. critical, warning, info
- **-m, --max MAX** Max Results
- **-o, --offset OFFSET** Offset Results
- **-s, --search PHRASE** Search Phrase
- **-S, --sort ORDER** Sort Order
- **-D, --desc** Reverse Sort Order
- **--last-updated TIME** Filter by Last Updated (gte)
- **-yaml** YAML Output
- **--csv** CSV Output
- **--csv-delim CHAR** Delimiter for CSV Output values. Default: ','
- **--csv-newline [CHAR]** Delimiter for CSV Output rows. Default: '\n'
- **--csv-quotes [CHAR]** Wrap CSV values with ''. Default: false
- **--csv-no-header** Exclude header for CSV Output.
- **--fields x,y,z** Filter Output to a limited set of fields. Default is all fields.
- **-j, --json** JSON Output
- **-d, --dry-run** Dry Run, print the API request instead of executing it
- **--curl** Dry Run to output API request as a curl command.
- **--scrub** Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- **-r, --remote REMOTE** Remote name. The current remote is used by default.
- **--remote-url URL** Remote url. The current remote url is used by default.
- **-T, --token TOKEN** Access token for authentication with --remote. Saved credentials are used by default.
- **-U, --username USERNAME** Username for authentication.
- **-P, --password PASSWORD** Password for authentication.
- **-I, --insecure** Allow insecure HTTPS communication. i.e. bad SSL certificate.
- **-H, --header HEADER** Additional HTTP header to include with requests.
- **--timeout SECONDS** Timeout for api requests. Default is typically 30 seconds.
- **-C, --nocolor** Disable ANSI coloring
- **-B, --benchmark** Print benchmark time after the command is finished.
### 17.37.4 morpheus monitor-checks list

Usage: morpheus monitor-checks list

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-m, --max</td>
<td>Max Results</td>
</tr>
<tr>
<td>-s, --offset</td>
<td>Offset Results</td>
</tr>
<tr>
<td>-S, --sort</td>
<td>Sort Order</td>
</tr>
<tr>
<td>-D, --desc</td>
<td>Reverse Sort Order</td>
</tr>
<tr>
<td>-Q, --query</td>
<td>Query parameters. <code>PARAMS</code> format is <code>phrase=foobar&amp;category=web</code></td>
</tr>
<tr>
<td>-F, --fields x,y,z</td>
<td>Filter Output to a limited set of fields. Default is all fields.</td>
</tr>
<tr>
<td>-j, --json</td>
<td>JSON Output</td>
</tr>
<tr>
<td>-d, --dry-run</td>
<td>Dry Run, print the API request instead of executing it</td>
</tr>
<tr>
<td>-T, --token</td>
<td>Access token for authentication with --remote. Saved credentials are used by default.</td>
</tr>
<tr>
<td>-U, --username</td>
<td>Username for authentication.</td>
</tr>
<tr>
<td>-P, --password</td>
<td>Password for authentication.</td>
</tr>
<tr>
<td>-I, --insecure</td>
<td>Allow insecure HTTPS communication. i.e. bad SSL certificate.</td>
</tr>
<tr>
<td>-H, --header</td>
<td>Additional HTTP header to include with requests. Timeout for api requests. Default is typically 30 seconds.</td>
</tr>
<tr>
<td>-C, --nocolor</td>
<td>Disable ANSI coloring</td>
</tr>
<tr>
<td>-B, --benchmark</td>
<td>Print benchmark time after the command finished.</td>
</tr>
<tr>
<td>-V, --debug</td>
<td>Print extra output for debugging.</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Print this help</td>
</tr>
</tbody>
</table>

### 17.37.5 morpheus monitor-checks list-types

Usage: morpheus monitor-checks list-types

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-m, --max</td>
<td>Max Results</td>
</tr>
</tbody>
</table>
-o, --offset OFFSET Offset Results
-s, --search PHRASE Search Phrase
-S, --sort ORDER Sort Order
-D, --desc Reverse Sort Order
-j, --json JSON Output
-yaml YAML Output
--csv CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
--csv-quotes Wrap CSV values with '"'. Default: false
--csv-no-header Exclude header for CSV Output.
-F, --fields x,y,z Filter Output to a limited set of fields.
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Additional HTTP header to include with requests.
--header HEADER Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
--payload FILE Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting
-j, --json JSON Output

List monitoring check types.

17.37.6 morpheus monitor-checks mute

Usage: morpheus monitor-checks mute [name]
--disable Disable mute state instead, the same as unmute
-0, --option OPTION Option in the format -O field="value"
--prompt Always prompts. Use passed options as the default value.
-N, --no-prompt Skip prompts. Use default values for all optional fields.
--payload FILE Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting
-j, --json JSON Output
Morpheus Documentation, Release 3.6.2

-\(d\), --dry-run
  Dry Run, print the API request instead of executing it
  -\(--\)curl
    Dry Run to output API request as a curl command.
  -\(--\)scrub
    Mask secrets in output, such as the Authorization header. For use with --\(\)--curl and --\(\)dry-run.
  -\(--\)quiet
    No Output, do not print to stdout
  -\(--r\), --remote REMOTE
    Remote name. The current remote is used by default.
  -\(--\)remote-url URL
    Remote url. The current remote url is used by default.
  -\(--T\), --token TOKEN
    Access token for authentication with --\(\)remote.
  Saved credentials are used by default.
  -\(--U\), --username USERNAME
    Username for authentication.
  -\(--P\), --password PASSWORD
    Password for authentication.
  -\(--I\), --insecure
    Allow insecure HTTPS communication. i.e. bad SSL certificate.
  -\(--H\), --header HEADER
    Additional HTTP header to include with requests.
  -\(--\)timeout SECONDS
    Timeout for api requests. Default is typically 30 seconds.
  -\(--C\), --nocolor
    Disable ANSI coloring
  -\(--B\), --benchmark
    Print benchmark time after the command is finished.
  -\(--V\), --debug
    Print extra output for debugging.
  -\(--h\), --help
    Print this help

Mute a check. This prevents it from creating new incidents. [name] is required. This is the name or id of a check.

17.37.7 morpheus monitor-checks mute-all

Usage: morpheus monitor-checks mute-all
  --\(\)disable
  Disable mute state instead, the same as unmute-all
  --\(\)all
  Option in the format -O field="value"
  --prompt
  Always prompts. Use passed options as the default value.
  --\(\)no-prompt
  Skip prompts. Use default values for all optional fields.
  --\(\)payload FILE
  Payload from a local JSON or YAML file, skip all prompting
  --\(\)payload-dir DIRECTORY
  Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
  --\(\)json
  Payload JSON, skip all prompting
  --\(\)yaml
  Payload YAML, skip all prompting
  -\(--j\), --\(\)json
    JSON Output
  -\(--d\), --\(\)dry-run
    Dry Run, print the API request instead of executing it
  -\(--\)curl
    Dry Run to output API request as a curl command.
  -\(--\)scrub
    Mask secrets in output, such as the Authorization header. For use with --\(\)curl and --\(\)dry-run.
  -\(--\)quiet
    No Output, do not print to stdout
  -\(--r\), --remote REMOTE
    Remote name. The current remote is used by default.
  -\(--\)remote-url URL
    Remote url. The current remote url is used by default.
Morpheus Documentation, Release 3.6.2

(continued from previous page)

- **-T, --token** TOKEN  Access token for authentication with --remote.
- **-U, --username** USERNAME  Username for authentication.
- **-P, --password** PASSWORD  Password for authentication.
- **-I, --insecure**  Allow insecure HTTPS communication. i.e. bad
- **--ssl certificate**.
- **-H, --header** HEADER  Additional HTTP header to include with requests.
- **--timeout** SECONDS  Timeout for api requests. Default is typically
- 30 seconds.
- **-C, --nocolor**  Disable ANSI coloring
- **-B, --benchmark**  Print benchmark time after the command is finished.
- **-V, --debug**  Print extra output for debugging.
- **-h, --help**  Print this help

Mute all checks. This prevents the creation new incidents.

### 17.37.8 morpheus monitor-checks remove

Usage: morpheus monitor-checks remove [name]

- **-j, --json**  JSON Output
- **-d, --dry-run**  Dry Run, print the API request instead of executing it
- **--curl**  Dry Run to output API request as a curl command.
- **--scrub**  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- **-q, --quiet**  No Output, do not print to stdout
- **-r, --remote** REMOTE  Remote name. The current remote is used by default.
- **--remote-url** URL  Remote url. The current remote url is used by default.
- **-T, --token** TOKEN  Access token for authentication with --remote.
- **-U, --username** USERNAME  Username for authentication.
- **-P, --password** PASSWORD  Password for authentication.
- **-I, --insecure**  Allow insecure HTTPS communication. i.e. bad
- **--ssl certificate**.
- **--timeout** SECONDS  Timeout for api requests. Default is typically
- 30 seconds.
- **-C, --nocolor**  Disable ANSI coloring
- **-B, --benchmark**  Print benchmark time after the command is finished.
- **-V, --debug**  Print extra output for debugging.
- **-h, --help**  Print this help

### 17.37.9 morpheus monitor-checks unmute

Usage: morpheus monitor-checks unmute [name]

- **--payload** FILE  Payload from a local JSON or YAML file, skip all prompting
- **--payload-dir** DIRECTORY  Payload from a local directory containing 1-N JSON or YAML files, skip all prompting

(continues on next page)
## 17.37.10 morpheus monitor-checks unmute-all

Usage: morpheus monitor-checks unmute-all

```
--payload FILE          Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting
-j, --json             JSON Output
-d, --dry-run          Dry Run, print the API request instead of executing it
--curl                 Dry Run to output API request as a curl command.
--scrub                Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-q, --quiet            No Output, do not print to stdout
-r, --remote REMOTE    Remote name. The current remote is used by default.
--default.              Remote url. The current remote url is used by default.
-T, --token TOKEN      Access token for authentication with --remote.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
--insecure             Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER    Additional HTTP header to include with requests.
--timeout SECONDS      Timeout for api requests. Default is typically 30 seconds.
-c, --nocolor          Disable ANSI coloring
-B, --benchmark        Print benchmark time after the command is finished.
-V, --debug            Print extra output for debugging.
-h, --help             Print this help
```

Unmute a check.

[name] is required. This is the name or id of a check.
Unmute all checks.

17.37.11 morpheus monitor-checks update

Usage: morpheus monitor-checks update [name]
  --name VALUE Name
  --description VALUE Description
  --checkInterval VALUE Check Interval. Value is in milliseconds.
  --severity VALUE Max Severity. Determines the maximum severity
  --level this check can incur on an incident when failing. Default is critical
  --inUptime [on|off] Affects Availability. Default is on.
  -O, --option OPTION Option in the format -O field="value"
  --prompt Always prompts. Use passed options as the default value.
  -N, --no-prompt Skip prompts. Use default values for all optional fields.
  --payload FILE Payload from a local JSON or YAML file, skip all prompting
  --payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
  --payload-json JSON Payload JSON, skip all prompting
  --payload-yaml YAML Payload YAML, skip all prompting
  -j, --json JSON Output
  -d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -q, --quiet No Output, do not print to stdout
  -r, --remote REMOTE Remote name. The current remote is used by default.
  --remote-url URL Remote url. The current remote url is used by default.
  -T, --token TOKEN Access token for authentication with --remote.
  --saved credentials are used by default.
  -U, --username USERNAME Username for authentication.
  -P, --password PASSWORD Password for authentication.
  -I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
  -H, --header HEADER Additional HTTP header to include with requests.
  --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
  -C, --nocolor Disable ANSI coloring
  -B, --benchmark Print benchmark time after the command is finished.
  -h, --help Print this help
-V, --debug          Print extra output for debugging.
-h, --help           Print this help

Update a monitoring check.
[name] is required. This is the name or id of a check.
The available options vary by type.

17.38 morpheus monitor-contacts

Usage: morpheus monitor-contacts [command] [options]
Commands:
  add
  get
  list
  remove
  update

17.38.1 morpheus monitor-contacts add

Usage: morpheus monitor-contacts add [id]
  --name STRING       Contact name
  --email STRING      Contact email address
  --mobile STRING     Contact sms address
  --slackHook STRING  Contact slack hook
  -J, --json          JSON Output
  -d, --dry-run       Dry Run, print the API request instead of
  --executing it      Dry Run to output API request as a curl command.
  --scrub             Mask secrets in output, such as the
  --Authorization header. For use with --curl and --dry-run.
  -q, --quiet         No Output, do not print to stdout
  -C, --nocolor       Disable ANSI coloring
  -B, --benchmark     Print benchmark time after the command is
  --finished.
  -V, --debug         Print extra output for debugging.
  -h, --help          Print this help

17.38.2 morpheus monitor-contacts get

Usage: morpheus monitor-contacts get [id list]
  --history           Display History
  --notifications     Display Notifications
  -j, --json          JSON Output
  --csv               CSV Output
  --csv-delim CHAR    Delimiter for CSV Output values. Default: ','
  --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
  --csv-quotes        Wrap CSV values with '. Default: false
  --csv-no-header     Exclude header for CSV Output.
  -F, --fields x,y,z  Filter Output to a limited set of fields.
  Default is all fields.
17.38.3 morpheus monitor-contacts list

Usage: morpheus monitor-contacts list

- m, --max MAX Max Results
- o, --offset OFFSET Offset Results
- s, --search PHRASE Search Phrase
- S, --sort ORDER Sort Order
- D, --desc Reverse Sort Order
- Q, --query PARAMS Query parameters. PARAMS format is
  'phrase=foobar&category=web'
  --csv
  --csv-delim CHAR Delimiter for CSV Output values. Default: ','
  --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
  --csv-quotes Wrap CSV values with ". Default: false
  --csv-no-header Exclude header for CSV Output.
  --yaml YAML Output
-F, --fields x,y,z Filter Output to a limited set of fields.
 Default is all fields.
- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of executing it
- curl
- scrub
- Authorization header. For use with --curl and --dry-run.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help
17.38.4 morpheus monitor-contacts remove

Usage: morpheus monitor-contacts remove [id list]

- **-y, --yes** Auto Confirm
- **-q, --quiet** No Output, do not print to stdout
- **-j, --json** JSON Output
- **-d, --dry-run** Dry Run, print the API request instead of executing it
  - **--curl** Dry Run to output API request as a curl command.
  - **--scrub** Mask secrets in output, such as the Authorization header.
  - **--remote REMOTE** Remote name. The current remote is used by default.
  - **--remote-url URL** Remote url. The current remote url is used by default.
  - **--token TOKEN** Access token for authentication with --remote.
- Saved credentials are used by default.
  - **-U, --username USERNAME** Username for authentication.
  - **-P, --password PASSWORD** Password for authentication.
  - **-I, --insecure** Allow insecure HTTPS communication. i.e. bad SSL certificate.
- **-H, --header HEADER** Additional HTTP header to include with requests.
  - **--timeout SECONDS** Timeout for api requests. Default is typically 30 seconds.
  - **-C, --nocolor** Disable ANSI coloring
  - **-B, --benchmark** Print benchmark time after the command is finished.
- **-V, --debug** Print extra output for debugging.
- **-h, --help** Print this help

17.38.5 morpheus monitor-contacts update

Usage: morpheus monitor-contacts update [id]

- **--name STRING** Contact name
- **--email STRING** Contact email address
- **--mobile STRING** Contact sms address
- **--slackHook STRING** Contact slack hook
- **-j, --json** JSON Output
- **-d, --dry-run** Dry Run, print the API request instead of executing it
  - **--curl** Dry Run to output API request as a curl command.
  - **--scrub** Mask secrets in output, such as the Authorization header.
- **--remote REMOTE** Remote name. The current remote is used by default.
- Saved credentials are used by default.
  - **-U, --username USERNAME** Username for authentication.
  - **-P, --password PASSWORD** Password for authentication.
  - **-I, --insecure** Allow insecure HTTPS communication. i.e. bad SSL certificate.
- **-H, --header HEADER** Additional HTTP header to include with requests.
  - **--timeout SECONDS** Timeout for api requests. Default is typically 30 seconds.
  - **-C, --nocolor** Disable ANSI coloring
  - **-B, --benchmark** Print benchmark time after the command is finished.
- **-V, --debug** Print extra output for debugging.
- **-h, --help** Print this help
17.39  morpheus monitor-groups

Usage: morpheus monitor-groups [command] [options]

Commands:
    add
    get
    history
    list
    mute
    mute-all
    remove
    unmute
    unmute-all
    update

17.39.1 morpheus monitor-groups add

Usage: morpheus monitor-groups add [name]

    --name VALUE     Name
    --description VALUE Description
    --minHappy VALUE Min Checks. This specifies the minimum number of
    → checks within the group that must be happy to keep the group from becoming → unhealthy.
    --severity VALUE Max Severity. Determines the maximum severity → level this group can incur on an incident when failing. Default is critical
    → inUptime [on|off] Affects Availability. Default is on.
    → checks LIST Checks to include in this group, comma separated
    → list of IDs
    -O, --option OPTION Option in the format -O field="value"
    → prompt Always prompts. Use passed options as the → default value.
    -N, --no-prompt Skip prompts. Use default values for all
    → optional fields.
    → payload FILE Payload from a local JSON or YAML file, skip all
    → prompting
    → payload-dir DIRECTORY Payload from a local directory containing 1-N
    → JSON or YAML files, skip all prompting
    → payload-json JSON Payload JSON, skip all prompting
    → payload-yaml YAML Payload YAML, skip all prompting
    -j, --json JSON Output
    → d, --dry-run Dry Run, print the API request instead of → executing it
    → curl
    → scrub Mask secrets in output, such as the → Authorization header. For use with → curl and → dry-run.
    -r, --remote REMOTE Remote name. The current remote is used by → default.
    → remote-url URL Remote url. The current remote url is used by → default.
    → t, --token TOKEN Access token for authentication with → --remote.
    → saved credentials are used by default.
    → u, --username USERNAME Username for authentication.
    → p, --password PASSWORD Password for authentication.
    → i, --insecure Allow insecure HTTPS communication. i.e. bad

(continues on next page)
Create a new group of monitoring checks.
[name] is required and can be passed as --name instead.

17.39.2 morpheus monitor-groups get

Usage: morpheus monitor-groups get [id list]
   --history Display Check Group History
   -j, --json JSON Output
   --yaml YAML Output
   --csv CSV Output
   --csv-delim CHAR Delimiter for CSV Output values. Default: ','
   --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
   --csv-quotes Wrap CSV values with "". Default: false
   --csv-no-header Exclude header for CSV Output.
   -F, --fields x,y,z Filter Output to a limited set of fields.
   --d, --dry-run Dry Run, print the API request instead of
   --executing it
   --curl Dry Run to output API request as a curl command.
   --scrub Mask secrets in output, such as the
   --Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE Remote name. The current remote is used by
   --default.
   --remote-url URL Remote url. The current remote url is used by
   --default.
   -T, --token TOKEN Access token for authentication with --remote.
   --saved credentials are used by default.
   -U, --username USERNAME Username for authentication.
   -P, --password PASSWORD Password for authentication.
   -I, --insecure Allow insecure HTTPS communication. i.e. bad
   --SSL certificate.
   -H, --header HEADER Additional HTTP header to include with requests.
   --timeout SECONDS Timeout for api requests. Default is typically
   30 seconds.
   -C, --nocolor Disable ANSI coloring
   --benchmark Print benchmark time after the command is
   finished.
   -V, --debug Print extra output for debugging.
   -h, --help Print this help
### 17.39.3 morpheus monitor-groups history

Usage: `morpheus monitor-groups history [name] [options]`  
- **--severity LIST** Filter by severity. critical, warning, info  
- m, --max MAX Max Results  
- o, --offset OFFSET Offset Results  
- s, --search PHRASE Search Phrase  
- S, --sort ORDER Sort Order  
- D, --desc Reverse Sort Order  
- last-updated TIME Filter by Last Updated (gte)  
- yml YAML Output  
- csv CSV Output  
- csv-delim CHAR Delimiter for CSV Output values. Default: ','  
- csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'  
- csv-quotes Wrap CSV values with '"'. Default: false  
- csv-no-header Exclude header for CSV Output.  
- F, --fields x,y,z Filter Output to a limited set of fields. Default is all fields.  
- j, --json JSON Output  
- d, --dry-run Dry Run, print the API request instead of executing it  
- curl Dry Run to output API request as a curl command.  
- scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.  
- r, --remote REMOTE Remote name. The current remote is used by default.  
- remote-url URL Remote url. The current remote url is used by default.  
- token TOKEN Access token for authentication with --remote. Saved credentials are used by default.  
- username USERNAME Username for authentication.  
- password PASSWORD Password for authentication.  
- insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.  
- H, --header HEADER Additional HTTP header to include with requests.  
- timeout SECONDS Timeout for api requests. Default is typically 30 seconds.  
- C, --nocolor Disable ANSI coloring  
- B, --benchmark Print benchmark time after the command is finished.  
- V, --debug Print extra output for debugging.  
- h, --help Print this help

### 17.39.4 morpheus monitor-groups list

Usage: `morpheus monitor-groups list [name] [options]`  
- **--status VALUE** Filter by status. error,healthy,warning,muted  
- m, --max MAX Max Results  
- o, --offset OFFSET Offset Results  
- s, --search PHRASE Search Phrase  
- S, --sort ORDER Sort Order  
- D, --desc Reverse Sort Order  
- last-updated TIME Filter by Last Updated (gte)  
- Q, --query PARAMS Query parameters. PARAMS format is 'phrase=foobar&category=web'  

(continues on next page)
-j, --json JSON Output
--yaml YAML Output
--csv CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '
'
--csv-quotes Wrap CSV values with '. Default: false
--csv-no-header Exclude header for CSV Output.
-F, --fields x,y,z Filter Output to a limited set of fields.
Default is all fields.
-d, --dry-run Dry Run, print the API request instead of
executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the
Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by
--default.
--remote-url URL Remote url. The current remote url is used by
--default.
-T, --token TOKEN Access token for authentication with --remote.
--scrub Mask secrets in output, such as the
Authorization header. For use with --curl and --dry-run.
-d, --dry-run Dry Run, print the API request instead of
executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the
Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by
--default.
--remote-url URL Remote url. The current remote url is used by
--default.
-T, --token TOKEN Access token for authentication with --remote.
--scrub Mask secrets in output, such as the
Authorization header. For use with --curl and --dry-run.
-d, --dry-run Dry Run, print the API request instead of
executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the
Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by
--default.
--remote-url URL Remote url. The current remote url is used by
--default.

17.39.5 morpheus monitor-groups mute

Usage: morpheus monitor-groups mute [name]
   --disable Disable mute, the same as unmute
   -O, --option OPTION Option in the format -O field="value"
   --prompt Always prompts. Use passed options as the
   default value.
   -N, --no-prompt Skip prompts. Use default values for all
   optional fields.
   --payload FILE Payload from a local JSON or YAML file, skip all
   prompting
   --payload-dir DIRECTORY Payload from a local directory containing 1-N
   --JSON or YAML files, skip all prompting
   --payload-json JSON Payload JSON, skip all prompting
   --payload-yaml YAML Payload YAML, skip all prompting
   -j, --json JSON Output
   -d, --dry-run Dry Run, print the API request instead of
   executing it
   --curl Dry Run to output API request as a curl command.
   --scrub Mask secrets in output, such as the
   Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE Remote name. The current remote is used by
   --default.
--remote-url URL
-remote-url URL
Remote url. The current remote url is used by default.

-T, --token TOKEN
-token TOKEN
Access token for authentication with --remote.

-Saved credentials are used by default.
-Saved credentials are used by default.

-U, --username USERNAME
-username USERNAME
Username for authentication.

-P, --password PASSWORD
-password PASSWORD
Password for authentication.

-I, --insecure
--insecure
Allow insecure HTTPS communication. i.e. bad SSL certificate.

-SSL certificate.
-SSL certificate.

-H, --header HEADER
-header HEADER
Additional HTTP header to include with requests.

--timeout SECONDS
-timeout SECONDS
Timeout for api requests. Default is typically 30 seconds.

-q, --quiet
--quiet
No Output, do not print to stdout

-C, --nocolor
--nocolor
Disable ANSI coloring

-B, --benchmark
--benchmark
Print benchmark time after the command is finished.

-v, --debug
--debug
Print extra output for debugging.

-h, --help
-help
Print this help

Mute a check group. This prevents it from creating new incidents.
[name] is required. This is the name or id of a check group.

17.39.6 morpheus monitor-groups mute-all

Usage: morpheus monitor-groups mute-all

--disable
-disable
Disable mute, the same as unmute-all

-O, --option OPTION
-option OPTION
Option in the format -O field="value"

--prompt
--prompt
Always prompts. Use passed options as the default value.

-N, --no-prompt
-no-prompt
Skip prompts. Use default values for all optional fields.

--payload FILE
-payload FILE
Payload from a local JSON or YAML file, skip all prompting

--payload-dir DIRECTORY
-payload-dir DIRECTORY
Payload from a local directory containing 1-N JSON or YAML files, skip all prompting

--prompting
--prompting
Payload from a local JSON or YAML file, skip all prompting

--payload-yaml YAML
--payload-yaml YAML
Payload YAML, skip all prompting

-j, --json
--json
JSON Output

-d, --dry-run
-dry-run
Dry Run, print the API request instead of executing it

--curl
--curl
Dry Run to output API request as a curl command.

--scrub
--scrub
Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.

-r, --remote REMOTE
--remote REMOTE
Remote name. The current remote is used by default.

--remote-url URL
--remote-url URL
Remote url. The current remote url is used by default.

-T, --token TOKEN
-token TOKEN
Access token for authentication with --remote.

-Saved credentials are used by default.
-Saved credentials are used by default.

-U, --username USERNAME
-username USERNAME
Username for authentication.

-P, --password PASSWORD
-password PASSWORD
Password for authentication.

-I, --insecure
--insecure
Allow insecure HTTPS communication. i.e. bad SSL certificate.

-SSL certificate.
-SSL certificate.

-H, --header HEADER
-header HEADER
Additional HTTP header to include with requests.

--timeout SECONDS
-timeout SECONDS
Timeout for api requests. Default is typically 30 seconds.

(continues on next page)
Morpheus Documentation, Release 3.6.2

(continued from previous page)

-quiet
-nocolor
-benchmark

-no Output, do not print to stdout
-Disable ANSI coloring
-Print benchmark time after the command is finished.

-verbose
-help

- Print extra output for debugging.
- Print this help

Mute all check groups. This prevents the creation of new incidents.

### 17.39.7 morpheus monitor-groups remove

Usage: morpheus monitor-groups remove [name]

- j, --json
- d, --dry-run
- executing it
  - curl
  - scrub
- Authorization header. For use with --curl and --dry-run.
- q, --quiet
- C, --nocolor
- B, --benchmark

-no Output, do not print to stdout
-Disable ANSI coloring
-Print benchmark time after the command is finished.

-verbose
-help

- Print extra output for debugging.
- Print this help

### 17.39.8 morpheus monitor-groups unmute

Usage: morpheus monitor-groups unmute [name]

- r, --remote REMOTE
- default.
  - remote-url URL
- default.
- T, --token TOKEN
- Saved credentials are used by default.
- U, --username USERNAME
- default.
- P, --password PASSWORD
- default.
- I, --insecure
- SSL certificate.
- H, --header HEADER
- default.
- timeout SECONDS

Remote name. The current remote is used by default.
Username for authentication.
Password for authentication.
Allow insecure HTTPS communication. i.e. bad SSL certificate.
Additional HTTP header to include with requests.
Timeout for api requests. Default is typically

(continues on next page)
Morpheus Documentation, Release 3.6.2

Unmute a check group.
[ name ] is required. This is the name or id of a check.

17.39.9 morpheus monitor-groups unmute-all

Usage: morpheus monitor-groups unmute-all
   --payload FILE   Payload from a local JSON or YAML file, skip all prompting
   --payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
   --payload-json JSON Payload JSON, skip all prompting
   --payload-yaml YAML Payload YAML, skip all prompting
   -j, --json JSON Output
   -d, --dry-run Dry Run, print the API request instead of executing it
   --curl Dry Run to output API request as a curl command.
   --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE Remote name. The current remote is used by default.
   --remote-url URL Remote url. The current remote url is used by default.
   -T, --token TOKEN Access token for authentication with --remote.
   -U, --username USERNAME Username for authentication.
   -P, --password PASSWORD Password for authentication.
   -I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
   --header HEADER Additional HTTP header to include with requests.
   --timeout SECONDS Timeout for API requests. Default is typically 30 seconds.
   -q, --quiet No Output, do not print to stdout
   -C, --nocolor Disable ANSI coloring
   -B, --benchmark Print benchmark time after the command is finished.
   -V, --debug Print extra output for debugging.
   -h, --help Print this help

Unmute all check groups.

17.39.10 morpheus monitor-groups update

Usage: morpheus monitor-groups update [ name ]
   --name VALUE Name for this check group
   --description VALUE Description

(continues on next page)
--minHappy VALUE Min Checks. This specifies the minimum number of checks within the group that must be happy to keep the group from becoming unhealthy.
--severity VALUE Max Severity. Determines the maximum severity level this group can incur on an incident when failing. Default is critical
--inUptime [on|off] Affects Availability. Default is on.
--checks LIST Checks to include in this group, comma separated
--list of IDs
-O, --option OPTION Option in the format -O field="value"
--prompt Always prompts. Use passed options as the default value.
-N, --no-prompt Skip prompts. Use default values for all optional fields.
--payload FILE Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
--saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
--header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-q, --quiet No Output, do not print to stdout
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Update a check group.
[name] is required. This is the name or id of a check group.

17.40 morpheus monitor-incidents

Usage: morpheus monitor-incidents [command] [options]
Commands:
close
get
17.40.1 morpheus monitor-incidents close

Usage: morpheus monitor-incidents close [id list]
   -y, --yes   Auto Confirm
   -q, --quiet No Output, do not print to stdout
   -j, --json  JSON Output
   -d, --dry-run Dry Run, print the API request instead of executing it
   --curl Dry Run to output API request as a curl command.
   --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE Remote name. The current remote is used by default.
   --remote-url URL Remote url. The current remote url is used by default.
   --token TOKEN Access token for authentication with --remote.
   -U, --username USERNAME Username for authentication.
   -P, --password PASSWORD Password for authentication.
   -I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
   --header HEADER Additional HTTP header to include with requests.
   --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
   -C, --nocolor Disable ANSI coloring
   -B, --benchmark Print benchmark time after the command is finished.
   -V, --debug Print extra output for debugging.
   -h, --help Print this help

17.40.2 morpheus monitor-incidents get

Usage: morpheus monitor-incidents get [id list]
   --history Display Incident History
   --notifications Display Incident Notifications
   -j, --json JSON Output
   -y, --yaml YAML Output
   --csv CSV Output
   --csv-delim CHAR Delimiter for CSV Output values. Default: ','
   --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
   --csv-quotes Wrap CSV values with '. Default: false

(continues on next page)
17.40.3 morpheus monitor-incidents history

Usage: morpheus monitor-incidents history [id] [options]

--severity LIST Filter by severity. critical, warning, info
--max MAX Max Results
--offset OFFSET Offset Results
--search PHRASE Search Phrase
--sort ORDER Sort Order
--desc Reverse Sort Order
--last-updated TIME Filter by Last Updated (gte)
--csv CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
--csv-quotes Wrap CSV values with ''. Default: false
--csv-no-header Exclude header for CSV Output.
--yaml YAML Output
--fields x,y,z Filter Output to a limited set of fields.
--json JSON Output
--dry-run Dry Run, print the API request instead of
--executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the
--Authorization header. For use with --curl and --dry-run.
--remote REMOTE Remote name. The current remote is used by
--remote-url URL Remote url. The current remote url is used by
--default.
--token TOKEN Access token for authentication with --remote.
--username USERNAME Username for authentication.
--password PASSWORD Password for authentication.
--insecure Allow insecure HTTPS communication. i.e. bad
--header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically
--30 seconds.
--nocolor Disable ANSI coloring
--benchmark Print benchmark time after the command is
--finished.
--debug Print extra output for debugging.
--help Print this help
17.40.4 morpheus monitor-incidents list

Usage: morpheus monitor-incidents list

- --status LIST Filter by status. open, closed
- --severity LIST Filter by severity. critical, warning, info
- --max MAX Max Results
- --offset OFFSET Offset Results
- --search PHRASE Search Phrase
- --sort ORDER Sort Order
- --desc Reverse Sort Order
- --query PARAMS Query parameters. PARAMS format is
  'phrase=foobars&category=web'
- --last-updated TIME Filter by Last Updated (gte)
- --json JSON Output
- --yaml YAML Output
- --csv CSV Output
- --csv-delim CHAR Delimiter for CSV Output values. Default: ','
- --csv-newline CHAR Delimiter for CSV Output rows. Default: '\n'
- --csv-quotes Wrap CSV values with ''. Default: false
- --csv-no-header Exclude header for CSV Output.
- --fields x,y,z Filter Output to a limited set of fields.
- --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with
    --curl and --dry-run.
- --remote REMOTE Remote name. The current remote is used by default.
- --remote-url URL Remote url. The current remote url is used by default.
- --token TOKEN Access token for authentication with --remote.
- --username USERNAME Username for authentication.
- --password PASSWORD Password for authentication.
- --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- --header HEADER Additional HTTP header to include with requests.
- --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- --nocolor Disable ANSI coloring
- --benchmark Print benchmark time after the command is finished.
- --debug Print extra output for debugging.
- --help Print this help
17.40.5 morpheus monitor-incidents mute

Usage: morpheus monitor-incidents mute [id]

- disable Disable mute state instead, the same as unmute
- O, --option OPTION Option in the format -O field="value"
- prompt Always prompts. Use passed options as the default value.
- N, --no-prompt Skip prompts. Use default values for all optional fields.
- payload FILE Payload from a local JSON or YAML file, skip all prompting
- payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
- payload-json JSON Payload JSON, skip all prompting
- payload-yaml YAML Payload YAML, skip all prompting
- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of executing it
- curl Dry Run to output API request as a curl command.
- scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- q, --quiet No Output, do not print to stdout
- r, --remote REMOTE Remote name. The current remote is used by default.
- remote-url URL Remote url. The current remote url is used by default.
- T, --token TOKEN Access token for authentication with --remote.
- saved credentials are used by default.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
- timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

Mute an incident.

[id] is required. This is the id of an incident.
17.40.6 morpheus monitor-incidents mute-all

Usage: morpheus monitor-incidents mute-all

--disable
- all
- -O, --option OPTION Option in the format -O field="value"
- -prompt
- -default value.
- -N, --no-prompt Skip prompts. Use default values for all
- -optional fields.
- --payload FILE Payload from a local JSON or YAML file, skip all
- -prompting
- --payload-dir DIRECTORY Payload from a local directory containing 1-N
- -JSON or YAML files, skip all prompting
- --payload-json JSON Payload JSON, skip all prompting
- --payload-yaml YAML Payload YAML, skip all prompting
- -j, --json JSON Output
- -d, --dry-run Dry Run, print the API request instead of
- -executing it
- --curl
- -scrub Mask secrets in output, such as the
- -Authorization header. For use with --curl and --dry-run.
- -q, --quiet No Output, do not print to stdout
- -r, --remote REMOTE Remote name. The current remote is used by
- -default.
- --remote-url URL Remote url. The current remote url is used by
- -default.
- -T, --token TOKEN Access token for authentication with --remote.
- -saved credentials are used by default.
- -U, --username USERNAME Username for authentication.
- -P, --password PASSWORD Password for authentication.
- -I, --insecure Allow insecure HTTPS communication. i.e. bad
- -SSL certificate.
- -H, --header HEADER Additional HTTP header to include with requests.
- --timeout SECONDS Timeout for api requests. Default is typically
- -30 seconds.
- -C, --nocolor Disable ANSI coloring
- -B, --benchmark Print benchmark time after the command is
- -finished.
- -V, --debug Print extra output for debugging.
- -h, --help

Mute all open incidents.

17.40.7 morpheus monitor-incidents notifications

Usage: morpheus monitor-incidents notifications [id] [options]

- -m, --max MAX Max Results
- -o, --offset OFFSET Offset Results
- -s, --search PHRASE Search Phrase
- -S, --sort ORDER Sort Order
- -D, --desc Reverse Sort Order
- -csv
- --csv-delim CHAR Delimiter for CSV Output values. Default: ','
- --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
### 17.40.8 morpheus monitor-incidents reopen

Usage: `morpheus monitor-incidents reopen [id list]`

- `-y`, `--yes` Auto Confirm
- `-q`, `--quiet` No Output, do not print to stdout
- `-j`, `--json` JSON Output
- `-d`, `--dry-run` Dry Run, print the API request instead of
- `--executing it` Dry Run to output API request as a curl command.
- `-r`, `--remote REMOTE` Remote name. The current remote is used by.
- `--remote-url URL` Remote url. The current remote url is used by.
- `--timeout SECONDS` Timeout for api requests. Default is typically
- `-30 seconds.`
- `-C`, `--nocolor` Disable ANSI coloring
- `-B`, `--benchmark` Print benchmark time after the command is
- `--finished.`
- `-V`, `--debug` Print extra output for debugging.
- `-h`, `--help` Print this help
-B, --benchmark  Print benchmark time after the command is finished.
-V, --debug  Print extra output for debugging.
-h, --help  Print this help

17.40.9 morpheus monitor-incidents stats

Usage: morpheus monitor-incidents stats
-m, --max MAX  Max Results
-o, --offset OFFSET  Offset Results
-s, --search PHRASE  Search Phrase
-S, --sort ORDER  Sort Order
-D, --desc  Reverse Sort Order
-j, --json  JSON Output
-yaml  YAML Output
-csv  CSV Output
--csv-delim CHAR  Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR]  Delimiter for CSV Output rows. Default: '\n'
--csv-quotes  Wrap CSV values with '"'. Default: false
--csv-no-header  Exclude header for CSV Output.
-F, --fields x,y,z  Filter Output to a limited set of fields.
-d, --dry-run  Dry Run, print the API request instead of executing it
--curl  Dry Run to output API request as a curl command.
--scrub  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE  Remote name. The current remote is used by default.
--remote-url URL  Remote url. The current remote url is used by default.
-T, --token TOKEN  Access token for authentication with --remote.
--payload FILE  Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY  Payload from a local directory containing 1-N JSON or YAML files, skip all prompting

17.40.10 morpheus monitor-incidents unmute

Usage: morpheus monitor-incidents unmute [id]
--payload FILE  Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY  Payload from a local directory containing 1-N JSON or YAML files, skip all prompting

(continues on next page)
### 17.40.11 morpheus monitor-incidents unmute-all

Usage: morpheus monitor-incidents unmute-all

```
  --payload FILE  Payload from a local JSON or YAML file, skip all prompting
  --payload-dir DIRECTORY  Payload from a local directory containing 1-N
  --payload-json JSON  Payload JSON, skip all prompting
  --payload-yaml YAML  Payload YAML, skip all prompting
  -j, --json JSON Output
  -d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -q, --quiet No Output, do not print to stdout
  -r, --remote REMOTE Remote name. The current remote is used by default.
  --remote-url URL Remote url. The current remote url is used by default.
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -q, --quiet No Output, do not print to stdout
  -r, --remote REMOTE Remote name. The current remote is used by default.
  --remote-url URL Remote url. The current remote url is used by default.
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -q, --quiet No Output, do not print to stdout
  -r, --remote REMOTE Remote name. The current remote is used by default.
  --remote-url URL Remote url. The current remote url is used by default.
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -q, --quiet No Output, do not print to stdout
  -r, --remote REMOTE Remote name. The current remote is used by default.
  --remote-url URL Remote url. The current remote url is used by default.
```

Unmute an incident.

[id] is required. This is the id of an incident.
-I, --insecure
-SSL certificate.
-H, --header HEADER
---timeout SECONDS
---30 seconds.
-C, --nocolor
-B, --benchmark
---finished.
-V, --debug
-h, --help

Unmute all open incidents.

17.40.12 morpheus monitor-incidents update

Usage: morpheus monitor-incidents update [id]
-c, --comment STRING
--resolution STRING
--status STATUS
--severity STATUS
--name STRING
--startDate TIME
--endDate TIME
--inUptime BOOL
--insecure
-H, --header HEADER
---timeout SECONDS
---30 seconds.
--header HEADER
---timeout SECONDS
---30 seconds.
-C, --nocolor
-B, --benchmark
---finished.
-V, --debug
-h, --help

Unmute all open incidents.

17.40. morpheus monitor-incidents
17.41 morpheus network-domains

Usage: morpheus network-domains [command] [options]
Commands:
    add
    get
    list
    remove
    update

17.41.1 morpheus network-domains add

Usage: morpheus network-domains add
    --name VALUE Name for this network domain
    --description VALUE Description for this network domain
    --public-zone [on|off] Public Zone
    --domain-controller [on|off] Join Domain Controller
    --domain-username VALUE Domain Username
    --domain-password VALUE Domain Password
    --dc-server VALUE DC Server
    --ou-path VALUE OU Path
    --visibility [private|public] Visibility
    --tenant ID Tenant Account ID
    -O, --option OPTION Option in the format -O field="value"
    --prompt Always prompts. Use passed options as the default value.
    -N, --no-prompt Skip prompts. Use default values for all optional fields.
    --payload FILE Payload from a local JSON or YAML file, skip all prompting
    --payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
    --payload-json JSON Payload JSON, skip all prompting
    --payload-yaml YAML Payload YAML, skip all prompting
    --executing it Dry Run, print the API request instead of executing it
    --curl Dry Run to output API request as a curl command.
    --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
    -q, --quiet No Output, do not print to stdout
    -r, --remote REMOTE Remote name. The current remote is used by default.
    --remote-url URL Remote url. The current remote url is used by default.
    -I, --token TOKEN Access token for authentication with --remote.
    --saved credentials are used by default.
    -U, --username USERNAME Username for authentication.
    -P, --password PASSWORD Password for authentication.
    --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
    --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.

(continues on next page)
Create a new network domain.  
[name] is required and can be passed as --name instead.

17.41.2 morpheus network-domains get

Usage: morpheus network-domains get [network-domain]  
-j, --json JSON Output  
--yaml YAML Output  
--csv CSV Output  
--csv-delim CHAR Delimiter for CSV Output values. Default: ','  
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'  
--csv-quotes Wrap CSV values with '. Default: false  
--csv-no-header Exclude header for CSV Output.  
-F, --fields x,y,z Filter Output to a limited set of fields. Default is all fields.  
-d, --dry-run Dry Run, print the API request instead of executing it  
--curl Dry Run to output API request as a curl command.  
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.  
-r, --remote REMOTE Remote name. The current remote is used by default.  
--remote-url URL Remote url. The current remote url is used by default.  
-T, --token TOKEN Access token for authentication with --remote. Saved credentials are used by default.  
-U, --username USERNAME Username for authentication.  
P, --password PASSWORD Password for authentication.  
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.  
-H, --header HEADER Additional HTTP header to include with requests.  
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.  
--timeout SECONDS  
-C, --nocolor Disable ANSI coloring  
-B, --benchmark Print benchmark time after the command is finished.  
-V, --debug Print extra output for debugging.  
-h, --help Print this help  

Get details about a network domain.  
[network-domain] is required. This is the name or id of a network domain.

17.41.3 morpheus network-domains list

Usage: morpheus network-domains list  
-m, --max MAX Max Results
17.41.4 morpheus network-domains remove

Usage: morpheus network-domains remove [network-domain]

- Account Name
  - Account ID
  - Auto Confirm
  - JSON Output
  - Dry Run, print the API request instead of executing it
    - --curl
    - --scrub
    - Remote name. The current remote is used by default.
    - Remote url. The current remote url is used by default.
    - Access token for authentication with --remote.
    - Username for authentication.
    - Password for authentication.
    - Allow insecure HTTPS communication. i.e. bad SSL certificate.
    - Additional HTTP header to include with requests.
    - Timeout for api requests. Default is typically 30 seconds.
    - Disable ANSI coloring
    - Print benchmark time after the command is finished.
    - Print extra output for debugging.
    - Print this help

List network domains.
-T, --token TOKEN  Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME  Username for authentication.
-P, --password PASSWORD  Password for authentication.
-I, --insecure  Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER  Additional HTTP header to include with requests.
--timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor  Disable ANSI coloring
-B, --benchmark  Print benchmark time after the command is finished.
-V, --debug  Print extra output for debugging.
-h, --help  Print this help

Delete a network domain.
[network-domain] is required. This is the name or id of a network domain.

17.41.5 morpheus network-domains update

Usage: morpheus network-domains update [network-domain] [options]
--name VALUE  Name for this network domain
--type VALUE  Type of network domain
--ip-ranges LIST  IP Ranges, comma separated list IP ranges in the following format:
--format start-end.
-O, --option OPTION  Option in the format -O field="value"
--prompt  Always prompts. Use passed options as the default value.
-N, --no-prompt  Skip prompts. Use default values for all optional fields.
--payload FILE  Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY  Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-json JSON  Payload JSON, skip all prompting
--payload-yaml YAML  Payload YAML, skip all prompting
--json  JSON Output
--dry-run  Dry Run, print the API request instead of executing it
--curl  Dry Run to output API request as a curl command.
--scrub  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE  Remote name. The current remote is used by default.
--remote-url URL  Remote url. The current remote url is used by default.
-T, --token TOKEN  Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME  Username for authentication.
-P, --password PASSWORD  Password for authentication.
-I, --insecure  Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER  Additional HTTP header to include with requests.
--timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
Update a network domain. 

[network-domain] is required. This is the id of a network domain.

17.42 morpheus network-groups

Usage: morpheus network-groups [command] [options]

Commands:

- add
- get
- list
- remove
- update

17.42.1 morpheus network-groups add

Usage: morpheus network-groups add --networks [id,id,id]

--name VALUE Name for this network group
--description VALUE Description of network group
--networks LIST Networks in the group, comma separated list of
--network IDs
--group-access-all [on|off] Toggle Access for all groups.
--group-access LIST Group Access, comma separated list of group IDs.
--group-defaults LIST Group Default Selection, comma separated list of
--group IDs
--tenants LIST Tenant Access, comma separated list of account
--accounts LIST alias for --tenants
--visibility [private|public] Visibility
--option OPTION Option in the format -O field="value"
--prompt Always prompts. Use passed options as the default value.
--no-prompt Skip prompts. Use default values for all optional fields.
--payload FILE Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting
--json JSON Output
--dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-q, --quiet
No Output, do not print to stdout
-r, --remote REMOTE
Remote name. The current remote is used by default.
--remote-url URL
Remote url. The current remote url is used by default.
-T, --token TOKEN
Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME
Username for authentication.
-P, --password PASSWORD
Password for authentication.
-I, --insecure
Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER
Additional HTTP header to include with requests.
--timeout SECONDS
Timeout for api requests. Default is typically 30 seconds.
--json
JSON Output
--yaml
YAML Output
--csv
CSV Output
--csv-delim CHAR
Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR]
Delimiter for CSV Output rows. Default: '\n'
--csv-quotes
Wrap CSV values with "". Default: false
--csv-no-header
Exclude header for CSV Output.
-F, --fields x,y,z
Filter Output to a limited set of fields.
-D, --dry-run
Dry Run, print the API request instead of executing it
--curl
Dry Run to output API request as a curl command.
--scrub
Mask secrets in output, such as the Authorization header.
For use with --curl and --dry-run.
-r, --remote REMOTE
Remote name. The current remote is used by default.
--remote-url URL
Remote url. The current remote url is used by default.
-T, --token TOKEN
Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME
Username for authentication.
-P, --password PASSWORD
Password for authentication.
-I, --insecure
Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER
Additional HTTP header to include with requests.
--timeout SECONDS
Timeout for api requests. Default is typically 30 seconds.
--json
JSON Output
--yaml
YAML Output
--csv
CSV Output
--csv-delim CHAR
Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR]
Delimiter for CSV Output rows. Default: '\n'
--csv-quotes
Wrap CSV values with "". Default: false
--csv-no-header
Exclude header for CSV Output.
-F, --fields x,y,z
Filter Output to a limited set of fields.
-D, --dry-run
Dry Run, print the API request instead of executing it
--curl
Dry Run to output API request as a curl command.
--scrub
Mask secrets in output, such as the Authorization header.
For use with --curl and --dry-run.
-r, --remote REMOTE
Remote name. The current remote is used by default.
--remote-url URL
Remote url. The current remote url is used by default.
-T, --token TOKEN
Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME
Username for authentication.
-P, --password PASSWORD
Password for authentication.
-I, --insecure
Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER
Additional HTTP header to include with requests.
--timeout SECONDS
Timeout for api requests. Default is typically 30 seconds.
--json
JSON Output
--yaml
YAML Output
--csv
CSV Output
--csv-delim CHAR
Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR]
Delimiter for CSV Output rows. Default: '\n'
--csv-quotes
Wrap CSV values with "". Default: false
--csv-no-header
Exclude header for CSV Output.
-F, --fields x,y,z
Filter Output to a limited set of fields.
-D, --dry-run
Dry Run, print the API request instead of executing it
--curl
Dry Run to output API request as a curl command.
--scrub
Mask secrets in output, such as the Authorization header.
For use with --curl and --dry-run.
-r, --remote REMOTE
Remote name. The current remote is used by default.
--remote-url URL
Remote url. The current remote url is used by default.
Get details about a network group. 

[network-group] is required. This is the name or id of a network group.

17.42.3 morpheus network-groups list

Usage: morpheus network-groups list

-\[--max MAX\] Max Results
-\[--offset OFFSET\] Offset Results
-\[--search PHRASE\] Search Phrase
-\[--sort ORDER\] Sort Order
-\[--desc\] Reverse Sort Order
-\[--yaml\] YAML Output
-\[--csv\] CSV Output
-\[--csv-delim CHAR\] Delimiter for CSV Output values. Default: ','
-\[--csv-newline [CHAR]\] Delimiter for CSV Output rows. Default: '
'
-\[--csv-quotes \] Wrap CSV values with '"'. Default: false
-\[--csv-no-header\] Exclude header for CSV Output.
-\[--fields x,y,z\] Filter Output to a limited set of fields. Default is all fields.
-\[--json\] JSON Output
-\[--dry-run\] Dry Run, print the API request instead of executing it
-\[--curl\] Dry Run to output API request as a curl command.
-\[--scrub\] Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-\[--remote ACCOUNT\] Account Name
-\[--remote-url URL\] Remote url. The current remote is used by default.
-\[--token TOKEN\] Access token for authentication with --remote.
-\[--insecure\] Allow insecure HTTPS communication. i.e. bad SSL certificate.
-\[--header HEADER\] Additional HTTP header to include with requests.
-\[--timeout SECONDS\] Timeout for api requests. Default is typically 30 seconds.
-\[--nocolor\] Disable ANSI coloring
-\[--benchmark\] Print benchmark time after the command is finished.
-\[--debug\] Print extra output for debugging.
-\[--help\] Print this help

17.42.4 morpheus network-groups remove

Usage: morpheus network-groups remove [network-group]

-\[--account ACCOUNT\] Account Name
Delete a network group.

[Network-group] is required. This is the name or id of a network group.

17.42.5 morpheus network-groups update

Usage: morpheus network-groups update [network-group] [options]

--name VALUE Name for this network group
--description VALUE Description of network group
--networks LIST Networks in the group, comma separated list of
--network IDs
--group-access-all [on|off] Toggle Access for all groups.
--group-access LIST Group Access, comma separated list of group IDs.
--group-defaults LIST Group Default Selection, comma separated list of
--group IDs
--tenants LIST Tenant Access, comma separated list of account IDs
--accounts LIST alias for --tenants
--visibility [private|public] Visibility
--option OPTION Option in the format -O field="value"
--prompt Always prompts. Use passed options as the default value.
--no-prompt Skip prompts. Use default values for all optional fields.
--payload FILE Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
Update a network group.

[network-group] is required. This is the id of a network group.

17.43 morpheus network-pool-servers

Usage: morpheus network-pool-servers [command] [options]

Commands:
  add
  get
  list
  remove
  update

17.43.1 morpheus network-pool-servers add

Usage: morpheus network-pool-servers add
  --name VALUE      Name for this network pool server
  --type VALUE      Type of network pool server
  -O, --option OPTION Option in the format -O field="value"
  -P, --prompt      Always prompts. Use passed options as the default value.
  -N, --no-prompt   Skip prompts. Use default values for all optional fields.
  --payload FILE    Payload from a local JSON or YAML file, skip all prompting

(continues on next page)
Create a new network pool server. 

[name] is required and can be passed as --name instead.

17.43.2 morpheus network-pool-servers get

Usage: morpheus network-pool-servers get [network-pool-server]

-j, --json JSON Output
--yaml YAML Output
--csv CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
--csv-quotes Wrap CSV values with "". Default: false
--csv-no-header Exclude header for CSV Output.
-d, --dry-run Dry Run, print the API request instead of
executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the
Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by
default.
--remote-url URL Remote url. The current remote url is used by
default.
-T, --token TOKEN Access token for authentication with --remote.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad
SSL certificate.
--header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically
30 seconds.
-C, --nocolor Disable ANSI coloring
--benchmark Print benchmark time after the command is
finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Saved credentials are used by default.
Get details about a network pool server. 
[network-pool-server] is required. This is the name or id of a network pool server.

### 17.43.3 morpheus network-pool-servers list

Usage: morpheus network-pool-servers list

- `-m`, `--max` MAX
- `-o`, `--offset` OFFSET
- `-s`, `--search` PHRASE
- `-S`, `--sort` ORDER
- `-D`, `--desc`
- `--yaml`
- `--csv`
- `--csv-delim` CHAR
- `--csv-newline` [CHAR]
- `--csv-quotes`
- `--csv-no-header`
- `-F`, `--fields` x,y,z
- `--json`
- `-d`, `--dry-run`
- `-r`, `--remote` REMOTE
- `--remote-url` URL
- `-T`, `--token` TOKEN
- Saved credentials are used by default.
- `-U`, `--username` USERNAME
- `-P`, `--password` PASSWORD
- `-I`, `--insecure`
- `--ssl certificate`
- `-H`, `--header` HEADER
- `--timeout` SECONDS
- `-30` seconds.
- `-C`, `--nocolor`
- `-B`, `--benchmark`
- `--benchmark-enabled`
- `-V`, `--debug`
- `-h`, `--help`

(continues on next page)
17.43.4 morpheus network-pool-servers remove

Usage: morpheus network-pool-servers remove [network-pool-server]

- -a, --account ACCOUNT Account Name
- -A, --account-id ID Account ID
- -y, --yes Auto Confirm
- -j, --json JSON Output
- -d, --dry-run Dry Run, print the API request instead of executing it
  - -curl Dry Run to output API request as a curl command.
  - -scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- -r, --remote REMOTE Remote name. The current remote is used by default.
  - -remote-url URL Remote url. The current remote url is used by default.
- -T, --token TOKEN Access token for authentication with --remote.
  Saved credentials are used by default.
- -u, --username USERNAME Username for authentication.
- -p, --password PASSWORD Password for authentication.
- -i, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- -H, --header HEADER Additional HTTP header to include with requests.
  - -timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- -C, --nocolor Disable ANSI coloring
- -B, --benchmark Print benchmark time after the command is finished.
- -v, --debug Print extra output for debugging.
- -h, --help Print this help

Delete a network pool server. [network-pool-server] is required. This is the name or id of a network pool server.

17.43.5 morpheus network-pool-servers update

Usage: morpheus network-pool-servers update [network-pool-server] [options]

- -O, --option OPTION Option in the format -O field="value"
- -prompt Always prompts. Use passed options as the default value.
  - -no-prompt Skip prompts. Use default values for all optional fields.
- -payload FILE Payload from a local JSON or YAML file, skip all prompting
- -payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
- -payload-json JSON Payload JSON, skip all prompting

(continues on next page)
17.44 morpheus network-pools

Usage: morpheus network-pools [command] [options]
Commands:
  add
  get
  list
  remove
  update

17.44.1 morpheus network-pools add

Usage: morpheus network-pools add
  --name VALUE             Name for this network pool
  --type VALUE             Type of network pool
  --ip-ranges LIST         IP Ranges, comma separated list IP ranges in the
  --format start-end.
  -O, --option OPTION      Option in the format -O field="value"
  --prompt                 Always prompts. Use passed options as the
  --default value.
  -N, --no-prompt          Skip prompts. Use default values for all
  --optional fields.
  --payload FILE           Payload from a local JSON or YAML file, skip all
Create a new network pool.

[17.44.2] morpheus network-pools get

Usage: morpheus network-pools get [network-pool]

- j, --json JSON Output
- yam YAML Output
- csv CSV Output
- csv-delim CHAR Delimiter for CSV Output values. Default: ','
- csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
- csv-quotes Wrap CSV values with "". Default: false
- csv-no-header Exclude header for CSV Output.
- f, --fields x,y,z Filter Output to a limited set of fields.
- d, --dry-run Dry Run, print the API request instead of executing it
- r, --remote REMOTE Remote name. The current remote is used by default.
- u, --username USERNAME Username for authentication.
- p, --password PASSWORD Password for authentication.
- i, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- h, --header HEADER Additional HTTP header to include with requests.
- timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- C, --ncolno Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- v, --debug Print extra output for debugging.
- h, --help Print this help

Created credentials are used by default.
Get details about a network pool.

[network-pool] is required. This is the name or id of a network pool.

### 17.44.3 morpheus network-pools list

Usage: morpheus network-pools list

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-m, --max MAX</td>
<td>Max Results</td>
</tr>
<tr>
<td>-o, --offset OFFSET</td>
<td>Offset Results</td>
</tr>
<tr>
<td>-s, --search PHRASE</td>
<td>Search Phrase</td>
</tr>
<tr>
<td>-S, --sort ORDER</td>
<td>Sort Order</td>
</tr>
<tr>
<td>-D, --desc</td>
<td>Reverse Sort Order</td>
</tr>
<tr>
<td>--yaml</td>
<td>YAML Output</td>
</tr>
<tr>
<td>--csv</td>
<td>CSV Output</td>
</tr>
<tr>
<td>--csv-delim CHAR</td>
<td>Delimiter for CSV Output values. Default: ','</td>
</tr>
<tr>
<td>--csv-newline [CHAR]</td>
<td>Delimiter for CSV Output rows. Default: '\n'</td>
</tr>
<tr>
<td>--csv-quotes</td>
<td>Wrap CSV values with &quot;&quot;. Default: false</td>
</tr>
<tr>
<td>--csv-no-header</td>
<td>Exclude header for CSV Output.</td>
</tr>
<tr>
<td>-F, --fields x,y,z</td>
<td>Filter CSV Output to a limited set of fields.</td>
</tr>
<tr>
<td>--json</td>
<td>JSON Output</td>
</tr>
<tr>
<td>--dry-run</td>
<td>Dry Run, print the API request instead of executing it</td>
</tr>
<tr>
<td>--curl</td>
<td>Dry Run to output API request as a curl command.</td>
</tr>
<tr>
<td>--scrub</td>
<td>Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.</td>
</tr>
<tr>
<td>-r, --remote REMOTE</td>
<td>Remote name. The current remote is used by default.</td>
</tr>
<tr>
<td>--remote-url URL</td>
<td>Remote url. The current remote url is used by default.</td>
</tr>
<tr>
<td>-T, --token TOKEN</td>
<td>Access token for authentication with --remote.</td>
</tr>
<tr>
<td>-U, --usernameUSERNAME</td>
<td>Username for authentication.</td>
</tr>
<tr>
<td>-P, --password PASSWORD</td>
<td>Password for authentication.</td>
</tr>
<tr>
<td>-I, --insecure</td>
<td>Allow insecure HTTPS communication. i.e. bad SSL certificate.</td>
</tr>
<tr>
<td>-H, --header HEADER</td>
<td>Additional HTTP header to include with requests.</td>
</tr>
<tr>
<td>--timeout SECONDS</td>
<td>Timeout for api requests. Default is typically 30 seconds.</td>
</tr>
<tr>
<td>-C, --nocolor</td>
<td>Disable ANSI coloring</td>
</tr>
<tr>
<td>-B, --benchmark</td>
<td>Print benchmark time after the command is finished.</td>
</tr>
<tr>
<td>-V, --debug</td>
<td>Print extra output for debugging.</td>
</tr>
</tbody>
</table>
-h, --help                  Print this help
List network pools.

17.44.4 morpheus network-pools remove

Usage: morpheus network-pools remove [network-pool]
-a, --account ACCOUNT    Account Name
-A, --account-id ID      Account ID
-y, --yes                Auto Confirm
-j, --json               JSON Output
-d, --dry-run            Dry Run, print the API request instead of executing it
--curl                   Dry Run to output API request as a curl command.
--scrub                  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE     Remote Name. The current remote is used by default.
--remote-url URL        Remote url. The current remote url is used by default.
-T, --token TOKEN       Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure           Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER     Additional HTTP header to include with requests.
--timeout SECONDS       Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor           Disable ANSI coloring
-B, --benchmark         Print benchmark time after the command is finished.
-V, --debug             Print extra output for debugging.
-h, --help              Print this help

Delete a network pool.
[network-pool] is required. This is the name or id of a network pool.

17.44.5 morpheus network-pools update

Usage: morpheus network-pools update [network-pool] [options]
--name VALUE             Name for this network pool
--type VALUE             Type of network pool
--ip-ranges LIST        IP Ranges, comma separated list IP ranges in the format start-end.
-format start-end.
-O, --option OPTION     Option in the format -O field="value"
--prompt                 Always prompts. Use passed options as the default value.
-N, --no-prompt          Skip prompts. Use default values for all optional fields.
--payload FILE          Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting

(continues on next page)
Update a network pool.

[network-pool] is required. This is the id of a network pool.

17.45 morpheus network-proxies

Usage: morpheus network-proxies [command] [options]

Commands:
add
get
list
remove
update

17.45.1 morpheus network-proxies add

Usage: morpheus network-proxies add
--name VALUE Name for this network proxy
--proxy-host VALUE Proxy Host
--proxy-port VALUE Proxy Port
--proxy-user VALUE Proxy User
--proxy-password VALUE Proxy Password
--proxy-domain VALUE Proxy Domain
--proxy-workstation VALUE Proxy Workstation
--visibility [private|public] Visibility

(continues on next page)
Create a new network proxy.

[name] is required and can be passed as --name instead.

### 17.45.2 morpheus network-proxies get

Usage: morpheus network-proxies get [network-proxy]

- j, --json          JSON Output
- y, --yaml          YAML Output
- c, --csv           CSV Output
- csv-delim CHAR     Delimiter for CSV Output values. Default: ','
- csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
- csv-quotes [CHAR]  Wrap CSV values with ''. Default: false
- csv-no-header      Exclude header for CSV Output.
- F, --fields x,y,z  Filter Output to a limited set of fields.
- d, --dry-run       Dry Run, print the API request instead of
- h, --help          Print this help
Get details about a network proxy.

[network-proxy] is required. This is the name or id of a network proxy.

17.45.3 morpheus network-proxies list

Usage: morpheus network-proxies list

\ `-m, --max MAX` Max Results
\ `-o, --offset OFFSET` Offset Results
\ `-s, --search PHRASE` Search Phrase
\ `-S, --sort ORDER` Sort Order
\ `-D, --desc` Reverse Sort Order
\ `--yaml` YAML Output
\ `--csv` CSV Output
\ `--csv-delim CHAR` Delimiter for CSV Output values. Default: ','
\ `--csv-newline [CHAR]` Delimiter for CSV Output rows. Default: '\n'
\ `--csv-quotes` Wrap CSV values with "". Default: false
\ `--csv-no-header` Exclude header for CSV Output.
\ `-F, --fields x,y,z` Filter Output to a limited set of fields.

Default is all fields.

\ `-j, --json` JSON Output
\ `-d, --dry-run` Dry Run, print the API request instead of executing it
\ `--curl` Dry Run to output API request as a curl command.
\ `--scrub` Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
\ `-r, --remote REMOTE` Remote name. The current remote is used by default.
\ `--remote-url URL` Remote url. The current remote url is used by default.
\ `-T, --token TOKEN` Access token for authentication with --remote.
\ `-U, --username USERNAME` Username for authentication.
\ `-P, --password PASSWORD` Password for authentication.
\ `-I, --insecure` Allow insecure HTTPS communication. i.e. bad SSL certificate.
\ `--header HEADER` Additional HTTP header to include with requests.
\ `--timeout SECONDS` Timeout for api requests. Default is typically 30 seconds.
\ `--nocolor` Disable ANSI coloring
\ `-B, --benchmark` Print benchmark time after the command is finished.
\ `-v, --debug` Print extra output for debugging.
\ `-h, --help` Print this help
-I, --insecure
Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER
Additional HTTP header to include with requests.
--timeout SECONDS
Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor
Disable ANSI coloring
-B, --benchmark
Print benchmark time after the command is finished.
-V, --debug
Print extra output for debugging.
-h, --help
Print this help

List network proxies.

17.45.4 morpheus network-proxies remove

Usage: morpheus network-proxies remove [network-proxy]
-a, --account ACCOUNT
Account Name
-A, --account-id ID
Account ID
-y, --yes
Auto Confirm
-j, --json
JSON Output
-d, --dry-run
Dry Run, print the API request instead of executing it
--curl
Dry Run to output API request as a curl command.
--scrub
Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE
Remote name. The current remote is used by default.
--remote-url URL
Remote url. The current remote url is used by default.
-T, --token TOKEN
Access token for authentication with --remote. Saved credentials are used by default.
-U, --username USERNAME
Username for authentication.
-P, --password PASSWORD
Password for authentication.
-I, --insecure
Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER
Additional HTTP header to include with requests.
--timeout SECONDS
Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor
Disable ANSI coloring
-B, --benchmark
Print benchmark time after the command is finished.
-V, --debug
Print extra output for debugging.
-h, --help
Print this help

Delete a network proxy. [network-proxy] is required. This is the name or id of a network proxy.

17.45.5 morpheus network-proxies update

Usage: morpheus network-proxies update [network-proxy] [options]
--name VALUE
Name for this network proxy
--proxy-host VALUE
Proxy Host

(continues on next page)
### 17.46 morpheus network-services

Update a network proxy.

[network-proxy] is required. This is the id of a network proxy.

#### Usage: morpheus network-services [command] [options]

#### Commands:
- list
17.46.1 morpheus network-services list

Usage: morpheus network-services list

-\( m \), --max MAX  
Max Results
-\( o \), --offset OFFSET  
Offset Results
-\( s \), --search PHRASE  
Search Phrase
-\( S \), --sort ORDER  
Sort Order
-\( D \), --desc  
Reverse Sort Order
-\( y \), --yaml  
YAML Output
-\( c \), --csv  
CSV Output
-\( c \), --csv-delim CHAR  
Delimiter for CSV Output values. Default: ','
-\( c \), --csv-newline [CHAR]  
Delimiter for CSV Output rows. Default: '\n'
-\( c \), --csv-quotes  
Wrap CSV values with "". Default: false
-\( c \), --csv-no-header  
Exclude header for CSV Output.
-\( F \), --fields x,y,z  
Filter Output to a limited set of fields.
-\( j \), --json  
JSON Output
-\( d \), --dry-run  
Dry Run, print the API request instead of
-\( e \), --executing it  
Dry Run to output API request as a curl command.
-\( s \), --scrub  
Mask secrets in output, such as the
-\( v \), --remote REMOTE  
Remote name. The current remote is used by
-\( u \), --remote-url URL  
Remote url. The current remote url is used by
-\( t \), --token TOKEN  
Access token for authentication with --remote.
-\( u \), --username USERNAME  
Username for authentication.
-\( p \), --password PASSWORD  
Password for authentication.
-\( i \), --insecure  
Allow insecure HTTPS communication. i.e. bad
 SSL certificate.
-\( h \), --header HEADER  
Additional HTTP header to include with requests.
-\( c \), --timeout SECONDS  
Timeout for api requests. Default is typically 30 seconds.
-\( n \), --nocolor  
Disable ANSI coloring
-\( b \), --benchmark  
Print benchmark time after the command is finished.
-\( v \), --debug  
Print extra output for debugging.
-\( h \), --help  
Print this help

List network services.

17.47 morpheus networks

Usage: morpheus networks [command] [options]
Commands:
  add
  get
  list
  remove
  update
17.47.1 morpheus networks add

Usage: morpheus networks add -t TYPE

-\c, --cloud CLOUD Cloud Name or ID
-t, --type ID Network Type Name or ID
-name VALUE Name for this network
-description VALUE Description of network
-gateway VALUE Gateway
-\dns-primary VALUE DNS Primary
-\dns-secondary VALUE DNS Secondary
-cidr VALUE CIDR
-vlan-id VALUE VLAN ID
-pool ID Network Pool
-dhcp-server [on|off] DHCP Server
-allow-ip-override [on|off] Allow IP Override
-domain VALUE Network Domain ID
-scan [on|off] Scan Network
-proxy-bypass [on|off] Bypass Proxy for Appliance URL
-no-proxy LIST No Proxy Addresses
-group-access-all [on|off] Toggle Access for all groups.
-group-access LIST Group Access, comma separated list of group IDs.
-group-defaults LIST Group Default Selection, comma separated list of group IDs
--tenants LIST Tenant Access, comma separated list of account IDs
--accounts LIST alias for --tenants
-visibility [private|public] Visibility
-O, --option OPTION Option in the format -O field="value"
-prompt Always prompts. Use passed options as the default value.
-N, --no-prompt Skip prompts. Use default values for all optional fields.
-prompting
--payload FILE Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-q, --quiet No Output, do not print to stdout
-r, --remote REMOTE Remote name. The current remote is used by default.
-remote-url URL Remote url. The current remote url is used by default.
-t, --token TOKEN Access token for authentication with --remote.
-S, --saved-credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.

(continues on next page)
**-H, --header HEADER** Additional HTTP header to include with requests.

**-30 seconds.**

**-C, --nocolor** Disable ANSI coloring

**-B, --benchmark** Print benchmark time after the command is finished.

**-V, --debug** Print extra output for debugging.

**-h, --help** Print this help

Create a new network.

[name] is required and can be passed as --name instead.

### 17.47.2 morpheus networks get

Usage: morpheus networks get [network]

- **-j, --json** JSON Output
- **-y, --yaml** YAML Output
- **-csv** CSV Output
- **-csv-delim CHAR** Delimiter for CSV Output values. Default: ','
- **-csv-newline [CHAR]** Delimiter for CSV Output rows. Default: '\n'
- **-csv-quotes** Wrap CSV values with ''. Default: false
- **-csv-no-header** Exclude header for CSV Output.
- **-F, --fields x,y,z** Filter Output to a limited set of fields. Default is all fields.
- **-d, --dry-run** Dry Run, print the API request instead of executing it
- **--curl** Dry Run to output API request as a curl command.
- **--scrub** Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- **-r, --remote REMOTE** Remote name. The current remote is used by default.
- **--remote-url URL** Remote url. The current remote url is used by default.
- **-T, --token TOKEN** Access token for authentication with --remote.

Saved credentials are used by default.

- **-U, --username USERNAME** Username for authentication.
- **-P, --password PASSWORD** Password for authentication.
- **-I, --insecure** Allow insecure HTTPS communication. i.e. bad SSL certificate.
- **-H, --header HEADER** Additional HTTP header to include with requests.
- **-30 seconds.**
- **-C, --nocolor** Disable ANSI coloring
- **-B, --benchmark** Print benchmark time after the command is finished.
- **-V, --debug** Print extra output for debugging.
- **-h, --help** Print this help

Get details about a network.

[network] is required. This is the name or id of a network.
17.47.3 morpheus networks list

Usage: morpheus networks list

- --cidr VALUE Filter by cidr, matches beginning of value.
- -m, --max MAX Max Results
- -o, --offset OFFSET Offset Results
- -s, --search PHRASE Search Phrase
- -S, --sort ORDER Sort Order
- -D, --desc Reverse Sort Order
- -yaml YAML Output
- --csv CSV Output
- --csv-delim CHAR Delimiter for CSV Output values. Default: ','
- --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
- --csv-quotes Wrap CSV values with "". Default: false
- --csv-no-header Exclude header for CSV Output.
- -F, --fields x,y,z Filter Output to a limited set of fields. Default is all fields.
- -j, --json JSON Output
- -d, --dry-run Dry Run, print the API request instead of executing it
- --curl Dry Run to output API request as a curl command.
- --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- -r, --remote REMOTE Remote name. The current remote is used by default.
- -A, --account-id ID Account ID
- -y, --yes Auto Confirm
- -j, --json JSON Output
- -d, --dry-run Dry Run, print the API request instead of executing it
- -h, --help Print this help

List networks.

17.47.4 morpheus networks remove

Usage: morpheus networks remove [network]

- -a, --account ACCOUNT Account Name
- -A, --account-id ID Account ID
- -y, --yes Auto Confirm
- -j, --json JSON Output
- -d, --dry-run Dry Run, print the API request instead of executing it
- --curl Dry Run to output API request as a curl command.
- --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.

(continues on next page)
Delete a network.

[network] is required. This is the name or id of a network.

17.47.5 morpheus networks update

Usage: morpheus networks update [network] [options]

--name VALUE Name for this network
--description VALUE Description of network
--gateway VALUE Gateway
--dns-primary VALUE DNS Primary
--dns-secondary VALUE DNS Secondary
--cidr VALUE CIDR
--vlan-id VALUE VLAN ID
--pool ID Network Pool
--dhcp-server [on|off] DHCP Server
--allow-ip-override [on|off] Allow IP Override
--domain VALUE Network Domain ID
--scan [on|off] Scan Network
--proxy-bypass [on|off] Bypass Proxy for Appliance URL
--no-proxy LIST No Proxy Addresses
--group-access-all [on|off] Toggle Access for all groups.
--group-access LIST Group Access, comma separated list of group IDs.
--group-defaults LIST Group Default Selection, comma separated list of group IDs.
--group IDs
--tenants LIST Tenant Access, comma separated list of account IDs
--IDs
--accounts LIST alias for --tenants
--visibility [private|public]
  --active [on|off] Can be used to disable a network
  --option OPTION Option in the format -O field="value"
  --prompt Always prompts. Use passed options as the default value.
  --no-prompt Skip prompts. Use default values for all optional fields.
  --payload FILE Payload from a local JSON or YAML file, skip all prompting
Update a network.

[network] is required. This is the id of a network.

**17.48 morpheus passwd**

Usage: morpheus passwd [username] [options]

- **-a, --account ACCOUNT** Account Name
- **-A, --account-id ID** Account ID
- **-O, --option OPTION** Option in the format -O field="value"
  - **--prompt** Always prompts. Use passed options as the
- **default value.**
- **-N, --no-prompt** Skip prompts. Use default values for all
  - **optional fields.**
- **-j, --json** JSON Output
- **-d, --dry-run** Dry Run, print the API request instead of
  - **executing it**
  - **--curl** Dry Run to output API request as a curl command.
  - **--scrub** Mask secrets in output, such as the
  - **Authorization header. For use with --curl and --dry-run.**
  - **-r, --remote REMOTE** Remote name. The current remote is used by
  - **default.**
  - **--remote-url URL** Remote url. The current remote url is used by
  - **default.**
  - **-T, --token TOKEN** Access token for authentication with --remote.
  - **Saved credentials are used by default.**
  - **-U, --username USERNAME** Username for authentication.
  - **-P, --password PASSWORD** Password for authentication.
  - **--insecure** Allow insecure HTTPS communication. i.e. bad
  - **SSL certificate.**
  - **-H, --header HEADER** Additional HTTP header to include with requests.
  - **--timeout SECONDS** Timeout for api requests. Default is typically
  - **30 seconds.**
  - **--nocolor** Disable ANSI coloring
  - **-B, --benchmark** Print benchmark time after the command is
  - **finished.**
  - **-V, --debug** Print extra output for debugging.
  - **-h, --help** Print this help

Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password  PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header  HEADER Additional HTTP header to include with requests.
               --timeout SECONDS Timeout for API requests. Default is typically 30 seconds.
-g, --quiet No Output, do not print to stdout
-y, --yes Auto Confirm
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Change your password or the password of another user.
[username] is optional. This is the username of the user to update. Default is your own.
Be careful with this command, the default behavior is to update your own password.

17.49 morpheus policies

Usage: morpheus policies [command] [options]
Commands:
    add
    get
    get-type
    list
    list-types
    remove
    update

17.49.1 morpheus policies add

Usage: morpheus policies add -t TYPE
    -g, --group  GROUP Group Name or ID, for scoping the policy to a group.
    -c, --cloud  CLOUD Cloud Name or ID, for scoping the policy to a cloud.
    -u, --user  USER Username or ID, for scoping the policy to a user.
    -t, --type ID Policy Type Name or ID
    --name VALUE Name for this policy
    --description VALUE Description of policy
    --accounts LIST Tenant accounts, comma separated list of account IDs
    --enabled [on|off] Can be used to disable a policy
    --config JSON Policy Config JSON
    --config-yaml YAML Policy Config YAML
    --config-file FILE Policy Config from a local JSON or YAML file
    -O, --option OPTION Option in the format -O field=value
    --prompt Always prompts. Use passed options as the default value.
    --default value.

(continues on next page)
-N, --no-prompt Skip prompts. Use default values for all optional fields.
--payload FILE Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --quiet No Output, do not print to stdout
--remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-S, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Create a new policy.
[name] is optional and can be passed as --name instead.

17.49.2 morpheus policies get

Usage: morpheus policies get [policy]
-g, --group GROUP Group Name or ID
-c, --cloud CLOUD Cloud Name or ID
-j, --json JSON Output
--yaml YAML Output
--csv CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
--csv-quotes Wrap CSV values with ''. Default: false
--csv-no-header Exclude header for CSV Output.
-F, --fields x,y,z Filter Output to a limited set of fields. Default is all fields.
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
Get details about a policy.

17.49.3 morpheus policies get-type

Usage: morpheus policies get-type [policy-type]

- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of executing it
  -c, --curl Dry Run to output API request as a curl command.
  -s, --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by default.
- d, --remote-url URL Remote url. The current remote url is used by default.
- t, --token TOKEN Access token for authentication with --remote.
- u, --username USERNAME Username for authentication.
- p, --password PASSWORD Password for authentication.
- i, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- h, --header HEADER Additional HTTP header to include with requests.
  - t, --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- c, --nocolor Disable ANSI coloring
- b, --benchmark Print benchmark time after the command is finished.
- v, --debug Print extra output for debugging.
- h, --help Print this help

Get details about a policy type.

[policy-type] is required. This is ID of a policy type.
17.49.4 morpheus policies list

Usage: morpheus policies list

- **-g, --group GROUP** Group Name or ID
- **-c, --cloud CLOUD** Cloud Name or ID
- **-u, --user USER** Username or ID
- **-G, --global** Global policies only
- **-m, --max MAX** Max Results
- **-o, --offset OFFSET** Offset Results
- **-s, --search PHRASE** Search Phrase
- **-S, --sort ORDER** Sort Order
- **-D, --desc Reverse Sort Order**
- **-Q, --query PARAMS** Query parameters. PARAMS format is `'phrase=foobar&category=web'`
- **-j, --json** JSON Output
- **--yaml** YAML Output
- **--csv** CSV Output
- **--csv-delim CHAR** Delimiter for CSV Output. Default: ','
- **--csv-newline [CHAR]** Delimiter for CSV Output rows. Default: '\n'
- **--csv-quotes Wrap CSV values with "".** Default: false
- **--csv-no-header Exclude header for CSV Output.**
- **-F, --fields x,y,z** Filter Output to a limited set of fields. Default is all fields.
- **-d, --dry-run** Dry Run, print the API request instead of executing it.
- **--curl** Dry Run to output API request as a curl command.
- **--scrub Mask secrets in output, such as the Authorization header.** For use with --curl and --dry-run.
- **-r, --remote REMOTE** Remote name. The current remote is used by default.
- **--remote-url URL** Remote url. The current remote url is used by default.
- **-T, --token TOKEN** Access token for authentication with --remote.
- **Saved credentials are used by default.**
- **-U, --username USERNAME** Username for authentication.
- **-P, --password PASSWORD** Password for authentication.
- **-I, --insecure** Allow insecure HTTPS communication. i.e. bad SSL certificate.
- **-H, --header HEADER** Additional HTTP header to include with requests.
- **--timeout SECONDS** Timeout for api requests. Default is typically 30 seconds.
- **--nocolor Disable ANSI coloring**
- **--benchmark** Print benchmark time after the command is finished.
- **-v, --debug** Print extra output for debugging.
- **-h, --help** Print this help

List policies.

17.49.5 morpheus policies list-types

Usage: morpheus policies list-types

- **-j, --json** JSON Output
- **--yaml** YAML Output
- **--csv** CSV Output
### 17.49.6 morpheus policies remove

**Usage:** morpheus policies remove [policy]

- `-a`, `--account ACCOUNT` Account Name
- `-A`, `--account-id ID` Account ID
- `-y`, `--yes` Auto Confirm
- `-j`, `--json` JSON Output
- `-d`, `--dry-run` Dry Run, print the API request instead of executing it
  - `--curl` Dry Run to output API request as a curl command.
  - `--scrub` Mask secrets in output, such as the Authorization header.
- `-r`, `--remote REMOTE` Remote name. The current remote is used by default.
- `-r`, `--remote-url URL` Remote url. The current remote url is used by default.
- `-T`, `--token TOKEN` Access token for authentication with `--remote`.
- `-U`, `--username USERNAME` Username for authentication.
- `-P`, `--password PASSWORD` Password for authentication.
- `-I`, `--insecure` Allow insecure HTTPS communication. i.e. bad SSL certificate.
- `-H`, `--header HEADER` Additional HTTP header to include with requests.
- `-D`, `--timeout SECONDS` Timeout for api requests. Default is typically 30 seconds.
- `-C`, `--nocolor` Disable ANSI coloring
- `-B`, `--benchmark` Print benchmark time after the command is finished.
- `-V`, `--debug` Print extra output for debugging.
- `-h`, `--help` Print this help
Delete a policy.
[policy] is required. This is the id of a policy.

17.49.7 morpheus policies update

Usage: morpheus policies update [policy] [options]
   --name VALUE Name for this policy
   --description VALUE Description of policy
   --accounts LIST Tenant accounts, comma separated list of account IDs
   --enabled [on|off] Can be used to disable a policy
   --config JSON Policy Config JSON
   --config-yaml YAML Policy Config YAML
   --config-file FILE Policy Config from a local JSON or YAML file
   -O, --option OPTION Option in the format -O field="value"
   --prompt Always prompts. Use passed options as the default value.
   -N, --no-prompt Skip prompts. Use default values for all optional fields.
   -j, --json JSON Output
   -d, --dry-run Dry Run, print the API request instead of executing it
     --curl Dry Run to output API request as a curl command.
     --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE Remote name. The current remote is used by default.
   --remote-url URL Remote url. The current remote url is used by default.
   --token TOKEN Access token for authentication with --remote.
   --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
   -C, --nocolor Disable ANSI coloring
   -B, --benchmark Print benchmark time after the command is finished.
   -V, --debug Print extra output for debugging.
   -h, --help Print this help

Update a policy.
[policy] is required. This is the id of a policy.
17.50 morpheus power-schedules

Usage: morpheus power-schedules [command] [options]
Commands:
   add
   add-hosts
   add-instances
   get
   list
   remove
   remove-hosts
   remove-instances
   update

17.50.1 morpheus power-schedules add

Usage: morpheus power-schedules add [name]
   --name VALUE Name
   --description VALUE Description
   --type [power|power off] Type of Schedule. Default is 'power'
   --timezone CODE The timezone. Default is UTC.
   --sundayOn [0-24] Sunday start hour. Default is 0.
   --sundayOff [0-24] Sunday end hour. Default is 24.
   --mondayOn [0-24] Monday start hour. Default is 0.
   --tuesdayOn [0-24] Tuesday start hour. Default is 0.
   --tuesdayOff [0-24] Tuesday end hour. Default is 24.
   --wednesdayOn [0-24] Wednesday start hour. Default is 0.
   --thursdayOn [0-24] Thursday start hour. Default is 0.
   --fridayOn [0-24] Friday start hour. Default is 0.
   --fridayOff [0-24] Friday end hour. Default is 24.
   --saturdayOn [0-24] Saturday start hour. Default is 0.
   --saturdayOff [0-24] Saturday end hour. Default is 24.
   --enabled [on|off] Can be used to disable it
   -O, --option OPTION Option in the format -O field="value"
   --prompt Always prompts. Use passed options as the default value.
   -N, --no-prompt Skip prompts. Use default values for all optional fields.
   --payload FILE Payload from a local JSON or YAML file, skip all prompting
   --payload-dir DIRECTORY Payload from a local directory containing 1-N files, skip all prompting
   --payload-json JSON Payload JSON, skip all prompting
   --payload-yaml YAML Payload YAML, skip all prompting
   -j, --json JSON Output
   -d, --dry-run Dry Run, print the API request instead of executing it
   --curl Dry Run to output API request as a curl command.
   --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE Remote name. The current remote is used by default.

(continues on next page)
Create a new power schedule.

[name] is required and can be passed as --name instead.

17.50.2 morpheus power-schedules add-hosts

Usage: morpheus power-schedules add-hosts [name] [host]

--payload FILE Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-q, --quiet No Output, do not print to stdout
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help
Assign hosts to a power schedule.
[name] is required. This is the name or id of a power schedule.
[host] is required. This is the name or id of a host. More than one can be passed.

17.50.3 morpheus power-schedules add-instances

Usage: morpheus power-schedules add-instances [name] [instance]
--payload FILE Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
-curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-t, --token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-u, --username USERNAME Username for authentication.
-p, --password PASSWORD Password for authentication.
-i, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-q, --quiet No Output, do not print to stdout
-C, --nocolor Disable ANSI coloring
-b, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Assign instances to a power schedule.
[name] is required. This is the name or id of a power schedule.
[instance] is required. This is the name or id of an instance. More than one can be passed.

17.50.4 morpheus power-schedules get

Usage: morpheus power-schedules get [name]
--max-instances VALUE Display a limited number of instances in schedule. Default is 25
--max-hosts VALUE Display a limited number of hosts in schedule. Default is 25
-j, --json JSON Output
17.50.5 morpheus power-schedules list

Usage: morpheus power-schedules list

-\(m\), --max MAX Max Results
-\(o\), --offset OFFSET Offset Results
-\(s\), --search PHRASE Search Phrase
-\(S\), --sort ORDER Sort Order
-\(D\), --desc Reverse Sort Order
-\(j\), --json JSON Output
-\(y\), --yaml YAML Output
-\(c\), --csv CSV Output
-\(c\)-delim CHAR Delimiter for CSV Output values. Default: ','
-\(c\)-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
-\(c\)-quotes Wrap CSV values with '. Default: false
-\(c\)-no-header Exclude header for CSV Output.
-\(F\), --fields x,y,z Filter Output to a limited set of fields.
-\(d\), --dry-run Dry Run, print the API request instead of executing it
-\(c\)url Dry Run to output API request as a curl command.
-\(s\)crub Mask secrets in output, such as the Authorization header.
-\(T\), --token TOKEN Access token for authentication with --remote.
-\(U\), --username USERNAME Username for authentication.
-\(P\), --password PASSWORD Password for authentication.
-\(I\), --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-\(H\), --header HEADER Additional HTTP header to include with requests.
-\(t\), --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-\(C\), --nocolor Disable ANSI coloring
-\(B\), --benchmark Print benchmark time after the command is finished.
-\(V\), --debug Print extra output for debugging.
-\(h\), --help Print this help

(continues on next page)
Morpheus Documentation, Release 3.6.2

(continued from previous page)

```
--remote-url URL  Remote url. The current remote url is used by
  default.
-T, --token TOKEN Access token for authentication with --remote.  
  Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad
  SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
  --timeout SECONDS Timeout for api requests. Default is typically
  30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is
  finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help
```

17.50.6 morpheus power-schedules remove

```
Usage: morpheus power-schedules remove [name]
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of
  executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the
  Authorization header. For use with --curl and --dry-run.
-q, --quiet No Output, do not print to stdout
-y, --yes Auto Confirm
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is
  finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help
```

17.50.7 morpheus power-schedules remove-hosts

```
Usage: morpheus power-schedules remove-hosts [name] [host]
  --payload FILE Payload from a local JSON or YAML file, skip all
  prompting
  --payload-dir DIRECTORY Payload from a local directory containing 1-N
  payload files, skip all prompting
  --payload-json JSON Payload JSON, skip all prompting
  --payload-yaml YAML Payload YAML, skip all prompting
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of
  executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the
  Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by
  default.
  --remote-url URL Remote url. The current remote url is used by
  default.
```

(continues on next page)
Remove hosts from a power schedule.

[name] is required. This is the name or id of a power schedule.

[host] is required. This is the name or id of a host. More than one can be passed.

### 17.50.8 morpheus power-schedules remove-instances

Usage: morpheus power-schedules remove-instances [name] [instance]

- **--payload FILE** Payload from a local JSON or YAML file, skip all prompting
- **--payload-dir DIRECTORY** Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
- **--payload-json JSON** Payload JSON, skip all prompting
- **--payload-yaml YAML** Payload YAML, skip all prompting
- **--json** JSON Output
- **--dry-run** Dry Run, print the API request instead of executing it
- **--curl** Dry Run to output API request as a curl command.
- **--scrub** Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- **--remote REMOTE** Remote name. The current remote is used by default.
- **--remote-url URL** Remote url. The current remote url is used by default.
- **-T, --token TOKEN** Access token for authentication with --remote. Saved credentials are used by default.
- **-U, --username USERNAME** Username for authentication.
- **-P, --password PASSWORD** Password for authentication.
- **-I, --insecure** Allow insecure HTTPS communication. i.e. bad SSL certificate.
- **-H, --header HEADER** Additional HTTP header to include with requests.
- **--timeout SECONDS** Timeout for api requests. Default is typically 30 seconds.
- **-q, --quiet** No Output, do not print to stdout
- **-C, --nocolor** Disable ANSI coloring
- **-B, --benchmark** Print benchmark time after the command is finished.
- **-V, --debug** Print extra output for debugging.
- **-h, --help** Print this help
Remove instances from a power schedule.

[name] is required. This is the name or id of a power schedule.

[instance] is required. This is the name or id of an instance. More than one can be passed.

17.50.9 morpheus power-schedules update

Usage: morpheus power-schedules update [name]

--name VALUE Name
--description VALUE Description
--type [power|power off] Type of Schedule. Default is 'power'
--timezone CODE The timezone. Default is UTC.
--sundayOn [0-24] Sunday on hour. Default is 0.
--mondayOn [0-24] Monday on hour. Default is 0.
--tuesdayOn [0-24] Tuesday on hour. Default is 0.
--wednesdayOn [0-24] Wednesday on hour. Default is 0.
--thursdayOn [0-24] Thursday on hour. Default is 0.
--fridayOn [0-24] Friday on hour. Default is 0.
--saturdayOn [0-24] Saturday on hour. Default is 0.
--enabled [on|off] Can be used to disable it

-O, --option OPTION Option in the format -O field="value"
--prompt Always prompts. Use passed options as the default value.
--no-prompt Skip prompts. Use default values for all optional fields.
-payload FILE Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting
--dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
--remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
--token TOKEN Access token for authentication with --remote.
--insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
--header HEADER Additional HTTP header to include with requests.
--timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
-q, --quiet       No Output, do not print to stdout
-C, --nocolor     Disable ANSI coloring
-B, --benchmark   Print benchmark time after the command is finished.
-V, --debug       Print extra output for debugging.
-h, --help        Print this help

Update a power schedule. [name] is required. This is the name or id of a power schedule.

17.51 morpheus process

Usage: morpheus process [command] [options]
Commands:
  get
  get-event
  list

17.51.1 morpheus process get

Usage: morpheus process get [id]
  --details       Display more details. Shows everything, untruncated.
  -Q, --query PARAMS  Query parameters. PARAMS format is 'phrase=foobar&category=web'
  -j, --json       JSON Output
  --yaml           YAML Output
  --csv            CSV Output
  --csv-delim CHAR Delimiter for CSV Output values. Default: ','
  --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
  --csv-quotes     Wrap CSV values with '. Default: false
  --csv-no-header  Exclude header for CSV Output.
  -F, --fields x,y,z Filter Output to a limited set of fields. Default is all fields.
  -d, --dry-run    Dry Run, print the API request instead of executing it
  --curl           Dry Run to output API request as a curl command.
  --scrub          Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -r, --remote REMOTE Remote name. The current remote is used by default.
  --remote-url URL Remote url. The current remote url is used by default.
  --token TOKEN    Access token for authentication with --remote.
  -U, --username USERNAME Username for authentication.
  -P, --password PASSWORD Password for authentication.
  -I, --insecure   Allow insecure HTTPS communication. i.e. bad SSL certificate.
  -H, --header HEADER Additional HTTP header to include with requests.
Display details for a specific process.

**17.51.2 morpheus process get-event**

```
Usage: morpheus process get-event [event-id]

-Q, --query PARAMS  Query parameters. PARAMS format is
                        'phrase=foobar&category=web'
-j, --json           JSON Output
-y, --yaml           YAML Output
 -c, --csv            CSV Output
 -d, --csv-delim CHAR Delimiter for CSV Output values. Default: ','
 -n, --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
 -q, --csv-quotes     Wrap CSV values with '". Default: false
 -h, --csv-no-header  Exclude header for CSV Output.
 -f, --fields x,y,z   Filter Output to a limited set of fields.
-d, --dry-run        Dry Run, print the API request instead of
 executing it
 -r, --remote REMOTE Remote name. The current remote is used by default.
 -u, --remote-url URL Remote url. The current remote url is used by default.
 -t, --token TOKEN    Access token for authentication with --remote.
-S, --saved-credentials are used by default.
-k, --key KEYNAME     Username for authentication.
 -p, --password PASSWORD Password for authentication.
 -i, --insecure       Allow insecure HTTPS communication. i.e. bad SSL certificate.
 -H, --header HEADER  Additional HTTP header to include with requests.
 -30, --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor        Disable ANSI coloring
-B, --benchmark      Print benchmark time after the command is finished.
-V, --debug          Print extra output for debugging.
-h, --help           Print this help
```

Display details for a specific process event. [event-id] is required. This is the id of the process event.
# 17.51.3 morpheus process list

```
Usage: morpheus process list
       --events Display sub processes (events).
       --output Display process output.
       --details Display all details. Includes sub processes, output and error data is not truncated.
       --app ID Limit results to specific app(s).
       --instance ID Limit results to specific instance(s).
       --container ID Limit results to specific container(s).
       --host ID Limit results to specific host(s).
       --cloud ID Limit results to specific cloud(s).
       -m, --max MAX Max Results
       -o, --offset OFFSET Offset Results
       -s, --sort PHRASE Sort Order
       -D, --desc Reverse Sort Order
       -Q, --query PARAMS Query parameters. PARAMS format is 'phrase=foobar&category=web'
       -j, --json JSON Output
       --yaml YAML Output
       --csv CSV Output
       --csv-delim CHAR Delimiter for CSV Output values. Default: ','
       --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
       --csv-quotes Wrap CSV values with "". Default: false
       --csv-no-header Exclude header for CSV Output.
       -F, --fields x,y,z Filter Output to a limited set of fields. Default is all fields.
       -d, --dry-run Dry Run, print the API request instead of executing it
       --curl Dry Run to output API request as a curl command.
       --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
       -r, --remote REMOTE Remote name. The current remote is used by default.
       --remote-url URL Remote url. The current remote url is used by default.
       -T, --token TOKEN Access token for authentication with --remote.
       --username USERNAME Username for authentication.
       --password PASSWORD Password for authentication.
       -I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
       --header HEADER Additional HTTP header to include with requests.
       --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
       -c, --nocolor Disable ANSI coloring
       -b, --benchmark Print benchmark time after the command is finished.
       -V, --debug Print extra output for debugging.
       -h, --help Print this help

List historical processes.
```
17.52 morpheus recent-activity

Usage: morpheus recent-activity
--start TIMESTAMP Start timestamp. Default is 30 days ago.
--end TIMESTAMP End timestamp. Default is now.
-a, --account ACCOUNT Account Name
-A, --account-id ID Account ID
-m, --max MAX Max Results
-o, --offset OFFSET Offset Results
-s, --search PHRASE Search Phrase
-S, --sort ORDER Sort Order
-D, --desc Reverse Sort Order
-j, --json JSON Output
-xml YAML Output
-csv CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
--csv-quotes Wrap CSV values with ". Default: false
--csv-no-header Exclude header for CSV Output.
-F, --fields x,y,z Filter Output to a limited set of fields.
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--default.
--remote-url URL Remote url. The current remote url is used by default.
--default.
-T, --token TOKEN Access token for authentication with --remote.
--saved credentials are used by default.
-u, --username USERNAME Username for authentication.
-p, --password PASSWORD Password for authentication.
-i, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-c, --nocolor Disable ANSI coloring
-b, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.53 morpheus remote

Usage: morpheus remote [command] [options]
Commands:
add
check
current
get
list

(continues on next page)
17.53.1 morpheus remote add

Usage: morpheus remote add [name] [url]

    [name] The name for your appliance. eg. mymorph
    --mycompany.com
    --use Make this the current appliance
    -d, --default Does the same thing as --use
    --secure Prevent insecure HTTPS communication. This is
    → enabled by default.
    --insecure Allow insecure HTTPS communication. i.e. Ignore
    → SSL errors.
    -q, --quiet No Output, do not print to stdout
    -C, --nocolor Disable ANSI coloring
    -B, --benchmark Print benchmark time after the command is
    → finished.
    -V, --debug Print extra output for debugging.
    -h, --help Print this help

This will add a new remote appliance to your morpheus client configuration.
If the new remote is your one and only, --use is automatically applied and
it will be made the current remote appliance.
This command will prompt you to login and/or setup a fresh appliance.
Prompting can be skipped with use of the --quiet option.

17.53.2 morpheus remote check

Usage: morpheus remote check [name]

    [name] is required. This is the name of the remote. Use 'current' to check the
    → active appliance."
    -a, --all Refresh all appliances
    -q, --quiet No Output, do not print to stdout
    -C, --nocolor Disable ANSI coloring
    -B, --benchmark Print benchmark time after the command is
    → finished.
    -V, --debug Print extra output for debugging.
    -h, --help Print this help

This can be used to refresh a remote appliance.
It makes an api request to the configured appliance url to check the status and
→ version.

[name] is required. This is the name of the remote. Use 'current' to check the active
→ appliance."

Chapter 17. Morpheus Commands
17.53.3 morpheus remote current

Usage: morpheus remote current
-n, --name Print only the name.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Print details about the current remote appliance. The default behavior is the same as 'remote get current'.

17.53.4 morpheus remote get

Usage: morpheus remote get [name]
-j, --json JSON Output
--csv CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
--csv-quotes Wrap CSV values with '. Default: false
--csv-no-header Exclude header for CSV Output.
-F, --fields x,y,z Filter Output to a limited set of fields. Default is all fields.
-q, --quiet No Output, do not print to stdout
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.53.5 morpheus remote list

Usage: morpheus remote list
-a, --all Show all the appliance activity details
-j, --json JSON Output
--yaml YAML Output
--csv CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
--csv-quotes Wrap CSV values with '. Default: false
--csv-no-header Exclude header for CSV Output.
-F, --fields x,y,z Filter Output to a limited set of fields. Default is all fields.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

This outputs a list of the configured remote appliances. It also indicates the current appliance. The current appliance is where morpheus will send its commands by default. That is, in absence of the '--remote' option.
17.53.6 morpheus remote remove

```
Usage: morpheus remote remove [name]
    -y, --yes          Auto Confirm
    -q, --quiet        No Output, do not print to stdout
    -C, --nocolor      Disable ANSI coloring
    -B, --benchmark    Print benchmark time after the command is finished.
    -V, --debug        Print extra output for debugging.
    -h, --help         Print this help
```

This will delete an appliance from your list.

17.53.7 morpheus remote setup

```
Usage: morpheus remote setup
    -O, --option OPTION Option in the format -O field="value"
    -P, --prompt        Always prompts. Use passed options as the default value.
    -N, --no-prompt     Skip prompts. Use default values for all optional fields.
    -j, --json          JSON Output
    -d, --dry-run       Dry Run, print the API request instead of executing it.
    -C, --nocolor       Disable ANSI coloring
    -B, --benchmark     Print benchmark time after the command is finished.
    -V, --debug         Print extra output for debugging.
    -h, --help          Print this help
    -I, --insecure      Allow insecure HTTPS communication. i.e. bad SSL certificate.
```

This can be used to initialize a new appliance.
You will be prompted to create the master account.
This is only available on a new, freshly installed, remote appliance.

17.53.8 morpheus remote unuse

```
Usage: morpheus remote unuse
    -C, --nocolor      Disable ANSI coloring
    -B, --benchmark    Print benchmark time after the command is finished.
    -V, --debug        Print extra output for debugging.
    -h, --help         Print this help
```

This clears the current remote appliance.
You will need to use an appliance, or pass the --remote option to your commands.
17.53.9 morpheus remote update

Usage: morpheus remote update [name]
- --url URL Update the url of your remote appliance
- --secure Prevent insecure HTTPS communication. This is enabled by default
- --insecure Allow insecure HTTPS communication. i.e. Ignore SSL errors.
- --use Make this the current appliance
- -q, --quiet No Output, do not print to stdout
- -C, --nocolor Disable ANSI coloring
- -B, --benchmark Print benchmark time after the command is finished.
- -V, --debug Print extra output for debugging.
- -h, --help Print this help

This can be used to update remote appliance settings.

17.53.10 morpheus remote use

Usage: morpheus remote use [name]
- -q, --quiet No Output, do not print to stdout
- -C, --nocolor Disable ANSI coloring
- -B, --benchmark Print benchmark time after the command is finished.
- -V, --debug Print extra output for debugging.
- -h, --help Print this help

Make an appliance the current remote appliance.
This allows you to switch between your different appliances.
You may override this with the --remote option in your commands.

17.54 morpheus roles

Usage: morpheus roles [command] [options]
Commands:
- add
- get
- list
- remove
- update
- update-blueprint-access
- update-cloud-access
- update-feature-access
- update-global-blueprint-access
- update-global-cloud-access
- update-global-group-access
- update-global-instance-type-access
- update-group-access
- update-instance-type-access
17.54.1 morpheus roles add

Usage: morpheus roles add [options]

--authority VALUE Name
--description VALUE Description (optional)
--roleType VALUE Role Type (optional) Default: user
--baseRole VALUE Copy From Role (optional)
--multitenant on|off Multitenant (optional) Default: off - A
  Multitenant role is automatically copied into all existing subaccounts as well as
  placed into a subaccount when created. Useful for providing a set of predefined
  roles a Customer can use
--instanceLimits.maxStorage VALUE Max Storage (bytes) (optional)
--instanceLimits.maxMemory VALUE Max Memory (bytes) (optional)
--instanceLimits.maxCpu VALUE CPU Count (optional)
-O, --option OPTION Option in the format -O field="value"
-prompt Always prompts. Use passed options as the
  default value.
-N, --no-prompt Skip prompts. Use default values for all
optional fields.
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of
executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the
Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by
default.
--remote-url URL Remote url. The current remote url is used by
default.
-T, --token TOKEN Access token for authentication with --remote.
Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad
SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically
30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is
finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.54.2 morpheus roles get

Usage: morpheus roles get [name]

-f, --feature-access Display Feature Access
-g, --group-access Display Group Access
-c, --cloud-access Display Cloud Access
-i, --instance-type-access Display Instance Type Access
-b, --blueprint-access Display Blueprint Access

(continues on next page)
-a, --all-access  Display All Access Lists
-j, --json      YAML Output
--yaml        YAMl Output
--csv         CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '
'
--csv-quotes Wrap CSV values with '. Default: false
--csv-no-header Exclude header for CSV Output.
-F, --fields x,y,z Filter Output to a limited set of fields.
-d, --dry-run Dry Run, print the API request instead of exectuting it
--crl         Dry Run to output API request as a curl command.
--scrub       Mask secrets in output, such as the Authorization header. For use with --crl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-u, --username USERNAME Username for authentication.
-p, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-c, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Get details about a role. [name] is required. This is the name or id of a role.

17.54.3 morpheus roles list

Usage: morpheus roles list
-m, --max MAX Max Results
-o, --offset OFFSET Offset Results
-s, --search PHRASE Search Phrase
-S, --sort ORDER Sort Order
-D, --desc Reverse Sort Order
-Q, --query PARAMS Query parameters. PARAMS format is 'phrase=foobar&category=web'
-j, --json      YAML Output
--yaml        YAMl Output
--csv         CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '
'
--csv-quotes Wrap CSV values with '. Default: false
--csv-no-header Exclude header for CSV Output.
-F, --fields x,y,z Filter Output to a limited set of fields.

Default is all fields.
-d, --dry-run
  Dry Run, print the API request instead of executing it
  --curl
  Dry Run to output API request as a curl command.
  --scrub
  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -r, --remote REMOTE
  Remote name. The current remote is used by default.
  --remote-url URL
  Remote url. The current remote url is used by default.
  --default.
  --token TOKEN
  Access token for authentication with --remote.
  Saved credentials are used by default.
  -U, --username USERNAME
  Username for authentication.
  -P, --password PASSWORD
  Password for authentication.
  -I, --insecure
  Allow insecure HTTPS communication. i.e. bad SSL certificate.
  -H, --header HEADER
  Additional HTTP header to include with requests.
  --timeout SECONDS
  Timeout for api requests. Default is typically 30 seconds.
  -c, --nocolor
  Disable ANSI coloring
  -B, --benchmark
  Print benchmark time after the command is finished.
  -v, --debug
  Print extra output for debugging.
  -h, --help
  Print this help

List roles.

17.54.4 morpheus roles remove

Usage: morpheus roles remove [name]
  -y, --yes
  Auto Confirm
  -j, --json
  JSON Output
  -d, --dry-run
  Dry Run, print the API request instead of executing it
  --curl
  Dry Run to output API request as a curl command.
  --scrub
  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -r, --remote REMOTE
  Remote name. The current remote is used by default.
  --remote-url URL
  Remote url. The current remote url is used by default.
  --default.
  --token TOKEN
  Access token for authentication with --remote.
  Saved credentials are used by default.
  -U, --username USERNAME
  Username for authentication.
  -P, --password PASSWORD
  Password for authentication.
  -I, --insecure
  Allow insecure HTTPS communication. i.e. bad SSL certificate.
  -H, --header HEADER
  Additional HTTP header to include with requests.
  --timeout SECONDS
  Timeout for api requests. Default is typically 30 seconds.
  -c, --nocolor
  Disable ANSI coloring
  -B, --benchmark
  Print benchmark time after the command is finished.
  -v, --debug
  Print extra output for debugging.
  -h, --help
  Print this help
17.54.5 morpheus roles update

Usage: morpheus roles update [name] [options]
- --authority VALUE Name
- --description VALUE Description (optional)
- --multitenant on|off Multitenant (optional) Default: off - A Multitenant role is automatically copied into all existing subaccounts as well as placed into a subaccount when created. Useful for providing a set of predefined roles a Customer can use
- --instanceLimits.maxStorage VALUE Max Storage (bytes) (optional)
- --instanceLimits.maxMemory VALUE Max Memory (bytes) (optional)
- --instanceLimits.maxCpu VALUE CPU Count (optional)
- -O, --option OPTION Option in the format -O field="value"
- --prompt Always prompts. Use passed options as the default value.
- -N, --no-prompt Skip prompts. Use default values for all optional fields.
- -j, --json JSON Output Dry Run, print the API request instead of executing it
- -d, --dry-run Dry Run to output API request as a curl command.
- --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- -r, --remote REMOTE Remote name. The current remote is used by default.
- -r, --remote-url URL Remote url. The current remote url is used by default.
- -T, --token TOKEN Access token for authentication with --remote. Saved credentials are used by default.
- -U, --username USERNAME Username for authentication.
- -P, --password PASSWORD Password for authentication.
- -I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- -H, --header HEADER Additional HTTP header to include with requests.
- --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- -C, --nocolor Disable ANSI coloring
- -B, --benchmark Print benchmark time after the command is finished.
- -V, --debug Print extra output for debugging.
- -h, --help Print this help

17.54.6 morpheus roles update-blueprint-access

Usage: morpheus roles update-blueprint-access [name]
- --blueprint ID Blueprint ID or Name
- --all Update all blueprints at once.
- --access VALUE Access value [full|read|none]
- -j, --json JSON Output
- -d, --dry-run Dry Run, print the API request instead of executing it
- --curl Dry Run to output API request as a curl command.
Update role access for an blueprint or all blueprints.

Usage: morpheus roles update-cloud-access [name]

- name is required. This is the name or id of a role.
- --blueprint or --all is required. This is the name or id of a blueprint.
- --access is required. This is the new access value.

### 17.54.7 morpheus roles update-cloud-access
Update role access for a cloud or all clouds. [name] is required. This is the name or id of a role. --cloud or --all is required. This is the name or id of a cloud. --access is required. This is the new access value.

17.54.8 morpheus roles update-feature-access

Usage: morpheus roles update-feature-access [name] [code] [full|read|user|yes|no|none]
   -j, --json JSON Output
   -d, --dry-run Dry Run, print the API request instead of executing it
     --curl Dry Run to output API request as a curl command.
     --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE Remote name. The current remote is used by default.
   --remote-url URL Remote url. The current remote url is used by default.
   -T, --token TOKEN Access token for authentication with --remote.
   -U, --username USERNAME Username for authentication.
   -P, --password PASSWORD Password for authentication.
   --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
   --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
   -C, --nocolor Disable ANSI coloring
   -B, --benchmark Print benchmark time after the command is finished.
   -V, --debug Print extra output for debugging.
   -h, --help Print this help

17.54.9 morpheus roles update-global-blueprint-access

Usage: morpheus roles update-global-blueprint-access [name] [full|custom|none]
   -j, --json JSON Output
   -d, --dry-run Dry Run, print the API request instead of executing it
     --curl Dry Run to output API request as a curl command.
     --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE Remote name. The current remote is used by default.
   --remote-url URL Remote url. The current remote url is used by default.
   -T, --token TOKEN Access token for authentication with --remote.
   -U, --username USERNAME Username for authentication.
17.54.10 morpheus roles update-global-cloud-access

Usage: morpheus roles update-global-cloud-access [name] [full|custom|none]

-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
--default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
--timeout SECONDS
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.54.11 morpheus roles update-global-group-access

Usage: morpheus roles update-global-group-access [name] [code] [full|read|custom|none]

-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--default.
--remote-url URL Remote url. The current remote url is used by default.

-T, --token TOKEN
Saved credentials are used by default.
-U, --username USERNAME
Username for authentication.
-P, --password PASSWORD
Password for authentication.
-I, --insecure
Allow insecure HTTPS communication. i.e. bad
SSL certificate.
-H, --header HEADER
Additional HTTP header to include with requests.
--timeout SECONDS
Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor
Disable ANSI coloring
-B, --benchmark
Print benchmark time after the command is finished.
-V, --help
Print this help

17.54.12 morpheus roles update-global-instance-type-access

Usage: morpheus roles update-global-instance-type-access [name] [full|custom|none]
-j, --json
JSON Output
-d, --dry-run
Dry Run, print the API request instead of executing it
  --curl
  Dry Run to output API request as a curl command.
  --scrub
  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE
Remote name. The current remote is used by default.
-remote-url URL
Remote url. The current remote url is used by default.
-T, --token TOKEN
Saved credentials are used by default.
-U, --username USERNAME
Username for authentication.
-P, --password PASSWORD
Password for authentication.
-I, --insecure
Allow insecure HTTPS communication. i.e. bad
SSL certificate.
-H, --header HEADER
Additional HTTP header to include with requests.
--timeout SECONDS
Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor
Disable ANSI coloring
-B, --benchmark
Print benchmark time after the command is finished.
-V, --help
Print extra output for debugging.
-h, --help
Print this help

17.54.13 morpheus roles update-group-access

Usage: morpheus roles update-group-access [name] [all|group]
-g, --group GROUP
Group name or id
--all
Update all groups at once.
-access VALUE
Access value [full|read|none]
-j, --json
JSON Output
-d, --dry-run
Dry Run, print the API request instead of executing it
  --curl
  Dry Run to output API request as a curl command.
--scrub
   Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE
   Remote name. The current remote is used by default.
-remote-url URL
   Remote url. The current remote url is used by default.
-T, --token TOKEN
   Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME
   Username for authentication.
-P, --password PASSWORD
   Password for authentication.
-I, --insecure
   Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER
   Additional HTTP header to include with requests.
-timeout SECONDS
   Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor
   Disable ANSI coloring
-B, --benchmark
   Print benchmark time after the command is finished.
-V, --debug
   Print extra output for debugging.
-h, --help
   Print this help

Update role access for a group or all groups.
[name] is required. This is the name or id of a role.
--group or --all is required. This is the name or id of a group.
--access is required. This is the new access value.

17.54.14 morpheus roles update-instance-type-access

Usage: morpheus roles update-instance-type-access [name]
   --instance-type INSTANCE_TYPE
     Instance Type name
   --all
     Update all instance types at once.
   -j, --json
     JSON Output
   -d, --dry-run
     Dry Run, print the API request instead of executing it
   --executing it
   --curl
     Dry Run to output API request as a curl command.
   --scrub
     Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE
     Remote name. The current remote is used by default.
   -remote-url URL
     Remote url. The current remote url is used by default.
   --token TOKEN
     Access token for authentication with --remote.
   --Saved credentials are used by default.
   -U, --username USERNAME
     Username for authentication.
   -P, --password PASSWORD
     Password for authentication.
   -I, --insecure
     Allow insecure HTTPS communication. i.e. bad SSL certificate.
   -H, --header HEADER
     Additional HTTP header to include with requests.
   --timeout SECONDS
     Timeout for api requests. Default is typically 30 seconds.
   -C, --nocolor
     Disable ANSI coloring
   -B, --benchmark
     Print benchmark time after the command is finished.
   -h, --help
     Print this help
Update role access for an instance type or all instance types.

[name] is required. This is the name or id of a role.
--instance-type or --all is required. This is the name of an instance type.
--access is required. This is the new access value.

### 17.55 morpheus security-group-rules

[Usage: morpheus security-group-rules [command] [options]]

Commands:
- add-custom-rule
- add-instance-rule
- list
- remove

### 17.55.1 morpheus security-group-rules add-custom-rule

[Usage: morpheus security-group-rules add-custom-rule SOURCE_CIDR PORT_RANGE PROTOCOL
  --[options]]

- SOURCE_CIDR: CIDR to white-list
- PORT_RANGE: Port value (i.e. 123) or port range (i.e. 1-65535)
- PROTOCOL: tcp, udp, icmp
- -s, --secgroup SECGROUP Security Group ID (Use will use security as set
- with 'security-groups use id'
- -j, --json JSON Output
- -d, --dry-run Dry Run, print the API request instead of
- executing it
- --curl Dry Run to output API request as a curl command.
- --scrub Mask secrets in output, such as the
- Authorization header. For use with --curl and --dry-run.
- -r, --remote REMOTE Remote name. The current remote is used by
- default.
- --remote-url URL Remote url. The current remote url is used by
- default.
- -T, --token TOKEN Access token for authentication with --remote.
- Saved credentials are used by default.
- -U, --username USERNAME Username for authentication.
- -P, --password PASSWORD Password for authentication.
- -I, --insecure Allow insecure HTTPS communication. i.e. bad
- SSL certificate.
- -H, --header HEADER Additional HTTP header to include with requests.
- --timeout SECONDS Timeout for api requests. Default is typically
- 30 seconds.
- -C, --nocolor Disable ANSI coloring
- -B, --benchmark Print benchmark time after the command is
- finished.
- -V, --debug Print extra output for debugging.
- -h, --help Print this help
17.55.2 `morpheus security-group-rules add-instance-rule`

Usage: morpheus security-group-rules add_instance_rule SOURCE_CIDR INSTANCE_TYPE_ID

- [options]
  - [options]
    - SOURCE_CIDR: CIDR to white-list
    - INSTANCE_TYPE_ID: ID of the Instance Type to access
    - -s, --secgroup secgroup Security Group ID (Use will use security as set
    - with 'security-groups use id'
    - -j, --json JSON Output
    - -d, --dry-run Dry Run, print the API request instead of
    - --executing it
    - --curl Dry Run to output API request as a curl command.
    - --scrub Mask secrets in output, such as the
    - Authorization header. For use with --curl and --dry-run.
    - -r, --remote REMOTE Remote name. The current remote is used by
    - -default.
    - --remote-url URL Remote url. The current remote url is used by
    - -default.
    - -T, --token TOKEN Access token for authentication with --remote.
  - Saved credentials are used by default.
    - -U, --username USERNAME Username for authentication.
    - -P, --password PASSWORD Password for authentication.
    - -I, --insecure Allow insecure HTTPS communication. i.e. bad
    - SSL certificate.
    - -H, --header HEADER Additional HTTP header to include with requests.
    - --timeout SECONDS Timeout for api requests. Default is typically
    - 30 seconds.
    - -C, --nocolor Disable ANSI coloring
    - -B, --benchmark Print benchmark time after the command is
    - finished.
    - -V, --debug Print extra output for debugging.
    - -h, --help Print this help

17.55.3 `morpheus security-group-rules list`

Usage: morpheus security-group-rules list [id]

- [options]
  - [options]
    - -j, --json JSON Output
    - -d, --dry-run Dry Run, print the API request instead of
    - --executing it
    - --curl Dry Run to output API request as a curl command.
    - --scrub Mask secrets in output, such as the
    - Authorization header. For use with --curl and --dry-run.
    - -r, --remote REMOTE Remote name. The current remote is used by
    - -default.
    - --remote-url URL Remote url. The current remote url is used by
    - -default.
    - -T, --token TOKEN Access token for authentication with --remote.
  - Saved credentials are used by default.
    - -U, --username USERNAME Username for authentication.
    - -P, --password PASSWORD Password for authentication.
    - -I, --insecure Allow insecure HTTPS communication. i.e. bad
    - SSL certificate.
    - -H, --header HEADER Additional HTTP header to include with requests.
    - --timeout SECONDS Timeout for api requests. Default is typically
    - 30 seconds.
17.55.4 morpheus security-group-rules remove

Usage: morpheus security-group-rules remove [id] [options]

- `-s`, `--secgroup secgroup` Security Group ID (Use will use security as set
  with 'security-groups use id'
- `-j`, `--json` JSON Output
- `-d`, `--dry-run` Dry Run, print the API request instead of.
- `-c`, `--curl` Dry Run to output API request as a curl command.
- `-r`, `--remote REMOTE` Remote name. The current remote is used by
  default.
- `-u`, `--remote-url URL` Remote url. The current remote url is used by
  default.
- `-t`, `--token TOKEN` Access token for authentication with --remote.
- `-U`, `--username USERNAME` Username for authentication.
- `-P`, `--password PASSWORD` Password for authentication.
- `-I`, `--insecure` Allow insecure HTTPS communication. i.e. bad
  SSL certificate.
- `-H`, `--header HEADER` Additional HTTP header to include with requests.
- `-c`, `--nocolor` Disable ANSI coloring
- `-B`, `--benchmark` Print benchmark time after the command is
  finished.
- `-V`, `--debug` Print extra output for debugging.
- `-h`, `--help` Print this help

17.56 morpheus security-groups

Usage: morpheus security-groups [command] [options]
Commands:
  add
get
list
remove
unuse
use
17.56.1 morpheus security-groups add

Usage: morpheus security-groups add [name] [options]

- `--description Description` Description of the security group
- `-j, --json` JSON Output
- `-d, --dry-run` Dry Run, print the API request instead of executing it
  - `--curl` Dry Run to output API request as a curl command.
  - `--scrub` Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- `-C, --nocolor` Disable ANSI coloring
- `-B, --benchmark` Print benchmark time after the command is finished.
- `-V, --debug` Print extra output for debugging.
- `-h, --help` Print this help

17.56.2 morpheus security-groups get

Usage: morpheus security-groups get [id]

- `-j, --json` JSON Output
- `-d, --dry-run` Dry Run, print the API request instead of executing it
  - `--curl` Dry Run to output API request as a curl command.
  - `--scrub` Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- `-C, --nocolor` Disable ANSI coloring
- `-B, --benchmark` Print benchmark time after the command is finished.
- `-V, --debug` Print extra output for debugging.
- `-h, --help` Print this help

17.56.3 morpheus security-groups list

Usage: morpheus security-groups list

- `-m, --max MAX` Max Results
- `-o, --offset OFFSET` Offset Results
- `-s, --search PHRASE` Search Phrase
- `-S, --sort ORDER` Sort Order
- `-D, --desc` Reverse Sort Order
- `-Q, --query PARAMS` Query parameters. PARAMS format is 'phrase=foobar&category=web'
- `-j, --json` JSON Output
- `--yaml` YAML Output
- `--csv` CSV Output
  - `--csv-delim CHAR` Delimiter for CSV Output values. Default: ','
  - `--csv-newline [CHAR]` Delimiter for CSV Output rows. Default: '\n'
  - `--csv-quotes` Wrap CSV values with "". Default: false
  - `--csv-no-header` Exclude header for CSV Output.
  - `-F, --fields x,y,z` Filter Output to a limited set of fields. Default is all fields.
- `-d, --dry-run` Dry Run, print the API request instead of executing it
  - `--curl` Dry Run to output API request as a curl command.

(continues on next page)
17.56.4 morpheus security-groups remove

Usage: morpheus security-groups remove [id]
  -j, --json               JSON Output
  -d, --dry-run            Dry Run, print the API request instead of executing it
                           --curl
                           --scrub                Dry Run to output API request as a curl command.
  --authorization header. For use with --curl and --dry-run.
  -C, --nocolor            Disable ANSI coloring
  -B, --benchmark          Print benchmark time after the command is finished.
  -V, --debug              Print extra output for debugging.
  -h, --help               Print this help

17.56.5 morpheus security-groups unuse

Usage: morpheus security-groups use [id] [--none]
  --none                  Do not use an active group.
  -C, --nocolor           Disable ANSI coloring
  -B, --benchmark         Print benchmark time after the command is finished.
  -V, --debug             Print extra output for debugging.
  -h, --help              Print this help
17.57 morpheus shell

Usage: morpheus shell
-e, --exec EXPRESSION  Execute the command(s) expression and exit.
     --norc            Do not read and execute the personal
                     initialization script .morpheusrc
-I, --insecure        Allow for insecure HTTPS communication i.e. bad
-Z, --incognito      Incognito mode. Use a temporary shell. Remotes
                     are loaded without without saved credentials or history logging.
-C, --nocolor        Disable ANSI coloring
-V, --debug          Print extra output for debugging.
-B, --benchmark      Print benchmark time after each command is
                     finished, including shell itself.
-h, --help           Print this help

17.58 morpheus storage-buckets

Usage: morpheus storage-buckets [command] [options]
Commands:
    add
    download
    get
    list
    list-files
    ls
    read
    remove
    remove-file
    rm
    update
    upload

17.58.1 morpheus storage-buckets add

Usage: morpheus storage-buckets add
     --name VALUE          Name for this storage bucket
     --type code VALUE     Storage Bucket Type Code
     --bucket-name VALUE   Bucket Name
     --default-backup-target [on|off] Default Backup Target
     --default-deployment-target [on|off] Default Deployment Target

(continues on next page)
--default-virtual-image-target [on|off]  Default Virtual Image Store
--copy-to-store [on|off]  Archive Snapshots
-O, --option OPTION  Option in the format -O field="value"
--prompt  Always prompts. Use passed options as the default value.
-N, --no-prompt  Skip prompts. Use default values for all optional fields.
--optional-fields.  Payload from a local JSON or YAML file, skip all prompting
--payload FILE  Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-dir DIRECTORY  Payload JSON, skip all prompting
--payload-json JSON  Payload YAML, skip all prompting
-j, --json  JSON Output
-d, --dry-run  Dry Run, print the API request instead of executing it
--curl  Dry Run to output API request as a curl command.
--scrub  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-q, --quiet  No Output, do not print to stdout
-r, --remote REMOTE  Remote name. The current remote is used by default.
--remote-url URL  Remote url. The current remote url is used by default.
-T, --token TOKEN  Access token for authentication with --remote.
-U, --username USERNAME  Username for authentication.
-P, --password PASSWORD  Password for authentication.
-I, --insecure  Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER  Additional HTTP header to include with requests.
--timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor  Disable ANSI coloring
-B, --benchmark  Print benchmark time after the command is finished.
-V, --debug  Print extra output for debugging.
-h, --help  Print this help

Create a new storage bucket.
[name] is required and can be passed as --name instead.

17.58.2 morpheus storage-buckets download

Usage: morpheus storage-buckets download [provider:/path] [local-file]
-f, --force  Overwrite existing [local-file] if it exists.
-p, --mkdir  Create missing directories for [local-file] if they do not exist.
-d, --dry-run  Dry Run, print the API request instead of executing it
--curl  Dry Run to output API request as a curl command.
--scrub  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-q, --quiet  No Output, do not print to stdout
-C, --nocolor	Disable ANSI coloring
-B, --benchmark	Print benchmark time after the command is finished.
-V, --debug	Print extra output for debugging.
-h, --help	Print this help

Download a file or directory.
[provider:/path] is required. This is the name or id of the provider and /path the file or folder to be downloaded.
[local-file] is required. This is the full local filepath for the downloaded file. Directories will be downloaded as a .zip file, so you'll want to specify a [local-file] with a .zip extension.

17.58.3 morpheus storage-buckets get

Usage: morpheus storage-buckets get [storage-bucket]
  -j, --json	JSON Output
  --yaml	YAML Output
  --csv	CSV Output
  --csv-delim CHAR	Delimiter for CSV Output values. Default: ','
  --csv-newline [CHAR]	Delimiter for CSV Output rows. Default: '
'
  --csv-quotes Wrap CSV values with "". Default: false
  --csv-no-header	Exclude header for CSV Output.
  -F, --fields x,y,z	Filter Output to a limited set of fields. Default is all fields.
  -d, --dry-run	Dry Run, print the API request instead of executing it
  --curl	Dry Run to output API request as a curl command.
  --scrub	Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -r, --remote REMOTE	Remote name. The current remote is used by default.
  --remote-url URL	Remote url. The current remote url is used by default.
  -T, --token TOKEN	Access token for authentication with --remote.
  -U, --username USERNAME	Username for authentication.
  -P, --password PASSWORD	Password for authentication.
  -I, --insecure	Allow insecure HTTPS communication. i.e. bad SSL certificate.
  -H, --header HEADER	Additional HTTP header to include with requests.
  --timeout SECONDS	Timeout for api requests. Default is typically 30 seconds.
  -C, --nocolor	Disable ANSI coloring
  -B, --benchmark	Print benchmark time after the command is finished.
  -V, --debug	Print extra output for debugging.
  -h, --help	Print this help

Get details about a storage bucket.
[storage-bucket] is required. This is the name or id of a storage bucket.
17.58.4 morpheus storage-buckets list

Usage: morpheus storage-buckets list

    -m, --max MAX  Max Results
    -o, --offset OFFSET  Offset Results
    -s, --search PHRASE  Search Phrase
    -S, --sort ORDER  Sort Order
    -D, --desc  Reverse Sort Order
    -y, --yaml YAML Output
    --csv  CSV Output
    --csv-delim CHAR  Delimiter for CSV Output values. Default: ', '
    --csv-newline [CHAR]  Delimiter for CSV Output rows. Default: '\n'
    --csv-quotes  Wrap CSV values with '"'. Default: false
    --csv-no-header  Exclude header for CSV Output.
    -F, --fields x,y,z  Filter Output to a limited set of fields.
    -j, --json JSON Output
    -d, --dry-run  Dry Run, print the API request instead of executing it
    --curl  Dry Run to output API request as a curl command.
    --scrub  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
    -r, --remote REMOTE  Remote name. The current remote is used by default.
    --remote-url URL  Remote url. The current remote url is used by default.
    -T, --token TOKEN  Access token for authentication with --remote.
    -U, --username USERNAME  Username for authentication.
    -P, --password PASSWORD  Password for authentication.
    -I, --insecure  Allow insecure HTTPS communication. i.e. bad SSL certificate.
    --header HEADER  Additional HTTP header to include with requests.
    --timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
    -C, --nocolor  Disable ANSI coloring
    -b, --benchmark  Print benchmark time after the command is finished.
    -v, --debug  Print extra output for debugging.
    -h, --help  Print this help

List storage buckets.

17.58.5 morpheus storage-buckets list-files

Usage: morpheus storage-buckets list-files [provider:/path]

    -a, --all  Show all files, including subdirectories under the /path.
    -m, --max MAX  Max Results
    -o, --offset OFFSET  Offset Results
    -s, --search PHRASE  Search Phrase
    -S, --sort ORDER  Sort Order
    -D, --desc  Reverse Sort Order
    -Q, --query PARAMS  Query parameters. PARAMS format is 'phrase=foobar&category=web'

(continues on next page)
17.58.6 morpheus storage-buckets ls

Usage: morpheus storage-buckets ls [bucket/path]

-a, --all Show all files, including subdirectories under.
-t, --the /path.
-l, --long Lists files in the long format, which contains.
 lots of useful information, e.g. the exact size of the file, the file type, and.
 when it was last modified.
-H, --human Humanized file sizes. The default is just the.
-l, --oneline One file per line. The default delimiter is a.
single space.
-m, --max MAX Max Results
-o, --offset OFFSET Offset Results
-s, --search PHRASE Search Phrase
-S, --sort ORDER Sort Order
-D, --desc Reverse Sort Order
-j, --json JSON Output
-F, --fields x,y,z Filter Output to a limited set of fields.
-d, --dry-run Dry Run, print the API request instead of.
-executing it
--curl
--scrub

Authorization header. For use with --curl and --dry-run.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is.
-finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Print filenames for a given location.
Pass storage location in the format bucket/path.

17.58.7 morpheus storage-buckets read

Usage: morpheus storage-buckets read [provider:/path]
  -d, --dry-run          Dry Run, print the API request instead of executing it
  --curl                 Dry Run to output API request as a curl command.
  --scrub                Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -C, --nocolor          Disable ANSI coloring
  -B, --benchmark        Print benchmark time after the command is finished.
  -V, --debug            Print extra output for debugging.
  -h, --help             Print this help

Print the contents of a storage file.
[provider:/path] is required. This is the name or id of the provider and /path the file or folder to be downloaded.
This confirmation can be skipped with the -y option.

17.58.8 morpheus storage-buckets remove

Usage: morpheus storage-buckets remove [storage-bucket]
  -y, --yes               Auto Confirm
  -j, --json              JSON Output
  -d, --dry-run           Dry Run, print the API request instead of executing it
  --curl                 Dry Run to output API request as a curl command.
  --scrub                Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -r, --remote REMOTE    Remote name. The current remote is used by default.
  -t, --token TOKEN      Access token for authentication with --remote.
  -U, --username USERNAME Username for authentication.
  -P, --password PASSWORD Password for authentication.
  -I, --insecure         Allow insecure HTTPS communication. i.e. bad SSL certificate.
  -H, --header HEADER    Additional HTTP header to include with requests.
  -T, --timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
  -C, --nocolor          Disable ANSI coloring
  -B, --benchmark        Print benchmark time after the command is finished.
  -V, --debug            Print extra output for debugging.
  -h, --help             Print this help

Delete a storage bucket.
[storage-bucket] is required. This is the name or id of a storage bucket.
17.58.9 morpheus storage-buckets remove-file

Usage: morpheus storage-buckets remove-file [provider:/path]

- R, --recursive Delete a directory and all of its files. This must be passed if specifying a directory.
- y, --yes Auto Confirm
- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of executing it
- curl
- scrub
- Authorization header. For use with curl and dry-run.
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

Delete a storage file or directory.

17.58.10 morpheus storage-buckets rm

Usage: morpheus storage-buckets rm [provider:/path]

- R, --recursive Delete a directory and all of its files. This must be passed if specifying a directory.
- y, --yes Auto Confirm
- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of executing it
- curl
- scrub
- Authorization header. For use with curl and dry-run.
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

Delete a storage file or directory.

17.58.11 morpheus storage-buckets update

Usage: morpheus storage-buckets update [storage-bucket] [options]

--name VALUE Name for this storage bucket
--type code Storage Bucket Type Code
--bucket-name VALUE Bucket Name
--default-backup-target [on|off] Default Backup Target
--default-deployment-target [on|off] Default Deployment Target
--default-virtual-image-target [on|off] Default Virtual Image Store
--copy-to-store [on|off] Archive Snapshots

(continues on next page)
17.58.12 morpheus storage-buckets upload

Usage: morpheus storage-buckets upload [local-file] [provider:/path]

-R, --recursive          Upload a directory and all of its files. This
--must be passed if [local-file] is a directory.
--ignore-files PATTERN   Pattern of files to be ignored when uploading a
--directory.
-y, --yes                Auto Confirm
-j, --json               JSON Output
-d, --dry-run            Dry Run, print the API request instead of
--executing it
--curl                   Dry Run to output API request as a curl command.
--scrub                  Mask secrets in output, such as the
--Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE      Remote name. The current remote is used by
--default.
--remote-url URL         Remote url. The current remote url is used by
--default.
-t, --token TOKEN        Access token for authentication with --remote.
--Saved credentials are used by default.
-u, --username USERNAME  Username for authentication.
-p, --password PASSWORD  Password for authentication.
-i, --insecure           Allow insecure HTTPS communication. i.e. bad
--SSL certificate.
-h, --header HEADER      Additional HTTP header to include with requests.
--timeout SECONDS        Timeout for api requests. Default is typically
30 seconds.
-c, --nocolor            Disable ANSI coloring
-b, --benchmark          Print benchmark time after the command is
--finished.
-v, --debug              Print extra output for debugging.
-h, --help               Print this help

Update a storage bucket.
[storage-bucket] is required. This is the id of a storage bucket.
Upload a local file or folder to a storage bucket.
The first argument [local-file] should be the path of a local file or directory.
The second argument [provider:/path] should contain the name or id of the provider.
The [:/path] component is optional and can be used to specify the destination of the
uploaded file or folder.
The default destination is the same name as the [local-file], under the root
directory '/'.
This will overwrite any existing remote files that match the destination /path.

17.59 morpheus tasks

Usage: morpheus tasks [command] [options]
Commands:
  add
  get
  list
  remove
  types
  update

17.59.1 morpheus tasks add

Usage: morpheus tasks add [name] -t TASK_TYPE
  -t, --type TASK_TYPE Task Type
  --name NAME Task Name
  --file FILE File containing the script. This can be used
    instead of --O taskOptions.script
  -O, --option OPTION Option in the format -O field="value"
    Always prompts. Use passed options as the
    default value.
  -N, --no-prompt Skip prompts. Use default values for all
    optional fields.
  --payload FILE Payload from a local JSON or YAML file, skip all
    prompting
  --payload-dir DIRECTORY Payload from a local directory containing 1-N
    JSON or YAML files, skip all prompting
  --payload-json JSON Payload JSON, skip all prompting
  --payload-yaml YAML Payload YAML, skip all prompting
  -j, --json JSON Output
  -d, --dry-run Dry Run, print the API request instead of
    executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the
    Authorization header. For use with --curl and --dry-run.
  -q, --quiet No Output, do not print to stdout
  -r, --remote REMOTE Remote name. The current remote is used by
    default.
  --remote-url URL Remote url. The current remote url is used by
    default.
-T, --token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad
-SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
-timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.59.2 morpheus tasks get

Usage: morpheus tasks get [workflow]
- j, --json JSON Output
- y, --yaml YAML Output
- csv CSV Output
- csv-delim CHAR Delimiter for CSV Output values. Default: ','
- csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
- csv-quotes Wrap CSV values with '"'. Default: false
- csv-no-header Exclude header for CSV Output.
- F, --fields x,y,z Filter Output to a limited set of fields. Default is all fields.
- d, --dry-run Dry Run, print the API request instead of executing it
- curl Dry Run to output API request as a curl command.
- scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by default.
- remote-url URL Remote url. The current remote url is used by default.
- T, --token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad
-SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
-timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help
17.59.3 morpheus tasks list

Usage: morpheus tasks list

- --types x,y,z Filter by task type code(s)
- -m, --max MAX Max Results
- -o, --offset OFFSET Offset Results
- -s, --search PHRASE Search Phrase
- -S, --sort ORDER Sort Order
- -D, --desc Reverse Sort Order
- -Q, --query PARAMS Query parameters. PARAMS format is
  'phrase=foobar&category=web'
- -j, --json JSON Output
- --yaml YAML Output
- -csv CSV Output
- --csv-delim CHAR Delimiter for CSV Output values. Default: ','
- --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
- --csv-quotes Wrap CSV values with ''. Default: false
- --csv-no-header Exclude header for CSV Output.
- -F, --fields x,y,z Filter Output to a limited set of fields.
  Default is all fields.
- -d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header.
- -r, --remote REMOTE Remote name. The current remote is used by default.
  --remote-url URL Remote url. The current remote url is used by default.
- -t, --token TOKEN Access token for authentication with --remote.
  Saved credentials are used by default.
- -U, --username USERNAME Username for authentication.
- -P, --password PASSWORD Password for authentication.
- -i, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- -H, --header HEADER Additional HTTP header to include with requests.
  --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- -C, --nocolor Disable ANSI coloring
- -B, --benchmark Print benchmark time after the command is finished.
- -V, --debug Print extra output for debugging.
- -h, --help Print this help

17.59.4 morpheus tasks remove

Usage: morpheus tasks remove [task]

- -y, --yes Auto Confirm
- -j, --json JSON Output
- -d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header.
- -q, --quiet No Output, do not print to stdout
17.59.5 morpheus tasks types

Usage: morpheus tasks types

- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of 
- executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the
- Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by 
- default.
  --remote-url URL Remote url. The current remote url is used by 
- default.
- T, --token TOKEN Access token for authentication with --remote.
- Saved credentials are used by default.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. i.e. bad
- SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
  --timeout SECONDS Timeout for api requests. Default is typically
- 30 seconds.
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is
- finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help
- f, --force Force Delete

17.59.6 morpheus tasks update

Usage: morpheus tasks update [task] [options]

- O, --option OPTION Option in the format -O field="value"

(continues on next page)
--prompt Always prompts. Use passed options as the default value.
-N, --no-prompt Skip prompts. Use default values for all optional fields.
--optional fields.
--json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
--token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
--insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
-timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
--nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

17.60 morpheus tenants

Usage: morpheus tenants [command] [options]
Commands:
add
count
get
groups
list
remove
update

17.60.1 morpheus tenants add

Usage: morpheus tenants add [options]
   --name VALUE Name
   --description VALUE Description (optional)
   --role VALUE Base Role (optional)
   --currency VALUE Currency (optional)
   --option OPTION Option in the format -O field="value"
   --prompt Always prompts. Use passed options as the default value.
   -N, --no-prompt Skip prompts. Use default values for all optional fields.
   --optional fields.
Get the number of tenants.

Usage: morpheus tenants count [options]
    -Q, --query PARAMS Query parameters. PARAMS format is
        'phrase=foobar&category=web'
    -r, --remote REMOTE Remote name. The current remote is used by default.
        --remote-url URL Remote url. The current remote url is used by default.
    -T, --token TOKEN Access token for authentication with --remote.
    Saved credentials are used by default.
    -U, --username USERNAME Username for authentication.
    -P, --password PASSWORD Password for authentication.
    -I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
    -H, --header HEADER Additional HTTP header to include with requests.
        --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
    -d, --dry-run Dry Run, print the API request instead of executing it
        --curl Dry Run to output API request as a curl command.
        --scrub Mask secrets in output, such as the Authorization header. For use with
        --curl and --dry-run.
    -C, --nocolor Disable ANSI coloring
    -B, --benchmark Print benchmark time after the command is finished.
    -V, --debug Print extra output for debugging.
    -h, --help Print this help

17.60.2 morpheus tenants count

Usage: morpheus tenants count [options]
17.60.3 morpheus tenants get

Usage: morpheus tenants get [name]

- j, --json JSON Output
- r, --remote REMOTE Remote name. The current remote is used by default.
- default.
- remote-url URL Remote url. The current remote url is used by default.
- default.
- T, --token TOKEN Access token for authentication with --remote.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. I.e. bad SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
- timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- d, --dry-run Dry Run, print the API request instead of executing it
- --curl Dry Run to output API request as a curl command.
- --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

17.60.4 morpheus tenants groups

Usage: morpheus tenants groups [command] [options]

Commands:
  add
  add-cloud
  get
  list
  remove
  remove-cloud
  update

17.60.5 morpheus tenants list

Usage: morpheus tenants list

- m, --max MAX Max Results
- o, --offset OFFSET Offset Results
- s, --search PHRASE Search Phrase
- S, --sort ORDER Sort Order
- D, --desc Reverse Sort Order
- Q, --query PARAMS Query parameters. PARAMS format is 'phrase=foobar&category=web'
- j, --json JSON Output
- r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL
-remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN
-token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME
--username USERNAME Username for authentication.
-P, --password PASSWORD
-password PASSWORD Password for authentication.
-I, --insecure
-insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER
--header HEADER Additional HTTP header to include with requests.
--timeout SECONDS
-timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-d, --dry-run
-dry-run Dry Run, print the API request instead of executing it.
--curl
-curl Dry Run to output API request as a curl command.
--scrub
-scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
--no-color
--no-color Disable ANSI coloring
-B, --benchmark
--benchmark Print benchmark time after the command is finished.
-V, --debug
--debug Print extra output for debugging.
-h, --help
-help Print this help

List tenants (accounts).

17.60.6 morpheus tenants remove

Usage: morpheus tenants remove [name]

- y, --yes
- j, --json
- r, --remote REMOTE
- default.
-remote-url URL
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN
-token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME
--username USERNAME Username for authentication.
-P, --password PASSWORD
-password PASSWORD Password for authentication.
-I, --insecure
-insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER
--header HEADER Additional HTTP header to include with requests.
--timeout SECONDS
-timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-d, --dry-run
-dry-run Dry Run, print the API request instead of executing it.
--curl
-curl Dry Run to output API request as a curl command.
--scrub
-scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
--no-color
--no-color Disable ANSI coloring
-B, --benchmark
--benchmark Print benchmark time after the command is finished.
-V, --debug
--debug Print extra output for debugging.
-h, --help
-help Print this help
### 17.60.7 morpheus tenants update

**Usage:** morpheus tenants update [name] [options]

- `--active [on|off]` Can be used to disable a network
- `--name VALUE` Name
- `--description VALUE` Description (optional)
- `--role VALUE` Base Role (optional)
- `--currency VALUE` Currency (optional)
- `-O, --option OPTION` Option in the format `-O field="value"
- `--prompt` Always prompts. Use passed options as the default value.
- `-W, --no-prompt` Skip prompts. Use default values for all optional fields.
- `-j, --json` JSON Output
- `-r, --remote REMOTE` Remote name. The current remote is used by default.
- `-O, --option OPTION` Option in the format `-O field="value"
- `--default` Remote url. The current remote url is used by default.
- `--remote-url URL` Remote url. The current remote url is used by default.
- `-T, --token TOKEN` Access token for authentication with --remote.
- `-U, --username USERNAME` Username for authentication.
- `-P, --password PASSWORD` Password for authentication.
- `-I, --insecure` Allow insecure HTTPS communication. i.e. bad SSL certificate.
- `-H, --header HEADER` Additional HTTP header to include with requests.
- `-t, --timeout SECONDS` Timeout for api requests. Default is typically 30 seconds.
- `-d, --dry-run` Dry Run, print the API request instead of executing it
- `--curl` Dry Run to output API request as a curl command.
- `--scrub` Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- `-C, --nocolor` Disable ANSI coloring
- `-B, --benchmark` Print benchmark time after the command is finished.
- `-V, --debug` Print extra output for debugging.
- `-h, --help` Print this help

### 17.61 morpheus user-groups

**Usage:** morpheus user-groups [command] [options]

**Commands:**

- add
- add-user
- get
- list
- remove
- remove-user
- update
17.61.1 morpheus user-groups add

Usage: morpheus user-groups add [name]

  --name VALUE   Name
  --description VALUE   Description
  --sudoUser [on|off]   Sudo Access
  --serverGroup VALUE   Server Group
  --users LIST   Users to include in this group, comma separated
  -O, --option OPTION   Option in the format -O field="value"
  --prompt   Always prompts. Use passed options as the default value.
  -N, --no-prompt   Skip prompts. Use default values for all optional fields.
  --payload FILE   Payload from a local JSON or YAML file, skip all prompting
  --payload-dir DIRECTORY   Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
  --payload-json JSON   Payload JSON, skip all prompting
  --payload-yaml YAML   Payload YAML, skip all prompting
  -j, --json JSON Output
  -d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
  -r, --remote REMOTE   Remote name. The current remote is used by default.
  --remote-url URL   Remote url. The current remote url is used by default.
  -T, --token TOKEN   Access token for authentication with --remote.
  --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
  --header HEADER Additional HTTP header to include with requests.
  --timeout SECONDS   Timeout for api requests. Default is typically 30 seconds.
  -q, --quiet No Output, do not print to stdout
  -C, --nocolor Disable ANSI coloring
  -B, --benchmark Print benchmark time after the command is finished.
  -V, --debug Print extra output for debugging.
  -h, --help Print this help

Create a new user group.
[name] is required and can be passed as --name instead.

17.61.2 morpheus user-groups add-user

Usage: morpheus user-groups add-user [name] [user]

  -j, --json JSON Output
  -d, --dry-run Dry Run, print the API request instead of executing it

(continues on next page)
Add a user to a user group. 

[name] is required. This is the name or id of a user group. 

[user] is required. This is the username or id of a user. More than one can be passed.

17.61.3 morpheus user-groups get

Usage: morpheus user-groups get [name]

-j, --json JSON Output 
--yaml YAML Output 
-csv CSV Output 
--csv-delim CHAR Delimiter for CSV Output values. Default: ',' 
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
--csv-quotes Wrap CSV values with ". Default: false 
--csv-no-header Exclude header for CSV Output. 
-F, --fields x,y,z Filter Output to a limited set of fields. Default is all fields. 
-d, --dry-run Dry Run, print the API request instead of executing it 
--curl Dry Run to output API request as a curl command. 
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run. 
-r, --remote REMOTE Remote name. The current remote is used by default. 
--remote-url URL Remote url. The current remote url is used by default. 
-T, --token TOKEN Access token for authentication with --remote. Saved credentials are used by default. 
-U, --username USERNAME Username for authentication. 
-P, --password PASSWORD Password for authentication. 
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate. 
-H, --header HEADER Additional HTTP header to include with requests. 
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds. 
-g, --quiet No Output, do not print to stdout 
-C, --nocolor Disable ANSI coloring 
-B, --benchmark Print benchmark time after the command is finished. 
-V, --debug Print extra output for debugging. 
-h, --help Print this help

(continues on next page)
17.61.4 morpheus user-groups list

Usage: morpheus user-groups list

-m, --max MAX Max Results
-s, --offset OFFSET Offset Results
-s, --search PHRASE Search Phrase
-S, --sort ORDER Sort Order
-B, --desc Reverse Sort Order
-j, --json JSON Output
-yaml YAML Output
--csv CSV Output
--csv-delim CHAR Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
--csv-quotes Wrap CSV values with "". Default: false
--csv-no-header Exclude header for CSV Output.
-F, --fields x,y,z Filter Output to a limited set of fields.
-D, --desc Reverse Sort Order
-F, --fields x,y,z Filter Output to a limited set of fields.
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-S, --fields x,y,z Filter Output to a limited set of fields.
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-S, --fields x,y,z Filter Output to a limited of fields.
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-S, --fields x,y,z Filter Output to a limited set of fields.
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-S, --fields x,y,z Filter Output to a limited set of fields.
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-S, --fields x,y,z Filter Output to a limited set of fields.
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-S, --fields x,y,z Filter Output to a limited set of fields.
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
-d, --dry-run
Dry Run, print the API request instead of executing it
--curl
Dry Run to output API request as a curl command.
--scrub
Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-g, --quiet
No Output, do not print to stdout
-y, --yes
Auto Confirm
-C, --nocolor
Disable ANSI coloring
-B, --benchmark
Print benchmark time after the command is finished.
-V, --debug
Print extra output for debugging.
-h, --help
Print this help

17.61.6 morpheus user-groups remove-user

Usage: morpheus user-groups remove-user [name] [user]
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl
Dry Run to output API request as a curl command.
--scrub
Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote. Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests. Timeout for api requests. Default is typically 30 seconds.
-q, --quiet No Output, do not print to stdout
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Remove a user from a user group.
[name] is required. This is the name or id of a user group.
[user] is required. This is the username or id of a user. More than one can be passed.

17.61.7 morpheus user-groups update

Usage: morpheus user-groups update [name]
--name VALUE Name for this user group
--description VALUE Description
--sudoUser [on|off] Sudo Access. Default is off.

(continues on next page)
Update a user group.

[name] is required. This is the name or id of a user group.

### 17.62 morpheus user-settings

Usage: morpheus user-settings [command] [options]

Commands:
- clear-access-token
- get
- list-clients
- regenerate-access-token
- update
- update-avatar
- view-avatar
17.62.1 morpheus user-settings clear-access-token

Usage: morpheus user-settings clear-access-token [client-id]
    --payload FILE            Payload from a local JSON or YAML file, skip all prompting
    --payload-dir DIRECTORY   Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
    --payload-json JSON       Payload JSON, skip all prompting
    --payload-yaml YAML       Payload YAML, skip all prompting
    -O, --option OPTION       Option in the format -O field="value"
    --prompt                  Always prompts. Use passed options as the default value.
    -N, --no-prompt           Skip prompts. Use default values for all optional fields.
    -j, --json                JSON Output
    -d, --dry-run             Dry Run, print the API request instead of executing it
    --curl                    Dry Run to output API request as a curl command.
    --scrub                   Mask secrets in output, such as the Authorization header. Use with --curl and --dry-run.
    -q, --quiet               No Output, do not print to stdout
    -r, --remote REMOTE       Remote name. The current remote is used by default.
    --remote-url URL          Remote url. The current remote url is used by default.
    -T, --token TOKEN         Access token for authentication with --remote.
    -U, --username USERNAME   Username for authentication.
    -P, --password PASSWORD   Password for authentication.
    -I, --insecure            Allow insecure HTTPS communication. i.e. bad SSL certificate.
    -H, --header HEADER       Additional HTTP header to include with requests.
    --timeout SECONDS         Timeout for api requests. Default is typically 30 seconds.
    -C, --nocolor             Disable ANSI coloring
    -B, --benchmark           Print benchmark time after the command is finished.
    -V, --debug               Print extra output for debugging.
    -h, --help                Print this help

Clear API access token for a specific client. [client-id] is required. This is the id of an api client.

17.62.2 morpheus user-settings get

Usage: morpheus user-settings get
    -Q, --query PARAMS        Query parameters. PARAMS format is 'phrase=foobar&category=web'
    -j, --json                JSON Output
    -y, --yaml                YAML Output
    -c, --csv                 CSV Output
    --csv-delim CHAR          Delimiter for CSV Output values. Default: ','
    --csv-newline [CHAR]     Delimiter for CSV Output rows. Default: '\n'
    --csv-quotes              Wrap CSV values with "". Default: false
    --csv-no-header           Exclude header for CSV Output.

(continues on next page)
-F, --fields x,y,z  Filter Output to a limited set of fields.
-d, --dry-run  Dry Run, print the API request instead of executing it.
--curl  Dry Run to output API request as a curl command.
--scrub  Mask secrets in output, such as the Authorization header.
--remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Get your user settings.

17.62.3 morpheus user-settings list-clients

Usage: morpheus user-settings list-clients
-Q, --query PARAMS  Query parameters. PARAMS format is 'phrase=foobar&category=web'
-j, --json  JSON Output
--yaml  YAML Output
--csv  CSV Output
--csv-delim CHAR  Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR]  Delimiter for CSV Output rows. Default: '\n'
--csv-quotes  Wrap CSV values with '. Default: false
--csv-no-header  Exclude header for CSV Output.
-F, --fields x,y,z  Filter Output to a limited set of fields.
-d, --dry-run  Dry Run, print the API request instead of executing it.
--curl  Dry Run to output API request as a curl command.
--scrub  Mask secrets in output, such as the Authorization header.
--remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.

(continues on next page)
Regenerate API access token for a specific client.

[client-id] is required. This is the id of an api client.

17.62.4 morpheus user-settings regenerate-access-token

Usage: morpheus user-settings regenerate-access-token [client-id]

--payload FILE Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting
-O, --option OPTION Option in the format -O field="value"
--prompt Always prompts. Use passed options as the default value.
-N, --no-prompt Skip prompts. Use default values for all optional fields.
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of executing it
--curl Dry Run to output API request as a curl command.
--scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-q, --quiet No Output, do not print to stdout
-r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
-T, --token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help
17.62.5 morpheus user-settings update

Usage: morpheus user-settings update [options]
- --payload FILE  Payload from a local JSON or YAML file, skip all
  prompting
- --payload-dir DIRECTORY Payload from a local directory containing 1-N
  JSON or YAML files, skip all prompting
- --payload-json JSON Payload JSON, skip all prompting
- --payload-yaml YAML Payload YAML, skip all prompting
- -O, --option OPTION Option in the format -O field="value"
  Always prompts. Use passed options as the default value.
- -N, --no-prompt Skip prompts. Use default values for all optional fields.
- -j, --json JSON Output
- -d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- -q, --quiet No Output, do not print to stdout
- -r, --remote REMOTE Remote name. The current remote is used by default.
- --remote-url URL Remote url. The current remote url is used by default.
- -T, --token TOKEN Access token for authentication with --remote.
- -U, --username USERNAME Username for authentication.
- -P, --password PASSWORD Password for authentication.
- -I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- -H, --header HEADER Additional HTTP header to include with requests.
  --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- -C, --nocolor Disable ANSI coloring
- -B, --benchmark Print benchmark time after the command is finished.
- -V, --debug Print extra output for debugging.
- -h, --help Print this help

Update your user settings.

17.62.6 morpheus user-settings update-avatar

Usage: morpheus user-settings update-avatar [file]
- -j, --json JSON Output
- -d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- -q, --quiet No Output, do not print to stdout
- -r, --remote REMOTE Remote name. The current remote is used by default.
- --remote-url URL Remote url. The current remote url is used by default.

(continues on next page)
-T, --token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

Update your avatar profile image.
[File] is required. This is the local path of a file to upload [png|jpg|svg].

17.62.7 morpheus user-settings view-avatar

Usage: morpheus user-settings view-avatar
-r, --remote REMOTE Remote name. The current remote is used by default.
--default.
-remote-url URL Remote url. The current remote url is used by default.
--default.
-T, --token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help

View your avatar profile image.
This opens the avatar image url with a web browser.

17.63 morpheus user-sources

Usage: morpheus user-sources [command] [options]
Commands:
  activate
  add
  deactivate
  get
  get-type

(continues on next page)
list
list-types
remove
update
update-subdomain

### 17.63.1 morpheus user-sources activate

```
Usage: morpheus user-sources activate [name]

-O, --option OPTION Option in the format -O field="value"
  Always prompts. Use passed options as the
--prompt default value.
-N, --no-prompt Skip prompts. Use default values for all
--optional fields.
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of
--executing it
-curl
--scrub Dry Run to output API request as a curl command.
Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE Remote name. The current remote is used by
--default.
--remote-url URL Remote url. The current remote url is used by
--default.
-T, --token TOKEN Access token for authentication with --remote.
--saved credentials are used by default.
-u, --username USERNAME Username for authentication.
-p, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad
--SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
--timeout SECONDS Timeout for api requests. Default is typically
--30 seconds.
-c, --nocolor Disable ANSI coloring
-b, --benchmark Print benchmark time after the command is
--finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help
```

Activate a user source.
[name] is required. This is the name or id of a user source.

### 17.63.2 morpheus user-sources add

```
Usage: morpheus user-sources add [account] [name]

--account ID Account this user source belongs to
--type CODE User Source Type
--name VALUE Name for this user source
--description VALUE Description
--role-mappings MAPPINGS Role Mappings FQN in the format id1:FQN1,id2:FQN2
--role-mapping-names MAPPINGS Role Mapping Names in the format id1:Name1,
```

(continues on next page)
Morpheus Documentation, Release 3.6.2

(continued from previous page)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--default-role ID</td>
<td>Default Role ID</td>
</tr>
<tr>
<td>-O, --option OPTION</td>
<td>Option in the format -O field=&quot;value&quot;</td>
</tr>
<tr>
<td>--prompt</td>
<td>Always prompts. Use passed options as the default value.</td>
</tr>
<tr>
<td>-N, --no-prompt</td>
<td>Skip prompts. Use default values for all optional fields.</td>
</tr>
<tr>
<td>--payload FILE</td>
<td>Payload from a local JSON or YAML file, skip all prompting</td>
</tr>
<tr>
<td>--payload-dir DIRECTORY</td>
<td>Payload from a local directory containing 1-N JSON or YAML files, skip all prompting</td>
</tr>
<tr>
<td>-j, --json</td>
<td>JSON Output</td>
</tr>
<tr>
<td>-d, --dry-run</td>
<td>Dry Run, print the API request instead of executing it</td>
</tr>
<tr>
<td>--curl</td>
<td>Dry Run to output API request as a curl command.</td>
</tr>
<tr>
<td>--scrub</td>
<td>Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.</td>
</tr>
<tr>
<td>-r, --remote REMOTE</td>
<td>Remote name. The current remote is used by default.</td>
</tr>
<tr>
<td>--remote-url URL</td>
<td>Remote url. The current remote url is used by default.</td>
</tr>
<tr>
<td>-T, --token TOKEN</td>
<td>Access token for authentication with --remote. Saved credentials are used by default.</td>
</tr>
<tr>
<td>-U, --username USERNAME</td>
<td>Username for authentication.</td>
</tr>
<tr>
<td>-P, --password PASSWORD</td>
<td>Password for authentication.</td>
</tr>
<tr>
<td>-I, --insecure</td>
<td>Allow insecure HTTPS communication. i.e. bad SSL certificate.</td>
</tr>
<tr>
<td>-H, --header HEADER</td>
<td>Additional HTTP header to include with requests.</td>
</tr>
<tr>
<td>--timeout SECONDS</td>
<td>Timeout for api requests. Default is typically 30 seconds.</td>
</tr>
<tr>
<td>-C, --nocolor</td>
<td>Disable ANSI coloring</td>
</tr>
<tr>
<td>-B, --benchmark</td>
<td>Print benchmark time after the command is finished.</td>
</tr>
<tr>
<td>-V, --debug</td>
<td>Print extra output for debugging.</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Print this help</td>
</tr>
</tbody>
</table>

Create a new user source. [account] is required. This is the name or id of an account.

### 17.63.3 morpheus user-sources deactivate

Usage: morpheus user-sources deactivate [name]

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-O, --option OPTION</td>
<td>Option in the format -O field=&quot;value&quot;</td>
</tr>
<tr>
<td>--prompt</td>
<td>Always prompts. Use passed options as the default value.</td>
</tr>
<tr>
<td>-N, --no-prompt</td>
<td>Skip prompts. Use default values for all optional fields.</td>
</tr>
<tr>
<td>--payload FILE</td>
<td>Payload from a local JSON or YAML file, skip all prompting</td>
</tr>
<tr>
<td>--payload-dir DIRECTORY</td>
<td>Payload from a local directory containing 1-N JSON or YAML files, skip all prompting</td>
</tr>
<tr>
<td>-j, --json</td>
<td>JSON Output</td>
</tr>
<tr>
<td>-d, --dry-run</td>
<td>Dry Run, print the API request instead of executing it</td>
</tr>
<tr>
<td>--curl</td>
<td>Dry Run to output API request as a curl command.</td>
</tr>
<tr>
<td>--scrub</td>
<td>Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.</td>
</tr>
<tr>
<td>-r, --remote REMOTE</td>
<td>Remote name. The current remote is used by default.</td>
</tr>
<tr>
<td>--remote-url URL</td>
<td>Remote url. The current remote url is used by default.</td>
</tr>
</tbody>
</table>

(continues on next page)
Deactivate a user source.

[name] is required. This is the name or id of a user source.

17.63.4 morpheus user-sources get

Usage: morpheus user-sources get [name]

- j, --json
  JSON Output
- y, --yaml
  YAML Output
- c, --csv
  CSV Output
- csv-delim CHAR
  Delimiter for CSV Output rows. Default: ','
- csv-newline [CHAR]
  Delimiter for CSV Output rows. Default: '\n'
- csv-quotes
  Wrap CSV values with "". Default: false
- csv-no-header
  Exclude header for CSV Output.
- F, --fields x,y,z
  Filter Output to a limited set of fields.
- d, --dry-run
  Dry Run, print the API request instead of executing it
- c, --curl
  Dry Run to output API request as a curl command.
- s, --scrub
  Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE
  Remote name. The current remote is used by default.
- u, --remote-url URL
  Remote url. The current remote url is used by default.
- T, --token TOKEN
  Access token for authentication with --remote.
- U, --username USERNAME
  Username for authentication.
- P, --password PASSWORD
  Password for authentication.
- I, --insecure
  Allow insecure HTTPS communication. i.e. bad SSL certificate.
- H, --header HEADER
  Additional HTTP header to include with requests.
- t, --timeout SECONDS
  Timeout for api requests. Default is typically 30 seconds.
- C, --nocolor
  Disable ANSI coloring
- B, --benchmark
  Print benchmark time after the command is finished.
- v, --debug
  Print extra output for debugging.
- h, --help
  Print this help
Get details about an user source. 
[name] is required. This is the name or id of an user source.

17.63.5 morpheus user-sources get-type

Usage: morpheus user-sources get-type [type]

   -j, --json                JSON Output
   --yaml                   YAML Output
   --csv                    CSV Output
   --csv-delim CHAR         Delimiter for CSV Output values. Default: ','
   --csv-newline [CHAR]     Delimiter for CSV Output rows. Default: '\n'
   --csv-quotes             Wrap CSV values with '. Default: false
   --csv-no-header          Exclude header for CSV Output.
   -F, --fields x,y,z       Filter Output to a limited set of fields.
   -d, --dry-run            Dry Run, print the API request instead of
                           executing it
   --curl                   Dry Run to output API request as a curl command.
   --scrub                  Mask secrets in output, such as the
                           Authorization header. For use with --curl and
                           --dry-run.
   -r, --remote REMOTE      Remote name. The current remote is used by
   --default.               --remote-url URL      Remote url. The current remote url is used by
                           Default.
   --default.               --token TOKEN       Access token for authentication with --remote.
   --fields x,y,z           Filter Output to a limited set of fields.
   --fields x,y,z           Filter Output to a limited set of fields.
   --account ID             Filter by Tenant
   -m, --max MAX            Max Results
   -o, --offset OFFSET      Offset Results
   -s, --search PHRASE      Search Phrase
   -S, --sort ORDER         Sort Order
   -D, --desc               Reverse Sort Order

Get details about a user source type. 
[type] is required. This is the type identifier.

17.63.6 morpheus user-sources list

Usage: morpheus user-sources list

   --account ID Filter by Tenant
   -m, --max MAX Max Results
   -o, --offset OFFSET Offset Results
   -s, --search PHRASE Search Phrase
   -S, --sort ORDER Sort Order
   -D, --desc Reverse Sort Order

(continues on next page)
-Q, --query PARAMS
  Query parameters. PARAMS format is
  -->phrase=foobar&category=web'
-j, --json
  JSON Output
-yaml
  YAML Output
-csv
  CSV Output
--csv-delim CHAR
  Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR]
  Delimiter for CSV Output rows. Default: '\n'
--csv-quotes
  Wrap CSV values with ". Default: false
--csv-no-header
  Exclude header for CSV Output.
-F, --fields x,y,z
  Filter Output to a limited set of fields.
  Default is all fields.
-d, --dry-run
  Dry Run, print the API request instead of executing it
  --curl
    Dry Run to output API request as a curl command.
    Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-r, --remote REMOTE
  Remote name. The current remote is used by default.
--remote-url URL
  Remote url. The current remote url is used by default.
-T, --token TOKEN
  Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME
  Username for authentication.
-P, --password PASSWORD
  Password for authentication.
-I, --insecure
  Allow insecure HTTPS communication. i.e. bad SSL certificate.
-H, --header HEADER
  Additional HTTP header to include with requests. Timeout for api requests. Default is typically 30 seconds.
-C, --nocolor
  Disable ANSI coloring
-B, --benchmark
  Print benchmark time after the command is finished.
-V, --debug
  Print extra output for debugging.
-h, --help
  Print this help

List user sources.

17.63.7 morpheus user-sources list-types

Usage: morpheus user-sources list-types
-m, --max MAX
  Max Results
-o, --offset OFFSET
  Offset Results
-s, --search PHRASE
  Search Phrase
-S, --sort ORDER
  Sort Order
-D, --desc
  Reverse Sort Order
-j, --json
  JSON Output
-yaml
  YAML Output
-csv
  CSV Output
--csv-delim CHAR
  Delimiter for CSV Output values. Default: ','
--csv-newline [CHAR]
  Delimiter for CSV Output rows. Default: '\n'
--csv-quotes
  Wrap CSV values with ". Default: false
--csv-no-header
  Exclude header for CSV Output.
-F, --fields x,y,z
  Filter Output to a limited set of fields.
  Default is all fields.
-d, --dry-run
  Dry Run, print the API request instead of executing it
List user source types.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--curl</td>
<td>Dry Run to output API request as a curl command.</td>
</tr>
<tr>
<td>--scrub</td>
<td>Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.</td>
</tr>
<tr>
<td>-r, --remote REMOTE</td>
<td>Remote name. The current remote is used by default.</td>
</tr>
<tr>
<td>-d, --default</td>
<td>Remote url. The current remote url is used by default.</td>
</tr>
<tr>
<td>-T, --token TOKEN</td>
<td>Access token for authentication with --remote.</td>
</tr>
<tr>
<td>-U, --username USERNAME</td>
<td>Username for authentication.</td>
</tr>
<tr>
<td>-P, --password PASSWORD</td>
<td>Password for authentication.</td>
</tr>
<tr>
<td>-I, --insecure</td>
<td>Allow insecure HTTPS communication. i.e. bad SSL certificate.</td>
</tr>
<tr>
<td>-H, --header HEADER</td>
<td>Additional HTTP header to include with requests.</td>
</tr>
<tr>
<td>-C, --nocolor</td>
<td>Disable ANSI coloring</td>
</tr>
<tr>
<td>-B, --benchmark</td>
<td>Print benchmark time after the command is finished.</td>
</tr>
<tr>
<td>-V, --debug</td>
<td>Print extra output for debugging.</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Print this help</td>
</tr>
</tbody>
</table>

17.63.8 morpheus user-sources remove

Usage: morpheus user-sources remove [name]

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-y, --yes</td>
<td>Auto Confirm</td>
</tr>
<tr>
<td>-j, --json</td>
<td>JSON Output</td>
</tr>
<tr>
<td>-d, --dry-run</td>
<td>Dry Run, print the API request instead of executing it</td>
</tr>
<tr>
<td>-c, --curl</td>
<td>Dry Run to output API request as a curl command.</td>
</tr>
<tr>
<td>-s, --scrub</td>
<td>Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.</td>
</tr>
<tr>
<td>-r, --remote REMOTE</td>
<td>Remote name. The current remote is used by default.</td>
</tr>
<tr>
<td>-d, --default</td>
<td>Remote url. The current remote url is used by default.</td>
</tr>
<tr>
<td>-T, --token TOKEN</td>
<td>Access token for authentication with --remote.</td>
</tr>
<tr>
<td>-U, --username USERNAME</td>
<td>Username for authentication.</td>
</tr>
<tr>
<td>-P, --password PASSWORD</td>
<td>Password for authentication.</td>
</tr>
<tr>
<td>-I, --insecure</td>
<td>Allow insecure HTTPS communication. i.e. bad SSL certificate.</td>
</tr>
<tr>
<td>-H, --header HEADER</td>
<td>Additional HTTP header to include with requests.</td>
</tr>
<tr>
<td>-C, --timeout SECONDS</td>
<td>Timeout for api requests. Default is typically 30 seconds.</td>
</tr>
<tr>
<td>-C, --nocolor</td>
<td>Disable ANSI coloring</td>
</tr>
<tr>
<td>-B, --benchmark</td>
<td>Print benchmark time after the command is finished.</td>
</tr>
<tr>
<td>-V, --debug</td>
<td>Print extra output for debugging.</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Print this help</td>
</tr>
</tbody>
</table>

Delete a user_source.
## 17.63.9 morpheus user-sources update

Usage: `morpheus user-sources update [name] [options]`
- `--name VALUE` Name for this user source
- `--description VALUE` Description
- `--role-mappings MAPPINGS` Role Mappings in the format id1:FQN,id2:FQN2
- `--role-mapping-names MAPPINGS` Role Mapping Names in the format id1:Name1,id2:Name2
- `--option OPTION` Option in the format -O field="value"
- `--prompt` Always prompts. Use passed options as the default value.
- `--no-prompt` Skip prompts. Use default values for all optional fields.
- `--json JSON Output`
- `--dry-run` Dry Run, print the API request instead of executing it
- `--curl` Dry Run to output API request as a curl command.
- `--scrub` Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- `--remote REMOTE` Remote name. The current remote is used by default.
- `--remote-url URL` Remote url. The current remote url is used by default.
- `--token TOKEN` Access token for authentication with --remote.
- `--username USERNAME` Username for authentication.
- `--password PASSWORD` Password for authentication.
- `--insecure` Allow insecure HTTPS communication. i.e. bad SSL certificate.
- `--header HEADER` Additional HTTP header to include with requests.
- `--timeout SECONDS` Timeout for api requests. Default is typically 30 seconds.
- `--nocolor` Disable ANSI coloring
- `--benchmark` Print benchmark time after the command is finished.
- `--help` Print this help

Update a user source. [name] is required. This is the name or id of a user source.

## 17.63.10 morpheus user-sources update-subdomain

Usage: `morpheus user-sources update-subdomain [name]`
- `--subdomain VALUE` New subdomain for this user source
- `--option OPTION` Option in the format -O field="value"
- `--prompt` Always prompts. Use passed options as the default value.
- `--no-prompt` Skip prompts. Use default values for all optional fields.
- `--json JSON Output`
- `--dry-run` Dry Run, print the API request instead of executing it
- `--curl` Dry Run to output API request as a curl command.

(continues on next page)
Update subdomain for a user source.

[name] is required. This is the name or id of a user source.

17.64 morphesus users

Usage: morphesus users [command] [options]

Commands:
  add
  count
  get
  list
  passwd
  remove
  update

17.64.1 morphesus users add

Usage: morphesus users add [options]
  --firstName VALUE  First Name (optional)
  --lastName VALUE  Last Name (optional)
  --username VALUE  Username
  --email VALUE  Email
  --password VALUE  Password
  --passwordConfirmation VALUE  Confirm Password
  --instanceLimits.maxStorage VALUE  Max Storage (bytes) (optional)
  --instanceLimits.maxMemory VALUE  Max Memory (bytes) (optional)
  --instanceLimits.maxCpu VALUE  CPU Count (optional)
  --role VALUE  Role (optional) - Role names (comma separated)
17.64.2 morpheus users count

Usage: morpheus users count [options]
  -a, --account ACCOUNT Account Name
  -A, --account-id ID Account ID
  -O, --option OPTION Option in the format -O field="value"
  -P, --prompt Always prompts. Use passed options as the default value.
  -N, --no-prompt Skip prompts. Use default values for all optional fields.
  --payload FILE Payload from a local JSON or YAML file, skip all prompting
  --payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
  --payload-json JSON Payload JSON, skip all prompting
  --payload-yaml YAML Payload YAML, skip all prompting
  -j, --json JSON Output
  -d, --dry-run Dry Run, print the API request instead of executing it
  --dry-run Dry Run to output API request as a curl command.
  -C, --nocolor Disable ANSI coloring
  -B, --benchmark Print benchmark time after the command is finished.
  -T, --token TOKEN Access token for authentication with --remote.
  -U, --username USERNAME Username for authentication.
  -P, --password PASSWORD Password for authentication.
  -I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
  -H, --header HEADER Additional HTTP header to include with requests.
  --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
  -d, --dry-run Dry Run, print the API request instead of executing it
  --dry-run Dry Run to output API request as a curl command.
  -C, --nocolor Disable ANSI coloring
  -B, --benchmark Print benchmark time after the command is finished.
  -V, --debug Print extra output for debugging.
  -h, --help Print this help
Get the number of users.

### 17.64.3 morpheus users get

**Usage:** `morpheus users get [username]

- `f`, `--feature-access` Display Feature Access
- `a`, `--all-access` Display All Access Lists
- `A`, `--account ACCOUNT` Account Name
- `A`, `--account-id ID` Account ID
- `j`, `--json` JSON Output
- `yml`, `--yaml` YAML Output
- `csv`, `--csv` CSV Output
- `csv-delim CHAR` Delimiter for CSV Output values. Default: ','
- `csv-newline [CHAR]` Delimiter for CSV Output rows. Default: '\n'
- `csv-quotes` Wrap CSV values with '. Default: false
- `csv-no-header` Exclude header for CSV Output.
- `fields x,y,z` Filter Output to a limited set of fields.
- `d`, `--dry-run` Dry Run, print the API request instead of executing it
- `curl` Dry Run to output API request as a curl command.
- `scrub` Mask secrets in output, such as the Authorization header. For use with `--curl` and `--dry-run`.
- `r`, `--remote REMOTE` Remote name. The current remote is used by default.
- `remote-url URL` Remote url. The current remote url is used by default.
- `token TOKEN` Access token for authentication with `--remote`.
- `username USERNAME` Username for authentication.
- `password PASSWORD` Password for authentication.
- `insecure` Allow insecure HTTPS communication. i.e. bad SSL certificate.
- `header HEADER` Additional HTTP header to include with requests.
- `timeout SECONDS` Timeout for api requests. Default is typically 30 seconds.
- `nocolor` Disable ANSI coloring
- `benchmark` Print benchmark time after the command is finished.
- `debug` Print extra output for debugging.
- `help` Print this help

Get details about a user.

[username] is required. This is the username or id of a user.

### 17.64.4 morpheus users list

**Usage:** `morpheus users list

- `a`, `--account ACCOUNT` Account Name
- `A`, `--account-id ID` Account ID
- `m`, `--max MAX` Max Results
-o, --offset OFFSET Offset Results
-s, --search PHRASE Search Phrase
-S, --sort ORDER Sort Order
-D, --desc Reverse Sort Order
-Q, --query PARAMS Query parameters. PARAMS format is
  -phrase=foobar&category=web' YAML Output
  -yaml CSV Output
  --csv Delimiter for CSV Output rows. Default: ','
  --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '
  --csv-quotes Wrap CSV values with '. Default: false
  --csv-no-header Exclude header for CSV Output.
-F, --fields x,y,z Filter Output to a limited set of fields.
  -F Default is all fields.
-j, --json JSON Output
-d, --dry-run Dry Run, print the API request instead of
  -executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the
  -Authorization header. For use with --curl and --dry-run.
  -r, --remote REMOTE Remote name. The current remote is used by
  -default.
  --remote-url URL Remote url. The current remote url is used by
  -default.
-T, --token TOKEN Access token for authentication with --remote.
-Saved credentials are used by default.
-U, --username USERNAME Username for authentication.
-P, --password PASSWORD Password for authentication.
-I, --insecure Allow insecure HTTPS communication. i.e. bad
  -SSL certificate.
-H, --header HEADER Additional HTTP header to include with requests.
  --timeout SECONDS Timeout for api requests. Default is typically
  -30 seconds.
-C, --nocolor Disable ANSI coloring
-B, --benchmark Print benchmark time after the command is
  -finished.
-V, --debug Print extra output for debugging.
-h, --help Print this help List users.

17.64.5 morpheus users passwd

Usage: morpheus users passwd [username] [options]
  -a, --account ACCOUNT Account Name
  -A, --account-id ID Account ID
  -O, --option OPTION Option in the format -O field="value"
  --prompt Always prompts. Use passed options as the
  -default value.
  -N, --no-prompt Skip prompts. Use default values for all
  -optional fields.
  -j, --json JSON Output
  -d, --dry-run Dry Run, print the API request instead of
  --executing it
  --curl Dry Run to output API request as a curl command.
Morpheus Documentation, Release 3.6.2

(continued from previous page)

```
--scrub
   Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-<r, --remote REMOTE
   Remote name. The current remote is used by default.
-<default.
-<remote-url URL
   Remote url. The current remote url is used by default.
-<default.
-<T, --token TOKEN
   Access token for authentication with --remote.
-<Saved credentials are used by default.
-<U, --username USERNAME
   Username for authentication.
-<P, --password PASSWORD
   Password for authentication.
-<I, --insecure
   Allow insecure HTTPS communication. i.e. bad SSL certificate.
-<SSL certificate.
-<H, --header HEADER
   Additional HTTP header to include with requests.
-<timeout SECONDS
   Timeout for api requests. Default is typically 30 seconds.
-<g, --quiet
   No Output, do not print to stdout
-<C, --nocolor
   Disable ANSI coloring
-<B, --benchmark
   Print benchmark time after the command is finished.
-<V, --debug
   Print extra output for debugging.
-<h, --help
   Print this help
```

17.64.6 morpheus users remove

Usage: morpheus users remove [username]

-<a, --account ACCOUNT
   Account Name
-<A, --account-id ID
   Account ID
-<y, --yes
   Auto Confirm
-<j, --json
   JSON Output
-<d, --dry-run
   Dry Run, print the API request instead of executing it
   --curl
   Dry Run to output API request as a curl command.
   --scrub
   Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-<C, --nocolor
   Disable ANSI coloring
-<B, --benchmark
   Print benchmark time after the command is finished.
-<V, --debug
   Print extra output for debugging.
-<h, --help
   Print this help

17.64.7 morpheus users update

Usage: morpheus users update [username] [options]

-<firstName VALUE
   First Name (optional)
-<lastName VALUE
   Last Name (optional)
-<username VALUE
   Username (optional)
-<email VALUE
   Email (optional)
-<password VALUE
   Password (optional)
-<passwordConfirmation VALUE
   Confirm Password (optional)
-<instanceLimits.maxStorage VALUE
   Max Storage (bytes) (optional)
-<instanceLimits.maxMemory VALUE

(continues on next page)
Max Memory (bytes) (optional)

--instanceLimits.maxCpu VALUE

CPU Count (optional)

--role VALUE

Role (optional) - Role names (comma separated)

-a, --account ACCOUNT

Account Name

-A, --account-id ID

Account ID

-O, --option OPTION

Option in the format -O field="value"

-P, --prompt

Always prompts. Use passed options as the default value.

-N, --no-prompt

Skip prompts. Use default values for all optional fields.

--payload FILE

Payload from a local JSON or YAML file, skip all prompting

--payload-dir DIRECTORY

Payload from a local directory containing 1-N JSON or YAML files, skip all prompting

--payload-json JSON

Payload JSON, skip all prompting

--payload-yaml YAML

Payload YAML, skip all prompting

-j, --json

JSON Output

-d, --dry-run

Dry Run, print the API request instead of executing it

--curl

Dry Run to output API request as a curl command.

--scrub

Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.

-C, --nocolor

Disable ANSI coloring

-B, --benchmark

Print benchmark time after the command is finished.

-V, --debug

Print extra output for debugging.

-h, --help

Print this help

17.65 morpheus version

morpheus version

-v, --short

Print only the client version number

-C, --nocolor

Disable ANSI coloring

-B, --benchmark

Print benchmark time after the command is finished.

-V, --debug

Print extra output for debugging.

-h, --help

Print this help

17.66 morpheus virtual-images

Usage: morpheus virtual-images [command] [options]

Commands:

add
add-file
get
list
remove
remove-file
types
update
17.66.1 morpheus virtual-images add

```
Usage: morpheus virtual-images add [name] -t TYPE
      -f, --filename NAME          Image File Name. Specify a name for the uploaded
      -u, --url URL                Image File URL. This can be used instead of
      --tenants LIST              Tenant Access, comma separated list of account
      -O, --option OPTION         Option in the format -O field="value"
      --default value.            Always prompts. Use passed options as the
      --no-prompt                 Skip prompts. Use default values for all
      --optional fields.          JSON Output
      -j, --json                   Dry Run, print the API request instead of
      -d, --dry-run                Dry Run to output API request as a curl command.
      --executing it               Mask secrets in output, such as the
      --curl                      Authorization header. For use with --curl and --dry-run.
      --scrub                     --remote REMOTE          Remote name. The current remote is used by
      --default                   --remote-url URL        Remote url. The current remote url is used by
      -T, --token TOKEN           Access token for authentication with --remote.
      --header HEADER             Saved credentials are used by default.
      -U, --username USERNAME     Username for authentication.
      -P, --password PASSWORD     Password for authentication.
      -I, --insecure              Allow insecure HTTPS communication. i.e. bad
      --ssl certificate.          Additional HTTP header to include with requests.
      --timeout SECONDS          Timeout for api requests. Default is typically
      --30 seconds.               Disable ANSI coloring
      -c, --nocolor               --benchmark          Print benchmark time after the command is
      -b, --benchmark             --finished.            Print extra output for debugging.
      -V, --debug                 --help                Print this help

Create a virtual image.
```

17.66.2 morpheus virtual-images add-file

```
Usage: morpheus virtual-images add-file [name] [filepath]
      --filename FILENAME         Filename for uploaded file. Derived from
      [filepath] by default.
      --url URL                   Image File URL. This can be used instead of
      --gzip                      Compress uploaded file
      -j, --json                   JSON Output
      -d, --dry-run                Dry Run, print the API request instead of
      --executing it               Dry Run to output API request as a curl command.
      --curl                      Mask secrets in output, such as the
      --scrub                     Authorization header. For use with --curl and --dry-run.
```

(continues on next page)
### 17.66.3 morpheus virtual-images get

**Usage:** morpheus virtual-images get [name]

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-d, --dry-run</td>
<td>Dry Run, print the API request instead of executing it.</td>
</tr>
<tr>
<td>-r, --remote REMOTE</td>
<td>Remote name. The current remote is used by default.</td>
</tr>
<tr>
<td>--remote-url URL</td>
<td>Remote url. The current remote url is used by default.</td>
</tr>
<tr>
<td>-T, --token TOKEN</td>
<td>Access token for authentication with --remote.</td>
</tr>
<tr>
<td>Saved credentials are used by default.</td>
<td></td>
</tr>
<tr>
<td>-U, --username USERNAME</td>
<td>Username for authentication.</td>
</tr>
<tr>
<td>-P, --password PASSWORD</td>
<td>Password for authentication.</td>
</tr>
<tr>
<td>-I, --insecure</td>
<td>Allow insecure HTTPS communication. i.e. bad SSL certificate.</td>
</tr>
<tr>
<td>-H, --header HEADER</td>
<td>Additional HTTP header to include with requests.</td>
</tr>
<tr>
<td>--timeout SECONDS</td>
<td>Timeout for api requests. Default is typically 30 seconds.</td>
</tr>
<tr>
<td>-C, --nocolor</td>
<td>Disable ANSI coloring</td>
</tr>
<tr>
<td>-B, --benchmark</td>
<td>Print benchmark time after the command is finished.</td>
</tr>
<tr>
<td>-V, --debug</td>
<td>Print extra output for debugging.</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Print this help</td>
</tr>
</tbody>
</table>

Upload a virtual image file.

[name] is required. This is the name or id of a virtual image.

[filepath] or --url is required. This is location of the virtual image file.

---

17.66. morpheus virtual-images

---

819
Get details about a virtual image. [name] is required. This is the name or id of a virtual image.

17.66.4 morpheus virtual-images list

Usage: morpheus virtual-images list

-t, --type IMAGE_TYPE Image Type
-  --all All Images
-  --user User Images
-  --system System Images
-  --max MAX Max Results
-  --offset OFFSET Offset Results
-  --search PHRASE Search Phrase
-  --sort ORDER Sort Order
-  --desc Reverse Sort Order
-  --query PARAMS Query parameters. PARAMS format is
-  --phrase=foobar&category=web'
-  --json JSON Output
-  --yaml YAML Output
-  --csv CSV Output
-  --csv-delim CHAR Delimiter for CSV Output values. Default: ','
-  --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
-  --csv-quotes Wrap CSV values with '". Default: false
-  --csv-no-header Exclude header for CSV Output.
-  --fields x,y,z Filter Output to a limited set of fields. Default is all fields.
-  --dry-run Dry Run, print the API request instead of executing it
-  --curl Dry Run to output API request as a curl command.
-  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
-  --remote REMOTE Remote name. The current remote is used by default.
-  --remote-url URL Remote url. The current remote url is used by default.
-  --token TOKEN Access token for authentication with --remote.
-  --username USERNAME Username for authentication.
-  --password PASSWORD Password for authentication.
-  --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
-  --header HEADER Additional HTTP header to include with requests.
-  --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
-  --color Disable ANSI coloring
-  --benchmark Print benchmark time after the command is finished.
-  --debug Print extra output for debugging.
-  --help Print this help
List virtual images.

17.66.5 morpheus virtual-images remove

Usage: morpheus virtual-images remove [name]

- y, --yes Auto Confirm
- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by default.
- u, --remote-url URL Remote url. The current remote url is used by default.
- t, --token TOKEN Access token for authentication with --remote.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- i, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
- T, --timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- c, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- v, --debug Print extra output for debugging.
- h, --help Print this help

17.66.6 morpheus virtual-images remove-file

Usage: morpheus virtual-images remove-file [name] [filename]

- y, --yes Auto Confirm
- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by default.
- u, --remote-url URL Remote url. The current remote url is used by default.
- t, --token TOKEN Access token for authentication with --remote.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- i, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
17.66.7 morpheus virtual-images types

Usage: morpheus virtual-images types

- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
- r, --remote REMOTE Remote name. The current remote is used by default.
--remote-url URL Remote url. The current remote url is used by default.
- T, --token TOKEN Access token for authentication with --remote.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. i.e. bad SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
- timeout SECONDS Timeout for api requests. Default is typically 30 seconds.
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help

17.66.8 morpheus virtual-images update

Usage: morpheus virtual-images update [name] [options]

--tenants LIST Tenant Access, comma separated list of account IDs
-O, --option OPTION Option in the format -O field="value"
--prompt Always prompts. Use passed options as the default value.
-N, --no-prompt Skip prompts. Use default values for all optional fields.
--payload FILE Payload from a local JSON or YAML file, skip all prompting
--payload-dir DIRECTORY Payload from a local directory containing 1-N JSON or YAML files, skip all prompting
--payload-json JSON Payload JSON, skip all prompting
--payload-yaml YAML Payload YAML, skip all prompting

(continues on next page)
Update a virtual image.

[name] is required. This is the name or id of a virtual image.

Usage: morpheus whoami [options]

- n, --name
- f, --feature-access
- t, --token-only
- j, --json
- r, --remote REMOTE
- d, --dry-run
- c, --curl
- s, --scrub
- d, --dry-run
- c, --curl
- s, --scrub
- Authorization header. For use with --curl and --dry-run.
- U, --username USERNAME
- P, --password PASSWORD
- I, --insecure
- H, --header HEADER
- T, --token TOKEN
- R, --remote-url URL
- H, --header HEADER
- T, --token TOKEN
- R, --remote-url URL
- d, --dry-run
- d, --dry-run
- d, --dry-run
- d, --dry-run
- Authorization header. For use with --curl and --dry-run.
- q, --quiet

Update a virtual image.

17.67. morpheus whoami

Usage: morpheus whoami [options]

- n, --name
- f, --feature-access
- t, --token-only
- j, --json
- r, --remote REMOTE
- d, --dry-run
- c, --curl
- s, --scrub
- d, --dry-run
- c, --curl
- s, --scrub
- Authorization header. For use with --curl and --dry-run.
- q, --quiet

Update a virtual image.

[17.67. morpheus whoami]
17.68 morpheus workflows

Usage: morpheus workflows [command] [options]

Commands:
    add
    get
    list
    remove
    update

17.68.1 morpheus workflows add

Usage: morpheus workflows add [name] --tasks taskId:phase,taskId2:phase,taskId3:phase
    --name NAME Name for workflow
    --tasks x,y,z List of tasks to run in order, in the format
    --prompt Always prompts. Use passed options as the
    --option OPTION Option in the format -O field="value"
    --default value.
    --no-prompt Skip prompts. Use default values for all
    --optional fields.
    --payload FILE Payload from a local JSON or YAML file, skip all
    --prompting
    --payload-dir DIRECTORY Payload from a local directory containing 1-N
    --payload-json JSON Payload JSON, skip all prompting
    --payload-yaml YAML Payload YAML, skip all prompting
    --executing it
    --json JSON Output
    --dry-run Dry Run, print the API request instead of
    --performing
    --curl Dry Run to output API request as a curl command.
    --secret Mask secrets in output, such as the
    --Authorization header. For use with --curl and --dry-run.
    --quiet No Output, do not print to stdout
    --remote REMOTE Remote name. The current remote is used by
    --remote-url URL Remote url. The current remote url is used by
    --token TOKEN Access token for authentication with --remote.
    --saved credentials are used by default.
    --username USERNAME Username for authentication.
    --password PASSWORD Password for authentication.
    --insecure Allow insecure HTTPS communication. i.e. bad
    --SSL certificate.
    --header HEADER Additional HTTP header to include with requests.
    --timeout SECONDS Timeout for api requests. Default is typically
-C, --nocolor        Disable ANSI coloring
-B, --benchmark     Print benchmark time after the command is finished.
-V, --debug         Print extra output for debugging.
-h, --help          Print this help

17.68.2 morpheus workflows get

Usage: morpheus workflows get [workflow]
   -j, --json          JSON Output
   --yaml             YAML Output
   --csv              CSV Output
   --csv-delim CHAR   Delimiter for CSV Output values. Default: ','
   --csv-newline [CHAR] Delimiter for CSV Output rows. Default: '\n'
   --csv-quotes       Wrap CSV values with '"'. Default: false
   --csv-no-header    Exclude header for CSV Output.
   -F, --fields x,y,z Filter Output to a limited set of fields. Default is all fields.
   -d, --dry-run      Dry Run, print the API request instead of executing it
   --curl             Dry Run to output API request as a curl command.
   --scrub            Mask secrets in output, such as the Authorization header. For use with --curl and --dry-run.
   -r, --remote REMOTE Remote name. The current remote is used by default.
   --remote-url URL   Remote url. The current remote url is used by default.
   -T, --token TOKEN  Access token for authentication with --remote.
   -U, --username USERNAME Username for authentication.
   -P, --password PASSWORD Password for authentication.
   -I, --insecure     Allow insecure HTTPS communication. i.e. bad SSL certificate.
   --header HEADER    Additional HTTP header to include with requests.
   --timeout SECONDS  Timeout for api requests. Default is typically 30 seconds.
   -C, --nocolor      Disable ANSI coloring
   -B, --benchmark    Print benchmark time after the command is finished.
   -V, --debug        Print extra output for debugging.
   -h, --help         Print this help

17.68.3 morpheus workflows list

Usage: morpheus workflows list
   -m, --max MAX       Max Results
   -o, --offset OFFSET Offset Results
   -s, --search PHRASE Search Phrase
   -S, --sort ORDER    Sort Order
   -D, --desc          Reverse Sort Order
   -Q, --query PARAMS  Query parameters. PARAMS format is 'phrase=foobar&category=web'
   -j, --json          JSON Output

(continues on next page)
17.68.4 morpheus workflows remove

Usage: morpheus workflows remove [name]

- y, --yes Auto Confirm
- j, --json JSON Output
- d, --dry-run Dry Run, print the API request instead of
  executing it
  --curl Dry Run to output API request as a curl command.
  --scrub Mask secrets in output, such as the
  Authorization header. For use with --curl and --dry-run.
- u, --remote REMOTE Remote name. The current remote is used by
  default.
- r, --remote-url URL Remote url. The current remote url is used by
  default.
- T, --token TOKEN Access token for authentication with --remote.
- U, --username USERNAME Username for authentication.
- P, --password PASSWORD Password for authentication.
- I, --insecure Allow insecure HTTPS communication. i.e. bad
  SSL certificate.
- H, --header HEADER Additional HTTP header to include with requests.
- t, --timeout SECONDS Timeout for api requests. Default is typically
  30 seconds.
- C, --nocolor Disable ANSI coloring
- B, --benchmark Print benchmark time after the command is
  finished.
- V, --debug Print extra output for debugging.
- h, --help Print this help
17.68.5 morpheus workflows update

Usage: morpheus workflows update [name] --tasks taskId:phase,taskId2:phase,
-<taskId3:phase
-<tasks x,y,z
-<Phase>. Phase is optional, the default is 'provision'.
-<name NAME New name for workflow
-<O, --option OPTION Option in the format -O field="value"
-<prompt Always prompts. Use passed options as the
-<default value.
-<N, --no-prompt Skip prompts. Use default values for all
-<optional fields.
-<payload FILE Payload from a local JSON or YAML file, skip all
-<prompting
-<payload-dir DIRECTORY Payload from a local directory containing 1-N
-<JSON or YAML files, skip all prompting
-<payload-json JSON Payload JSON, skip all prompting
-<payload-yaml YAML Payload YAML, skip all prompting
-<j, --json JSON Output
-<d, --dry-run Dry Run, print the API request instead of
-<executing it
-<curl Dry Run to output API request as a curl command.
-<scrub Mask secrets in output, such as the
-<Authorization header. For use with --curl and --dry-run.
-<q, --quiet No Output, do not print to stdout.
-<r, --remote REMOTE Remote name. The current remote is used by
-<default.
-<remote-url URL Remote url. The current remote url is used by
-<default.
-<T, --token TOKEN Access token for authentication with --remote.
-<Saved credentials are used by default.
-<U, --username USERNAME Username for authentication.
-<P, --password PASSWORD Password for authentication.
-<I, --insecure Allow insecure HTTPS communication. i.e. bad
-<SSL certificate.
-<H, --header HEADER Additional HTTP header to include with requests.
-<timeout SECONDS Timeout for api requests. Default is typically
-<30 seconds.
-<C, --nocolor Disable ANSI coloring
-<B, --benchmark Print benchmark time after the command is
-<finished.
-<V, --debug Print extra output for debugging.
-<h, --help Print this help
Morpheus has only one environment variable that it uses.

### 18.1 MORPHEUS_CLI_HOME

The `MORPHEUS_CLI_HOME` variable is where morpheus CLI stores its configuration files. This can be set to allow a single system user to maintain many different configurations. If the directory does not exist, morpheus will attempt to create it.

The default home directory is `$HOME/.morpheus`

To see how this works, run the following:

```
MORPHEUS_CLI_HOME=~/.morpheus_test morpheus shell
```

Now, in your new morpheus shell, you can see that it is a fresh environment. There are no remote appliances configured.

```
morpheus> remote list
Morpheus Appliances
==============
You have no appliances configured. See the `remote add` command.
```

You can use this to create isolated environments (sandboxes), within which to execute your morpheus commands.

```
export MORPHEUS_CLI_HOME=~/.morpheus_test
morpheus remote add myremote https://testmorpheusappliance.mycompany.com --insecure
morpheus instances list
```

Morpheus saves the remote appliance information, including api access tokens, to the `$MORPHEUS_HOME_DIRECTORY`. These files are saved with file permissions `6000`. So, only one system user
should be allowed to execute morpheus with that home directory. See *Configuration* for more information on the files morpheus reads and writes.
Morpheus reads and writes several configuration files within the $MORPHEUS_CLI_HOME directory. 

**Note:** These files are maintained by the program. It is not recommended for you to manipulate them.

### 19.1 appliances file

The `appliances` YAML file contains a list of known appliances, keyed by name.

**Example:**

```yaml
qa:
  :host: https://qa.mycoolsite.com
  :active: true
production:
  :host: https://morpheus.mycoolsite.com
  :active: false
```

### 19.2 credentials file

The `.morpheus/credentials` YAML file contains access tokens for each known appliance.

### 19.3 groups file

The `.morpheus/groups` YAML file contains the active group information for each known appliance.
Startup scripts

When Morpheus starts, it executes the commands in a couple of dot files. These scripts are written in morpheus commands, not bash, so they can only execute morpheus commands and aliases.

20.1 .morpheus_profile file

It looks for $MORPHEUS_CLI_HOME/.morpheus_profile, and reads and executes it (if it exists). This may be inhibited by using the --noprofile option.

20.2 .morpheusrc file

When started as an interactive shell with the morpheus shell command, Morpheus reads and executes $MORPHEUS_CLI_HOME/.morpheusrc (if it exists). This may be inhibited by using the --norc option.

An example startup script might look like this:

```bash
# .morpheusrc

# aliases
alias our-instances='instances list -c "Our Cloud"

# switch to our appliance that we created with 'remote add morphapp1'
remote use morphapp1

# greeting
echo "Welcome back human, have fun!"

# print current user information
whoami
```

(continues on next page)
# print the list of instances in our cloud
our-instances
Welcome to the Morpheus API Documentation. Morpheus is a powerful provisioning service complete with monitoring, logging, backups, and application deployment strategies.

We currently provide an OAUTH 2.0 based authentication model and are working on language specific libraries for ruby and a few others.

The current version is 3.6.2
Authentication

The Morpheus API follows the OAuth 2.0 Specification and acts as an OAUTH 2.0 provider. To authorize your account you will need to use the same credentials you normally use to login to morpheus which will provide you with an accessToken as well as a refreshToken.

To authorize, use this code:

```bash
# With shell, you can just pass the correct header with each request
```

```json
#Returns:
{
  "access_token": "d0cc2cc4-f7f5-4713-a874-34491e7707de",
  "expires_in": 31535996,
  "refresh_token": "cda88865-f88d-4ed9-a621-424d9361beb2",
  "scope": "write",
  "token_type": "bearer"
}
```

Make sure to replace morpheus-appliance-url with your appliance url and meow with your username and password.

Morpheus expects all api requests to use the resultant access_token from the request made during authentication. This can be passed via the Authorization header. Be sure to replace the access_token with the actual token received from the OAuth request.

Authorization: Bearer access_token
23.1 Accounts

Provides API interfaces for managing the creation and modification of accounts within Morpheus (Typically only accessible by the Master Account)

23.1.1 Get All Accounts

```
curl "https://api.gomorpheus.com/api/accounts"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
    "accounts": [
    {
        "id": 1,
        "name": "Root Account",
        "description": "The master account",
        "currency": "USD",
        "instanceLimits": null,
        "lastUpdated": "2015-11-10T18:58:55+0000",
        "dateCreated": "2015-11-10T18:58:55+0000",
        "role": {
            "id": 1,
            "authority": "System Admin",
            "description": "Super User"
        },
        "active": true
    }
    ],
    "meta": {
```

(continues on next page)
This endpoint retrieves all accounts.

**HTTP Request**

GET https://api.gomorpheus.com/api/accounts

**Query Parameters**

**23.1.2 Get a Specific Account**

```
curl "https://api.gomorpheus.com/api/accounts/1" \  
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
    "account": {
        "id": 1,
        "name": "Root Account",
        "description": "The master account",
        "currency": "USD",
        "instanceLimits": null,
        "externalId": null,
        "lastUpdated": "2015-11-10T18:58:55+0000",
        "dateCreated": "2015-11-10T18:58:55+0000",
        "role": {
            "id": 1,
            "authority": "System Admin",
            "description": "Super User"
        },
        "active": true
    }
}
```

This endpoint will retrieve a specific account by id if the user has permission to access it.

**HTTP Request**

GET https://api.gomorpheus.com/api/accounts/:id

**23.1.3 Create an Account**
curl -XPOST "https://api.gomorpheus.com/api/accounts" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"account":{  "name": "My New Account",  "description": "My description",  "instanceLimits": {  "maxCpu": 0,  "maxMemory": 0,  "maxStorage": 0  },  "role": {  "id": 2  }}}'

The above command returns JSON structured like getting a single account:

**HTTP Request**

POST https://api.gomorpheus.com/api/accounts

**JSON Account Parameters**

23.1.4 Updating an Account

curl -XPUT "https://api.gomorpheus.com/api/accounts/2" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"account":{  "name": "My New Account",  "description": "My new description",  "instanceLimits": {  "maxCpu": 0,  "maxMemory": 0,  "maxStorage": 0  },  "role": {  "id": 3  }}'}

The above command returns JSON structured like getting a single account:

**HTTP Request**

PUT https://api.gomorpheus.com/api/accounts/:id
JSON Account Parameters

23.1.5 Delete an Account

```
curl -XDELETE "https://api.gomorpheus.com/api/accounts/1" \
   -H "Authorization: BEARER access_token"
```

The above command returns JSON Structured like this:

```
{
   "success": true
}
```

If an account still has users or instances tied to it, the delete will fail.

HTTP Request

DELETE https://api.gomorpheus.com/api/accounts/:id

23.2 Apps

Apps are groupings of instances that are linked together to form a full application stack. They can be created with existing templates or new templates, as well as from existing instances.

23.2.1 Get All Apps

```
curl "https://api.gomorpheus.com/api/apps"
   -H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
   "apps": [ 
   {
      "id": 1, 
      "name": "My Test App", 
      "description": "Sample Description", 
      "accountId": 1, 
      "account": { 
         "id": 1, 
         "name": "root" 
      }, 
      "siteId": 1, 
      "group": { 
         "id": 1, 
         "name": "My Group" 
      }, 
      "blueprint": { 
         "id": 135, 
         "name": "Grails Example", 
         "type": "morpheus" 
      }, 
   }
   ]
```

(continues on next page)
This endpoint retrieves all apps and the correlated instances. Server data is encrypted in the database.
HTTP Request

GET https://api.gomorpheus.com/api/apps

Query Parameters

23.2.2 Get a Specific App

curl "https://api.gomorpheus.com/api/apps/4" \
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```json
{
  "app": {
    "id": 1,
    "name": "My Test App",
    "description": "Sample Description",
    "accountId": 1,
    "account": {
      "id": 1,
      "name": "root"
    },
    "siteId": 1,
    "group": {
      "id": 1,
      "name": "My Group"
    },
    "blueprint": {
      "id": 135,
      "name": "Grails Example",
      "type": "morpheus"
    },
    "status": "running",
    "instanceCount": 2,
    "containerCount": 2,
    "dateCreated": "2015-06-09T20:59:17Z",
    "lastUpdated": "2015-06-09T21:00:19Z",
    "appTiers": [
      {
        "tier": {
          "id": 2,
          "name": "App"
        },
        "appInstances": [
        {
          "instance": {
            "id": 53,
            "name": "Test App - Grails"
          }
        }
      },
      {
        "tier": {
          "id": 5,
          "name": "Herb"
        },
        "appInstances": [
        {
          "instance": {
            "id": 54,
            "name": "Herb"
          }
        }
      }
    ]
  }
}
```
This endpoint retrieves a specific app.

**HTTP Request**

GET https://api.gomorpheus.com/api/apps/:id

### 23.2.3 Create an App

```bash
curl -X POST "https://api.gomorpheus.com/api/apps" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"app":{  "name": "sampleapp",  "description": "A sample app",  "group": {  "id": 1  }  }}'
```

The above command returns JSON structured like getting a single app.

**HTTP Request**

POST https://api.gomorpheus.com/api/apps
JSON App Parameters

23.2.4 Updating an App Name or Description

```
curl -XPUT "https://api.gomorpheus.com/api/apps/2" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"app": {
    "name": "My Sample App",
    "description": "A new description of this app",
}}' 
```

The above command returns JSON structured like getting a single app.

HTTP Request

PUT https://api.gomorpheus.com/api/apps/:id

JSON App Parameters

23.2.5 Add Existing Instance to App

```
curl -XPOST "https://api.gomorpheus.com/api/apps/1/add-instance" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"instanceId": 55, tierName: "App"}'
```

The above command returns JSON structured like getting a single app.

HTTP Request

POST https://api.gomorpheus.com/api/apps/:id/add-instance

JSON Parameters

23.2.6 Remove Instance from App

```
curl -XPOST "https://api.gomorpheus.com/api/apps/1/remove-instance" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"instanceId": 55}'
```

The above command returns JSON structured like getting a single app.

HTTP Request

POST https://api.gomorpheus.com/api/apps/:id/remove-instance
JSON Parameters

23.2.7 Get Security Groups

curl -XGET "https://api.gomorpheus.com/api/apps/1/security-groups" \
-H "Authorization: BEARER access_token"

The above command returns JSON structure like this:

```
{
   "success": true,
   "firewallEnabled": true,
   "securityGroups": [
      {
         "id": 19,
         "accountId": 1,
         "name": "All Tomcat Access",
         "description": "Allow everyone to access Tomcat"
      }
   ]
}
```

This returns a list of all of the security groups applied to an app and whether the firewall is enabled.

HTTP Request

GET https://api.gomorpheus.com/api/apps/:id/security-groups

23.2.8 Set Security Groups

curl -XPOST "https://api.gomorpheus.com/api/apps/1/security-groups" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{ "securityGroupIds": [19, 2] }'

The above command returns JSON structure similar to the ‘get’ of security groups.

HTTP Request

POST https://api.gomorpheus.com/api/apps/:id/security-groups

JSON Parameters

23.2.9 Delete an App

curl -XDELETE "https://api.gomorpheus.com/api/apps/1" \
-H "Authorization: BEARER access_token"

The above command returns JSON structure like this:
Will delete an app. Use removeInstances=on to also delete the instances in the app and all associated monitors and backups.

HTTP Request

DELETE https://api.gomorpheus.com/api/apps/:id

Query Parameters

23.3 Archives

Archives provides a way to store your files and make them available for download by your Scripts and Users. Archives are organized by buckets. Each bucket has a unique name that is used to identify it in URLs and Scripts.

23.3.1 Get All Archive Buckets

curl "https://api.gomorpheus.com/api/archives/buckets" \
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```json
{
    "archiveBuckets": [
    {
        "id": 1,
        "name": "testbucket",
        "description": "a test archive with local storage",
        "storageProvider": {
            "id": 2,
            "name": "testdrive2"
        },
        "owner": {
            "id": 1,
            "name": "root"
        },
        "createdBy": null,
        "isPublic": true,
        "visibility": "private",
        "code": "454ed1af504f",
        "filePath": "morpheus-archives/454ed1af504f/",
        "rawSize": 65154,
        "fileCount": 16,
        "accounts": [
        ],
        "dateCreated": "2017-06-14T14:09:01Z",
    }
    ]
}
```
This endpoint retrieves all archive buckets associated with the account.

**HTTP Request**

GET https://api.gomorpheus.com/api/archives/buckets

**Query Parameters**

**23.3.2 Get a Specific Archive Bucket**

```
curl "https://api.gomorpheus.com/api/archives/buckets/1" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
    "archiveBucket":{
        ...
    }
}
```
This endpoint retrieves a specific archive bucket.

**HTTP Request**

GET https://api.gomorpheus.com/api/archives/buckets/:id

**URL Parameters**

**23.3.3 Create an Archive Bucket**

```
curl -X POST "https://api.gomorpheus.com/api/archives/buckets" 
-H "Authorization: BEARER access_token" 
-H "Content-Type: application/json" 
-d '{
  "archiveBucket": {
    "name": "mybucket",
    "description": "my archive bucket",
    "storageProvider": {
      "id": 2
    },
    "visibility": "private",
    "isPublic": false
  }
}'
```

The above command returns JSON structured like getting a single archive bucket:
HTTP Request

POST https://api.gomorpheus.com/api/archives/buckets

JSON Parameters

23.3.4 Update an Archive Bucket

curl -XPUT "https://api.gomorpheus.com/api/archives/buckets/1" \
  -H "Authorization: BEARER access_token" \
  -H "Content-Type: application/json" \
  -d '{
      "archiveBucket": {
        "description": "our secure file store",
        "isPublic": false
      }
    }'

The above command returns JSON structured like getting a single archive bucket.

HTTP Request

PUT https://api.gomorpheus.com/api/archives/buckets/1

URL Parameters

JSON Parameters

23.3.5 Delete an Archive Bucket

curl -XDELETE "https://api.gomorpheus.com/api/archives/buckets/1" \
  -H "Authorization: BEARER access_token"

The above command returns JSON Structured like this:

```json
{
  "success": true
}
```

Will delete an archive bucket from the system and make it no longer usable.

HTTP Request

DELETE https://api.gomorpheus.com/api/archives/buckets/:id
URL Parameters

23.3.6 Get All Archive Files


The above command returns JSON structured like this:

```json
{
    "parentDirectory": null,
    "archiveFiles": [
        {
            "id": 951,
            "name": "myapp",
            "filePath": "myapp",
            "archiveBucket": {
                "id": 38,
                "name": "mybucket",
                "isPublic": false
            },
            "createdBy": {
                "username": "admin"
            },
            "isDirectory": true,
            "status": "Active",
            "rawSize": 26719,
            "contentType": null,
            "dateCreated": "2018-07-26T19:38:17Z",
            "lastUpdated": "2018-07-26T19:38:17Z"
        },
        {
            "id": 933,
            "name": "readme.txt",
            "filePath": "readme.txt",
            "archiveBucket": {
                "id": 38,
                "name": "mybucket",
                "isPublic": false
            },
            "createdBy": {
                "username": "admin"
            },
            "isDirectory": false,
            "status": "Active",
            "rawSize": 47104,
            "contentType": "text/plain",
            "dateCreated": "2018-07-20T04:07:33Z",
            "lastUpdated": "2018-07-20T04:07:33Z"
        }
    ],
    "archiveBucket": {
        "id": 38,
        "name": "mybucket",
        "description": "a test bucket with local storage",
        "storageProvider": {
            "id": 10,
        }
    }
}
```

(continues on next page)
This endpoint retrieves all files in an archive bucket under the specified `filePath`.

**HTTP Request**

```
GET https://api.gomorpheus.com/api/archives/buckets/:bucket/files/:filePath
```

**URL Parameters**

**Query Parameters**

```
curl "https://api.gomorpheus.com/api/archives/files/954"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
  "archiveFile": {
    "id": 954,
    "name": "articles_controller.rb",
    "filePath": "future/app/controllers/agents_controller.rb",
    "archiveBucket": {
      "id": 38,
      "name": "b10",
      "isPublic": false
    },
    "createdBy": {
```

(continues on next page)
Get details about a specific archive file.

**HTTP Request**

GET https://api.gomorpheus.com/api/archives/files/:fileId

**URL Parameters**

**23.3.7 Upload Archive File**

```bash
curl -XPOST "https://api.gomorpheus.com/api/archives/buckets/mybucket/files/myapp/config/?filename=application.rb" \
-H "Authorization: BEARER access_token" \
--data-binary '@/path/to/file'
```

The above command returns JSON structure like this:

```
{
   "success": true
}
```

Upload a file to the specified archive bucket and file path.

**HTTP Request**

POST https://api.gomorpheus.com/api/archives/buckets/:bucket/files/:filePath

**URL Parameters**

**Query Parameters**

**23.3.8 Download an Archive File**

```bash
curl -XGET "https://api.gomorpheus.com/api/archives/download/mybucket/myapp/config/application.rb" \
-H "Authorization: BEARER access_token"
```

Download an archive file.
The above command returns the contents of the specified file as an attachment with Content-Type dictated by the file.

Download the file as an authorized user with access to the bucket.

**HTTP Request**

GET https://api.gomorpheus.com/api/archives/download/:bucket/:filePath

**URL Parameters**

### 23.3.9 Download a Public Archive File

```bash
```

The above command returns the contents of the file as an attachment with Content-Type dictated by the file.

Files in an archive bucket that has Public URL enabled can be downloaded via this endpoint without any authentication, anonymously.

**HTTP Request**

GET https://api.gomorpheus.com/public-archives/download/:bucket/:filePath

**URL Parameters**

### 23.3.10 Delete Archive File

```bash
curl -XDELETE "https://api.gomorpheus.com/api/archives/files/99" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structure like this:

```
{
  "success": true
}
```

Permanently delete a file or directory.

**HTTP Request**

DELETE https://api.gomorpheus.com/api/archives/files/:fileId

**URL Parameters**

### 23.3.11 Get Archive File Links

23.3. Archives
HTTP Request

curl "https://api.gomorpheus.com/api/archives/files/1/links"
-H "Authorization: BEARER access_token"

```
{
  "archiveFileLinks": [
    {
      "id": 2,
      "secretAccessKey": "6e37727235041746",
      "archiveFile": {
        "id": 1,
        "name": "config.ini",
        "filePath": "config.ini"
      },
      "createdBy": {
        "username": "admin"
      },
      "dateCreated": "2018-09-20T21:15:38Z",
      "lastUpdated": "2018-09-20T21:15:38Z",
      "lastAccessDate": null,
      "expirationDate": null,
      "downloadCount": 0
    },
    {
      "id": 1,
      "secretAccessKey": "6562129e9e546b9",
      "archiveFile": {
        "id": 1,
        "name": "config.ini",
        "filePath": "config.ini"
      },
      "createdBy": {
        "username": "admin"
      },
      "dateCreated": "2018-09-20T21:06:04Z",
      "lastUpdated": "2018-09-20T21:09:26Z",
      "lastAccessDate": null,
      "expirationDate": "2018-09-20T21:26:04Z",
      "downloadCount": 1
    }
  ],
  "meta": {
    "size": 2,
    "total": 2,
    "offset": 0,
    "max": 50
  }
}
```

This endpoint retrieves the links that have been created for the specified file.

**HTTP Request**

GET https://api.gomorpheus.com/api/archives/files/:fileId/links
23.3.12 Create an Archive File Link

```sh
curl -XPOST "https://api.gomorpheus.com/api/archives/files/:fileId/links" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json"
```

The above command returns JSON structured like this:

```
{
  "success": true,
  "secretAccessKey": "45a214fce9a546b9"
}
```

This returns a secret token that can be used to download the file via a public url, without any other authentication or authorization. File links can be set to expire after a certain amount of time.

See Download an Archive File Link

HTTP Request

POST https://api.gomorpheus.com/api/archives/files/:fileId/links

URL Parameters

Query Parameters

23.3.13 Delete an Archive File Link

```sh
curl -XDELETE "https://api.gomorpheus.com/api/archives/files/1/links/1" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json"
```

The above command returns JSON structured like this:

```
{
  "success": true
}
```

This will delete the link from the system, so it can no longer be used.

HTTP Request

DELETE https://api.gomorpheus.com/api/archives/files/:fileId/links/:linkId

URL Parameters

23.3.14 Download an Archive File Link
The above command returns the contents of the file as an attachment with Content-Type dictated by the file.

**Download an archive file**

**HTTP Request**


**URL Parameters**

**23.4 Billing**

Provides API interfaces for viewing billing information by account, zone, instance or server. By default, the information returned is from the beginning of the current month until now. The date range is parameterized but the end date cannot exceed the current date.

**23.4.1 By Account**

The above command returns JSON structured like this:

```json
{
  "success": true,
  "billingInfo": {
    "accountId": 1,
    "name": "morpheus",
    "startDate": "2017-02-01T07:00:00Z",
    "endDate": "2017-02-22T23:03:13Z",
    "priceUnit": "hour",
    "price": 0,
    "cost": 0,
    "zones": [
      {
        "computeServers": [
          {
            "servers": [
              {
                "usages": []
              }
            }
          }
        ],
        "instances": [
          {
            "instances": []
          }
        ]
      }
    ]
  }
}
```

(continues on next page)
Retrieves billing information for the requesting user’s account.

**HTTP Request**

GET https://api.gomorpheus.com/api/billing/account

**Query Parameters**

**23.4.2 For a Sub Account**

Will retrieve billing information for a specific account, if it is the current account or a sub account of the requesting user’s account.

```
curl "https://api.gomorpheus.com/api/billing/account/1" \ 
   -H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
  "success": true,
  "billingInfo": {
    "accountId": 1,
    "name": "morpheus",
    "startDate": "2017-02-01T07:00:00Z",
    "endDate": "2017-02-22T23:03:13Z",
    "priceUnit": "hour",
    "price": 0,
    "cost": 0,
    "zones": [
      {
        "computeServers": [
          {
            "servers": [
              {
                "usages": []
              }]
          }
        },
        "instances": []
      }
    ]
  }
}
```
This endpoint will retrieve a specific account by id if the user has permission to access it.

**HTTP Request**

GET https://api.gomorpheus.com/api/billing/account/:id

**Query Parameters**

23.4.3 For All Zones

```
curl "https://api.gomorpheus.com/api/billing/zones"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
   "success": true,
   "billingInfo": {
      "startDate": "2017-02-01T07:00:00Z",
      "endDate": "2017-02-22T23:03:13Z",
      "priceUnit": "hour",
      "price": 0,
      "cost": 0,
      "zones": [
      {
         "computeServers": [
         {
            "servers": [
            {
               "usages": [ ]
            }
            ]
         }
         ]
      },
      "instances": [
      ]
      ]
   }
}
```
Retrieves billing information for all zones on the requestor’s account.

**HTTP Request**

```
GET https://api.gomorpheus.com/api/billing/zones
```

**Query Parameters**

**23.4.4 For a Specific Zone**

```
curl "https://api.gomorpheus.com/api/billing/zones/1"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
  "success": true,
  "billingInfo": {
    "zoneName": "",
    "zoneId": 1,
    "startDate": "2017-01-01T00:00:00Z",
    "endDate": "2017-01-31T23:59:59Z",
    "priceUnit": "hour",
    "computeServers": {
      "servers": [
        {
          "usages": []
        }
      ],
    },
    "instances": {
      "instances": [
        {
          "containers": {
            "usages": [
          ]
        }
      ]
    }
  }
}
```
"applicablePrices": [ ]

Retrieves billing information for a specific zone in the requestor’s account.

**HTTP Request**

GET https://api.gomorpheus.com/api/billing/zones/:id

**Query Parameters**

23.4.5 For All Servers

```bash
curl "https://api.gomorpheus.com/api/billing/servers"
   -H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
   "success": true,
   "billingInfo": {
      "price": 0,
      "cost": 0,
      "startDate": "2017-03-01T07:00:00Z",
      "endDate": "2017-03-09T22:03:28Z",
      "servers": [ {
         "refType": "computeServer",
         "refId": 1,
         "startDate": "2017-01-01T00:00:00Z",
         "endDate": "2017-01-31T23:59:59Z",
         "cost": 0,
         "price": 0,
         "usages": [ ],
         "numUnits": 0,
         "unit": "hour",
         "name": "name"
      } ]
   }
}
```

Retrieves billing information for all servers on the requestor’s account.
HTTP Request

GET https://api.gomorpheus.com/api/billing/servers

Query Parameters

23.4.6 For a Specific Server

curl "https://api.gomorpheus.com/api/billing/servers/1"
   -H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```
{
   "success": true,
   "billingInfo": {
      "refType": "computeServer",
      "refId": 1,
      "startDate": "2017-01-01T00:00:00Z",
      "endDate": "2017-01-31T23:59:59Z",
      "cost": 0,
      "price": 0,
      "usages": [],
      "numUnits": 0,
      "unit": "hour",
      "name": "name"
   }
}
```

Retrieves billing information for a specific server in the requestor's account.

HTTP Request

GET https://api.gomorpheus.com/api/billing/servers/:id

Query Parameters

23.4.7 For All Instances

curl "https://api.gomorpheus.com/api/billing/instances"
   -H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```
{
   "success": true,
   "billingInfo": {
      "price": 0.0,
      "cost": 0.0,
      "startDate": "2017-01-01T00:00:00Z",
      "endDate": "2017-01-31T23:59:59Z",
      "instances": [  
```

(continues on next page)
Retrieves billing information for all instances on the requestor’s account.

**HTTP Request**

GET https://api.gomorpheus.com/api/billing/instances

**Query Parameters**

23.4.8 For a Specific Instance

```
curl "https://api.gomorpheus.com/api/billing/instances/1"
  -H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
  "success": true,
  "billingInfo": {
    "instanceId": 11,
    "startDate": "2017-01-01T00:00:00Z",
    "endDate": "2017-01-31T23:59:59Z",
    "name": "name",
    "price": 0,
    "cost": 0,
    "containers": [
      {
        "usages": [
          {
            "applicablePrices": [
              
            ],
            "numUnits": 0.0,
            "unit": "hour",
            "name": "name"
          }
        ]
      }
    ]
  }
}
```
Retrieves billing information for a specific server in the requestor’s account.

**HTTP Request**

GET https://api.gomorpheus.com/api/billing/instances/:id

**Query Parameters**

### 23.5 Blueprints

Blueprints are templates for creating apps. They are a set of instance configurations, organized by tier, and scoped by group, cloud and environment.

#### 23.5.1 Get All Blueprints

```bash
curl "https://api.gomorpheus.com/api/blueprints"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
  "blueprints": [
    {
      "id": 135,
      "name": "test",
      "type": "morpheus",
      "description": null,
      "category": null,
      "config": {
        "image": "/assets/apps/template.png",
        "tiers": {
          "Web": {
            "linkedTiers": [],
            "tierIndex": 1,
            "instances": [
              {
                "instance": {
                  "type": "nginx"
                },
                "groups": {
                  "My Group": {
                    "clouds": {
                      "My Cloud": {
                        "instance": {
                          "layout": {
                            "code": "nginx-vmware-1.9-single",
                            "id": 179
                          },
                          "name": "test-nginx-${sequence}",
                          "allowExisting": false,
                          "createUser": "on"
                        }
                      }
                    }
                  }
                }
              }
            ]
          }
        }
      }
    }
  ]
}
```

(continues on next page)
"type": "nginx",
"userGroup": {
  "id": "",
},
"networkInterfaces": [
  {
    "ipMode": "",
    "primaryInterface": true,
    "network": {
      "id": "",
      "hasPool": false
    },
    "networkInterfaceTypeId": 4,
    "networkInterfaceTypeIdName": "VMXNET 3"
  }
],
"volumes": [
  {
    "vId": 255,
    "controllerMountPoint": "46:0:4:0",
    "size": 10,
    "maxIOPS": null,
    "name": "root",
    "rootVolume": true,
    "storageType": 1,
    "datastoreId": "autoCluster",
    "maxStorage": 0
  }
],
"config": {
  "resourcePoolId": "resgroup-123",
  "createUser": true
},
"plan": {
  "code": "vm-1024",
  "id": 76
}
"name": "test",
"templateImage": "",
"type": "morpheus",
"config": {
  "isVpcSelectable": true,
  "isEC2": false
}
},
"visibility": "private",
"resourcePermission": {
  "all": true,
This endpoint retrieves all blueprints.

**HTTP Request**

GET https://api.gomorpheus.com/api/blueprints

**Query Parameters**

```
23.5.2 Get a Specific Blueprint
```

```
curl "https://api.gomorpheus.com/api/blueprints/4" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
   "blueprint": {
      "id": 135,
      "name": "test",
      "type": "morpheus",
      "description": null,
      "category": null,
      "config": {
         "image": "/assets/apps/template.png",
         "tiers": {
            "Web": {
               "linkedTiers": [
               ],
               "tierIndex": 1,
               "instances": [
               {
                  "instance": {
                     "type": "nginx"
                  },
               ],
               "groups": {
                  "My Group": {
                     "clouds": {
                        "My Cloud": {
```

(continues on next page)

23.5. Blueprints
"instance": {
    "layout": {
        "code": "nginx-vmware-1.9-single",
        "id": 179
    },
    "name": "test-nginx-${sequence}",
    "allowExisting": false,
    "createUser": "on",
    "type": "nginx",
    "userGroup": {
        "id": ""
    }
},
"networkInterfaces": [
    {
        "ipMode": "",
        "primaryInterface": true,
        "network": {
            "id": "",
            "hasPool": false
        },
        "networkInterfaceTypeId": 4,
        "networkInterfaceTypeIdName": "VMXNET 3"
    }
],
"volumes": [
    {
        "vId": 255,
        "controllerMountPoint": "46:0:4:0",
        "size": 10,
        "maxIOPS": null,
        "name": "root",
        "rootVolume": true,
        "storageType": 1,
        "datastoreId": "autoCluster",
        "maxStorage": 0
    }
],
"config": {
    "resourcePoolId": "resgroup-123",
    "createUser": true
},
"plan": {
    "code": "vm-1024",
    "id": 76
}
}
This endpoint retrieves a specific blueprint.

HTTP Request

GET https://api.gomorpheus.com/api/blueprints/:id

23.5.3 Create a Blueprint

```
curl -XPOST "https://api.gomorpheus.com/api/blueprints" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{
    "name": "sample",
    "description": "A sample blueprint",
    "type": "morpheus",
    "tiers": {
        "Web": {
            "linkedTiers": [ ],
            "tierIndex": 1,
            "instances": [ { 
                "instance": { 
                    "type": "nginx"
                },
                "groups": { 
                    "My Group": { 
                        "clouds": { 
                            "My Cloud": { 
                                "instance": { 
                                    "layout": { 
                                        "code": "nginx-vmware-1.9-single",
                                        "id": 179
                                    },
                                    "name": "test-nginx-${sequence}",
                                    "allowExisting": false,
                                    "createUser": "on",
                                    "type": "nginx",
                                    "type": "nginx"
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```
"userGroup": {  
  "id": ""  
},  

"networkInterfaces": [  
  {  
    "ipMode": "",  
    "primaryInterface": true,  
    "network": {  
      "id": "",  
      "hasPool": false  
    },  
    "networkInterfaceTypeId": 4,  
    "networkInterfaceTypeIdName": "VMXNET 3"  
  }  
],  

"volumes": [  
  {  
    "vId": 255,  
    "controllerMountPoint": "46:0:4:0",  
    "size": 10,  
    "maxIOPS": null,  
    "name": "root",  
    "rootVolume": true,  
    "storageType": 1,  
    "datastoreId": "autoCluster",  
    "maxStorage": 0  
  }  
],  

"config": {  
  "resourcePoolId": "resgroup-123",  
  "createUser": true  
},  

"plan": {  
  "code": "vm-1024",  
  "id": 76  
}

The above command returns JSON structured like getting a single blueprint.

**HTTP Request**

`POST https://api.gomorpheus.com/api/blueprints`

**JSON Blueprint Parameters**
Blueprint Tiers Configuration

The blueprint tiers can be structured so that instance configurations are scoped to a specific environment, group and/or cloud. The environments key is the environment name. The groups key is the group name. The clouds key is the cloud name. The order of scoping must always be done in the order: environments, groups, and then clouds.

Example:

```
{
    "name": "sample blueprint",
    "type": "morpheus",
    "tiers": {
        "App": {
            "tierIndex": 1,
            "linkedTiers": [],
            "instances": [
                {
                    "instance": {
                        "type": "activemq"
                    },
                    "environments": {
                        "Dev": {
                            "groups": {
                                "My Group": {
                                    "clouds": {
                                        "My Cloud": {
                                            "instance": {
                                                "layout": {
                                                    "code": "nginx-vmware-1.9-single",
                                                    "id": 179
                                                },
                                                "name": "test-nginx-$sequence"
                                            },
                                            "plan": {
                                                "code": "vm-1024",
                                                "id": 76
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            ]
        }
    }
}
```

23.5.4 Updating a Blueprint

```
curl -XPUT "https://api.gomorpheus.com/api/blueprints/2" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{
    "name": "sample",
}
```
"description": "A sample nginx blueprint",
"type": "morpheus",
"tiers": {
  "Web": {
    "linkedTiers": [
    ],
    "tierIndex": 1,
    "instances": [
    {
      "instance": {
        "type": "nginx"
      },
      "groups": {
        "My Group": {
          "clouds": {
            "My Cloud": {
              "instance": {
                "layout": {
                  "code": "nginx-vmware-1.9-single",
                  "id": 179
                },
                "name": "test-nginx-${sequence}"
              }
            }
          }
        }
      },
      "networkInterfaces": [
      {
        "ipMode": "",
        "primaryInterface": true,
        "network": {
          "id": "",
          "hasPool": false
        },
        "networkInterfaceTypeId": 4,
        "networkInterfaceTypeIdName": "VMXNET 3"
      }
      ],
      "volumes": [
      {
        "vId": 255,
        "controllerMountPoint": "46:0:4:0",
        "size": 10,
        "maxIOPS": null,
        "name": "root",
        "rootVolume": true,
        "storageType": 1,
        "datastoreId": "autoCluster",
        "maxStorage": 0
      }
      ],
      "config": {
        "resourcePoolId": "resgroup-123",
        "resourcePoolName": "autoCluster",
        "resourcePoolType": "Cluster",
        "resourcePoolTypeNamespace": "autoCluster",
        "resourcePoolTypeNamespaceId": 179,
        "resourcePoolTypeNamespaceName": "autoCluster"
      }
    }
  }
  ]
}
The above command returns JSON structured like getting a single blueprint.

**HTTP Request**

PUT https://api.gomorpheus.com/api/blueprints/:id

**JSON Blueprint Parameters**

Same as *Create*.

This overwrites the entire config, so the entire blueprint config should be passed.

### 23.5.5 Update Blueprint Permissions

```bash
curl -XPOST "https://api.gomorpheus.com/api/blueprints/:id/update-permissions" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{ "resourcePermission": { 
   "all":false, 
   "sites": [ 
     {"id": 1} 
   ] 
 }}'
```

The above command returns JSON structured like getting a single blueprint.

**HTTP Request**

POST https://api.gomorpheus.com/api/blueprints/:id/update-permissions
### JSON Parameters

#### 23.5.6 Update Blueprint Image

```
curl -XPOST "https://api.gomorpheus.com/api/blueprints/1/image" \
-H "Authorization: BEARER access_token" \
-F 'templateImage=@filename'
```

The above command returns JSON structured like getting a single blueprint.

**HTTP Request**

POST https://api.gomorpheus.com/api/blueprints/:id/image

**Parameters**

Upload a new logo image. Expects multipart form data as the request format, not JSON.

#### 23.5.7 Delete a Blueprint

```
curl -XDELETE "https://api.gomorpheus.com/api/blueprints/1" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structure like this:

```
{
  "success": true
}
```

**HTTP Request**

DELETE https://api.gomorpheus.com/api/blueprints/:id

#### 23.6 Boot Scripts

Boot Scripts are used in the Image Builder service. See *Image Builds*

#### 23.6.1 Get All Boot Scripts

```
curl "https://api.gomorpheus.com/api/boot-scripts" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:
This endpoint retrieves all boot scripts associated with the account.

**HTTP Request**

GET https://api.gomorpheus.com/api/boot-scripts

**Query Parameters**

**23.6.2 Get a Specific Boot Script**

```
curl "https://api.gomorpheus.com/api/boot-scripts/1" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
    "bootScript": {
        "id": 1,
        "account": {
            "id": 1,
            "name": "root"
        },
        "fileName": "debian standard",
        "description": null,
        "content": "...",
        "createdBy": {
            "username": "admin"
        },
        "visibility": "private"
    }
}
```

(continues on next page)
This endpoint retrieves a specific boot script.

**HTTP Request**

GET https://api.gomorpheus.com/api/boot-scripts/:id

**URL Parameters**

23.6.3 Create a Boot Script

```
curl -XPOST "https://api.gomorpheus.com/api/boot-scripts" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{
  "bootScript": {
    "fileName": "debian standard",
    "content": "<esc><wait>install <wait> preseed/url=<%=preseedUrl%> <wait>debian- installer=en_US.UTF-8 <wait>auto <wait>locale=en_US.UTF-8 <wait>kbd-chooser/ method=us <wait>keyboard-configuration/xkb-keymap=us <wait>netcfg/get_hostname= <wait>%=container.hostname% <wait>netcfg/get_domain=morpheusdata.com <wait>fb=false <wait>console-setup/ask_detect=false <wait>console-keymaps-at/keymap=us <wait>grub-installer/bootdev=/dev/sda <wait><enter> <wait>
  }
}
''
```

The above command returns JSON structured like getting a single boot script:

**HTTP Request**

POST https://api.gomorpheus.com/api/boot-scripts

**JSON Parameters**

23.6.4 Update a Boot Script

```
curl -XPUT "https://api.gomorpheus.com/api/boot-scripts/1" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{
  "bootScript": {
    "fileName": "debian default"
  }
}
''
```

The above command returns JSON structured like getting a single boot script:
HTTP Request

PUT https://api.gomorpheus.com/api/boot-scripts/1

URL Parameters

JSON Parameters

Same as Create.

23.6.5 Delete a Boot Script

```
curl -XDELETE "https://api.gomorpheus.com/api/boot-scripts/1" \ 
-H "Authorization: BEARER access_token"
```

The above command returns JSON Structured like this:

```
{
  "success": true
}
```

Will delete a boot script from the system and make it no longer usable.

HTTP Request

DELETE https://api.gomorpheus.com/api/boot-scripts/:id

URL Parameters

23.7 Check Types

A set of APIs for fetching a list of available check types is also provided. This API can make it useful for associating a check type code to an ID for check GET and POST requests.

23.7.1 Get All Check Types

```
curl "https://api.gomorpheus.command/api/monitoring/check-types" 
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this

```
{
  "checkTypes": [
    {
      "id": 1,
      "code": "webGetCheck",
      "createIncident": true,
      "defaultInterval": 60000,
      "iconPath": null,
      etc.
    }
  ]
}
```

(continues on next page)
"iconType": "upload",
"inUptime": true,
"metricName": "response",
"name": "Web Check",
"tunnelSupported": true
},
{
"id": 2,
"code": "mysqlCheck",
"createIncident": true,
"defaultInterval": 60000,
"iconPath": null,
"iconType": "upload",
"inUptime": true,
"metricName": "result",
"name": "MySQL Check",
"tunnelSupported": true
},
{
"id": 3,
"code": "mongoCheck",
"createIncident": true,
"defaultInterval": 300000,
"iconPath": null,
"iconType": "upload",
"inUptime": true,
"metricName": "result",
"name": "Mongo Check",
"tunnelSupported": true
},
{
"id": 4,
"code": "elasticSearchCheck",
"createIncident": true,
"defaultInterval": 60000,
"iconPath": null,
"iconType": "upload",
"inUptime": true,
"metricName": "cluster status",
"name": "Elastic Search Check",
"tunnelSupported": true
},
{
"id": 5,
"code": "riakCheck",
"createIncident": true,
"defaultInterval": 300000,
"iconPath": null,
"iconType": "upload",
"inUptime": true,
"metricName": "write time",
"name": "Riak Check",
"tunnelSupported": true
},
{
"id": 6,
"code": "redisCheck",
"createIncident": true,
"createIncident": true,
"defaultInterval": 300000,
"iconPath": null,
"iconType": "upload",
"inUptime": true,
"metricName": "key count",
"name": "Redis Check",
"tunnelSupported": true
},
{
"id": 7,
"code": "rabbitCheck",
"createIncident": true,
"defaultInterval": 60000,
"iconPath": null,
"iconType": "upload",
"inUptime": true,
"metricName": "queue count",
"name": "Rabbit MQ Check",
"tunnelSupported": true
},
{
"id": 9,
"code": "postgresCheck",
"createIncident": true,
"defaultInterval": 300000,
"iconPath": null,
"iconType": "upload",
"inUptime": true,
"metricName": "result",
"name": "Postgres Check",
"tunnelSupported": true
},
{
"id": 10,
"code": "sqlCheck",
"createIncident": true,
"defaultInterval": 300000,
"iconPath": null,
"iconType": "upload",
"inUptime": true,
"metricName": "result",
"name": "Microsoft SQL Server",
"tunnelSupported": true
},
{
"id": 11,
"code": "socketCheck",
"createIncident": true,
"defaultInterval": 60000,
"iconPath": null,
"iconType": "upload",
"inUptime": true,
"metricName": "response",
"name": "Socket Check",
"tunnelSupported": true
},

HTTP Request

GET https://api.gomorpheus.com/api/monitoring/check-types

23.7.2 Get Specific Check Type

curl "https://api.gomorpheus.com/api/monitoring/check-types/10"
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this

```json
{
   "success": true,
   "checkType": {
      "id": 10,
      "code": "sqlCheck",
      "createIncident": true,
      "defaultInterval": 300000,
      "iconPath": null,
      "iconType": "upload",
      "inUptime": true,
      "metricName": "result",
      "name": "Microsoft SQL Server",
      "tunnelSupported": true
   }
}
```
HTTP Request

GET https://api.gomorpheus.com/api/monitoring/check-types/1

23.8 Compute Servers

A Compute Server is either a bare metal machine or virtual machine that is provisioned into morpheus via Chef. These servers are setup as Docker Hosts and used to provision containers into. They also run the morphd agent which reports server statistics and logs back to the morpheus stack.

23.8.1 Get All Servers

```bash
curl "https://api.gomorpheus.com/api/servers"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
    "servers": [
        {
            "id": 1,
            "accountId": 1,
            "name": "dre-matrix-1",
            "visibility": "public",
            "description": "dre-matrix-1",
            "zoneId": 1,
            "siteId": 1,
            "sshHost": "10.100.54.2",
            "internalIp": "10.100.54.2",
            "externalIp": "10.100.54.2",
            "sshPort": 22,
            "volumeId": null,
            "platform": null,
            "platformVersion": null,
            "sshUsername": "vagrant",
            "sshPassword": "***",
            "osDevice": "/dev/sda",
            "dataDevice": "/dev/sdb",
            "apiKey": "a3914182-0f2f-4e9c-a6d2-63822747b9cd",
            "softwareRaid": false,
            "config": null,
            "capacityInfo": {
                "class": "com.morpheus.ComputeCapacityInfo",
                "id": 1,
                "maxCores": null,
                "maxMemory": 2099228672,
                "maxStorage": 42945478656,
                "server": {
                    "class": "com.morpheus.ComputeServer",
                    "id": 1
                },
                "usedMemory": 1073741824,
                "usedStorage": 3221225472
            }
        }
    ]
}
```

(continues on next page)
"dateCreated": "2015-06-09T12:43:51Z",
"lastUpdated": "2015-06-09T12:47:42Z",
"status": "provisioned",
"interfaces": [
{
"id": 1,
"active": true,
"dhcp": true,
"ipAddress": "10.100.54.2",
"ipSubnet": null,
"ipv6Address": null,
"ipv6Subnet": null,
"name": "eth1",
"network": null,
"networkPosition": null,
"primaryInterface": true,
"publicIpAddress": null,
"publicIpv6Address": null,
"server": {
"id": 1
}
},
"zone": {
"id": 1,
"accountId": 1,
"groupId": 1,
"name": "Davids Laptop",
"description": "My Laptop Vagrant",
"location": null,
"visibility": "public",
"zoneTypeId": 1
}
],
{
"id": 2,
"accountId": 1,
"name": "dre-matrix-2",
"visibility": "public",
"description": "dre-matrix-2",
"zoneId": 1,
"siteId": 1,
"sshHost": "10.100.54.3",
"internalIp": "10.100.54.3",
"externalIp": "10.100.54.3",
"sshPort": 22,
"volumeId": null,
"platform": null,
"platformVersion": null,
"sshUsername": "vagrant",
"sshPassword": "****",
"osDevice": "/dev/sda",
"dataDevice": "/dev/sdb",
(continues on next page)
"apiKey": "c3c12af8-1db2-44b3-930d-87f914b14577",
"softwareRaid": false,
"config": null,
"capacityInfo": {
  "id": 2,
  "maxCores": null,
  "maxMemory": 2099228672,
  "maxStorage": 42945478656,
  "server": {
    "id": 2,
  },
  "usedMemory": 1073741824,
  "usedStorage": 3221225472
},
"dateCreated": "2015-06-09T14:07:57Z",
"lastUpdated": "2015-06-09T14:17:51Z",
"lastStats": "{\"cpuIdleTime\":130016650,\"cpuSystemTime\":1041990,\n\"cpuTotalTime\":131172760,\"cpuUsage\":0.1677870750427246,\"cpuUserTime\":114120,\n\"freeMemory\":215248896,\"freeSwap\":0,\"ts\":\"2015-06-10T13:18:33+0000\",\n\"usedMemory\":1883979776,\"usedSwap\":0}",
"status": "provisioned",
"interfaces": [
  {
    "id": 2,
    "active": true,
    "dhcp": true,
    "ipAddress": "10.100.54.3",
    "ipSubnet": null,
    "ipv6Address": null,
    "ipv6Subnet": null,
    "name": "eth1",
    "network": null,
    "networkPosition": null,
    "primaryInterface": true,
    "publicIpAddress": null,
    "publicIpv6Address": null,
    "server": {
      "id": 2
    }
  }
],
"zone": {
  "id": 1,
  "accountId": 1,
  "groupId": 1,
  "name": "Davids Laptop",
  "description": "My Laptop Vagrant",
  "location": null,
  "visibility": "public",
  "zoneTypeId": 1
}
},
"serverCount": 2,
"stats": {
  "1": {
    "usedStorage": 48861184,
This endpoint retrieves all servers and their JSON encoded configuration attributes based on check type. Server data is encrypted in the database.

**HTTP Request**

GET https://api.gomorpheus.com/api/servers

**Query Parameters**

23.8.2 Get a Specific Server

curl "https://api.gomorpheus.com/api/servers/1" \
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```json
{
"success": true,
"server": {
"id": 1,
"accountId": 1,
"name": "dre-matrix-1",
"visibility": "public",
"description": "dre-matrix-1",
"zoneId": 1,
"siteId": 1,
"sshHost": "10.100.54.2",
"internalIp": "10.100.54.2",
"externalIp": "10.100.54.2",
"sshPort": 22,
"volumeId": null,
"platform": null,
"platformVersion": null,
"sshUsername": "vagrant",
"sshPassword": "****",
"osDevice": "/dev/sda",
```
"dataDevice": "/dev/sdb",
"apiKey": "a3914182-0f2f-4e9c-a6d2-63822747b9cd",
"softwareRaid": false,
"config": null,
"capacityInfo": {
  "class": "com.morpheus.ComputeCapacityInfo",
  "id": 1,
  "maxCores": null,
  "maxMemory": 2099228672,
  "maxStorage": 42945478656,
  "server": {
    "class": "com.morpheus.ComputeServer",
    "id": 1
  },
  "usedMemory": 1073741824,
  "usedStorage": 3221225472
},
"dateCreated": "2015-06-09T12:43:51Z",
"lastUpdated": "2015-06-09T12:47:42Z",
"lastStats": {
  "cpuIdleTime": 131294520,
  "cpuSystemTime": 76390,
  "cpuTotalTime": 131476290,
  "cpuUsage": 0.10046958923339844,
  "cpuUserTime": 105380,
  "freeMemory": 91181056,
  "freeSwap": 0,
  "ts": "2015-06-10T13:14:45+0000",
  "usedMemory": 2008047616,
  "usedSwap": 0
},
"status": "provisioned",
"interfaces": [
  {
    "id": 1,
    "active": true,
    "dhcp": true,
    "ipAddress": "10.100.54.2",
    "ipSubnet": null,
    "ipv6Address": null,
    "ipv6Subnet": null,
    "name": "eth1",
    "network": null,
    "networkPosition": null,
    "primaryInterface": true,
    "publicIpAddress": null,
    "publicIpv6Address": null,
    "server": {
      "id": 1
    }
  }
],
"zone": {
  "id": 1,
  "accountId": 1,
  "groupId": 1,
  "name": "Davids Laptop",
  "description": "My Laptop Vagrant",
  "location": null,
  "visibility": "public",
  "zoneTypeId": 1
}
This endpoint retrieves a specific server.

**HTTP Request**

GET https://api.gomorpheus.com/api/servers/:id

**URL Parameters**

### 23.8.3 Get Available Service Plans for a Server

```bash
curl -X GET "https://api.gomorpheus.com/api/servers/service-plans?zoneId=2&serverTypeId=60" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structure like this:

```json
{
  "plans": [
    {
      "id": 75,
      "name": "1 CPU, 512MB Memory",
      "value": 75,
      "code": "vm-512",
      "maxStorage": 10737418240,
      "maxMemory": 536870912,
      "maxCpu": 1,
      "maxCores": 1,
      "customCpu": false,
      "customMaxMemory": false,
      "customMaxStorage": true,
      "customMaxDataStorage": true,
      "customCoresPerSocket": false,
      "storageTypes": [
        {
          "id": 1,
          "editable": false,
          "optionTypes": [

          ],
          "displayName": "Disk",
          "volumeType": "disk",
          "minStorage": null,
          "deletable": false,
          "defaultType": true,
          "createDatastore": null,
          "Resizable": false,
          "storageType": null,
          "allowSearch": true,
          "volumeOptionSource": null,
          "displayName": "Disk",
          "minIOPS": null,
          "maxIOPS": null,
          "hasDatastore": true,
        }
      ]
    }
  ]
}
```
"customSize": true,
"autoDelete": true,
"name": "Standard",
"configurableIOPS": false,
"customLabel": true,
"enabled": true,
"description": "Standard",
"volumeCategory": "disk",
"externalId": null,
"maxStorage": null
},
"rootStorageTypes": [
{
"id": 1,
"editable": false,
"optionTypes": [
],
"displayOrder": 1,
"code": "standard",
"volumeType": "disk",
"minStorage": null,
"deletable": false,
"defaultType": true,
"createDatastore": null,
"resizable": false,
"storageType": null,
"allowSearch": true,
"volumeOptionSource": null,
"displayName": "Disk",
"minIOPS": null,
"maxIOPS": null,
"hasDatastore": true,
"customSize": true,
"autoDelete": true,
"name": "Standard",
"configurableIOPS": false,
"customLabel": true,
"enabled": true,
"description": "Standard",
"volumeCategory": "disk",
"externalId": null,
"maxStorage": null
}
],
"addVolumes": true,
"customizeVolume": true,
"rootDiskCustomizable": true,
"noDisks": false,
"hasDatastore": true,
"minDisk": 0,
"maxDisk": null,
"lvmSupported": null,
"datastores": {
"cluster": [
]}}
This returns a list of all of the service plans available for a server type. The response includes details about the plans and their configuration options. The parameters `zoneId` and `serverTypeId` are required.

This endpoint can be used to get the list of plans available for provisioning a new server or resizing a server.

**HTTP Request**

GET https://api.gomorpheus.com/api/servers/service-plans
Query Parameters

23.8.4 Provision a Server

```bash
curl -XPOST "https://api.gomorpheus.com/api/servers" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{ "server": { 
  "name": "dre-matrix-3", 
  "description": "dre-matrix-3", 
  "zone": {"id":1}, 
  "sshHost": "10.100.54.4", 
  "sshUsername": "vagrant", 
  "sshPassword": "vagrant", 
  "dataDevice": "/dev/sdb"
 ),
  "network": { 
    "name": "eth1"
  }}'
```

The above command returns a similar JSON structure when submitting a GET request for a single check

HTTP Request

POST https://api.gomorpheus.com/api/servers

JSON Server Parameters

23.8.5 Updating a Server

```bash
curl -XPUT "https://api.gomorpheus.com/api/servers/1" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{ "server": { 
  "name": "dre-matrix-3", 
  "description": "dre-matrix-3" 
}}'
```

The above command returns a similar JSON structure when submitting a GET request for a single check

HTTP Request

PUT https://api.gomorpheus.com/api/servers/:id

JSON Server Parameters

23.8.6 Install Agent
curl -XPUT "https://api.gomorpheus.com/api/servers/1/install-agent" \
   -H "Authorization: BEARER access_token" \
   -H "Content-Type: application/json" \
   -d '{ "server": { 
      "sshUsername": "admin",
      "sshPassword": "asafepassword",
      "serverOs": { "id": 1 }
   }}' \

   The above command returns JSON structure like this:

   ```json
   {
     "success": true
   }
   ```

   This will make the server a managed server, and install the agent.

   **HTTP Request**

   PUT https://api.gomorpheus.com/api/servers/:id/install-agent

   **JSON Server Parameters**

   23.8.7 Upgrade Agent

   curl -XPUT "https://api.gomorpheus.com/api/servers/1/upgrade" \
   -H "Authorization: BEARER access_token"

   The above command returns JSON structure like this:

   ```json
   {
     "success": true
   }
   ```

   This will upgrade the version of the install installed on the server.

   **HTTP Request**

   PUT https://api.gomorpheus.com/api/servers/:id/upgrade

   23.8.8 Resize a Server

   curl -XPUT "https://api.gomorpheus.com/api/servers/1/resize" \
   -H "Authorization: BEARER access_token" \
   -H "Content-Type: application/json" \
   -d '{ "server": { 
      "id": 82,
      "plan": { 
      "id": 76
   }' \

   (continues on next page)
The above command returns JSON structure like this:

```
{
  "success": true
}
```

Will resize a server asynchronously.

**HTTP Request**

PUT https://api.gomorpheus.com/api/servers/:id/resize

**JSON Server Parameters**

**23.8.9 Delete a Server**

```
curl -XDELETE "https://api.gomorpheus.com/api/servers/1" \
  -H "Authorization: BEARER access_token"
```

The above command returns JSON structure like this:

```
{
  "success": true
}
```

Will delete a server asynchronously and remove from the hosted chef system.

**HTTP Request**

DELETE https://api.gomorpheus.com/api/servers/:id

**Query Parameters**

**23.9 Compute Server Types**

A Compute Server Type is the description of the technology (bare metal or virtual) being deployed onto.
NOTE: A Server Type in the API is equivalent to a Host Type within the morpheus UI.

23.9.1 Get All Server Types

```
curl "https://api.gomorpheus.com/api/server-types"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
   "serverTypes": [
      {
         "id": 19,
         "code": "softlayerVm",
         "name": "Softlayer Instance",
         "description": "",
         "platform": "linux",
         "nodeType": "morpheus-vm-node",
         "managed": true,
         "enabled": true,
         "vmHypervisor": false,
         "containerHypervisor": false,
         "displayOrder": 0,
         "selectable": false,
         "controlPower": true,
         "controlSuspend": true,
         "hasAgent": true,
         "creatable": false,
         "optionTypes": [
         ]
      },
      {
         "id": 23,
         "code": "amazonVm",
         "name": "Amazon Instance",
         "description": "",
         "platform": "linux",
         "nodeType": "morpheus-vm-node",
         "managed": true,
         "enabled": true,
         "vmHypervisor": false,
         "containerHypervisor": false,
         "displayOrder": 0,
         "selectable": false,
         "controlPower": true,
         "controlSuspend": true,
         "hasAgent": true,
         "creatable": false,
         "optionTypes": [
         ]
      },
      {
         "id": 31,
         "code": "vmwareVm",
         "name": "VMware Instance",
         "description": "",
         "platform": "vmware",
         "nodeType": "morpheus-vm-node",
         "managed": true,
         "enabled": true,
         "vmHypervisor": false,
         "containerHypervisor": false,
         "displayOrder": 0,
         "selectable": false,
         "controlPower": true,
         "controlSuspend": true,
         "hasAgent": true,
         "creatable": false,
         "optionTypes": [
         ]
      }
   ]
}
```
"name":"Vmware Instance",
"description":"",
"platform":"linux",
"nodeType":"morpheus-vm-node",
"managed":true,
"enabled":true,
"vmHypervisor":false,
"containerHypervisor":false,
"displayOrder":0,
"selectable":false,
"controlPower":true,
"controlSuspend":true,
"hasAgent":true,
"creatable":false,
"optionTypes":"[
]
},
{
"id":39,
"code":"nutanixVm",
"name":"Nutanix Instance",
"description":"",
"platform":"linux",
"nodeType":"morpheus-vm-node",
"managed":true,
"enabled":true,
"vmHypervisor":false,
"containerHypervisor":false,
"displayOrder":0,
"selectable":false,
"controlPower":true,
"controlSuspend":true,
"hasAgent":true,
"creatable":false,
"optionTypes":"[
]
},
{
"id":43,
"code":"xenserverVm",
"name":"Xen Instance",
"description":"",
"platform":"linux",
"nodeType":"morpheus-vm-node",
"managed":true,
"enabled":true,
"vmHypervisor":false,
"containerHypervisor":false,
"displayOrder":0,
"selectable":false,
"controlPower":true,
"controlSuspend":true,
"hasAgent":true,
"creatable":false,
"optionTypes":"[
]"}
},
],

{
  "id": 14,
  "code": "metapodLinux",
  "name": "Metapod Linux Node",
  "description": ",",
  "platform": "linux",
  "nodeType": "morpheus-node",
  "managed": true,
  "enabled": true,
  "vmHypervisor": false,
  "containerHypervisor": false,
  "displayOrder": 2,
  "selectable": false,
  "controlPower": true,
  "controlSuspend": true,
  "hasAgent": true,
  "creatable": true,
  "optionTypes": [
    {
      "name": "osUsr",
      "type": "text",
      "defaultValue": null,
      "placeHolder": null,
      "required": true,
      "fieldName": "osUsr",
      "fieldContext": "config"
    },
    {
      "name": "flavorId",
      "type": "selectOsFlavor",
      "defaultValue": null,
      "placeHolder": null,
      "required": true,
      "fieldName": "flavorId",
      "fieldContext": "config"
    }
  ]
},

{
  "id": 8,
  "code": "openstackLinux",
  "name": "Openstack Linux Node",
  "description": ",",
  "platform": "linux",
  "nodeType": "morpheus-node",
  "managed": true,
  "enabled": true,
  "vmHypervisor": false,
  "containerHypervisor": false,
  "displayOrder": 4,
  "selectable": false,
  "controlPower": true,
  "controlSuspend": true,
  "hasAgent": true,
  "creatable": true,
  "optionTypes": ["name": "osUsr",
      "type": "text",
      "defaultValue": null,
      "placeHolder": null,
      "required": true,
      "fieldName": "osUsr",
      "fieldContext": "config"
    },
    {
      "name": "flavorId",
      "type": "selectOsFlavor",
      "defaultValue": null,
      "placeHolder": null,
      "required": true,
      "fieldName": "flavorId",
      "fieldContext": "config"
    }
  ]
}
23.9. Compute Server Types
HTTP Request

GET https://api.gomorpheus.com/api/server-types

Query Parameters

23.9.2 Get a Specific Server Type

curl "https://api.gomorpheus.com/api/server-types/1" \
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this:
This endpoint will retrieve a specific server type by id

**HTTP Request**

GET https://api.gomorpheus.com/api/server-types/:id

**URL Parameters**

**23.10 Compute Zones (Clouds)**

**NOTE:** A Zone in the API is equivalent to a Cloud within the Morpheus UI.

Zones are a means of zoning various servers based on provisioning type or subnets. Typically a Zone belongs to a zone and a zone can have many zones. There are several supported zone types that can be used for hardware/vm procurement such as the OpenStack zone type. The zone holds the credentials necessary to provision virtual machines on the open stack api. Amazon is another openstack zone type currently in the works. Of course, we also have the Standard zone type which allows for manual vm procurement.

**23.10.1 Get All Zones**

```
curl "https://api.gomorpheus.com/api/zones"
   -H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
   "zones": [
      {
         "id": 1,
         "code": "softlayerVm",
         "name": "Softlayer Instance",
         "description": "",
         "platform": "linux",
         "nodeType": "morpheus-vm-node",
         "managed": true,
         "enabled": true,
         "vmHypervisor": false,
         "containerHypervisor": false,
         "displayOrder": 0,
         "selectable": false,
         "controlPower": true,
         "controlSuspend": true,
         "hasAgent": true,
         "creatable": false,
         "optionTypes": []
      }
   ]
}
```
This endpoint retrieves all zones and a list of zones associated with the zone by id.

**HTTP Request**

GET https://api.gomorpheus.com/api/zones

**Query Parameters**

23.10.2 Get a Specific Zone

```bash
curl "https://api.gomorpheus.com/api/zones/1" \  
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
   "success": true,
   "zone": {
      "id": 1,
      "accountId": 1,
      "groupId": 1,
      "name": "Davids Laptop",
      "description": "My Laptop Vagrant",
      "location": null,
      "visibility": "public",
      "zoneTypeId": 1
   }
}
```

This endpoint retrieves a specific zone.

**HTTP Request**

GET https://api.gomorpheus.com/api/zones/:id
URL Parameters

23.10.3 Create a Zone

```
curl -XPOST "https://api.gomorpheus.com/api/zones" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"zone":{
   "name": "My Zone",
   "code": "myzone",
   "description": "My description",
   "location": "US EAST",
   "zoneType": {"code": "standard"},
   "groupId": 1
}}'
```

The above command returns JSON structured like getting a single zone:

HTTP Request

POST https://api.gomorpheus.com/api/zones

JSON Check Parameters

Additional properties are dynamic for the most part depending on the zone/cloud type. To determine what these are please look at the `optionTypes` list on the `ZoneType` record.

23.10.4 Updating a Zone

```
curl -XPUT "https://api.gomorpheus.com/api/zones/1" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"zone":{
   "name": "My Zone",
   "description": "My description",
   "location": "US EAST",
   "zoneType": {"code": "standard"},
   "groupId": 1,
   "config": null
}}'
```

The above command returns JSON structured like getting a single zone:

HTTP Request

PUT https://api.gomorpheus.com/api/zones/:id

JSON Check Parameters

Additional properties are dynamic for the most part depending on the zone/cloud type. To determine what these are please look at the `optionTypes` list on the `ZoneType` record.
23.10.5 Delete a Zone

```
curl -XDELETE "https://api.gomorpheus.com/api/zones/1" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON Structured like this:

```
{
  "success": true
}
```

If a zone has zones or servers still tied to it, a delete action will fail.

**HTTP Request**

DELETE https://api.gomorpheus.com/api/zones/:id

23.10.6 Get Security Groups

```
curl -XGET "https://api.gomorpheus.com/api/zones/1/security-groups" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structure like this:

```
{
  "success": true,
  "firewallEnabled": true,
  "securityGroups": [
    {
      "id": 19,
      "accountId": 1,
      "name": "All Tomcat Access",
      "description": "Allow everyone to access Tomcat"
    }
  ]
}
```

This returns a list of all of the security groups applied to a zone and whether the firewall is enabled.

**HTTP Request**

GET https://api.gomorpheus.com/api/zones/:id/security-groups

23.10.7 Set Security Groups

```
curl -XPOST "https://api.gomorpheus.com/api/zones/1/security-groups" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{ "securityGroupIds": [19, 2] }'
```

The above command returns JSON structure similar to the ‘get’ of security groups.
HTTP Request

POST https://api.gomorpheus.com/api/zones/:id/security-groups

JSON Parameters

23.11 Compute Zone (Cloud) Types

Provides a means to find out which zone types are available for zone provisioning and in the future what config properties are required.

NOTE: A Zone Type in the API Is equivalent to a Cloud Type within the morpheus UI.

23.11.1 Get All Zone Types

curl "https://api.gomorpheus.com/api/zone-types"
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this

```json
{
   "zoneTypes": [
      {
         "id": 3,
         "name": "Amazon",
         "code": "amazon",
         "description": "amazon zone",
         "serverTypes": [],
         "optionTypes": []
      },
      {
         "id": 2,
         "name": "Openstack",
         "code": "openstack",
         "description": "openstack zone",
         "serverTypes": [],
         "optionTypes": []
      },
      {
         "id": 1,
         "name": "Standard",
         "code": "standard",
         "description": "Standard zone - manually managed servers or virtual machines",
         "serverTypes": [],
         "optionTypes": []
      }
   ]
}
```

HTTP Request

GET https://api.gomorpheus.com/api/zone-types
Query Parameters

### 23.11.2 Get Specific Zone Type

```bash
curl "https://api.gomorpheus.com/api/zone-types/1"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this

```json
{
  "success": true,
  "zoneType": {
    "id": 1,
    "name": "Standard",
    "code": "standard",
    "description": "Standard zone - manually managed servers or virtual machines"
  },
  "serverTypes": [],
  "optionTypes": []
}
```

### HTTP Request

GET https://api.gomorpheus.com/api/zone-types/:id

---

**23.12 Cypher**

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.

### 23.12.1 Cypher Authentication

The cypher api endpoints allow for authentication via an special headers or the standard `Authentication: bear access_token`. Instead of an access token, an execution lease token can be used to authenticate. An execution lease will be issued by Morpheus for certain tasks, such as Ansible, which can then use the token to read cypher keys.

Cypher has the following headers and url parameters available for authentication:

### 23.12.2 List Cypher Keys

```bash
curl "https://api.gomorpheus.com/api/cypher/v1?list=true"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:
This endpoint retrieves all cypher keys associated with the account, or user.

**HTTP Request**

GET https://api.gomorpheus.com/api/cypher/v1/:key?list=true

**URL Parameters**

**Query Parameters**

23.12.3 Read a Cypher Key

```
curl "https://api.gomorpheus.com/api/cypher/v1/secret/foo" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:
This endpoint retrieves a specific cypher key. The value of the key is decrypted and returned as data. It may be a String or an object with many "key":"value" pairs. The type depends on the cypher engine's capabilities and what type of data was written to the key. For example the secret/ engine allows either a string or an object, while the password/ engine will always store and return a string.

The leaseTimeout and ttl parameters are only relevant if the cypher engine will be creating a key that does not exist. eg. password/

HTTP Request

GET https://api.gomorpheus.com/api/cypher/v1/:key

URL Parameters

Query Parameters

23.12.4 Read a Cypher with Lease

The above command returns JSON structured like reading a key with normal authentication:
HTTP Request

GET https://api.gomorpheus.com/api/cypher/v1/:key

HTTP Headers

See Cypher Authentication for details on specifying a lease token.

URL Parameters

Query Parameters

23.12.5 Write a Cypher

```
curl -XPOST "https://api.gomorpheus.com/api/cypher/v1/secret/mymsg" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"msg":"hello world"}'
```

The above command returns JSON structured like readding a cypher key:

HTTP Request

POST https://api.gomorpheus.com/api/cypher/v1/:key

Query Parameters

JSON Parameters

The following parameters are available under the root context of the JSON body.

The secret engine is capable of storing the entire JSON object as key=value pairs, or just a single string. To pass a string, use the value query parameter instead of JSON.

The ttl payload key is a special key that if present will be parsed and used as the ttl parameter (lease duration in seconds).

Key

The key includes a mount prefix separated by a /. For example, the key secret/foo uses the secret mount.
Available Mountpoints

Keys can have different behaviors depending on the specified mountpoint.

Lease Time

Quick MS Time Reference:

23.12.6 Delete a Cypher

```
curl -XDELETE "https://api.gomorpheus.com/api/cypher/v1/secret/foo" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON Structured like this:

```
{
  "success": true
}
```

Will delete a cypher from the system and make it no longer usable.

HTTP Request

DELETE https://api.gomorpheus.com/api/cypher/v1/:key

URL Parameters

23.13 Data Stores

Data Stores can be managed for each Compute Zone (Cloud) in your infrastructure.

23.13.1 Get All Data Stores for Cloud

```
curl "https://api.gomorpheus.com/api/zones/5/data-stores" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
  "datastores": [
    {
      "id": 50,
      "name": "datastore1",
      "zone": {
        "id": 34,
        "name": "test-vmware"
      },
      "type": "generic",
      "freeSpace": 421317836800,
    },
    ...
  ]
}
```

(continues on next page)
This endpoint retrieves all data stores under a cloud.

**HTTP Request**

```
GET https://api.gomorpheus.com/api/zones/:zoneId/data-stores
```

**URL Parameters**

**Query Parameters**

**23.13.2 Get a Specific Data Store**

```
curl "https://api.gomorpheus.com/api/zones/5/data-stores/50" \\
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
  "datastore": {
    "id": 50,
    "name": "datastore1",
    "zone": {
      "id": 34,
      "name": "test-vmware"
    },
    "type": "generic",
    "freeSpace": 421317836800,
    "online": true,
    "active": true,
    "visibility": "private",
    "tenants": [
      { "id": 1,
        "name": "root"
    ]
  }
}
```

(continues on next page)
This endpoint retrieves a specific data store.

**HTTP Request**

GET https://api.gomorpheus.com/api/zones/:zoneId/data-stores/:id

**URL Parameters**

**23.13.3 Updating a Data Store**

```
curl -XPUT "https://api.gomorpheus.com/api/zones/5/data-stores/50" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"datastore":{
  "active": true,
  "visibility": "private",
  "tenantPermissions": {
    "accounts": [1,2,3,4,5]
  },
  "resourcePermissions": {
    "all": false,
    "sites": [
      {"id": 1}, {"id": 2}, {"id": 3}
    ]
  }
}}'
```

The above command returns JSON structured like getting a single data store:

This endpoint allows updating settings for a data store.

**HTTP Request**

PUT https://api.gomorpheus.com/api/zones/:zoneId/data-stores/:id
23.14 Deployment Management

Provides API’s used for creating “Deployment archives” for use with deployable instance types (i.e. Tomcat, Nginx, Apache, etc.). These endpoints also provide a means to trigger a deploy as well as to rollback from a failed deploy. There is a flow to creating a deployment archive. First you must create an appDeploy record. Then you can freely upload files to that deployment archive. Once your upload is complete it is easy to simply trigger the deploy.

23.14.1 Get all Deployments

The above command returns JSON structured like this:

```
{
   "appDeploys": [
   {
      "config": null,
      "dateCreated": "2015-11-14T23:49:24Z",
      "deployDate": "2015-11-14T23:49:47Z",
      "deployGroup": null,
      "deployType": "browser",
      "fetchUrl": null,
      "gitRef": null,
      "gitUrl": null,
      "id": 2,
      "instanceId": 5,
      "keyPair": {
         "id": 1,
         "name": null
      },
      "lastUpdated": "2015-11-14T23:49:47Z",
      "status": "committed",
      "userVersion": null
   },
   ],
   "success": true
}
```

This endpoint retrieves all deploys that were created for a given instance

HTTP Request

GET https://api.gomorpheus.com/api/instances/1/deploy
23.14.2 Create a new Deployment

```bash
curl -XPOST "https://api.gomorpheus.com/api/instances/1/deploy"
-H "Authorization: BEARER access_token"
-H "Content-Type: application/json"
-d '{"appDeploy":{
    "deployType": "browser",
    "gitUrl": null,
    "fetchUrl": null,
    "gitRef": null,
    "userVersion": "1.0.0"
}}'
```

The above command returns JSON structured like this:

```
{
    "appDeploy": {
        "deployType": "browser",
        "gitUrl": null,
        "fetchUrl": null,
        "gitRef": null,
        "userVersion": "1.0.0",
        "status": "open",
        "dateCreated": null,
        "lastUpdated": null,
        "config": null,
        "deployGroup": null,
        "userVersion": "1.0.0"
    }
}
```

This endpoint will create a new AppDeploy entry configured for the specific instance in the url. Depending on the deployment type you may want to upload files to the archive.

HTTP Request

POST https://api.gomorpheus.com/api/instances/:id/deploy

JSON Check Parameters

23.14.3 Upload Files to Deployment Archive

23.15 Errors

The Morpheus API uses the following error codes:

23.16 Execute Schedules

Execute Schedules are definitions for recurring schedules. These schedules can be used in your backup jobs.
23.16.1 Get All Execute Schedules

```bash
curl "https://api.gomorpheus.com/api/execute-schedules" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
  "schedules": [
    {
      "id": 1,
      "name": "daily",
      "description": "Daily at Midnight",
      "enabled": true,
      "scheduleType": "execute",
      "scheduleTimezone": "America/New_York",
      "cron": "0 0 * * *",
      "dateCreated": "2018-03-01T07:56:38+0000",
      "lastUpdated": "2018-09-13T21:38:19+0000"
    },
    {
      "id": 2,
      "name": "weekly",
      "description": "Weekly on Sunday at Midnight",
      "enabled": true,
      "scheduleType": "execute",
      "scheduleTimezone": "America/New_York",
      "cron": "0 0 * * 7",
      "dateCreated": "2018-03-01T07:56:38+0000",
      "lastUpdated": "2018-09-13T21:38:19+0000"
    }
  ],
  "meta": {
    "size": 1,
    "total": 1,
    "max": 25,
    "offset": 0
  }
}
```

This endpoint retrieves all execute schedules associated with the account.

**HTTP Request**

GET https://api.gomorpheus.com/api/execute-schedules

**Query Parameters**

23.16.2 Get a Specific Execute Schedule

```bash
curl "https://api.gomorpheus.com/api/execute-schedules/1" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:
This endpoint retrieves a specific execute schedule.

**HTTP Request**

GET https://api.gomorpheus.com/api/execute-schedules/:id

**URL Parameters**

**23.16.3 Create an Execute Schedule**

```bash
curl -XPOST "https://api.gomorpheus.com/api/execute-schedules" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{
    "schedule": {
        "name": "Friday at Midnight",
        "description": null,
        "enabled": true,
        "scheduleType": "execute",
        "scheduleTimezone": "UTC",
        "cron": "0 0 * * 5"
    }
}'
```

The above command returns JSON structured like getting a single execute schedule:

**HTTP Request**

POST https://api.gomorpheus.com/api/execute-schedules

**JSON Parameters**

**23.16.4 Update an Execute Schedule**

```bash
curl -XPUT "https://api.gomorpheus.com/api/execute-schedules/2" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
```

(continues on next page)
{-
"schedule": {
   "description": "Daily at 2AM",
   "cron": "0 2 * * *
}
}

The above command returns JSON structured like getting a single execute schedule:

**HTTP Request**

PUT https://api.gomorpheus.com/api/execute-schedules/:id

**JSON Parameters**

See Create.

### 23.16.5 Delete an Execute Schedule

```sh
curl -XDELETE "https://api.gomorpheus.com/api/execute-schedules/1" \ 
   -H "Authorization: BEARER access_token"
```

The above command returns JSON Structured like this:

```json
{
   "success": true
}
```

Will delete an execute schedule from the system and make it no longer usable.

**HTTP Request**

DELETE https://api.gomorpheus.com/api/execute-schedules/:id

**URL Parameters**

### 23.17 Execution Request

Provides API interfaces for executing an arbitrary script or command on an instance, container or host.

#### 23.17.1 Create an Execution Request

```sh
curl -XPOST "https://api.gomorpheus.com/api/execution-request/execute?instanceId=256" \ 
   -H "Authorization: BEARER access_token" \ 
   -H "Content-Type: application/json" \ 
   -d '{
```

(continues on next page)
The above command returns JSON structured like this:

```
{
  "executionRequest": {
    "id": 24,
    "uniqueId": "f22e1292-4407-44c0-b2c7-698ee2241491",
    "containerId": null,
    "serverId": null,
    "instanceId": 256,
    "stdOut": "Linux apachetest 3.19.0-69-generic #77~14.04.1-Ubuntu SMP Tue Aug 30 01:29:21 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux",
    "stdErr": null,
    "exitCode": 0,
    "status": "complete",
    "expiresAt": "2018-11-30T18:23:02+0000",
    "createdById": 1
  }
}
```

HTTP Request

POST https://api.gomorpheus.com/api/execution-request/execute

Query Parameters

JSON Parameters

This endpoint executes the provided script on the specified instance, container or server.

23.17.2 Get a Specific Execution Request

```
curl "https://api.gomorpheus.com/api/execution-request/f22e1292-4407-44c0-b2c7-
-698ee2241491" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
  "executionRequest": {
    "id": 24,
    "uniqueId": "f22e1292-4407-44c0-b2c7-698ee2241491",
    "containerId": null,
    "serverId": null,
    "instanceId": 256,
    "stdOut": "Linux apachetest 3.19.0-69-generic #77~14.04.1-Ubuntu SMP Tue Aug 30 01:29:21 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux",
    "stdErr": null,
    "exitCode": 0,
    "status": "complete",
    "expiresAt": "2018-11-30T18:23:02+0000",
```

(continues on next page)
HTTP Request

GET https://api.gomorpheus.com/api/execution-request/:uniqueId

URL Parameters

This endpoint retrieves a specific execution request.

23.18 Groups

Groups are used to organize provisioned servers in your infrastructure. When a user on the system provisions an instance like MySQL, they can select which group to provision the instance into. This can be used to scope servers by environment or by region.

23.18.1 Get All Groups

curl "https://api.gomorpheus.com/api/groups"
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```json
{
  "groups": [
    {
      "id": 1,
      "accountId": 1,
      "name": "Amazon East",
      "code": "amazon",
      "active": true,
      "location": null,
      "visibility": "public",
      "zones": [
        {
          "id": 1,
          "accountId": 1,
          "groupId": 1,
          "name": "VPC 1a",
          "description": "1a VPC Subnet",
          "location": null,
          "visibility": "public",
          "zoneTypeId": 1
        }
      ]
    }
  ]
}
```
This endpoint retrieves all groups and a list of zones associated with the group by id.

**HTTP Request**

GET https://api.gomorpheus.com/api/groups

**Query Parameters**

23.18.2 Get a Specific Group

```bash
curl "https://api.gomorpheus.com/api/groups/1" \ 
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
  "success": true,
  "group": {
    "id": 1,
    "accountId": 1,
    "name": "Vagrant",
    "code": null,
    "active": true,
    "location": null,
    "visibility": "public",
    "zones": [
      {
        "id": 1,
        "accountId": 1,
        "groupId": 1,
        "name": "Davids Laptop",
        "description": "My Laptop Vagrant",
        "location": null,
        "visibility": "public",
        "zoneTypeId": 1
      }
    ]
  }
}
```

This endpoint retrieves a specific group.

**HTTP Request**

GET https://api.gomorpheus.com/api/groups/:id

**URL Parameters**

23.18.3 Create a Group
curl -XPOST "https://api.gomorpheus.com/api/groups" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"group":{
   "name": "My Group",
   "description": "My description",
   "location": "US EAST"
}}'

The above command returns JSON structured like getting a single group:

HTTP Request
POST https://api.gomorpheus.com/api/groups

JSON Check Parameters

23.18.4 Updating a Group

curl -XPUT "https://api.gomorpheus.com/api/groups/1" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"group":{
   "name": "My Group",
   "location": "US EAST"
}}'

The above command returns JSON structured like getting a single group:

HTTP Request
PUT https://api.gomorpheus.com/api/groups/:id

JSON Check Parameters

23.18.5 Updating Group Zones

curl -XPUT "https://api.gomorpheus.com/api/groups/1/update-zones" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"group":{
   "zones": [
      {"id": 1}, {"id": 2}, {"id": 5}
   ]
}}'

The above command returns JSON Structured like this:

{  
   "success": true
}
This will update the zones that are assigned to the group. Any zones that are not passed in the `zones` parameter will be removed from the group.

**HTTP Request**

PUT https://api.gomorpheus.com/api/groups/:id/update-zones

**JSON Check Parameters**

**23.18.6 Delete a Group**

```
curl -XDELETE "https://api.gomorpheus.com/api/groups/1" \
   -H "Authorization: BEARER access_token"
```

The above command returns JSON Structured like this:

```
{
   "success": true
}
```

If a group has zones or servers still tied to it, a delete action will fail

**HTTP Request**

DELETE https://api.gomorpheus.com/api/groups/:id

**23.19 Subtenant Groups**

Groups belonging to a subtenant can be managed by the master account.

**23.19.1 Get All Groups for Subtenant**

```
curl "https://api.gomorpheus.com/api/accounts/20/groups" \
   -H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
   "groups": [
      {
         "id": 365,
         "name": "testgroup",
         "code": "testgroup",
         "location": "West",
         "accountId": 20,
         "visibility": "public",
         "active": true,
         "dateCreated": "2018-03-20T20:34:22+0000",
         "lastUpdated": "2018-03-31T18:32:56+0000",
         "zones": [
```
This endpoint retrieves all groups and a list of zones associated with the group by id.

**HTTP Request**

GET https://api.gomorpheus.com/api/accounts/:accountId/groups

**URL Parameters**

**Query Parameters**

23.19.2 Get a Specific Group for Subtenant

```
curl "https://api.gomorpheus.com/api/accounts/20/groups/365" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
  "group": {
    "id": 365,
    "name": "testgroup",
    "code": "testgroup",
    "location": "West",
    "accountId": 20,
    "visibility": "public",
    "active": true,
    "dateCreated": "2018-03-20T20:34:22+0000",
    "lastUpdated": "2018-03-31T18:32:56+0000",
    "zones": [
      {
        "id": 32,
        "name": "test-google"
      },
      {
        "id": 33,
        "name": "test-vmware"
      }
    ]
  },
  "meta": {
    "offset": 0,
    "max": 25,
    "size": 1,
    "total": 1
  }
}
```
This endpoint retrieves a specific group.

**HTTP Request**

GET https://api.gomorpheus.com/api/accounts/:accountId/groups/:id

**URL Parameters**

**23.19.3 Create a Group for Subtenant**

```
curl -XPOST "https://api.gomorpheus.com/api/accounts/20/groups" \
  -H "Authorization: BEARER access_token" \
  -H "Content-Type: application/json" \
  -d '{"group":{ 
    "name": "My Group", 
    "description": "My description", 
    "location": "West" 
  }}'
```

The above command returns JSON structured like getting a single group:

**HTTP Request**

POST https://api.gomorpheus.com/api/accounts/:accountId/groups

**URL Parameters**

**JSON Parameters**

**23.19.4 Updating a Group for Subtenant**

```
curl -XPUT "https://api.gomorpheus.com/api/accounts/20/groups/365" \
  -H "Authorization: BEARER access_token" \
  -H "Content-Type: application/json" \
  -d '{"group":{ 
    "name": "My Group", 
    "description": "My description", 
    "location": "West" 
  }}'
```

The above command returns JSON structured like getting a single group:
HTTP Request

PUT https://api.gomorpheus.com/api/accounts/:accountId/groups/:id

URL Parameters

JSON Parameters

23.19.5 Updating Group Zones for Subtenant

curl -XPUT "https://api.gomorpheus.com/api/accounts/20/groups/365/update-zones" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"group":{ 
 "zones": [ 
 {"id": 32}, {"id": 33}, {"id": 34} 
 ] 
}}'

The above command returns JSON Structured like this:

```json
{
   "success": true
}
```

This will update the zones that are assigned to the group. Any zones that are not passed in the zones parameter will be removed from the group.

HTTP Request

PUT https://api.gomorpheus.com/api/accounts/:id/groups/:groupId/update-zones

JSON Parameters

23.19.6 Delete a Group for Subtenant

curl -XDELETE "https://api.gomorpheus.com/api/accounts/20/groups/365" \
-H "Authorization: BEARER access_token"

The above command returns JSON Structured like this:

```json
{
   "success": true
}
```

HTTP Request

DELETE https://api.gomorpheus.com/api/accounts/:id/groups/:groupId

If a group has zones or servers still tied to it, a delete action will fail
23.20 Image Builds

Image Builds are used to generate Virtual Images for your Morpheus Library.

23.20.1 Get All Image Builds

```
curl "https://api.gomorpheus.com/api/image-builds" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
   "imageBuilds": [
   {
      "id": 1,
      "account": {
         "id": 1,
         "name": "root"
      },
      "type": {
         "id": 1,
         "code": "vmware",
         "name": "VMware"
      },
      "site": {
         "id": 1,
         "name": "my-group"
      },
      "zone": {
         "id": 1,
         "name": "my-vmware"
      },
      "name": "testbuild",
      "description": "a test build",
      "bootScript": {
         "id": 2,
         "fileName": "debian standard"
      },
      "bootCommand": null,
      "preseedScript": {
         "id": 2,
         "fileName": "debian 8"
      },
      "scripts": [
      {
         "id": 114,
         "name": "blah.txt",
         "type": "bash",
         "phase": "postProvision"
      }
      ],
      "sshUsername": "builderbot",
      "sshPassword": "************",
      "storageProvider": null,
      "buildOutputName": "mytestbuild",
      ...
   }
   ...
]
```
"conversionFormats": null,
"isCloudInit": false,
"vmToolsInstalled": true,
"keepResults": 2,
"config": {},
"lastResult": {
  "id": 70,
  "imageBuild": {
    "id": 21,
    "name": "testbuild"
  },
  "buildNumber": 6,
  "startDate": "2017-09-28T05:48:03Z",
  "endDate": null,
  "statusMessage": "Initializing",
  "statusPercent": 0.0,
  "statusEta": null,
  "status": "running",
  "errorMessage": null,
  "createdBy": {
    "username": "admin"
  },
  "tempInstance": null,
  "virtualImages": [ ]
},
"executionCount": 2,
"meta": {
  "size": 1,
  "total": 1,
  "offset": 0,
  "max": 25
}

This endpoint retrieves all image builds associated with the account.

**HTTP Request**

GET https://api.gomorpheus.com/api/image-builds

**Query Parameters**

**23.20.2 Get a Specific Image Build**

```
curl "https://api.gomorpheus.com/api/image-builds/4" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this *(config omitted)* :


```json
{
    "imageBuild": {
        "id": 1,
        "account": {
            "id": 1,
            "name": "root"
        },
        "type": {
            "id": 1,
            "code": "vmware",
            "name": "VMware"
        },
        "site": {
            "id": 1,
            "name": "my-group"
        },
        "zone": {
            "id": 1,
            "name": "my-vmware"
        },
        "name": "testbuild",
        "description": "a test build",
        "bootScript": {
            "id": 2,
            "fileName": "debian standard"
        },
        "bootCommand": null,
        "preseedScript": {
            "id": 2,
            "fileName": "debian 8"
        },
        "scripts": [
            {
                "id": 114,
                "name": "cleanup.sh",
                "type": "bash",
                "phase": "postProvision"
            }
        ],
        "sshUsername": "builderbot",
        "sshPassword": "************",
        "storageProvider": null,
        "buildOutputName": null,
        "conversionFormats": null,
        "isCloudInit": false,
        "vmToolsInstalled": true,
        "keepResults": 2,
        "config": {
            "instance": {
                "layout": {
                    "code": "vmware-1.0-single",
                    "id": 142
                },
                "type": "vmware",
                "userGroup": {
                    "id": ""
                }
            }
        }
    }
}
```

(continues on next page)
"networkInterfaces": [
  {
    "primaryInterface": true,
    "network": {
      "id": "network-147"
    }
  }
],
"volumes": [
  {
    "vId": 1752,
    "size": 10,
    "maxIOPS": null,
    "name": "root",
    "rootVolume": true,
    "storageType": 1,
    "datastoreId": "auto"
  }
],
"storageControllers": [],
"zoneId": 1,
"config": {
  "template": 1752,
  "vmwareResourcePoolId": "resgroup-123",
  "expose": 8080
},
"plan": {
  "code": "vm-512",
  "id": 75
},
"lastResult": {
  "id": 70,
  "imageBuild": {
    "id": 21,
    "name": "testbuild"
  },
  "buildNumber": 6,
  "startDate": "2017-09-28T05:48:03Z",
  "endDate": null,
  "statusMessage": "Booting",
  "statusPercent": 20.0,
  "statusEta": null,
  "status": "running",
  "errorMessage": null,
  "createdBy": {
    "username": "admin"
  },
  "tempInstance": null,
  "virtualImages": []
},
"executionCount": 4
}
This endpoint retrieves a specific image build.
HTTP Request

GET https://api.gomorpheus.com/api/image-builds/:id

URL Parameters

23.20.3 Create an Image Build

curl -X POST "https://api.gomorpheus.com/api/image-builds" \\n-H "Authorization: BEARER access_token" \\n-H "Content-Type: application/json" \\n-d '{
  "imageBuild": {
    "type": "vmware",
    "name": "builder test",
    "description": null,
    "site": {
      "id": 1
    },
    "zone": {
      "id": 1
    },
    "config": {
      "zoneId": 1,
      "instance": {
        "name": "builder test",
        "site": {
          "id": 1
        },
        "type": "vmware",
        "instanceType": {
          "code": "vmware"
        },
        "layout": {
          "id": 142
        },
        "plan": {
          "id": 76
        },
        "networkDomain": {
        }
      },
      "config": {
        "resourcePoolId": "resgroup-123",
        "template": 1232,
        "nestedVirtualization": "off",
        "expose": "8080"
      },
      "volumes": [
        {
          "id": -1,
          "rootVolume": true,
          "name": "root",
          "size": 10,
          "storageType": 1,
          "datastoreId": "autoCluster"
        }
      ]
    }
  }
}'}
The above command returns JSON structured like getting a single image build.

**HTTP Request**

**POST** https://api.gomorpheus.com/api/image-builds

**JSON Parameters**

23.20.4 Update an Image Build

```bash
curl -XPUT "https://api.gomorpheus.com/api/image-builds/1" \ 
-H "Authorization: Bearer access_token" \ 
-H "Content-Type: application/json" \ 
-d '{
   "imageBuild": {
      "description": "a good build",
      "keepResults": 5,
   }
}'
```

The above command returns JSON structured like getting a single image build.
HTTP Request

PUT https://api.gomorpheus.com/api/image-builds/1

URL Parameters

JSON Parameters

See Create.

## 23.20.5 Delete an Image Build

```bash
curl -XDELETE "https://api.gomorpheus.com/api/image-builds/1" \ 
-H "Authorization: BEARER access_token"
```

The above command returns JSON Structured like this:

```json
{
   "success": true
}
```

Will delete an image build from the system and make it no longer usable.

HTTP Request

DELETE https://api.gomorpheus.com/api/image-builds/:id

URL Parameters

## 23.20.6 Run an Image Build

```bash
curl -XPOST "https://api.gomorpheus.com/api/image-builds/1/run" \ 
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
   "success": true
}
```

Running an image build is done asynchronously. This api will kick off the new execution and update the image build status.

HTTP Request

POST https://api.gomorpheus.com/api/image-builds/1/run
URL Parameters

23.20.7 List Image Build Executions

curl "https://api.gomorpheus.com/api/image-builds/1/list-executions" \
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```
{
   "imageBuildExecutions": [
   {
      "id": 82,
      "imageBuild": {
         "id": 20,
         "name": "my-testbuild"
      },
      "buildNumber": 2,
      "startDate": "2017-09-29T15:30:07Z",
      "endDate": null,
      "statusMessage": "Installing",
      "statusPercent": 25.0,
      "statusEta": null,
      "status": "running",
      "errorMessage": null,
      "createdBy": {
         "username": "admin"
      },
      "tempInstance": null,
      "virtualImages": [
      ]
   },
   {
      "id": 81,
      "imageBuild": {
         "id": 20,
         "name": "my-testbuild"
      },
      "buildNumber": 1,
      "startDate": "2017-09-29T14:57:33Z",
      "endDate": "2017-09-29T15:26:41Z",
      "statusMessage": null,
      "statusPercent": 100.0,
      "statusEta": null,
      "status": "success",
      "errorMessage": null,
      "createdBy": {
         "username": "admin"
      },
      "tempInstance": null,
      "virtualImages": [
         {
            "id": 1850,
            "name": "my-testbuild-4-176-1506697891084"
         }
      ]
   }
   ]
```

(continues on next page)
HTTP Request

GET https://api.gomorpheus.com/api/image-builds/1/list-executions

URL Parameters
Query Parameters

23.21 Instances

Instances are sets of containers or vms (morpheus API represents a vm as a container attached to a server) of various types that can be provisioned across the Morpheus stack and offer a wide range of services. MySQL, Redis, Elastic-Search, PostgreSQL, Tomcat, nginx, Confluence, Jenkins, and more. There are a few important concept differentiators between what morpheus calls an instance and what amazon calls an instance. In morpheus an instance can represent many vms or containers that are of a set. For example. If you wanted to spin up a Mongo sharded replicaset, that requires 7 virtual machines or 7 docker containers. Morpheus represents this as a singular instance with a specified layout and then represents all the associated services running within that instance as containers. If, a container record is a docker container then the serverId it belongs to is representative of the Docker Host it was provisioned onto. If the container is a virtual machine then the serverId represents the compute resource it was provisioned onto, (i.e. the virtual machine).

23.21.1 Get All Instances

```
curl "https://api.gomorpheus.com/api/instances?max=3"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
  "instances": [
    {
      "id": 1530,
      "accountId": 1,
      "instanceType": {
        "id": 35,
        "code": "ubuntu",
    
```
```
"category": "os",
"name": "Ubuntu"
},
"group": {
  "id": 3,
  "name": "Demo"
},
"cloud": {
  "id": 6,
  "name": "San Mateo VMware"
},
"containers": [
  1798
],
"servers": [
  2
],
"connectionInfo": [
  {
    "ip": "192.168.162.59",
    "port": 22
  }
],
"layout": {
  "id": 105
},
"plan": {
  "id": 12,
  "code": "vm-2048"
},
"name": "ah-San Mateo VMware-ubuntu",
"description": null,
"instanceVersion": null,
"dateCreated": "2017-01-31T21:30:49+0000",
"lastUpdated": "2017-02-07T22:58:26+0000",
"hostName": "ah-San-Mateo-VMware-ubuntu",
"domainName": null,
"environmentPrefix": null,
"firewallEnabled": true,
"networkLevel": "container",
"autoScale": false,
"instanceContext": "production",
"currentDeployId": null,
"status": "running",
"statusMessage": null,
"errorMessage": null,
"statusDate": "2017-01-31T21:34:07+0000",
"statusPercent": null,
"statusEta": null,
"userStatus": null,
"createdBy": {
  "id": 38
}
},
{
  "id": 1653,
  "accountId": 1
}
```
"instanceType": { 
  "id": 35,
  "code": "ubuntu",
  "category": "os",
  "name": "Ubuntu"
},
"group": { 
  "id": 3,
  "name": "Demo"
},
"cloud": { 
  "id": 6,
  "name": "San Mateo VMware"
},
"containers": [ 
  1945
],
"servers": [ 
  2
],
"connectionInfo": [ 
  { 
    "ip": "192.168.163.55",
    "port": 22
  }
],
"layout": { 
  "id": 105
},
"plan": { 
  "id": 11,
  "code": "vm-1024"
},
"name": "ah-San Mateo VMware-ubuntu-PDNStest",
"description": null,
"instanceVersion": null,
"dateCreated": "2017-02-10T14:27:42+0000",
"lastUpdated": "2017-02-10T14:31:19+0000",
"hostName": "ah-san-mateo-vmware-ubuntu-pdnstest",
"domainName": null,
"environmentPrefix": null,
"firewallEnabled": true,
"networkLevel": "container",
"autoScale": false,
"instanceContext": "dev",
"currentDeployId": null,
"status": "running",
"statusMessage": null,
"errorMessage": null,
"statusDate": "2017-02-10T14:30:43+0000",
"statusPercent": null,
"statusEta": null,
"userStatus": null,
"createdBy": { 
  "id": 38
}
),
(continues on next page)
{  
  "id": 1624,  
  "accountId": 1,  
  "instanceType": {  
    "id": 21,  
    "code": "apache",  
    "category": "web",  
    "name": "Apache"  
  },  
  "group": {  
    "id": 163,  
    "name": "snow-approvals"  
  },  
  "cloud": {  
    "id": 6,  
    "name": "San Mateo VMware"  
  },  
  "containers": [  
    1912  
  ],  
  "servers": [  
    3  
  ],  
  "connectionInfo": [  
    {  
      "ip": "192.168.163.28",  
      "port": 10009  
    }  
  ],  
  "layout": {  
    "id": 48  
  },  
  "plan": {  
    "id": 3,  
    "code": "container-256"  
  },  
  "name": "approval-snow-test",  
  "description": null,  
  "instanceVersion": null,  
  "dateCreated": "2017-02-09T06:45:30+0000",  
  "lastUpdated": "2017-02-09T06:53:20+0000",  
  "hostName": "approval-snow-test",  
  "domainName": null,  
  "environmentPrefix": null,  
  "firewallEnabled": true,  
  "networkLevel": "container",  
  "autoScale": false,  
  "instanceContext": null,  
  "currentDeployId": null,  
  "status": "running",  
  "statusMessage": null,  
  "errorMessage": null,  
  "statusDate": "2017-02-09T06:53:20+0000",  
  "statusPercent": null,  
  "statusEta": null,  
  "userStatus": null,  
  "createdBy": {  
    "id": 934  
  }  
}
This endpoint retrieves all instances and their JSON encoded configuration attributes based on check type. Server data is encrypted in the database.

HTTP Request

GET https://api.gomorpheus.com/api/instances

Query Parameters

23.21.2 Get a Specific Instance

curl "https://api.gomorpheus.com/api/instances/1216" \
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this:
{  
  "success": true,
  "instance": {  
    "id": 1698,
    "accountId": 1,
    "instanceType": {  
      "id": 44,
      "code": "redis",
      "category": "cache",
      "name": "Redis"
    },
    "group": {  
      "id": 3,
      "name": "Demo"
    },
    "cloud": {  
      "id": 6,
      "name": "San Mateo VMware"
    },
    "containers": [19],
    "servers": [2],
    "connectionInfo": [{  
      "ip": "10.211.55.11",
      "port": 10000
    }],
    "layout": {  
      "id": 221
    },
    "plan": {  
      "id": 69,
      "code": "container-512"
    },
    "name": "redistest",
    "description": null,
    "instanceVersion": null,
    "tags": [],
    "maxMemory": 536870912,
    "maxStorage": 5368709120,
    "maxCores": 0,
    "maxCpu": null,
    "dateCreated": "2016-10-25T15:12:06+0000",
    "lastUpdated": "2017-02-13T19:22:00+0000",
    "hostName": "redistest",
    "domainName": null,
    "environmentPrefix": null,
    "firewallEnabled": true,
    "networkLevel": "container",
    "autoScale": false,
    "instanceContext": null,
  }
}
This endpoint retrieves a specific instance.

**HTTP Request**

GET https://api.gomorpheus.com/api/instances/:id

### 23.21.3 Get Env Variables

```bash
curl "https://api.gomorpheus.com/api/instances/1216/envs" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
  "envs": [
    {
      "export": false,
      "masked": false,
      "name": "DATABASE_NAME",
      "value": "spud_marketing"
    }
  ],
  "readOnlyEnvs": {
    "TOMCAT_HOST": {
      "export": true,
      "masked": false,
      "name": "TOMCAT_HOST",
      "value": "container1414"
    },
    "TOMCAT_HOST_2": {
      "export": true,
      "masked": false,
      "name": "TOMCAT_HOST_2",
```

(continues on next page)
"value": "container1759"
},
"TOMCAT_IP": {
  "export": true,
  "masked": false,
  "name": "TOMCAT_IP",
  "value": "192.168.163.232"
},
"TOMCAT_IP_2": {
  "export": true,
  "masked": false,
  "name": "TOMCAT_IP_2",
  "value": "192.168.163.233"
},
"TOMCAT_PORT": {
  "export": true,
  "masked": false,
  "name": "TOMCAT_PORT",
  "value": 10017
},
"TOMCAT_PORT_2": {
  "export": true,
  "masked": false,
  "name": "TOMCAT_PORT_2",
  "value": 10017
},
"TOMCAT_PORT_8080_TCP": {
  "export": true,
  "masked": false,
  "name": "TOMCAT_PORT_8080_TCP",
  "value": "tcp://192.168.163.232:10017"
},
"TOMCAT_PORT_8080_TCP_2": {
  "export": true,
  "masked": false,
  "name": "TOMCAT_PORT_8080_TCP_2",
  "value": "tcp://192.168.163.233:10017"
},
"TOMCAT_PORT_8080_TCP_ADDR": {
  "export": true,
  "masked": false,
  "name": "TOMCAT_PORT_8080_TCP_ADDR",
  "value": "192.168.163.232"
},
"TOMCAT_PORT_8080_TCP_ADDR_2": {
  "export": true,
  "masked": false,
  "name": "TOMCAT_PORT_8080_TCP_ADDR_2",
  "value": "192.168.163.233"
},
"TOMCAT_PORT_8080_TCP_PORT": {
  "export": true,
  "masked": false,
  "name": "TOMCAT_PORT_8080_TCP_PORT",
  "value": 10017
},
"TOMCAT_PORT_8080_TCP_PORT_2": {
"export": true,
"masked": false,
"name": "TOMCAT_PORT_8080_TCP_PORT_2",
"value": 10017
},
"TOMCAT_PORT_8080_TCP_PROTO": {
  "export": true,
  "masked": false,
  "name": "TOMCAT_PORT_8080_TCP_PROTO",
  "value": "tcp"
},
"TOMCAT_PORT_8080_TCP_PROTO_2": {
  "export": true,
  "masked": false,
  "name": "TOMCAT_PORT_8080_TCP_PROTO_2",
  "value": "tcp"
}
},
"importedEnvs": {
  "MYSQL_HOST": {
    "export": true,
    "masked": false,
    "name": "MYSQL_HOST",
    "value": "container1413"
  },
  "MYSQL_HOST_2": {
    "export": true,
    "masked": false,
    "name": "MYSQL_HOST_2",
    "value": "container1756"
  },
  "MYSQL_IP": {
    "export": true,
    "masked": false,
    "name": "MYSQL_IP",
    "value": "192.168.163.232"
  },
  "MYSQL_IP_2": {
    "export": true,
    "masked": false,
    "name": "MYSQL_IP_2",
    "value": "192.168.163.233"
  },
  "MYSQL_MASTER": {
    "export": true,
    "masked": false,
    "name": "MYSQL_HOST",
    "value": "container1413"
  },
  "MYSQL_PASSWORD": {
    "export": true,
    "masked": true,
    "name": "MYSQL_PASSWORD",
    "value": "morpheus"
  },
  "MYSQL_PASSWORD_2": {
    "export": true,
"masked": true,
"name": "MYSQL_PASSWORD",
"value": "morpheus"
},
"MYSQL_PORT": {
"export": true,
"masked": false,
"name": "MYSQL_PORT",
"value": 10016
},
"MYSQL_PORT_2": {
"export": true,
"masked": false,
"name": "MYSQL_PORT_2",
"value": 10016
},
"MYSQL_PORT_3306_TCP": {
"export": true,
"masked": false,
"name": "MYSQL_PORT_3306_TCP",
"value": "tcp://192.168.163.232:10016"
},
"MYSQL_PORT_3306_TCP_2": {
"export": true,
"masked": false,
"name": "MYSQL_PORT_3306_TCP_2",
"value": "tcp://192.168.163.233:10016"
},
"MYSQL_PORT_3306_TCP_ADDR": {
"export": true,
"masked": false,
"name": "MYSQL_PORT_3306_TCP_ADDR",
"value": "192.168.163.232"
},
"MYSQL_PORT_3306_TCP_ADDR_2": {
"export": true,
"masked": false,
"name": "MYSQL_PORT_3306_TCP_ADDR_2",
"value": "192.168.163.233"
},
"MYSQL_PORT_3306_TCP_PORT": {
"export": true,
"masked": false,
"name": "MYSQL_PORT_3306_TCP_PORT",
"value": 10016
},
"MYSQL_PORT_3306_TCP_PORT_2": {
"export": true,
"masked": false,
"name": "MYSQL_PORT_3306_TCP_PORT_2",
"value": 10016
},
"MYSQL_PORT_3306_TCP_PROTO": {
"export": true,
"masked": false,
"name": "MYSQL_PORT_3306_TCP_PROTO",
"value": "tcp"
This gets all the environment variables associated with the instance.

**HTTP Request**

GET https://api.gomorpheus.com/api/instances/:id/envs

### 23.21.4 Get Instance History

```
curl "https://api.gomorpheus.com/api/238/history" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
    "processes": [
    {
        "id": 250,
        "accountId": 1,
        "uniqueId": "cebc47ec-cb2f-417a-886e-dd60cf81db26",
        "processType": {
            "code": "provision",
            "name": "provision"
        },
        "description": null,
        "subType": null,
        "subId": null,
        "zoneId": 34,
        "integrationId": null,
        "instanceId": 238,
        "containerId": 240,
        "serverId": 601,
        "containerName": "apachetest",
        "displayName": "apachetest",
        "timerCategory": "vmware",
        "timerSubCategory": "28",
        "status": "failed",
        "reason": null,
        "percent": 100.0,
        "statusEta": 348246,
        "message": null,
        "output": null,
        "error": null,
    }
    ]
}
```

(continues on next page)
This endpoint retrieves the process history for a specific instance. Alternatively, the Process History endpoint can be used to get the same information.

**HTTP Request**

GET https://api.gomorpheus.com/api/instances/:id/history

**Query Parameters**

### 23.21.5 Get Container Details

```bash
curl "https://api.gomorpheus.com/api/instances/1216/containers" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
   "containers": [
   {
      "id": 292,
      "accountId": 1,
      "instance": {
         "id": 294,
         "name": "nginxtest"
      },
      "containerType": {
         "id": 187,
         "type": "web"  
      }
   }
   ]
}
```
"code": "nginx-vmware-1.9",
"category": "nginx",
"name": "NGINX 1.9"
},
"containerTypeSet": {
  "id": 193,
  "code": "nginx-vmware-1.9-set",
  "category": "nginx"
},
"server": {
  "id": 653,
  "name": "nginxtest"
},
"cloud": {
  "id": 34,
  "name": "myvmware"
},
"name": "nginxtest_292",
"ip": "10.30.20.50",
"internalIp": "10.30.20.50",
"internalHostname": "container292",
"externalHostname": "nginxtest",
"externalDomain": "localdomain",
"externalFqdn": "nginxtest.localdomain",
"ports": [
  {
    "index": 0,
    "external": 80,
    "internal": 80,
    "primaryPort": true,
    "displayName": "Http",
    "export": true,
    "visible": true,
    "loadBalance": true,
    "link": true,
    "exportName": "HTTP",
    "protocol": "http",
    "code": "nginx.80"
  },
  {
    "index": 1,
    "external": 443,
    "internal": 443,
    "primaryPort": false,
    "displayName": "Https",
    "export": true,
    "visible": true,
    "loadBalance": true,
    "link": true,
    "exportName": "HTTPS",
    "protocol": "https",
    "code": "nginx.443"
  }
],
"plan": {
  "id": 76,
  "code": "vm-1024"}
This can be valuable for evaluating the details of the compute server(s) running on an instance

**HTTP Request**

GET https://api.gomorpheus.com/api/instances/:id/containers
23.21.6 Get Available Service Plans for an Instance

```
curl -XGET "https://api.gomorpheus.com/api/instances/service-plans?zoneId=1&layoutId=75" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structure like this:

```
{
  "plans": [
    {
      "id": 75,
      "name": "1 CPU, 512MB Memory",
      "value": 75,
      "code": "vm-512",
      "maxStorage": 10737418240,
      "maxMemory": 536870912,
      "maxCpu": 1,
      "maxCores": 1,
      "customCpu": false,
      "customMaxMemory": false,
      "customMaxStorage": true,
      "customMaxDataStorage": true,
      "customCoresPerSocket": false,
      "coresPerSocket": 1,
      "storageTypes": [
        {
          "id": 1,
          "editable": false,
          "optionTypes": [
          ],
          "displayOrder": 1,
          "code": "standard",
          "volumeType": "disk",
          "minStorage": null,
          "deletable": false,
          "defaultType": true,
          "createDatastore": null,
          "resizable": false,
          "storageType": null,
          "allowSearch": true,
          "volumeOptionSource": null,
          "displayName": "Disk",
          "minIOPS": null,
          "maxIOPS": null,
          "hasDatastore": true,
          "customSize": true,
          "autoDelete": true,
          "name": "Standard",
          "configurableIOPS": false,
          "customLabel": true,
          "enabled": true,
          "description": "Standard",
          "volumeCategory": "disk",
          "externalId": null,
          "maxStorage": null
        }
      ]
    }
  ]
}
```

(continues on next page)


```
]
},
"rootStorageTypes": [
{
"id": 1,
"editable": false,
"optionTypes": [
],
"displayOrder": 1,
"code": "standard",
"volumeType": "disk",
"minStorage": null,
"deletable": false,
"defaultType": true,
"createDatastore": null,
"resizeable": false,
"storageType": null,
"allowSearch": true,
"volumeOptionSource": null,
"displayName": "Disk",
"minIOPS": null,
"maxIOPS": null,
"hasDatastore": true,
"customSize": true,
"autoDelete": true,
"name": "Standard",
"configurableIOPS": false,
"customLabel": true,
"enabled": true,
"description": "Standard",
"volumeCategory": "disk",
"externalId": null,
"maxStorage": null
}
],
"addVolumes": true,
"customizeVolume": true,
"rootDiskCustomizable": true,
"noDisks": false,
"hasDatastore": true,
"minDisk": 0,
"maxDisk": null,
"lvmSupported": true,
"datastores": {
"cluster": [
{
"id": 54,
"name": "demo-qnap - 4.3TB Free"
}
],
"store": [
{
"id": 50,
"name": "datastore1 - 463.4GB Free"
}
]
```

(continues on next page)
This returns a list of all of the service plans available for an instance type. The response includes details about the plans and their configuration options. The parameters `zoneId` and `layoutId` are required.

This endpoint can be used to get the list of plans available for creating a new instance or resizing an existing instance.

**HTTP Request**

GET https://api.gomorpheus.com/api/instances/service-plans

**Query Parameters**

**23.21.7 Create an Instance**

See *Provisioning* for details.

**HTTP Request**

POST https://api.gomorpheus.com/api/instances
23.21.8 Updating an Instance

```
$ curl -X PUT "https://api.gomorpheus.com/api/instances/1" \\
-H "Authorization: BEARER access_token" \\
-H "Content-Type: application/json" \\
-d '{ "instance": { "description": "my new redis" }}'
```

The above command returns a similar JSON structure when submitting a GET request for a single check

**HTTP Request**

PUT https://api.gomorpheus.com/api/instances/:id

**JSON Instance Parameters**

23.21.9 Updating Instance Notes

```
$ curl -X PUT "https://api.gomorpheus.com/api/instances/1/notes" \\
-H "Authorization: BEARER access_token" \\
-H "Content-Type: application/json" \\
-d '{ "instance": { "notes": "My notes about this instance." }}'
```

The above command returns a similar JSON structure when submitting a GET request for a single check

**HTTP Request**

PUT https://api.gomorpheus.com/api/instances/:id/notes

**JSON Instance Parameters**

23.21.10 Stop an Instance

```
$ curl -X PUT "https://api.gomorpheus.com/api/instances/1/stop" \\
-H "Authorization: BEARER access_token"
```

The above command returns JSON structure like this:

```
{
  "success": true
}
```

This will stop all containers running within an instance.

**HTTP Request**

PUT https://api.gomorpheus.com/api/instances/:id/stop
23.21.11 Start an Instance

```
curl -X PUT "https://api.gomorpheus.com/api/instances/1/start" \ 
-H "Authorization: BEARER access_token"
```

The above command returns JSON structure like this:

```
{
  "success": true
}
```

This will start all containers running within an instance.

**HTTP Request**

PUT https://api.gomorpheus.com/api/instances/:id/start

23.21.12 Restart an Instance

```
curl -X PUT "https://api.gomorpheus.com/api/instances/1/restart" \ 
-H "Authorization: BEARER access_token"
```

The above command returns JSON structure like this:

```
{
  "success": true
}
```

This will restart all containers running within an instance. This includes rebuilding the environment variables and applying settings to the docker containers.

**HTTP Request**

PUT https://api.gomorpheus.com/api/instances/:id/restart

23.21.13 Suspend an Instance

```
curl -X PUT "https://api.gomorpheus.com/api/instances/1/suspend" \ 
-H "Authorization: BEARER access_token"
```

The above command returns JSON structure like this:

```
{
  "success": true
}
```

This will suspend all containers in the instance.

**HTTP Request**

PUT https://api.gomorpheus.com/api/instances/:id/eject
23.21.14 Eject an Instance

```
curl -X PUT "https://api.gomorpheus.com/api/instances/1/eject" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structure like this:

```json
{
  "success": true
}
```

This will eject any ISO media on all containers in the instance.

**HTTP Request**

PUT https://api.gomorpheus.com/api/instances/:id/eject

23.21.15 Resize an Instance

```
curl -X PUT "https://api.gomorpheus.com/api/instances/1/resize" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{
  "instance": {
    "id": 1,
    "plan": {
      "id": 15
    },
  },
  "volumes": [
    {
      "id": "-1",
      "rootVolume": true,
      "name": "root",
      "size": 20,
      "sizeId": null,
      "storageType": null,
      "datastoreId": null
    }
  ],
  "deleteOriginalVolumes": true
}'
```

The above command returns JSON structure like this:

```json
{
  "success": true
}
```

It is possible to resize containers within an instance by increasing their memory plan or storage limit. This is done by assigning a new service plan to the container.
HTTP Request

PUT https://api.gomorpheus.com/api/instances/:id/resize

JSON Parameters

23.21.16 Clone an Instance

23.21.17 Backup an Instance
### 23.21.18 Get list of backups for an Instance

```bash
curl "https://api.gomorpheus.com/api/instances/1773/backups" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structure that looks like this:

```json
{
  "instance": {
    "id": 1773
  },
  "backups": []
}
```

**HTTP Request**

GET https://api.gomorpheus.com/api/instances/:id/backups

### 23.21.19 Import Snapshot of an Instance

```bash
curl -X PUT "https://api.gomorpheus.com/api/instances/1/import-snapshot" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{ "storageProviderId": 1
}
```

The above command returns JSON structure like this:

```json
{
  "success": true
}
```

It is possible to import a snapshot of an instance. This creates a Virtual Image of the instance as it currently exists.

**HTTP Request**

PUT https://api.gomorpheus.com/api/instances/:id/import-snapshot

**JSON Parameters**

### 23.21.20 Get Security Groups

```bash
curl -X GET "https://api.gomorpheus.com/api/instances/1/security-groups" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structure like this:

```json
{ "success": true
}
```
This returns a list of all of the security groups applied to an instance and whether the firewall is enabled.

**HTTP Request**

GET https://api.gomorpheus.com/api/instances/:id/security-groups

**23.21.21 Set Security Groups**

```bash
curl -X POST "https://api.gomorpheus.com/api/instances/1/security-groups" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{ "securityGroupIds": [19, 2] }'
```

The above command returns JSON structure similar to the ‘get’ of security groups.

**HTTP Request**

POST https://api.gomorpheus.com/api/instances/:id/security-groups

**JSON Parameters**

This defines the list of all the security groups applied to an instance.

**23.21.22 Delete an Instance**

```bash
curl -XDELETE "https://api.gomorpheus.com/api/instances/1" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structure like this:

```json
{
  "success": true
}
```

Will delete an instance and all associated monitors and backups.
HTTP Request

DELETE https://api.gomorpheus.com/api/instances/:id

Query Parameters

23.22 Instance Types

Provides a means to find out which instance types are available to your user account. These can vary in range from
database containers, to web containers, to custom containers.

23.22.1 Get All Instance Types

```bash
curl "https://api.gomorpheus.com/api/instance-types"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this

```
{
  "instanceTypes": [
    {
      "id": 12,
      "name": "ActiveMQ",
      "code": "activemq",
      "category": "messaging",
      "active": true,
      "versions": [
        "5.11"
      ],
      "instanceTypeLayouts": [
        {
          "id": 14,
          "code": "activemq-5.11",
          "name": "Single Process",
          "description": "This will provision a single process with no redundancy",
          "provisionType": { /* see provision types */ },
          "optionTypes": { /* see option types */ }
        }
      ],
      "instanceTypeLayouts": [
        {
          "id": 13,
          "name": "Cassandra",
          "code": "cassandra",
          "category": "nosql",
          "active": true,
          "versions": [
            "2.1"
          ],
          "instanceTypeLayouts": [
            {
              "id": 15,
              "code": "cassandra-2.1-single",
              "name": "Single Process",
```
"description": "This will provision a single process with no redundancy",
"provisionType": { /* see provision types */ },
"optionTypes": { /** see option types **/ }
}
},
{
"id": 10,
"name": "Confluence",
"code": "confluence",
"category": "utils",
"active": true,
"versions": [
"5.7"
],
"instanceTypeLayouts": [
{
"id": 12,
"code": "confluence-5.7",
"name": "Single Process",
"description": "This will provision a single process with no redundancy",
"provisionType": { /* see provision types */ },
"optionTypes": { /** see option types **/ }
}
],
"instanceTypeLayouts": [
{
"id": 3,
"code": "elasticsearch-1.5-single",
"name": "Single Process",
"description": "This will provision a single process with no redundancy",
"provisionType": { /* see provision types */ },
"optionTypes": { /** see option types **/ }
}]
},
{
"id": 5,
"name": "Elastic Search",
"code": "elasticsearch",
"category": "nosql",
"active": true,
"versions": [
"1.5"
],
"instanceTypeLayouts": [
{
"id": 3,
"code": "elasticsearch-1.5-single",
"name": "Single Process",
"description": "This will provision a single process with no redundancy",
"provisionType": { /* see provision types */ },
"optionTypes": { /** see option types **/ }
},
{
"id": 4,
"code": "elasticsearch-1.5-cluster",
"name": "Cluster",
"description": "This will provision two nodes, in multi master cluster",
"provisionType": { /* see provision types */ },
"optionTypes": { /** see option types **/ }
}]
},
{
"id": 7,
"name": "Jenkins",
"code": "jenkins",
"category": "dev",
"active": true,
"versions": [
"2.3.22"
],
"instanceTypeLayouts": [
{
"id": 8,
"code": "jenkins-2.3.22",
"name": "Jenkins",
"description": "This will provision a single process",
"provisionType": { /* see provision types */ },
"optionTypes": { /** see option types **/ }
}
]
"code": "jenkins",
"category": "utils",
"active": true,
"versions": [
  "1.596"
],
"instanceTypeLayouts": [
  {
    "id": 8,
    "code": "jenkins-1.596",
    "name": "Single Process",
    "description": "This will provision a single process with no redundancy",
    "provisionType": { /* see provision types */ },
    "optionTypes": { /* see option types */ }
  }
],
"id": 2,
"name": "Memcached",
"code": "memcached",
"category": "cache",
"active": true,
"versions": [
  "1.4"
],
"instanceTypeLayouts": [
  {
    "id": 11,
    "code": "memcached-1.4-single",
    "name": "Single Process",
    "description": "This will provision a single process with no redundancy",
    "provisionType": { /* see provision types */ },
    "optionTypes": { /* see option types */ }
  }
],
"id": 4,
"name": "Mongo",
"code": "mongo",
"category": "nosql",
"active": true,
"versions": [
  "3.0"
],
"instanceTypeLayouts": [
  {
    "id": 16,
    "code": "mongo-3.0-rs",
    "name": "ReplicaSet",
    "description": "This will provision a 3 node replicaSet",
    "provisionType": { /* see provision types */ },
    "optionTypes": { /* see option types */ }
  }
],
"id": 6,
"code": "mongo-3.0-single",
"name": "Single Process",
"description": "This will provision a single process with no redundancy",
"provisionType": { /* see provision types */ },
"optionTypes": { /* see option types */ }
}
],
"instanceTypeLayouts": [
{
"id": 3,
"name": "MySQL",
"code": "mysql",
"category": "sql",
"active": true,
"versions": [
"5.6"
],
"instanceTypeLayouts": [
{
"id": 5,
"code": "mysql-5.6-single",
"name": "Single Process",
"description": "This will provision a single process with no redundancy",
"provisionType": { /* see provision types */ },
"optionTypes": { /* see option types */ }
}
],
"instanceTypeLayouts": [
{
"id": 8,
"name": "Nexus",
"code": "nexus",
"category": "utils",
"active": true,
"versions": [
"2.11"
],
"instanceTypeLayouts": [
{
"id": 9,
"code": "nexus-2.11",
"name": "Single Process",
"description": "This will provision a single process with no redundancy",
"provisionType": { /* see provision types */ },
"optionTypes": { /* see option types */ }
}
],
"instanceTypeLayouts": [
{
"id": 14,
"name": "Nginx",
"code": "nginx",
"category": "web",
"active": true,
"versions": [
"1.9"
],
"instanceTypeLayouts": [
{
"id": 15,
"code": "nginx-1.9",
"name": "Single Process",
"description": "This will provision a single process with no redundancy",
"provisionType": { /* see provision types */ },
"optionTypes": { /* see option types */ }
}]}]}}

23.22. Instance Types
"instanceTypeLayouts": [
],
},
{
"id": 11,
"name": "Postgres",
"code": "postgres",
"category": "sql",
"active": true,
"versions": [
  "9.4"
],
"instanceTypeLayouts": [
  {
    "id": 13,
    "code": "postgres-9.4-single",
    "name": "Single Process",
    "description": "This will provision a single process with no redundancy",
    "provisionType": { /* see provision types */ },
    "optionTypes": { /** see option types */ }
  }
],
},
{
"id": 9,
"name": "RabbitMQ",
"code": "rabbitmq",
"category": "utils",
"active": true,
"versions": [
  "3.5"
],
"instanceTypeLayouts": [
  {
    "id": 10,
    "code": "rabbitmq-3.5",
    "name": "Single Process",
    "description": "This will provision a single process with no redundancy",
    "provisionType": { /* see provision types */ },
    "optionTypes": { /** see option types */ }
  }
],
},
{
"id": 1,
"name": "Redis",
"code": "redis",
"category": "cache",
"active": true,
"versions": [
  "3.0"
],
"instanceTypeLayouts": [
  {
    "id": 1,
    "code": "redis-3.0-single",
    (continues on next page)
### 23.22.2 Get Specific Instance Type

```
curl "https://api.gomorpheus.com/api/instance-types/12"  
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
  "success": true,
  "instanceType": {
    "id": 12,
    "name": "Single Process",
    "description": "This will provision a single process with no redundancy",
    "provisionType": { /* see provision types */ },
    "optionTypes": { /** see option types */ }
  }
}
```
HTTP Request

GET https://api.gomorpheus.com/api/instance-types/:id

23.23 Key Pairs

Morpheus provides a database for keeping track of Key Pairs in the system. These can be used for provisioning servers and auto assigning added keypairs.

23.23.1 Get All Key Pairs

curl "https://api.gomorpheus.com/api/key-pairs"
   -H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```json
{
   "keyPairCount": 1,
   "keyPairs": [
   {
      "accountId": 1,
      "id": 2,
      "name": "Test",
      "privateKey": null,
      "publicKey": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAABQAAbAABQDXhVj50e88bXPnA32iZ0j1gbTkCgTlKwDyGfOTBH56QR9gwU66Bmh+ceU/1m1jS0zNuHtGFlMabbL+7a+MgJ7HVuaV4CR2/a/cplyEzvvuJE61voG2diXIdaFsdFxcvdFadFvCnVOn+TW16rbJ6GR/IwuvS81GqS96z5/I/Jh4R51W5y+xvQ9xnv60JmwbC1N05UVps2KDBSC1ldlN1PR4AFrtYVDSjRmjv5J6YlgjHgc23bk1t0pknMphCfZ davydotcom@Davids-MacBook-Pro-2.local"
   }
   ]
}
```
This endpoint retrieves all key pairs associated with the account.

**HTTP Request**

GET https://api.gomorpheus.com/api/key-pairs

**Query Parameters**

**23.23.2 Get a Specific Key Pair**

```bash
curl "https://api.gomorpheus.com/api/key-pairs/2" \
  -H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
  "keyPair": {
    "accountId": 1,
    "id": 2,
    "name": "Test",
    "privateKey": null,
    "publicKey": "ssh-rsa
      AAAAB3NzaC1yc2EAAAADAQABAAABAQDXhVj50e88bXPmNA32i20ij1gbTkeCgnTkLwDyGfOTBH56QR9gwU668imh+ceU/
      lmlj90iNuHtgFIambbL+7a+Mgj7HVuaV4CR2/a/
      cplyEzvvuJE6IvoGzDiXidafadfxcdafVcEYon+TWl6rbS6GR/IwuvS81GqSj6Z5/
      IJh4R5IW5yzK6z8HTx+vQQN9xv60JmwBCIN05U6ps2KVDBSCi1dIN1PR4AFrtVYDPsJrmvj3DjGnJ6YlgjRgic23bk1t0pkn
      MphCKf2 davyd.com@Davids-MacBook-Pro-2.local"
  },
  "success": true
}
```

This endpoint retrieves a specific key.

**HTTP Request**

GET https://api.gomorpheus.com/api/key-pairs/:id

**URL Parameters**

**23.23.3 Create a KeyPair**

```bash
curl -X POST "https://api.gomorpheus.com/api/key-pairs" \
  -H "Authorization: BEARER access_token" \
  -H "Content-Type: application/json" \
  -d '{"keyPair":{
    "name": "My Key",
    "publicKey": "ssh-rsaAAAAB3NzaC1yc2EAAAADAQABAAABAQDXhVj50e88bXPmNA32i20ij1gbTkeCgnTkLwDyGfOTBH56QR9gwU668imh+ceU/
      lmlj90iNuHtgFIambbL+7a+Mgj7HVuaV4CR2/a/
      cplyEzvvuJE6IvoGzDiXidafadfxcdafVcEYon+TWl6rbS6GR/IwuvS81GqSj6Z5/
      IJh4R5IW5yzK6z8HTx+vQQN9xv60JmwBCIN05U6ps2KVDBSCi1dIN1PR4AFrtVYDPsJrmvj3DjGnJ6YlgjRgic23bk1t0pkn
      MphCKf2 davyd.com@Davids-MacBook-Pro-2.local"
  }
}
```

(continues on next page)
"publicKey": "ssh-rsa",
"privateKey": "privateKey Optional for most cases"
}

The above command returns JSON structured like getting a single keyPair:

**HTTP Request**

POST https://api.gomorpheus.com/api/key-pairs

**JSON Check Parameters**

**NOTE** The Public and Private key are stored in encrypted form in the database.

### 23.23.4 Delete a Key Pair

```
curl -XDELETE "https://api.gomorpheus.com/api/key-pairs/1" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON Structured like this:

```
{
  "success": true
}
```

Will delete a key pair from the system and make it no longer usable.

**HTTP Request**

DELETE https://api.gomorpheus.com/api/key-pairs/:id

### 23.24 Monitor Checks

These entities define what and when a check is executed within the Morpheus system. Morpheus supports a vast array of different check types (not solely web checks). The API provides a means to list all of an account’s checks in addition to create, modify, mute, and or delete them.

#### 23.24.1 Get All Checks

```
curl "https://api.gomorpheus.com/api/monitoring/checks" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:
This endpoint retrieves all checks and their JSON encoded configuration attributes based on check type. Check data is encrypted in the database.
HTTP Request

GET https://api.gomorpheus.com/api/monitoring/checks

Query Parameters

23.24.2 Get a Specific Check

curl "https://api.gomorpheus.com/api/monitoring/checks/1" \
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```json
{
  "success": true,
  "check": {
    "id": 798,
    "account": {
      "id": 1
    },
    "active": true,
    "availability": 99.9804109,
    "checkAgent": null,
    "checkIntegrations": [
    ],
    "checkInterval": 300,
    "checkSpec": null,
    "checkType": {
      "id": 1
    },
    "config": {"webMethod": "GET", "webUrl": "http://google.com"},
    "createIncident": true,
    "dateCreated": "2015-05-16T12:05:23Z",
    "deleted": false,
    "description": null,
    "health": 10,
    "history": "[1433339580607,14333395959119,1433339613169,1433339625412,1433339641010,1433339655209,1433339670178,1433339687802,1433339700471,1433339715171,1433339730710,1433339745351,1433339764299,1433339775508,1433339790377,1433339805373,1433339820944,1433339835996,1433339850317,1433339865833,1433339880884,1433339895489,1433339910554,1433339925660,1433339940875,1433339956143,1433339970551,1433339985179,143340000961,143340015765],"successList":true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,
    "description": null,
    "health": 10,
    "history": "[1433339580607,14333395959119,1433339613169,1433339625412,1433339641010,1433339655209,1433339670178,1433339687802,1433339700471,1433339715171,1433339730710,1433339745351,1433339764299,1433339775508,1433339790377,1433339805373,1433339820944,1433339835996,1433339850317,1433339865833,1433339880884,1433339895489,1433339910554,1433339925660,1433339940875,1433339956143,1433339970551,1433339985179,143340000961,143340015765],"successList":true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,
    "description": null,
    "health": 10,
    "history": "[1433339580607,14333395959119,1433339613169,1433339625412,1433339641010,1433339655209,1433339670178,1433339687802,1433339700471,1433339715171,1433339730710,1433339745351,1433339764299,1433339775508,1433339790377,1433339805373,1433339820944,1433339835996,1433339850317,1433339865833,1433339880884,1433339895489,1433339910554,1433339925660,1433339940875,1433339956143,1433339970551,1433339985179,143340000961,143340015765],"successList":true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true
```
This endpoint retrieves a specific check.

**HTTP Request**

GET https://api.gomorpheus.com/api/monitoring/checks/:id

**URL Parameters**

23.24.3 Create a Check

```
curl -XPOST "https://api.gomorpheus.com/api/monitoring/checks" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"check":{  "name": "My Check",  "checkType": {"code": "webGetCheck"},  "inUptime": true,  "severity": "critical",  "description": null,  "checkInterval": 300,  "checkAgent": null,  "active": true,  "config": {    "webMethod": "GET",    "webUrl": "http://google.com"  }}}'
```

The above command returns a similar JSON structure when submitting a GET request for a single check

**HTTP Request**

POST https://api.gomorpheus.com/api/monitoring/checks

**JSON Check Parameters**

23.24.4 Updating a Check
The above command returns a similar JSON structure when submitting a GET request for a single check

HTTP Request

PUT https://api.gomorpheus.com/api/monitoring/checks/:id

JSON Check Parameters

23.24.5 Check Types and Options

We support a wide variety of check types. Each check type varies in its configuration payload when determining how the check should be run.

- Creates a Web type Check

Web Get Check

```json
{
   "check": {
      "name": "My Web Check",
      "checkType": {"code": "webGetCheck"},
      "config": {
         "webMethod": "GET",
         "webUrl": "http://google.com",
         "checkUser": "basicUser",
         "checkPassword": "basicPassword",
         "webTextMatch": "Login",
         "textCheckOn": "on"
      }
   }
}
```

Code: `webGetCheck`

Web check type allows you to perform a standard web request and validate the response came back successfully. Additionally, you can check for matching text within the result. There are several `config` parameters available for use with this type of check.
MySQL Check

```
{
  "check": {
    "name": "MySQL Check",
    "checkType": {"code": "mysqlCheck"},
    "config": {
      "dbHost": "db.example.org", "dbPort": "3306", "dbUser": "basicUser",
      "dbPassword": "basicPassword", "dbName": "mydb", "dbQuery": "select 1",
      "checkOperator": "lt", "checkResult": 2
    }
  }
}
```

Code: mysqlCheck

MySQL check allows you to execute a query so that you may validate the value returned in addition to verifying the database is responding. This can be useful for doing a slow query check or just making sure something isn’t growing out of control.

SQL Server Check

```
{
  "check": {
    "name": "SQL Server Check",
    "checkType": {"code": "sqlCheck"},
    "config": {
      "dbHost": "db.example.org", "dbPort": "3306", "dbUser": "basicUser",
      "dbPassword": "basicPassword", "dbName": "mydb", "dbQuery": "select 1",
      "checkOperator": "lt", "checkResult": 2
    }
  }
}
```

Code: sqlCheck

SQL Server check allows to execute a query so that you may validate the value returned in addition to verifying the database is responding. This can be useful for doing a slow query check or just making sure something isn’t growing out of control.

PostgreSQL Check

```
{
  "check": {
    "name": "PostgreSQL Check",
    "checkType": {"code": "postgresCheck"},
    "config": {
      "dbHost": "db.example.org", "dbPort": "3306", "dbUser": "basicUser",
      "dbPassword": "basicPassword", "dbName": "mydb", "dbQuery": "select 1",
      "checkOperator": "lt", "checkResult": 2
    }
  }
}
```

Code: postgresCheck

PostgreSQL check allows to execute a query so that you may validate the value returned in addition to verifying the database is responding. This can be useful for doing a slow query check or just making sure something isn’t growing out of control.
Socket Check

```json
{
    "check": {
        "name": "Socket Check",
        "checkType": {"code": "socketCheck"},
        "config": {"host": "test.example.org", "port": "3306", "send": "blah", "responseMatch": "OK"}
    }
}
```

Code: socketCheck

Socket check confirms a certain TCP port is up and responding in your environment. It can be configured to do an initial send upon connect and compare and expected response of the service.

Elastic Search Check

```json
{
    "check": {
        "name": "Elastic Search Check",
        "checkType": {"code": "elasticSearchCheck"},
        "config": {"esHost": "test.example.org", "esPort": "9200"}
    }
}
```

Code: elasticSearchCheck

Elasticsearch check is capable of connecting to your Elasticsearch, cluster or node, verifying its health. In addition, Morpheus will also pull statistical information such as: document size, capacity, and CPU usage.

Push Check

```json
{
    "check": {
        "name": "Push Check",
        "checkType": {"code": "pushCheck"}
    }
}
```

Code: pushCheck

A Push check is a check that is updated by a web hook. An external source is responsible for periodically submitting a check status. Please see the section on Push Checks API for details.

23.24.6 SSH Tunneling

SSH tunneling options allow the different check types to tunnel to a host via a proxy, and execute checks relative to the proxy. A SSH tunnel can use your account generated public and private key-pairs or SSH password (we strongly recommend using a key-pair).

To enable SSH tunneling for a check, add the following parameters to any check type config as seen earlier in the Check Types section.
23.24.7 Mute a Check

curl -X PUT "https://api.gomorpheus.com/api/monitoring/checks/1/mute" \
-H "Authorization: Bearer access_token" \
-H "Content-Type: application/json" \
-d '{"enabled":true}'

The above command returns JSON structure like this:

```
{
  "muteState": "QUARANTINED",
  "success": true
}
```

This endpoint can be used to toggle the mute state of a check on and off.

**HTTP Request**

PUT https://api.gomorpheus.com/api/monitoring/checks/:id/mute

**JSON Parameters**

23.24.8 Mute All Checks

curl -X PUT "https://api.gomorpheus.com/api/monitoring/checks/mute-all" \
-H "Authorization: Bearer access_token" \
-H "Content-Type: application/json" \
-d '{"enabled":true}'

The above command returns JSON structure like this:

```
{
  "muteState": "QUARANTINED",
  "updated": 20,
  "success": true
}
```

This endpoint can be used to toggle the mute state on and off for all checks.

**HTTP Request**

PUT https://api.gomorpheus.com/api/monitoring/checks/mute-all
23.24.9  Delete a Check

```
curl -XDELETE "https://api.gomorpheus.com/api/monitoring/checks/1" \
  -H "Authorization: BEARER access_token"
```

The above command returns JSON structure like this:

```
{
    "success": true
}
```

A deleted check can be fetched from the API using the GET method to synchronize client side views, but can not be executed or updated.

23.25  Monitor Incidents

These entities are incidents that result from Checks. The API provides a means to list all of an account’s incidents and also update, mute, close, and reopen them.

23.25.1  Get All Incidents

```
curl "https://api.gomorpheus.com/api/monitoring/incidents"
  -H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
    "incidents": [
        {
            "id": 12,
            "account": {
                "id": 1,
            },
            "app": null,
            "autoClose": true,
            "channelId": "cdff5f78-19df-41e0-b6dc-2ab87cedeae5",
            "checkGroups": [
            ],
            "checks": [
            ],
            "comment": "",
            "displayName": "test-mysql",
            "duration": null,
            "endDate": null,
```

(continues on next page)
This endpoint retrieves all incidents.

**HTTP Request**

GET https://api.gomorpheus.com/api/monitoring/incidents

**Query Parameters**

23.25.2 Get a Specific Incident

```bash
curl "https://api.gomorpheus.com/api/monitoring/incidents/1" \ 
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
    "incident": {
        "id": 1,
        "account": {
            "id": 1
        },
        "app": null,
        "autoClose": true,
        "channelId": "3f2fb251-9f87-4e28-88f7-7e0df24f4d50",
        "checkGroups": [
            {
                "id": 129,
                "name": "test-nginx"
            }
        ],
        "checks": [  
```

(continues on next page)
This endpoint retrieves a specific incident.

**HTTP Request**

GET https://api.gomorpheus.com/api/monitoring/incidents/:id
23.25.3 Updating an Incident

```bash
curl -XPUT "https://api.gomorpheus.com/api/monitoring/incidents/1" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"incident":{
    "resolution": "We fixed the problem",
}}'
```

The above command returns JSON structure like this:

```json
{
    "success": true
}
```

This endpoint can be used to update certain properties of an incident.

HTTP Request

PUT https://api.gomorpheus.com/api/monitoring/incidents/:id

JSON Parameters

23.25.4 Mute an Incident

```bash
curl -XPUT "https://api.gomorpheus.com/api/monitoring/incidents/1/mute" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"enabled":true}"
```

The above command returns JSON structure like this:

```json
{
    "muteState": "QUARANTINED",
    "success": true
}
```

This endpoint can be used to toggle the mute state (inUptime) of an incident on and off.

HTTP Request

PUT https://api.gomorpheus.com/api/monitoring/incidents/:id/mute

JSON Parameters

23.25.5 Mute All Incidents
curl -XPUT "https://api.gomorpheus.com/api/monitoring/incidents/mute-all" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{}"}

The above command returns JSON structure like this:

```json
{
  "muteState": "QUARANTINED",
  "updated": 11,
  "success": true
}
```

This endpoint can be used to toggle the mute state (inUptime) of all open incidents.

**HTTP Request**

PUT https://api.gomorpheus.com/api/monitoring/incidents/mute-all

**JSON Parameters**

### 23.25.6 Close an Incident

curl -XDELETE "https://api.gomorpheus.com/api/monitoring/incidents/1" \
-H "Authorization: BEARER access_token"

The above command returns JSON structure like this:

```json
{
  "success": true,
  "msg": "Incident 1 is closed"
}
```

**HTTP Request**

DELETE https://api.gomorpheus.com/api/monitoring/incidents/:id

### 23.25.7 Reopen an Incident

curl -XPUT "https://api.gomorpheus.com/api/monitoring/incidents/1/reopen" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{}"'

The above command returns JSON structure like this:

```json
{
  "success": true,
  "msg": "Incident 1 is now open again"
}
```

This endpoint can be used to toggle the mute state (inUptime) of an incident on and off.
HTTP Request

PUT https://api.gomorpheus.com/api/monitoring/incidents/:id/reopen

23.26 Option Types

Morpheus has several objects that have dynamic models depending on the type of the object. This includes options when provisioning different instances or even options when defining tasks or creating docker hosts!. This section aims to describe what is contained in the option-types association as well as how to query morpheus for available options in certain option-type scenarios.

23.26.1 Example of an Option Type Record

```json
{
    "optionTypes": [  
        {
            "name": "subnet",
            "description": null,
            "fieldName": "subnetId",
            "fieldLabel": "Subnet",
            "fieldContext": "config",
            "fieldAddOn": null,
            "placeHolder": null,
            "helpBlock": "",
            "defaultValue": null,
            "optionSource": "amazonSubnet",
            "type": "select",
            "advanced": false,
            "required": true,
            "editable": false,
            "config": [],
            "displayOrder": 100
        },
        {
            "name": "security group",
            "description": null,
            "fieldName": "securityId",
            "fieldLabel": "Security Group",
            "fieldContext": "config",
            "fieldAddOn": null,
            "placeHolder": null,
            "helpBlock": "",
            "defaultValue": null,
            "optionSource": "amazonSecurityGroup",
            "type": "select",
            "advanced": false,
            "required": true,
            "editable": false,
            "config": [],
            "displayOrder": 101
        },
        {
            "name": "public key",
            "description": null,
            "fieldName": null,
            "fieldLabel": null,
            "fieldContext": null,
            "fieldAddOn": null,
            "placeHolder": null,
            "helpBlock": "",
            "defaultValue": null,
            "optionSource": null,
            "type": "null",
            "advanced": false,
            "required": false,
            "editable": false,
            "config": [],
            "displayOrder": 102
        }
    ]
}
```

(continues on next page)
Option types can easily represent some common input types, including text, number, radio, checkbox, and drop-down/multiple choice.

**JSON Parameters**

### 23.26.2 Get Option Source Data

```
curl "https://api.gomorpheus.com/api/options/keyPairs"  
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this

```
[
  {"name": "Davids Key Pair", "value": 1}
]
```

**HTTP Request**

GET https://api.gomorpheus.com/api/options/:optionSource

Returns a list of name/value pairs for option-type models. Some option-types depend on input data for proper representation. This typically includes zoneId or siteId for the item being provisioned as request parameters or sometimes previous option type parameters.

### 23.27 Policies

Provides API interfaces for managing Policies.

#### 23.27.1 Get All Policies

```
curl "https://api.gomorpheus.com/api/policies"  
-H "Authorization: BEARER access_token"
```
The above command returns JSON structured like this:

```json
{
    "policies": [
        {
            "id": 1,
            "name": null,
            "description": "global max containers",
            "policyType": {
                "id": 5,
                "code": "maxContainers",
                "name": "Max Containers"
            },
            "zone": null,
            "site": null,
            "user": null,
            "refType": null,
            "refId": null,
            "config": {
                "maxContainers": 999
            },
            "enabled": true,
            "owner": {
                "id": 1,
                "name": "root"
            },
            "accounts": [
            ]
        },
        {
            "id": 2,
            "name": null,
            "description": "dev budget",
            "policyType": {
                "id": 16,
                "code": "maxPrice",
                "name": "Budget"
            },
            "zone": null,
            "site": null,
            "user": null,
            "refType": null,
            "refId": null,
            "config": {
                "maxPrice": 1500,
                "maxPriceCurrency": "USD",
                "maxPriceUnit": "month"
            },
            "enabled": true,
            "owner": {
                "id": 1,
                "name": "root"
            },
            "accounts": [
            {
                "id": 2,
                "name": "dev"
            }
            ]
        }
    ]
}
```


```json
{
    "id": 3,
    "name": "test group maxcores",
    "description": null,
    "policyType": {
        "id": 3,
        "code": "maxCores",
        "name": "Max Cores"
    },
    "zone": null,
    "site": {
        "id": 2,
        "name": "test group"
    },
    "user": null,
    "refType": "ComputeSite",
    "refId": 1,
    "config": {
        "maxCores": 20
    },
    "enabled": true,
    "owner": {
        "id": 1,
        "name": "root"
    },
    "accounts": [
    ],
},
{
    "id": 4,
    "name": null,
    "description": "hulk max storage",
    "policyType": {
        "id": 2,
        "code": "maxStorage",
        "name": "Max Storage"
    },
    "zone": null,
    "site": null,
    "user": {
        "id": 26,
        "username": "hulk"
    },
    "refType": "User",
    "refId": 26,
    "config": {
        "maxStorage": 10000
    },
    "enabled": true,
    "owner": {
        "id": 1,
        "name": "root"
    },
}
```
This endpoint retrieves all policies associated with the account.

**HTTP Request**

GET https://api.gomorpheus.com/api/policies

**Query Parameters**

**23.27.2 Get a Specific Policy**

```
curl "https://api.gomorpheus.com/api/policies/4" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
  "policy": {
    "id": 4,
    "name": "hulk max storage",
    "description": "Limit the hulkster",
    "policyType": {
      "id": 2,
      "code": "maxStorage",
      "name": "Max Storage"
    },
    "zone": null,
    "site": null,
    "user": {
      "id": 26,
      "username": "hulk"
    },
    "refType": "User",
    "refId": 26,
    "config": {
      "maxStorage": "10000"
    },
    "enabled": true,
    "owner": {
      "id": 1,
      "name": "root"
    }
  }
}
```
This endpoint retrieves a specific policy.

**HTTP Request**

GET https://api.gomorpheus.com/api/policies/:id

**URL Parameters**

### 23.27.3 Policy Types

```bash
curl "https://api.gomorpheus.com/api/policy-types" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
  "policyTypes": [
    {
      "id": 15,
      "code": "createBackup",
      "name": "Backup Creation",
      "description": "",
      "category": "provision",
      "loadMethod": "loadBackupCreation",
      "enforceMethod": "enforceBackupCreation",
      "prepareMethod": "prepareBackupCreation",
      "validateMethod": "validateBackupCreation",
      "enforceOnProvision": true,
      "enforceOnManaged": false,
      "optionTypes": [
        {
          "code": "policyType.createBackupType",
          "name": "Enforcement Type",
          "description": null,
          "type": "select",
          "displayOrder": 1,
          "optionSource": "policyGenericType",
          "defaultValue": "",
          "placeHolder": null,
          "helpBlock": "",
          "required": true,
          "fieldLabel": "Enforcement Type",
          "fieldName": "createBackupType",
          "fieldContext": "config"
        },
        {
          "code": "policyType.createBackup",
```
This endpoint returns a list of all policy types.

**HTTP Request**

GET https://api.gomorpheus.com/api/policy-types

**Policy Type Options**

**Backup Creation (createBackup)**

**Budget (maxPrice)**

**Expiration (lifecycle)**

**Host Name (serverNaming)**

**Hostname (hostNaming)**

**Instance Name(naming)**

**Max Containers (maxContainers)**

**Max Cores (maxCores)**

**Max Hosts (maxHosts)**
Max Memory (maxMemory)

Max Storage (maxStorage)

Max VMs (maxVms)

Power Schedule (powerSchedule)

Provision Approval (provisionApproval)

Shutdown (shutdown)

User Creation (createUser)

23.27.4 Create a Policy

curl -XPOST "https://api.gomorpheus.com/api/policies" \
   -H "Authorization: BEARER access_token" \
   -H "Content-Type: application/json" \
   -d '{
   "policy": {
      "name": "max hosts",
      "description": null,
      "policyType": {
         "code": "maxHosts"
      },
      "config": {
         "maxHosts": 99
      },
      "enabled": true,
      "accounts": []
   }
}'

The above command returns JSON structured like getting a single policy:

HTTP Request

POST https://api.gomorpheus.com/api/policies

JSON Parameters

Create a Policy For a Group

Policies can be scoped to a group by passing the following:

Alternatively, the Group Policies endpoint can be used.

Create a Policy For a Cloud

Policies can be scoped to a cloud by passing the following:
Alternatively, the *Cloud Policies* endpoint can be used.

### Create a Policy For a User

Policies can be scoped to a cloud by passing the following:

#### 23.27.5 Update a Policy

```bash
curl -XPUT "https://api.gomorpheus.com/api/policies/1" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{
  "policy": {
    "name": "max containers 1000",
    "config": {
      "maxContainers": 1000
    },
  }
}'
```

The above command returns JSON structured like getting a single policy:

**HTTP Request**

PUT https://api.gomorpheus.com/api/policies/1

**URL Parameters**

**JSON Parameters**

#### 23.27.6 Delete a Policy

```bash
curl -XDELETE "https://api.gomorpheus.com/api/policies/1" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON Structured like this:

```json
{
  "success": true
}
```

Will delete a policy from the system and make it no longer usable.

**HTTP Request**

DELETE https://api.gomorpheus.com/api/policies/:id
URL Parameters

23.27.7 Group Policies

Policies scoped to a specific group can also be managed at another endpoint.

curl "https://api.gomorpheus.com/api/groups/1/policies"
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```json
{
  "policies": [
    {
      "id": 19,
      "name": "smallgroup max cores",
      "description": null,
      "policyType": {
        "id": 3,
        "code": "maxCores",
        "name": "Max Cores"
      },
      "zone": null,
      "site": {
        "id": 1,
        "name": "smallgroup"
      },
      "user": null,
      "refType": "ComputeSite",
      "refId": 1,
      "config": {
        "maxCores": 20
      },
      "enabled": true,
      "owner": {
        "id": 1,
        "name": "root"
      },
      "accounts": [
      ]
    }
  ],
  "meta": {
    "size": 1,
    "total": 1,
    "offset": 0,
    "max": 25
  }
}
```

HTTP Request

GET https://api.gomorpheus.com/api/groups/:groupId/policies
URL Parameters

23.27.8 Cloud Policies

Policies scoped to a specific cloud can also be managed at another endpoint.

```
curl "https://api.gomorpheus.com/api/zones/1/policies"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
  "policies": [
    {
      "id": 19,
      "name": "bigcloud max cores",
      "description": null,
      "policyType": {
        "id": 3,
        "code": "maxCores",
        "name": "Max Cores"
      },
      "zone": {
        "id": 1,
        "name": "bigcloud"
      },
      "site": null,
      "user": null,
      "refType": "ComputeZone",
      "refId": 1,
      "config": {
        "maxCores": 1500
      },
      "enabled": true,
      "owner": {
        "id": 1,
        "name": "root"
      },
      "accounts": []
    }
  ],
  "meta": {
    "size": 1,
    "total": 1,
    "offset": 0,
    "max": 25
  }
}
```

HTTP Request

GET https://api.gomorpheus.com/api/zones/:cloudId/policies
URL Parameters

23.28 Power Schedules

Power Schedules can be configured to automatically power on and off your instances and servers.

23.28.1 Get All Power Schedules

```
curl "https://api.gomorpheus.com/api/power-schedules" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
  "schedules": [
    {
      "id": 1,
      "name": "weekday daytime",
      "description": "weekday daytime hours",
      "enabled": true,
      "scheduleType": "power",
      "scheduleTimezone": "America/New_York",
      "sundayOn": 0.0,
      "sundayOff": 0.0,
      "mondayOn": 7.0,
      "mondayOff": 19.0,
      "tuesdayOn": 7.0,
      "tuesdayOff": 19.0,
      "wednesdayOn": 7.0,
      "wednesdayOff": 19.0,
      "thursdayOn": 7.0,
      "thursdayOff": 19.0,
      "fridayOn": 7.0,
      "fridayOff": 19.0,
      "saturdayOn": 0.0,
      "saturdayOff": 0.0,
      "totalMonthlyHoursSaved": 463.32,
      "dateCreated": "2018-03-01T07:56:38+0000",
      "lastUpdated": "2018-09-13T21:38:19+0000"
    }
  ],
  "meta": {
    "size": 1,
    "total": 1,
    "max": 25,
    "offset": 0
  }
}
```

This endpoint retrieves all power schedules associated with the account.

HTTP Request

GET https://api.gomorpheus.com/api/power-schedules
Query Parameters

23.28.2 Get a Specific Power Schedule

```
curl "https://api.gomorpheus.com/api/power-schedules/2" \ 
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
  "schedule": {
    "id": 2,
    "name": "my hours",
    "description": null,
    "enabled": true,
    "scheduleType": "power",
    "scheduleTimezone": "America/New_York",
    "sundayOn": 5.5,
    "sundayOff": 24.0,
    "mondayOn": 0.0,
    "mondayOff": 24.0,
    "tuesdayOn": 0.0,
    "tuesdayOff": 24.0,
    "wednesdayOn": 0.0,
    "wednesdayOff": 24.0,
    "thursdayOn": 0.0,
    "thursdayOff": 24.0,
    "fridayOn": 0.0,
    "fridayOff": 24.0,
    "saturdayOn": 0.0,
    "saturdayOff": 24.0,
    "totalMonthlyHoursSaved": 23.595,
    "dateCreated": "2018-03-07T18:34:08+0000",
    "lastUpdated": "2018-03-07T18:34:08+0000"
  },
  "instances": [
  ],
  "servers": [
  ]
}
```

This endpoint retrieves a specific power schedule.

HTTP Request

GET https://api.gomorpheus.com/api/power-schedules/:id

URL Parameters

23.28.3 Create a Power Schedule
The above command returns JSON structured like getting a single power schedule:

HTTP Request

POST https://api.gomorpheus.com/api/power-schedules

JSON Parameters

23.28.4 Update a Power Schedule

The above command returns JSON structured like getting a single power schedule:
HTTP Request

PUT https://api.gomorpheus.com/api/power-schedules/:id

JSON Parameters

See Create.

23.28.5 Delete a Power Schedule

curl -XDELETE "https://api.gomorpheus.com/api/power-schedules/1" \
-H "Authorization: BEARER access_token"

The above command returns JSON Structured like this:

```
{
  "success": true
}
```

Will delete a power schedule from the system and make it no longer usable.

HTTP Request

DELETE https://api.gomorpheus.com/api/power-schedules/:id

URL Parameters

23.28.6 Add Instances to a Power Schedule

curl -XPUT "https://api.gomorpheus.com/api/power-schedules/2/add-instances" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{
  "instances": [ 
    231, 232 
  ]
}'

The above command returns JSON structured like this:

```
{
  "success": true
}
```

Add one or many instances to a power schedule.

HTTP Request

PUT https://api.gomorpheus.com/api/power-schedules/:id/add-instances
23.28.7 Remove Instances from a Power Schedule

```
curl -XPUT "https://api.gomorpheus.com/api/power-schedules/2/remove-instances" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{
  "instances": [232]
}'
```

The above command returns JSON structured like this:

```
{
  "success": true
}
```

Remove one or many instances from a power schedule.

**HTTP Request**

PUT https://api.gomorpheus.com/api/power-schedules/:id/remove-instances

23.28.8 Add Servers to a Power Schedule

```
curl -XPUT "https://api.gomorpheus.com/api/power-schedules/2/add-servers" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{
  "servers": [6,7,8]
}'
```

The above command returns JSON structured like this:

```
{
  "success": true
}
```

Add one or many servers to a power schedule.

**HTTP Request**

PUT https://api.gomorpheus.com/api/power-schedules/:id/add-servers
**URL Parameters**

**JSON Parameters**

### 23.28.9 Remove Servers from a Power Schedule

```
curl -XPUT "https://api.gomorpheus.com/api/power-schedules/2/remove-servers" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{
  "servers": [
    7, 8
  ]
}'
```

The above command returns JSON structured like this:

```
{
  "success": true
}
```

Remove one or many servers from a power schedule.

**HTTP Request**

PUT https://api.gomorpheus.com/api/power-schedules/:id/remove-servers

**URL Parameters**

**JSON Parameters**

### 23.29 Preseed Scripts

Preseed Scripts are used in the Image Builder service. See Image Builds

#### 23.29.1 Get All Preseed Scripts

```
curl "https://api.gomorpheus.com/api/preseed-scripts" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
  "preseedScripts": [
    {
      "id": 1,
      "account": {
        "id": 1,
        "name": "root"
      },
      "fileName": "debian 8",
    }
  ]
}
```

(continues on next page)
This endpoint retrieves all preseed scripts associated with the account.

**HTTP Request**

GET https://api.gomorpheus.com/api/preseed-scripts

**Query Parameters**

23.29.2 Get a Specific Preseed Script

```
curl "https://api.gomorpheus.com/api/preseed-scripts/1" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
  "preseedScript": {
    "id": 1,
    "account": {
      "id": 1,
      "name": "root"
    },
    "fileName": "debian 8",
    "description": null,
    "content": "...",
    "createdBy": {
      "username": "admin"
    }
  }
}
```

This endpoint retrieves a specific preseed script.

**HTTP Request**

GET https://api.gomorpheus.com/api/preseed-scripts/:id
URL Parameters

23.29.3 Create a Preseed Script

```
curl -X POST "https://api.gomorpheus.com/api/preseed-scripts" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{
  "preseedScript": {
    "fileName": "ubuntu build",
    "content": "choose-mirror-bin mirror/http/proxy string\nd-i apt-setup/use_mirror
→boolean true\nd-i base-installer/kernel/override-image string linux-server\n→setup/utc default true\nset-auto\n→volatility yüklerini içeren bir disk için sadece bir disk kullanacağını belirtin. Eğer sistemde yalnızca bir diski varsa, kurulumun tarayıcı kullanır, aksi takdirde cihaz ismi belirtin.\
\n-VM parçalama, kullanıcı genellikle bir uyarı alır. Bu, preseeded away...
d-i partman-lvm/device_remove_lvm boolean true\n\n-VM parçalama için, seçilebilir herhangi bir volum grup için kullanılacak miktarı belirleyebilirsiniz:
d-i partman-auto-lvm/guided_size string max
\nYou can choose one of the three predefined partitioning recipes:
d-i partman-auto/choose_recipe select atomic
d-i partman-auto/expert_recipe
\npartman-auto/choose_recipe select single-root
d-i partman/mount_style select uuid
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true
d-i partman/confirm_write_new_label boolean true
d-i passwd/root-login boolean false
d-i passwd/root-password-again password password
d-i passwd/username string builderbot
\n-password passwords\n\n-pkgsel/include string openssh-server\n\npkgsel/install-language-support boolean false\n\npkgsel/update-policy select none\n\npackaged version of VirtualBox Guest Additions being installed\ndo not press/early_\n\nCommand string sed -i '/in-target/idiscover(){/sbin/discover|grep -v VirtualBox;}' /usr/lib/pre-pkgsel.d/20install-hwpackages\n\nTimezone string /etc/timezone
\nUser setup:\n\ninclude boolean false\ntasksel tasksel/first multiselect
\nstandard, ssh-server"

(continues on next page)```
The above command returns JSON structured like getting a single preseed script:

**HTTP Request**

**POST** https://api.gomorpheus.com/api/preseed-scripts

**JSON Parameters**

23.29.4 Update a Preseed Script

curl -XPUT "https://api.gomorpheus.com/api/preseed-scripts/1" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{
    "preseedScript": {
        "fileName": "good ubuntu"
    }
}

The above command returns JSON structured like getting a single preseed script:

**HTTP Request**

**PUT** https://api.gomorpheus.com/api/preseed-scripts/1

**URL Parameters**

**JSON Parameters**

See Create.

23.29.5 Delete a Preseed Script

curl -XDELETE "https://api.gomorpheus.com/api/preseed-scripts/1" \
-H "Authorization: BEARER access_token"

The above command returns JSON Structured like this:

```json
{
    "success": true
}
```

Will delete a preseed script from the system and make it no longer usable.
HTTP Request

DELETE https://api.gomorpheus.com/api/preseed-scripts/:id

URL Parameters

23.30 Process History

Provides API interfaces for viewing historical processes for instances.

23.30.1 Get All Processes

curl "https://api.gomorpheus.com/api/processes" \
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```json
{
    "processes": [
        {
            "id": 250,
            "accountId": 1,
            "uniqueId": "cebc47ec-cb2f-417a-886e-dd60cf81db26",
            "processType": {
                "code": "provision",
                "name": "provision"
            },
            "description": null,
            "subType": null,
            "subId": null,
            "zoneId": 34,
            "integrationId": null,
            "instanceId": 238,
            "containerId": 240,
            "serverId": 601,
            "containerName": "apachetest",
            "displayName": "apachetest",
            "timerCategory": "vmware",
            "timerSubCategory": "28",
            "status": "failed",
            "reason": null,
            "percent": 100.0,
            "statusEta": 348246,
            "message": null,
            "output": null,
            "error": null,
            "startDate": "2018-09-28T19:10:56+0000",
            "endDate": "2018-09-28T20:21:49+0000",
            "duration": 4253127,
            "dateCreated": "2018-09-28T19:10:56+0000",
            "lastUpdated": "2018-09-28T20:21:49+0000",
            "createdBy": {
                "username": "admin",
                "displayName": "Admin"
            }
        }
    ]
}
```

(continues on next page)
This endpoint retrieves all processes.

**HTTP Request**

GET https://api.gomorpheus.com/api/processes

**Query Parameters**

23.30.2 Get a Specific Process

```bash
curl "https://api.gomorpheus.com/api/processes/250" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
    "process": {
        "id": 250,
        "accountId": 1,
        "uniqueId": "cebc47ec-cb2f-417a-886e-dd60cf81db26",
        "processType": {
            "code": "provision",
            "name": "provision"
        },
        "description": null,
        "subType": null,
        "subId": null,
        "zoneId": 34,
        "integrationId": null,
        "instanceId": 238,
        "containerId": 240,
        "serverId": 601,
        "containerName": "apachetest",
        "displayName": "apachetest",
        "timerCategory": "vmware",
        "updatedBy": {
            "username": "admin",
            "displayName": "Admin"
        },
        "events": []
    }
}
```

(continues on next page)
"timerSubCategory": "28",
"status": "failed",
"reason": null,
"percent": 100.0,
"statusEta": 348246,
"message": null,
"output": null,
"error": null,
"startDate": "2018-09-28T19:10:56+0000",
"endDate": "2018-09-28T20:21:49+0000",
"duration": 4253127,
"dateCreated": "2018-09-28T19:10:56+0000",
"lastUpdated": "2018-09-28T20:21:49+0000",
"createdBy": {
  "username": "admin",
  "displayName": "Admin"
},
"updatedBy": {
  "username": "admin",
  "displayName": "Admin"
},
"events": [
  {
    "id": 940,
    "processId": 250,
    "accountId": 1,
    "uniqueId": "54bf6265-1e86-45b4-b1a7-d4b198b13c45",
    "processType": {
      "code": "provisionResources",
      "name": "prepare resources"
    },
    "description": null,
    "refType": "container",
    "refId": 240,
    "subType": null,
    "subId": null,
    "zoneId": 34,
    "integrationId": null,
    "instanceId": 238,
    "containerId": 240,
    "serverId": 601,
    "containerName": "apachetest",
    "displayName": "apachetest",
    "status": "complete",
    "reason": null,
    "percent": 100.0,
    "statusEta": 348246,
    "message": null,
    "output": null,
    "error": null,
    "startDate": "2018-09-28T19:10:56+0000",
    "endDate": "2018-09-28T19:10:57+0000",
    "duration": 921,
    "dateCreated": "2018-09-28T19:10:56+0000",
    "lastUpdated": "2018-09-28T19:10:57+0000",
    "createdBy": {
      "username": "admin",
      "displayName": "Admin"
    }
  },
"displayName": "Admin",
],
"updatedBy": {
  "username": "admin",
  "displayName": "Admin"
}
],
{
  "id": 941,
  "processId": 250,
  "accountId": 1,
  "uniqueId": "9a9791b7-0091-4ba7-be4d-e1586be3078c",
  "processType": {
    "code": "provisionImage",
    "name": "prepare image"
  },
  "description": null,
  "refType": "container",
  "refId": 240,
  "subType": null,
  "subId": null,
  "zoneId": 34,
  "integrationId": null,
  "instanceId": 238,
  "containerId": 240,
  "serverId": 601,
  "containerName": "apachetest",
  "displayName": "apachetest",
  "status": "complete",
  "reason": null,
  "percent": 100.0,
  "statusEta": 348246,
  "message": null,
  "output": null,
  "error": null,
  "startDate": "2018-09-28T19:10:57+0000",
  "endDate": "2018-09-28T19:11:01+0000",
  "duration": 3645,
  "dateCreated": "2018-09-28T19:11:01+0000",
  "lastUpdated": "2018-09-28T19:11:01+0000",
  "createdBy": {
    "username": "admin",
    "displayName": "Admin"
  },
  "updatedBy": {
    "username": "admin",
    "displayName": "Admin"
  }
],
{
  "id": 942,
  "processId": 250,
  "accountId": 1,
  "uniqueId": "f1905796-9387-4983-ae0d-0fee5bb81f56",
  "processType": {
    "code": "provisionConfig",
    "name": "configure instance"
  }
}
23.30. Process History
"percent": 100.0,
"statusEta": 348246,
"message": null,
"output": null,
"error": null,
"startDate": "2018-09-28T19:11:01+0000",
"endDate": "2018-09-28T19:11:33+0000",
"duration": 32219,
"dateCreated": "2018-09-28T19:11:01+0000",
"lastUpdated": "2018-09-28T19:11:33+0000",
"createdBy": {
  "username": "admin",
  "displayName": "Admin"
},
"updatedBy": {
  "username": "admin",
  "displayName": "Admin"
}
},
{
  "id": 944,
  "processId": 250,
  "accountId": 1,
  "uniqueId": "4f4088b0-7043-4a35-82c1-00456643beaa",
  "processType": {
    "code": "provisionResize",
    "name": "resize instance"
  },
  "description": null,
  "refType": "container",
  "refId": 240,
  "subType": null,
  "subId": null,
  "zoneId": 34,
  "integrationId": null,
  "instanceId": 238,
  "containerId": 240,
  "serverId": 601,
  "containerName": "apachetest",
  "displayName": "apachetest",
  "status": "complete",
  "reason": null,
  "percent": 100.0,
  "statusEta": 348246,
  "message": null,
  "output": null,
  "error": null,
  "startDate": "2018-09-28T19:11:33+0000",
  "endDate": "2018-09-28T19:11:36+0000",
  "duration": 2896,
  "dateCreated": "2018-09-28T19:11:33+0000",
  "lastUpdated": "2018-09-28T19:11:36+0000",
  "createdBy": {
    "username": "admin",
    "displayName": "Admin"
  },
  "updatedBy": {
  "username": "admin",
  "displayName": "Admin"
  }
}
"username": "admin",
"displayName": "Admin"
},
",
{ "id": 946,
"processId": 250,
"accountId": 1,
"uniqueId": "0081e523-bfea-4664-b582-d68076943a46",
"processType": { 
   "code": "provisionLaunch",
   "name": "power on"
},
"description": null,
"refType": "container",
"refId": 240,
"subType": null,
"subId": null,
"zoneId": 34,
"integrationId": null,
"instanceId": 238,
"containerId": 240,
"serverId": 601,
"containerName": "apachetest",
"displayName": "apachetest",
"status": "complete",
"reason": null,
"percent": 100.0,
"statusEta": 348246,
"message": null,
"output": null,
"error": null,
"startDate": "2018-09-28T19:42:00+0000",
"endDate": "2018-09-28T19:42:00+0000",
"duration": 6152,
"createdAt": "2018-09-28T19:42:00+0000",
"lastUpdated": "2018-09-28T19:42:00+0000",
"createdBy": { 
   "username": "admin",
   "displayName": "Admin"
},
"updatedBy": { 
   "username": "admin",
   "displayName": "Admin"
} },
"refId": 240,
"subType": null,
"subId": null,
"zoneId": 34,
"integrationId": null,
"instanceId": 238,
"containerId": 240,
"serverId": 601,
"containerName": "apachetest",
"displayName": "apachetest",
"status": "complete",
"reason": null,
"percent": 100.0,
"statusEta": 348246,
"message": null,
"output": null,
"error": null,
"startDate": "2018-09-28T19:11:42+0000",
"endDate": "2018-09-28T19:11:45+0000",
"duration": 2549,
"dateCreated": "2018-09-28T19:11:42+0000",
"lastUpdated": "2018-09-28T19:11:45+0000",
"createdBy": {
  "username": "admin",
  "displayName": "Admin"
},
"updatedBy": {
  "username": "admin",
  "displayName": "Admin"
},

"id": 947,
"processId": 250,
"accountId": 1,
"uniqueId": "de66729e-9580-43b0-950c-f2769cd86790",
"processType": {
  "code": "provisionNetwork",
  "name": "network wait"
},
"description": null,
"refType": "container",
"refId": 240,
"subType": null,
"subId": null,
"zoneId": 34,
"integrationId": null,
"instanceId": 238,
"containerId": 240,
"serverId": 601,
"containerName": "apachetest",
"displayName": "apachetest",
"status": "failed",
"reason": null,
"percent": 100.0,
"statusEta": 348246,
"message": null,
This endpoint retrieves a specific process.

**HTTP Request**

GET https://api.gomorpheus.com/api/processes/:id

**URL Parameters**

### 23.30.3 Get a Specific Process Event

```
curl "https://api.gomorpheus.com/api/processes/events/940" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
  "processEvent": {
    "id": 940,
    "processId": 250,
    "accountId": 1,
    "uniqueId": "54bf6265-1e86-45b4-b1a7-d4b198b13c45",
    "processType": {
      "code": "provisionResources",
      "name": "prepare resources"
    },
    "description": null,
    "refType": "container",
    "refId": 240,
    "subType": null,
    "subId": null,
    "zoneId": 34,
    "integrationId": null,
    "instanceId": 238,
  }
}
```

(continues on next page)
This endpoint retrieves a specific process event.

**HTTP Request**

GET https://api.gomorpheus.com/api/processes/events/:id

**URL Parameters**

23.31 Provision Types

Morpheus supports a diverse set of cloud APIs for provisioning compute and services. In order to facilitate some of these capabilities and preserve some of the diverse sets of feature sets across these platforms it is necessary to provide a means to dynamically specifying provisioning options depending on what is being provisioned. Morpheus calls these provision-types. Each InstanceTypeLayout that can be provisioned has a correlating ProvisionType and each CloudType (aka ZoneType) has a list of supported provision types it is capable of provisioning. This record contains optionTypes (see section on optionTypes for specifics on how to parse this data) as well as information for building out network parameters and storage parameters by listing different storage type information.

23.31.1 Get All Provision Types

```
curl "https://api.gomorpheus.com/api/provision-types"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this
{  
"provisionTypes": [
  {  
"id": 9,
"name": "Amazon",
"description": null,
"code": "amazon",
"aclEnabled": false,
"multiTenant": false,
"managed": true,
"hostNetwork": true,
"customSupported": false,
"mapPorts": false,
"exportServer": null,
"viewSet": "amazonCustom",
"serverType": "ami",
"hostType": "vm",
"addVolumes": true,
"hasDatastore": false,
"hasNetworks": null,
"maxNetworks": null,
"customizeVolume": true,
"rootDiskCustomizable": true,
"lvmSupported": true,
"hostDiskMode": "lvm",
"minDisk": 0,
"maxDisk": null,
"resizeCopiesVolumes": true,
"optionTypes": [
  {
"name": "subnet",
"description": null,
"fieldName": "subnetId",
"fieldLabel": "Subnet",
"fieldContext": "config",
"fieldAddOn": null,
"placeholder": null,
"helpBlock": "",
"defaultValue": null,
"optionSource": "amazonSubnet",
"type": "select",
"advanced": false,
"required": true,
"editable": false,
"config": [{}],
"displayOrder": 100
  },
  {
"name": "security group",
"description": null,
"fieldName": "securityId",
"fieldLabel": "Security Group",
"fieldContext": "config",
"fieldAddOn": null,
"placeholder": null,
"helpBlock": "",
"defaultValue": null,
"optionSource": "amazonSecurityGroup",
"type": "select",
"advanced": false,
"required": true,
"editable": false,
"config": [{}],
"displayOrder": 100
  }
  ]
  }
}

(continues on next page)
"optionSource": "amazonSecurityGroup",
  "type": "select",
  "advanced": false,
  "required": true,
  "editable": false,
  "config": [],
  "displayOrder": 101
},
{
  "name": "public key",
  "description": null,
  "fieldName": "publicKeyId",
  "fieldLabel": "Public Key",
  "fieldContext": "config",
  "fieldAddon": null,
  "placeholder": null,
  "helpBlock": "",
  "defaultValue": null,
  "optionSource": "keyPairs",
  "type": "select",
  "advanced": false,
  "required": false,
  "editable": false,
  "config": [],
  "displayOrder": 9
}
],
"customOptionTypes": [],
"networkTypes": [],
"storageTypes": [
  {
    "id": 7,
    "code": "amazon-sc1",
    "name": "sc1",
    "displayOrder": 4,
    "defaultType": false,
    "customLabel": true,
    "customSize": true,
    "customSizeOptions": null
  },
  {
    "id": 4,
    "code": "amazon-io1",
    "name": "io1",
    "displayOrder": 2,
    "defaultType": false,
    "customLabel": true,
    "customSize": true,
    "customSizeOptions": null
  },
  {
    "id": 5,
    "code": "amazon-gp2",
    "name": "gp2",
    "displayOrder": 1,
    "defaultType": true,
    "customLabel": true,
"customSize": true,
"customSizeOptions": null
},
{
"id": 6,
"code": "amazon-st1",
"name": "st1",
"displayOrder": 3,
"defaultType": false,
"customLabel": true,
"customSize": true,
"customSizeOptions": null
}
],
"rootStorageTypes": [
{
"id": 7,
"code": "amazon-scl",
"name": "scl",
"displayOrder": 4,
"defaultType": false,
"customLabel": true,
"customSize": true,
"customSizeOptions": null
}
],
"controllerTypes": []
}
HTTP Request

GET https://api.gomorpheus.com/api/provision-types

23.31.2 Get Specific Provision Type

curl "https://api.gomorpheus.com/api/provision-types/9"
   -H "Authorization: BEARER access_token"

The above command returns JSON structured like this

```json
{
   "success": true,
   "provisionType": {
      "id": 9,
      "name": "Amazon",
      "description": null,
      "code": "amazon",
      "aclEnabled": false,
      "multiTenant": false,
      "managed": true,
      "hostNetwork": true,
      "customSupported": false,
      "mapPorts": false,
      "exportServer": null,
      "viewSet": "amazonCustom",
      "serverType": "ami",
      "hostType": "vm",
      "addVolumes": true,
      "hasDatastore": false,
      "hasNetworks": null,
      "maxNetworks": null,
      "customizeVolume": true,
      "rootDiskCustomizable": true,
      "lvmSupported": true,
      "hostDiskMode": "lvm",
      "minDisk": 0,
      "maxDisk": null,
      "resizeCopiesVolumes": true,
      "optionTypes": [
         {
            "name": "subnet",
            "description": null,
            "fieldName": "subnetId",
            "fieldLabel": "Subnet",
            "fieldContext": "config",
            "fieldAddOn": null,
            "placeHolder": null,
            "helpBlock": "",
```
### Provision Types

```
{
    "defaultValue": null,
    "optionSource": "amazonSubnet",
    "type": "select",
    "advanced": false,
    "required": true,
    "editable": false,
    "config": [],
    "displayOrder": 100
},
{
    "name": "security group",
    "description": null,
    "fieldName": "securityId",
    "fieldLabel": "Security Group",
    "fieldContext": "config",
    "fieldAddOn": null,
    "placeholder": null,
    "helpBlock": "",
    "defaultValue": null,
    "optionSource": "amazonSecurityGroup",
    "type": "select",
    "advanced": false,
    "required": true,
    "editable": false,
    "config": [],
    "displayOrder": 101
},
{
    "name": "public key",
    "description": null,
    "fieldName": "publicKeyId",
    "fieldLabel": "Public Key",
    "fieldContext": "config",
    "fieldAddOn": null,
    "placeholder": null,
    "helpBlock": "",
    "defaultValue": null,
    "optionSource": "keyPairs",
    "type": "select",
    "advanced": false,
    "required": false,
    "editable": false,
    "config": [],
    "displayOrder": 9
}
```

23.31. Provision Types
"customSizeOptions": null
},
{
"id": 4,
"code": "amazon-io1",
"name": "io1",
"displayOrder": 2,
"defaultType": false,
"customLabel": true,
"customSize": true,
"customSizeOptions": null
},
{
"id": 5,
"code": "amazon-gp2",
"name": "gp2",
"displayOrder": 1,
"defaultType": true,
"customLabel": true,
"customSize": true,
"customSizeOptions": null
},
{
"id": 6,
"code": "amazon-st1",
"name": "st1",
"displayOrder": 3,
"defaultType": false,
"customLabel": true,
"customSize": true,
"customSizeOptions": null
}
],
"rootStorageTypes": [
{
"id": 7,
"code": "amazon-scl",
"name": "scl",
"displayOrder": 4,
"defaultType": false,
"customLabel": true,
"customSize": true,
"customSizeOptions": null
},
{
"id": 4,
"code": "amazon-io1",
"name": "io1",
"displayOrder": 2,
"defaultType": false,
"customLabel": true,
"customSize": true,
"customSizeOptions": null
},
{
"id": 5,
"code": "amazon-gp2",
"name": "gp2",
"displayOrder": 1,
"defaultType": true,
"customLabel": true,
"customSize": true,
"customSizeOptions": null
}
]
HTTP Request

GET https://api.gomorpheus.com/api/provision-types/:id

23.32 Provisioning

Provisioning options will depend heavily on the cloud you are provisioning to. This section is broken out into options based on the instance-type that is being created.

23.32.1 Provision an Instance

curl -X POST "https://api.gomorpheus.com/api/instances" \ 
-H "Authorization: BEARER access_token" \ 
-H "Content-Type: application/json" \ 
-d '{
  "zoneId": 6,
  "instance": {
    "name": "api-testing2",
    "site": {
      "id": 3
    },
    "instanceType": {
      "code": "Ubuntu"
    },
    "layout": {
      "id": 105
    },
    "plan": {
      "id": 75
    }
  }
}'
The above command returns a similar JSON structure when submitting a GET request for a single check

**HTTP Request**

POST https://api.gomorpheus.com/api/instances

**JSON Parameters**
Volumes

The (optional) `volumes` parameter is for LV configuration, can create additional LVs at provision. It should be passed as an array of Objects with the following attributes:

Network Interfaces

The `networkInterfaces` parameter is for network configuration.

The Options API `/api/options/zoneNetworkOptions?zoneId=5&provisionTypeId=10` can be used to see which options are available.

It should be passed as an array of Objects with the following attributes:

Config

The `config` parameter is for configuration options that are specific to each Provision Type. The Provision Types api can be used to see which options are available.

JSON Config Parameters for VMware

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>publicKeyId</td>
<td>N</td>
<td>null</td>
<td>ID of a public key to add to the instance</td>
</tr>
<tr>
<td>vmwareResourcePoolId</td>
<td>Y</td>
<td>null</td>
<td>ID of the resource group to use for instance</td>
</tr>
<tr>
<td>hostId</td>
<td>N</td>
<td>null</td>
<td>Specific host to deploy to if so desired</td>
</tr>
<tr>
<td>vmwareUsr</td>
<td>N</td>
<td>null</td>
<td>Additional user to provision to instance</td>
</tr>
<tr>
<td>vmwarePwd</td>
<td>N</td>
<td>null</td>
<td>Password for additional user</td>
</tr>
<tr>
<td>vmwareDomainName</td>
<td>N</td>
<td>null</td>
<td>Domain name to be given to instance</td>
</tr>
<tr>
<td>vmwareCustomSpec</td>
<td>N</td>
<td>null</td>
<td>Customization spec ID</td>
</tr>
</tbody>
</table>

Metadata

This is specific to AWS Metadata tags. Name-Values pairs can be anything you like and are added to the instance JSON as an array of n-v pairs per the example:

```
-d '{
    "zoneId": 6,
    "instance": {
        ...  
    }
    "metadata": [
        {
            "id": null,
            "name": "SampleName",
            "value": "SampleValue"
        }
    ]
    ...
}
```
Documentation on ALL of the provision types to come...

There can be additional properties to apply to the instance. For example mysql provisioning requires a set of initial credentials. You can get a list of what these input options are by fetching the instance-types list via the instance-types api and getting available layouts as well as the provision type option types associated with the layout. Currently these input options are available from the option-types map. These however, can be overridden in the event a config options map exists on the layout object within. **NOTE**: See the API Document on OptionTypes for figuring out how to build property maps from them.

### 23.33 Resource Folders

Resource Folders can be managed for each Compute Zone (Cloud) in your infrastructure.

#### 23.33.1 Get All Resource Folders for Cloud

```bash
curl "https://api.gomorpheus.com/api/zones/5/folders"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
  "folders": [
    {
      "id": 50,
      "name": "My Folder",
      "zone": {
        "id": 5,
        "name": "test-vmware"
      },
      "parent": null,
      "type": "default",
      "externalId": "group-v2342",
      "visibility": "private",
      "readOnly": false,
      "defaultFolder": false,
      "defaultStore": false,
      "active": true,
      "tenants": [
        {
          "id": 1,
          "name": "root",
          "defaultStore": false,
          "defaultTarget": false
        }
      ],
      "resourcePermission": {
        "all": true,
        "sites": [
          {
            "allPlans": true,
            "plans": [
            ]
          }
        ]
      }
    }
  ]
}
```

(continues on next page)
This endpoint retrieves all resource folders under a cloud.

**HTTP Request**

GET https://api.gomorpheus.com/api/zones/:zoneId/folders

**URL Parameters**

**Query Parameters**

23.33.2 Get a Specific Resource Folder

```bash
curl "https://api.gomorpheus.com/api/zones/5/folders/50" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
    "folder": {
        "id": 50,
        "name": "My Folder",
        "zone": {
            "id": 5,
            "name": "test-vmware"
        },
        "parent": null,
        "type": "default",
        "externalId": "group-v2342",
        "visibility": "private",
        "readOnly": false,
        "defaultFolder": false,
        "defaultStore": false,
        "active": true,
        "tenants": [,
            {
                "id": 1,
                "name": "root",
                "defaultStore": false,
                "defaultTarget": false
            }
        ],
    }
}
```

(continues on next page)
This endpoint retrieves a specific resource folder.

**HTTP Request**

GET https://api.gomorpheus.com/api/zones/:zoneId/folders/:id

**URL Parameters**

23.33.3 Updating a Resource Folder

```bash
curl -XPUT "https://api.gomorpheus.com/api/zones/5/folders/50" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"folder":{
   "active": true,
   "visibility": "private",
   "tenantPermissions": {
      "accounts": [1]
   },
   "resourcePermissions": {
      "all": false,
      "sites": [
         {"id": 1}, {"id": 2}, {"id": 3}
      ]
   }
}}'
```

The above command returns JSON structured like getting a single resource folder:

This endpoint allows updating settings for a resource folder.

**HTTP Request**

PUT https://api.gomorpheus.com/api/zones/:zoneId/folders/:id

**URL Parameters**
JSON Parameters

23.34 Resource Pools

Resource Pools can be managed for each Compute Zone (Cloud) in your infrastructure.

23.34.1 Get All Resource Pools for Cloud

```
curl "https://api.gomorpheus.com/api/zones/5/resource-pools"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
  "resourcePools": [
    {
      "id": 89,
      "name": "labs",
      "description": "labs - vpc-231b2e3c - 10.50.0.0/16",
      "zone": {
        "id": 8,
        "name": "labs-aws"
      },
      "parent": null,
      "type": "vpc",
      "externalId": "vpc-231b2e3c",
      "iacId": null,
      "visibility": "private",
      "readOnly": false,
      "defaultPool": false,
      "active": true,
      "status": "available",
      "config": {
        "cidrBlock": "10.50.0.0/16",
        "tenancy": "default"
      },
      "tenants": [
        {
          "id": 1,
          "name": "root"
        }
      ],
      "resourcePermission": {
        "all": true,
        "sites": [
        ],
        "allPlans": true,
        "plans": [
        ],
        "depth": 0
      }
    }
  ]
}
```

(continues on next page)
This endpoint retrieves all resource pools under a cloud.

**HTTP Request**

GET https://api.gomorpheus.com/api/zones/:zoneId/resource-pools

**URL Parameters**

**Query Parameters**

23.34.2 Get a Specific Resource Pool

```
curl "https://api.gomorpheus.com/api/zones/5/resource-pools/50" \ 
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
    "resourcePool": {
        "id": 89,
        "name": "labs",
        "description": "labs - vpc-231b2e3c - 10.50.0.0/16",
        "zone": {
            "id": 8,
            "name": "labs-aws"
        },
        "parent": null,
        "type": "vpc",
        "externalId": "vpc-231b2e3c",
        "iacId": null,
        "visibility": "private",
        "readOnly": false,
        "defaultPool": false,
        "active": true,
        "status": "available",
        "config": {
            "cidrBlock": "10.50.0.0/16",
            "tenancy": "default"
        },
        "tenants": [
            {
                "id": 1,
                "name": "root"
            }
        ]
    }
}
```
This endpoint retrieves a specific resource pool.

**HTTP Request**

GET https://api.gomorpheus.com/api/zones/:zoneId/resource-pools/:id

**URL Parameters**

**23.34.3 Create a Resource Pool**

```bash
curl -X POST "https://api.gomorpheus.com/api/zones/5/resource-pools" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"resourcePool": {   "name": "jdtestrp",   "config": {     "cidrBlock": "10.20.254.0/28",     "tenancy": "default"   },   "tenantPermissions": {     "accounts": [1,2,3,4,5]   },   "resourcePermissions": {     "all": false,     "sites": [       {"id": 1}, {"id": 2}, {"id": 3}     ]   } }}'
```

The above command returns JSON structured like getting a single resource pool:

This endpoint allows updating settings for a resource pool. Only certain types of clouds support creating and deleting resource pools. Configuration options vary by type.

**HTTP Request**

POST https://api.gomorpheus.com/api/zones/:zoneId/resource-pools
URL Parameters

JSON Parameters

JSON Parameters for Amazon Resource Pool

JSON Parameters for Cloud Foundry

23.34.4 Updating a Resource Pool

```
curl -XPUT "https://api.gomorpheus.com/api/zones/5/resource-pools/50" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"resourcePool":{ 
  "active": true, 
  "visibility": "private", 
  "tenantPermissions": { 
    "accounts": [1,2,3,4,5] 
  }, 
  "resourcePermissions": { 
    "all": false, 
    "sites": [ 
      {"id": 1}, {"id": 2}, {"id": 3} 
    ] 
  } 
}}'
```

The above command returns JSON structured like getting a single resource pool:

```
{
  "success": true
}
```

This endpoint allows updating settings for a resource pool.

HTTP Request

PUT https://api.gomorpheus.com/api/zones/:zoneId/resource-pools/:id

23.34.5 Delete a Resource Pool

```
curl -XDELETE "https://api.gomorpheus.com/api/zones/5/resource-pools/50" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structure like this:

```
{
  "success": true
}
```

HTTP Request

DELETE https://api.gomorpheus.com/api/zones/:zoneId/resource-pools/:id
URL Parameters

23.35 Roles

Provides API interfaces for managing the creation and modification of roles within Morpheus. This API is scoped to the roles owned by the current user’s account. System Admin users will also be able to access the system roles: System Admin and Account Admin.

23.35.1 Get All Roles

```
curl "https://api.gomorpheus.com/api/roles"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
  "roles": [
    {
      "id": 2,
      "authority": "Account Admin",
      "description": "Service account holder",
      "dateCreated": "2016-08-27T23:26:19+0000",
      "lastUpdated": "2016-08-27T23:26:19+0000",
      "scope": "Account",
      "instanceLimits": null,
      "ownerId": null,
      "owner": null
    },
    {
      "id": 1,
      "authority": "System Admin",
      "description": "Super User",
      "dateCreated": "2015-11-10T18:58:55+0000",
      "lastUpdated": "2015-11-10T18:58:55+0000",
      "scope": "Admin",
      "instanceLimits": null,
      "ownerId": null,
      "owner": null
    },
    {
      "id": 3,
      "authority": "Another Role",
      "description": "A custom role",
      "dateCreated": "2015-11-10T19:01:45+0000",
      "lastUpdated": "2015-11-10T19:02:01+0000",
      "scope": "Account",
      "instanceLimits": null,
      "ownerId": 1,
      "owner": {
        "id": 1,
        "name": "Root Account"
      }
    }
  ],
  "meta": {
    (continues on next page)
  }
}```
This endpoint retrieves all roles.

**HTTP Request**

GET https://api.gomorpheus.com/api/roles

**Query Parameters**

23.35.2 Get a Specific Role

```
curl "https://api.gomorpheus.com/api/roles/3" \ 
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
   "role": {
      "id": 3,
      "authority": "Another Role",
      "description": "A custom role",
      "dateCreated": "2015-11-10T19:01:45+0000",
      "lastUpdated": "2015-11-10T19:02:01+0000",
      "scope": "Account",
      "instanceLimits": null,
      "ownerId": 1,
      "owner": {
         "id": 1,
         "name": "Root Account"
      }
   },
   "featurePermissions": [
      {
         "id": 8,
         "code": "admin-users",
         "name": "Admin: Users",
         "access": "full"
      },
      {
         "id": 18,
         "code": "backups",
         "name": "Backups",
         "access": "full"
      },
      {
         "id": 19,
         "code": "dashboard",
```

(continues on next page)
The sample JSON above shows only a small subset of the featurePermissions and instanceTypePermissions that exist.
This endpoint will retrieve a specific role by id if the user has permission to access the role.

**HTTP Request**

GET https://api.gomorpheus.com/api/roles/:id

### 23.35.3 Create a Role

```bash
curl -XPOST "https://api.gomorpheus.com/api/roles" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{
  "role": {
    "authority": "Test Role",
    "description": "A test role",
    "baseRoleId": 2,
    "instanceLimits": {
      "maxCpu": 0,
      "maxMemory": 0,
      "maxStorage": 0
    }
  }
}
```

The above command returns JSON structured like getting a single role:

**HTTP Request**

POST https://api.gomorpheus.com/api/roles

### JSON Role Parameters

### 23.35.4 Updating Basic Role Settings

```bash
curl -XPUT "https://api.gomorpheus.com/api/roles/4" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{
  "role": {
    "authority": "Test Role",
    "description": "A new description of test role",
    "instanceLimits": {
      "maxCpu": 0,
      "maxMemory": 0,
      "maxStorage": 0
    }
  }
}
```

The above command returns JSON structured like getting a single role:

**HTTP Request**

PUT https://api.gomorpheus.com/api/roles/:id
## JSON Role Parameters

### 23.35.5 Updating Role Feature Permissions

The above command returns JSON Structured like this:

```json
{
  "success": true,
  "access": "read"
}
```

### HTTP Request

PUT https://api.gomorpheus.com/api/roles/:id/update-permission

### JSON Parameters

### 23.35.6 Global Group Access

Global Group Access must first be changed to custom as seen above.

The above command returns JSON Structured like this:

```json
{
  "success": true,
  "access": "custom"
}
```

### HTTP Request

PUT https://api.gomorpheus.com/api/roles/:id/update-permission

### JSON Parameters

### 23.35.7 Customizing Group Access

Global Group Access must first be changed to custom as seen above.
The above command returns JSON Structured like this:

```
{
  "success": true,
  "access": "full"
}
```

HTTP Request

PUT https://api.gomorpheus.com/api/roles/:id/update-group

JSON Parameters

23.35.8 Global Cloud Access

Global Cloud Access is controlled via the update-permission API

```
curl -XPUT "https://api.gomorpheus.com/api/roles/4/update-permission" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{
  "permissionCode": "ComputeZone",
  "access": "custom"
}'
```

HTTP Request

PUT https://api.gomorpheus.com/api/roles/:id/update-permission

JSON Parameters

23.35.9 Customizing Cloud Access

Global Cloud Access must first be changed to custom as seen above.

```
curl -XPUT "https://api.gomorpheus.com/api/roles/4/update-cloud" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{
  "cloudId": 2,
  "access": "full"
}'
```

The above command returns JSON Structured like this:

```
{
  "success": true,
  "access": "full"
}
```
Global Instance Type Access is controlled via the `update-permission` API.

```bash
curl -XPUT "https://api.gomorpheus.com/api/roles/4/update-permission" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{}
  "permissionCode": "InstanceType",
  "access": "custom"
}'
```

Global Instance Type Access must first be changed to `custom` as seen above.

```bash
curl -XPUT "https://api.gomorpheus.com/api/roles/4/update-instance-type" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{}
  "instanceTypeId": 1,
  "access": "full"
}'
```

The above command returns JSON Structured like this:

```json
{  "success": true,  "access": "full" }
```

Customizing Instance Type Access must first be changed to `custom` as seen above.
JSON Parameters

23.35.12 Global Blueprint Access

Global Blueprint Access is controlled via the update-permission API

```
curl -XPUT "https://api.gomorpheus.com/api/roles/4/update-permission" \  
-H "Authorization: BEARER access_token" \  
-H "Content-Type: application/json" \  
-d '{  
  "permissionCode": "AppTemplate",  
  "access": "custom"  
}';
```

HTTP Request

PUT https://api.gomorpheus.com/api/roles/:id/update-permission

JSON Parameters

23.35.13 Customizing Blueprint Access

Global Blueprint Access must first be changed to custom as seen above.

```
curl -XPUT "https://api.gomorpheus.com/api/roles/4/update-blueprint" \ 
-H "Authorization: BEARER access_token" \ 
-H "Content-Type: application/json" \ 
-d '{  
  "appTemplateId": 2,  
  "access": "full"  
}';
```

The above command returns JSON Structured like this:

```
{  
  "success": true,  
  "access": "full"  
}
```

HTTP Request

PUT https://api.gomorpheus.com/api/roles/:id/update-blueprint

JSON Parameters

23.35.14 Delete a Role

```
curl -XDELETE "https://api.gomorpheus.com/api/roles/4" \ 
-H "Authorization: BEARER access_token"
```

The above command returns JSON Structured like this:
HTTP Request
DELETE https://api.gomorpheus.com/api/roles/:id
If a role still has accounts or users tied to it, The delete will fail.

23.36 Security Group Rules

A Security Group Rule specifies that a certain CIDR is able to access a particular port (or port range) for a particular protocol. Or, that a particular CIDR is able to access all instances of a particular type (like MySql, Redis, etc). A Security Group Rule belongs to a Security Group and a Security Group is applied to either a Cloud, App, or Instance.

23.36.1 Get All Security Group Rules for a Security Group

curl "https://api.gomorpheus.com/api/security-groups/19/rules"
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```
{
  "success": true,
  "rules": [
    {
      "id": 31,
      "name": null,
      "securityGroupId": 19,
      "source": "50.22.10.10/32",
      "portRange": null,
      "protocol": null,
      "customRule": false,
      "instanceTypeId": 3
    },
    {
      "id": 30,
      "name": "port 99",
      "securityGroupId": 19,
      "source": "50.22.10.10/32",
      "portRange": "99",
      "protocol": "tcp",
      "customRule": true,
      "instanceTypeId": null
    }
  ]
}
```

This endpoint retrieves all security group rules for a Security Group.
HTTP Request
GET https://api.gomorpheus.com/api/security-groups/:id/rules

23.36.2 Get a Specific Security Group Rule

curl "https://api.gomorpheus.com/api/security-groups/19/rules/30" \
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```
{
  "success": true,
  "rule": {
    "id": 30,
    "name": "port 99",
    "securityGroupId": 19,
    "source": "50.22.10.10/32",
    "portRange": "99",
    "protocol": "tcp",
    "customRule": true,
    "instanceTypeId": null
  }
}
```

This endpoint retrieves a specific security group rule.

HTTP Request
GET https://api.gomorpheus.com/api/security-groups/:id/rules/:id

23.36.3 Create a Security Group Rule

curl -XPOST "https://api.gomorpheus.com/api/security-groups/19/rules" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{ "rule": {
    "name": "port 55",
    "source": "50.22.10.10/32",
    "portRange": "55",
    "protocol": "tcp",
    "customRule": true,
    "instanceTypeId": null
}}'

The above command returns a similar JSON structure when submitting a GET request for a single security group rule

Will create a security group rule and update all clouds, apps, and instances which are currently using the security group in which this rule belongs.
HTTP Request

POST https://api.gomorpheus.com/api/security-groups/:id/rules

JSON Security Group Rule Parameters

23.36.4 Updating a Security Group Rule

curl -XPUT "https://api.gomorpheus.com/api/security-groups/19/rules/30" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{ "rule": {  
  "source": "50.22.10.10/32",  
  "portRange": "55",  
  "protocol": "tcp",  
  "customRule": true,  
  "instanceTypeId": null  
}}'

The above command returns a similar JSON structure when submitting a GET request for a single security group rule

Will update a security group rule and update all clouds, apps, and instances which are currently using the security group in which this rule belongs.

HTTP Request

PUT https://api.gomorpheus.com/api/security-groups/:id/rules/:id

JSON Security Group Rule Parameters

Same parameters as specified in the creation of a Security Group Rule

23.36.5 Delete a Security Group Rule

curl -XDELETE "https://api.gomorpheus.com/api/security-groups/19/rules/30" \
-H "Authorization: BEARER access_token"

The above command returns JSON structure like this:

```
{
  "success": true
}
```

Will delete a security group rule and update all clouds, apps, and instances which are currently using the security group in which this rule belongs.
23.37 Security Groups

A Security Group is a grouping of rules. Each rule is a whitelist entry for a particular IP address to either a port range or a particular Morpheus instance type. A Security Group may be applied to multiple Clouds, Apps, and Instances.

23.37.1 Get All Security Groups

```
curl "https://api.gomorpheus.com/api/security-groups"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
   "success": true,
   "securityGroups": [
   {
      "id": 19,
      "accountId": 1,
      "name": "All Access to Tomcat",
      "description": ""
   },
   {
      "id": 18,
      "accountId": 1,
      "name": "Colorado office",
      "description": "All the Colorado office to access anywhere"
   }
   ],
   "securityGroupCount": 2
}
```

This endpoint retrieves all security groups and their JSON encoded configuration attributes.

HTTP Request

GET https://api.gomorpheus.com/api/security-groups

Query Parameters

23.37.2 Get a Specific Security Group

```
curl "https://api.gomorpheus.com/api/security-groups/18" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
   "success": true,
   "securityGroup": {
      "id": 18,
      "accountId": 1,
      "name": "Colorado office",
      "description": "All the Colorado office to access anywhere"
   }
}
```

(continues on next page)
This endpoint retrieves a specific security group.

**HTTP Request**

GET https://api.gomorpheus.com/api/security-groups/:id

### 23.37.3 Create a Security Group

```
curl -XPOST "https://api.gomorpheus.com/api/security-groups" \   
-H "Authorization: BEARER access_token" \   
-H "Content-Type: application/json" \   
-d '{ "securityGroup": { "name": "My New Security Group",   
"description": "My Description" }}'
```

The above command returns a similar JSON structure when submitting a GET request for a single security group.

**HTTP Request**

POST https://api.gomorpheus.com/api/security-groups

### JSON Security Group Parameters

### 23.37.4 Updating a Security Group

```
curl -XPUT "https://api.gomorpheus.com/api/security-groups/18" \   
-H "Authorization: BEARER access_token" \   
-H "Content-Type: application/json" \   
-d '{ "securityGroup": { "name": "My New Security Group",   
"description": "My Description" }}'
```

The above command returns a similar JSON structure when submitting a GET request for a single security group.

**HTTP Request**

PUT https://api.gomorpheus.com/api/security-groups/:id
JSON Security Group Parameters

23.37.5 Delete a Security Group

curl -XDELETE "https://api.gomorpheus.com/api/security-groups/1" \ 
-H "Authorization: BEARER access_token"

The above command returns JSON structure like this:

```json
{
  "success": true
}
```

Will delete a security group and update all clouds, apps, and instances which are currently using the security group.

HTTP Request

DELETE https://api.gomorpheus.com/api/security-groups/:id

23.38 Setup

The Morpheus API can be used to initialize a fresh installation of the morpheus appliance.

23.38.1 Check Appliance

curl "https://api.gomorpheus.com/api/setup/check"

The above command returns JSON structured like this:

```json
{
  "success":true,
  "buildVersion":"3.5.1",
  "setupNeeded":false
}
```

This endpoint can be used to check if the appliance needs to be setup or not, and what version it is running.

HTTP Request

GET https://api.gomorpheus.com/api/setup/check

23.38.2 Initialize Appliance

curl -XPOST "https://api.gomorpheus.com/api/setup/init" \ 
-H "Authorization: BEARER access_token" \ 
-H "Content-Type: application/json" \ 
-d '{
  "applianceName": "myenterprise-morpheus",
  "applianceUrl": "https://morpheus.myenterprise.com",
  "additionalProperties": []
}'

(continues on next page)
"accountName": "root",
"username": "admin",
"password": "69f49632b13e",
"email": "admin@myenterprise.com",
"firstName": "Admin"
}
```

The above command returns JSON Structured like this:

```
{
  "success": true
}
```

Initialize the appliance, creating the master account and user.

**HTTP Request**

`POST https://api.gomorpheus.com/api/setup/init`

**JSON Parameters**

### 23.39 SSL Certificates

Morpheus provides a database for keeping track of SSL Certificates in the system. These can be applied to various load balancers within the system and instances that use them.

#### 23.39.1 Get All SSL Certificates

```
curl "https://api.gomorpheus.com/api/certificates"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
  "certificateCount": 1,
  "certificates": [
    { 
      "accountId": 1,
      "certFile": "certFileContent",
      "domainName": "test.local",
      "generated": false,
      "id": 1,
      "keyFile": "keyFileContent",
      "name": "Test Cert",
      "wildcard": true
    }
  ]
}
```

This endpoint retrieves all key pairs associated with the account.
HTTP Request

GET https://api.gomorpheus.com/api/certificates

Query Parameters

23.39.2 Get a Specific Certificate

curl "https://api.gomorpheus.com/api/certificates/1" \ 
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```
{
    "certificate": {
        "accountId": 1,
        "certFile": "certFileContent",
        "domainName": "test.local",
        "generated": false,
        "id": 1,
        "keyFile": "keyFileContent",
        "name": "Test Cert",
        "wildcard": true
    },
    "success": true
}
```

This endpoint retrieves a specific key.

HTTP Request

GET https://api.gomorpheus.com/api/certificates/:id

URL Parameters

23.39.3 Create a Certificate

curl -XPOST "https://api.gomorpheus.com/api/certificates" \ 
-H "Authorization: BEARER access_token" \ 
-H "Content-Type: application/json" \ 
-d '{"certificate":{
    "name": "My Cert",
    "certFile": "my cert file contents",
    "keyFile": "My keyfile",
    "domainName": "Domain name of cert",
    "wildcard": false
}}'

The above command returns JSON structured like getting a single certificate:
23.39.4 Updating a Certificate

```
curl -XPUT "https://api.gomorpheus.com/api/certificates/1" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"certificate":{
  "name": "My Cert",
  "certFile": "my cert file contents",
  "keyFile": "My keyfile",
  "domainName": "Domain name of cert",
  "wildcard": false
}}'
```

The above command returns JSON structured like getting a single certificate:

**HTTP Request**

PUT https://api.gomorpheus.com/api/certificates/:id

**JSON Check Parameters**

23.39.5 Delete a Certificate

```
curl -XDELETE "https://api.gomorpheus.com/api/certificates/1" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON Structured like this:

```
{
  "success": true
}
```

Will delete a certificate from the system and make it no longer usable.

**HTTP Request**

DELETE https://api.gomorpheus.com/api/certificates/:id

23.40 Storage Buckets

Provides API interfaces for managing Storage Buckets (Object Stores and File Shares).

23.40.1 Get All Storage Buckets

```
curl "https://api.gomorpheus.com/api/storage/buckets" 
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:
This endpoint retrieves all storage buckets associated with the account.

HTTP Request

GET https://api.gomorpheus.com/api/storage/buckets

Query Parameters

23.40.2 Get a Specific Storage Bucket

curl "https://api.gomorpheus.com/api/storage/buckets/1" \
   -H "Authorization: BEARER access_token"
The above command returns JSON structured like this:

```
{
    "storageBucket": {
        "id": 1,
        "name": "s3 test",
        "accountId": 1,
        "providerType": "s3",
        "config": {
            "accessKey": "G429AED2C4L5Y2B7Q",
            "secretKey": "***********",
            "endpoint": ""
        },
        "bucketName": "morpheus-s3-test",
        "readOnly": false,
        "defaultBackupTarget": false,
        "defaultDeploymentTarget": false,
        "defaultVirtualImageTarget": false,
        "copyToStore": true,
        "retentionPolicyType": null,
        "retentionPolicyDays": null,
        "retentionProvider": null
    }
}
```

This endpoint retrieves a specific storage bucket.

**HTTP Request**

GET https://api.gomorpheus.com/api/storage/buckets/:id

**URL Parameters**

23.40.3 Create a Storage Bucket

```
curl -XPOST "https://api.gomorpheus.com/api/storage/buckets" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{
    "storageBucket": {
        "name": "test-storage",
        "providerType": "local",
        "config": {
            "basePath": "/tmp/test-storage"
        },
        "defaultBackupTarget": false,
        "copyToStore": true,
        "defaultDeploymentTarget": false,
        "defaultVirtualImageTarget": false,
        "retentionPolicyType": null,
        "retentionPolicyDays": null,
        "retentionProvider": null
    }
}
'
```

The above command returns JSON structured like getting a single storage bucket:
HTTP Request

POST https://api.gomorpheus.com/api/storage/buckets

JSON Parameters

Amazon S3 (s3)

Azure (azure) Parameters

CIFS (cifs) Parameters

Local Storage (local) Parameters

NFSv3 (nfs) Parameters

Openstack Swift (openstack) Parameters

Rackspace CDN (rackspace) Parameters

23.40.4 Update a Storage Bucket

curl -XPUT "https://api.gomorpheus.com/api/storage/buckets/1" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{
  "storageBucket": {
    "name": "my-storage",
    "copyToStore": true
  }
}"

The above command returns JSON structured like getting a single storage bucket:

HTTP Request

PUT https://api.gomorpheus.com/api/storage/buckets/1

JSON Parameters

See Create.

23.40.5 Delete a Storage Bucket

curl -XDELETE "https://api.gomorpheus.com/api/storage/buckets/1" \
-H "Authorization: BEARER access_token"

The above command returns JSON Structured like this:
Will delete a storage bucket from the system and make it no longer usable.

**HTTP Request**

DELETE https://api.gomorpheus.com/api/storage/buckets/:id

**URL Parameters**

**23.41 Tasks**

Provides API interfaces for managing the creation and modification of automation tasks. Tasks are used in workflows for automation.

**23.41.1 Get All Tasks**

```
curl "https://api.gomorpheus.com/api/tasks"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
  "tasks": [
    {
      "id": 5,
      "accountId": 1,
      "name": "aptitude upgrade",
      "taskType": {
        "id": 1,
        "code": "script",
        "name": "Shell Script"
      },
      "taskOptions": {
        "script": "apt-get upgrade -y"
      }
    }
  ],
  "meta": {
    "offset": 0,
    "max": 25,
    "size": 1,
    "total": 1
  }
}
```

This endpoint retrieves all tasks.
HTTP Request

GET https://api.gomorpheus.com/api/tasks

Query Parameters

23.41.2 Get a Specific Task

curl "https://api.gomorpheus.com/api/tasks/1" \ 
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```
{
    "task": {
        "id": 5,
        "accountId": 1,
        "name": "aptitude upgrade",
        "taskType": {
            "id": 1,
            "code": "script",
            "name": "Shell Script"
        },
        "taskOptions": {
            "script": "apt-get upgrade -y"
        }
    }
}
```

This endpoint will retrieve a specific task by id

HTTP Request

GET https://api.gomorpheus.com/api/tasks/:id

URL Parameters

23.41.3 Create a Task

curl -XPOST "https://api.gomorpheus.com/api/tasks" \ 
-H "Authorization: BEARER access_token" \ 
-H "Content-Type: application/json" \ 
-d '{"task": {
    "name": "cleanup tmp files",
    "taskType": {
        "code": "script"
    },
    "taskOptions": {
        "script": "rm -rf /var/www/app1/tmp/*\nrm -rf /var/www/app2/tmp/*"
    }
}}'

The above command returns JSON structured like getting a single task:
**HTTP Request**

POST https://api.gomorpheus.com/api/tasks

**JSON Parameters**

**23.41.4 Updating a Task**

```bash
curl -XPUT "https://api.gomorpheus.com/api/tasks/5" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"task":{
    "name": "my task",
}}'
```

The above command returns JSON structured like getting a single task:

**HTTP Request**

PUT https://api.gomorpheus.com/api/tasks/:id

**URL Parameters**

**JSON Parameters**

**23.41.5 Delete a Task**

```bash
curl -XDELETE "https://api.gomorpheus.com/api/tasks/1" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON Structured like this:

```json
{
    "success": true
}
```

**HTTP Request**

DELETE https://api.gomorpheus.com/api/tasks/:id

**URL Parameters**

If a task is still tied to workflows, the delete will fail.

**23.42 Task Types**

A Task Type is a type of automation task. Each type defines its own set of options to be configured for each task.
23.42.1 Get All Task Types

```bash
curl "https://api.gomorpheus.com/api/task-types"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
  "taskTypes": [
    {
      "id": 1,
      "code": "script",
      "name": "Shell Script",
      "category": "script",
      "description": null,
      "optionTypes": [
        {
          "id": 254,
          "name": "Script",
          "code": "script",
          "description": null,
          "fieldName": "script",
          "fieldLabel": "Script",
          "fieldContext": "taskOptions",
          "fieldGroup": null,
          "fieldClass": null,
          "fieldAddOn": null,
          "placeHolder": null,
          "helpBlock": null,
          "defaultValue": null,
          "optionSource": null,
          "type": "code-editor",
          "advanced": false,
          "required": false,
          "editable": false,
          "config": {},
          "displayOrder": 5,
          "wrapperClass": null,
          "enabled": true,
          "noBlank": null,
          "dependsOnCode": null,
          "contextualDefault": null
        }
      ]
    },
    {
      "id": 2,
      "code": "sshTask",
      "name": "SSH Script",
      "category": "script",
      "description": null,
      "optionTypes": [
        {
          "id": 258,
          "name": "Key",
          "code": "sshKey",
          "description": null,
          "fieldName": "key",
          "fieldLabel": "Key",
          "fieldContext": "taskOptions",
          "fieldGroup": null,
          "fieldClass": null,
          "fieldAddOn": null,
          "placeHolder": null,
          "helpBlock": null,
          "defaultValue": null,
          "optionSource": null,
          "type": "code-editor",
          "advanced": false,
          "required": false,
          "editable": false,
          "config": {},
          "displayOrder": 5,
          "wrapperClass": null,
          "enabled": true,
          "noBlank": null,
          "dependsOnCode": null,
          "contextualDefault": null
        }
      ]
    }
  ]
}
```
"description": null,
"fieldName": "sshKey",
"fieldLabel": "Key",
"fieldContext": "taskOptions",
"fieldGroup": null,
"fieldClass": null,
"fieldAddOn": null,
"placeHolder": null,
"helpBlock": null,
"defaultValue": null,
"optionSource": "keyPairs",
"type": "select",
"advanced": false,
"required": false,
"editable": false,
"config": {
},
"displayOrder": 2,
"wrapperClass": null,
"enabled": true,
"noBlank": null,
"dependsOnCode": null,
"contextualDefault": null
},
{
 "id": 254,
 "name": "Script",
 "code": "script",
 "description": null,
 "fieldName": "script",
 "fieldLabel": "Script",
 "fieldContext": "taskOptions",
 "fieldGroup": null,
 "fieldClass": null,
 "fieldAddOn": null,
 "placeHolder": null,
 "helpBlock": null,
 "defaultValue": null,
 "optionSource": null,
 "type": "code-editor",
 "advanced": false,
 "required": false,
 "editable": false,
 "config": {
 },
 "displayOrder": 5,
 "wrapperClass": null,
 "enabled": true,
 "noBlank": null,
 "dependsOnCode": null,
 "contextualDefault": null
},
{
 "id": 259,
 "name": "IP Address",
 "code": "host",
 "description": null,
"fieldName": "host",
"fieldLabel": "IP Address",
"fieldContext": "taskOptions",
"fieldGroup": null,
"fieldClass": null,
"fieldAddOn": null,
"placeHolder": null,
"helpBlock": null,
"defaultValue": null,
"optionSource": null,
"type": "text",
"advanced": false,
"required": false,
"editable": false,
"config": { 
},
"displayOrder": 0,
"wrapperClass": null,
"enabled": true,
"noBlank": null,
"dependsOnCode": null,
"contextualDefault": null
},
{
"id": 257,
"name": "Password",
"code": "password",
"description": null,
"fieldName": "password",
"fieldLabel": "Password",
"fieldContext": "taskOptions",
"fieldGroup": null,
"fieldClass": null,
"fieldAddOn": null,
"placeHolder": null,
"helpBlock": null,
"defaultValue": null,
"optionSource": null,
"type": "password",
"advanced": false,
"required": false,
"editable": false,
"config": { 
},
"displayOrder": 4,
"wrapperClass": null,
"enabled": true,
"noBlank": null,
"dependsOnCode": null,
"contextualDefault": null
},
{
"id": 260,
"name": "Port",
"code": "port",
"description": null,
"fieldName": "port",
"fieldLabel": "Port",
"fieldContext": "taskOptions",
"fieldGroup": null,
"fieldClass": null,
"fieldAddOn": null,
"placeHolder": null,
"helpBlock": null,
"defaultValue": null,
"optionSource": null,
"type": "text",
"advanced": false,
"required": false,
"editable": false,
"config": { 
},
"displayOrder": 6,
"wrapperClass": null,
"enabled": true,
"noBlank": null,
"dependsOnCode": null,
"contextualDefault": null
}
"fieldLabel": "Port",
"fieldContext": "taskOptions",
"fieldGroup": null,
"fieldClass": null,
"fieldAddOn": null,
"placeholder": null,
"helpBlock": null,
"defaultValue": null,
"optionSource": null,
"type": "text",
"advanced": false,
"required": false,
"editable": false,
"config": {
  
  
  
  },
"displayOrder": 1,
"wrapperClass": null,
"enabled": true,
"noBlank": null,
"dependsOnCode": null,
"contextualDefault": null
},
{
  "id": 256,
  "name": "Username",
  "code": "username",
  "description": null,
  "fieldName": "username",
  "fieldLabel": "Username",
  "fieldContext": "taskOptions",
  "fieldGroup": null,
  "fieldClass": null,
  "fieldAddOn": null,
  "placeholder": null,
  "helpBlock": null,
  "defaultValue": null,
  "optionSource": null,
  "type": "text",
  "advanced": false,
  "required": false,
  "editable": false,
  "config": {
  
    },
  "displayOrder": 3,
  "wrapperClass": null,
  "enabled": true,
  "noBlank": null,
  "dependsOnCode": null,
  "contextualDefault": null
}]}
HTTP Request

GET https://api.gomorpheus.com/api/task-types

Query Parameters

23.42.2 Get a Specific Task Type

curl "https://api.gomorpheus.com/api/task-types/1" \
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```
{
  "success": true,
  "taskType": {
    "id": 1,
    "code": "script",
    "name": "Shell Script",
    "category": "script",
    "description": null,
    "optionTypes": [
      {
        "id": 254,
        "name": "Script",
        "code": "script",
        "description": null,
        "fieldName": "script",
        "fieldLabel": "Script",
        "fieldContext": "taskOptions",
        "fieldGroup": null,
        "fieldClass": null,
        "fieldAddOn": null,
        "placeHolder": null,
        "helpBlock": null,
        "defaultValue": null,
        "optionSource": null,
        "type": "code-editor",
        "advanced": false,
        "required": false,
        "editable": false,
        "config": { },
        "displayOrder": 5,
        "wrapperClass": null,
        "enabled": true,
        "noBlank": null,
        "dependsOnCode": null,
        "contextualDefault": null
      }
    ]
  }
}
```

This endpoint will retrieve a specific task type by id
HTTP Request

GET https://api.gomorpheus.com/api/task-types/:id

URL Parameters

23.43 User Settings

Provides API for managing your own user settings and api access tokens.

23.43.1 Get User Settings

```
curl "https://api.gomorpheus.com/api/user-settings" \ 
-H "Authorization: Bearer access_token"
```

The above command returns JSON structured like this:

```
{
    "user": {
        "id": 1,
        "username": "admin",
        "firstName": "Admin",
        "lastName": ",",
        "email": "admin@morpheustestdata.com",
        "linuxUsername": "morphadmin",
        "linuxPassword": "***********",
        "linuxKeyPairId": null,
        "windowsUsername": null,
        "windowsPassword": null,
        "avatar": null,
        "receiveNotifications": true
    },
    "accessTokens": [
        {
            "clientId": "morph-cli",
            "username": "admin",
            "expiration": "2019-11-20T02:19:18Z",
            "tokenType": "bearer"
        }
    ]
}
```

This endpoint retrieves your user settings and API access token information.

HTTP Request

GET https://api.gomorpheus.com/api/user-settings

23.43.2 Update User Settings
curl -XPUT "https://api.gomorpheus.com/api/user-settings" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{
  "user": {
    "receiveNotifications": true
  }
}';

The above command returns JSON Structured like this:

```
{
  "success": true
}
```

**HTTP Request**

PUT https://api.gomorpheus.com/api/user-settings

**JSON Parameters**

### 23.43.3 Update Avatar Image

curl -XPOST "https://api.gomorpheus.com/api/user-settings/avatar" \
-H "Authorization: BEARER access_token" \
-F 'user.avatar=@filename'

The above command returns JSON Structured like this:

```
{
  "success": true
}
```

**HTTP Request**

POST https://api.gomorpheus.com/api/user-settings/avatar

**Parameters**

Upload a new avatar image. Expects multipart form data as the request format, not JSON.

### 23.43.4 Delete Avatar Image

curl -XDELETE "https://api.gomorpheus.com/api/user-settings/avatar" \
-H "Authorization: BEARER access_token"

The above command returns JSON Structured like this:
23.43.5 Regenerate API Access Token

```bash
curl -XPUT "https://api.gomorpheus.com/api/user-settings/regenerate-access-token?clientId=morph-api" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json"
```

The above command returns JSON Structured like this:

```json
{
  "success": true
}
```

**HTTP Request**

PUT https://api.gomorpheus.com/api/user-settings/regenerate-access-token?clientId=:clientId

**Query Parameters**

The above command returns JSON structured like this:

```json
{
  "success": true,
  "token": "a936c304-374d-42c3-8634-8f825756d240"
}
```

This endpoint regenerates your API access token for the specified client. If a current token exists, it is revoked and a new token is returned.

23.43.6 Revoke API Access Token

```bash
curl -XPUT "https://api.gomorpheus.com/api/user-settings/clear-access-token?clientId=morph-api" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json"
```

The above command returns JSON Structured like this:
HTTP Request

PUT https://api.gomorpheus.com/api/user-settings/clear-access-token?clientId=:clientId

Query Parameters

This endpoint revokes your API access token for the specified client.

23.43.7 Get Available API Clients

curl "https://api.gomorpheus.com/api/user-settings/api-clients" \
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```json
{
  "clients": [
    {
      "clientId": "morph-ios"
    },
    {
      "clientId": "morph-marketing"
    },
    {
      "clientId": "morph-customer"
    },
    {
      "clientId": "morph-cli"
    },
    {
      "clientId": "morph-api"
    }
  ]
}
```

HTTP Request

GET https://api.gomorpheus.com/api/user-settings/api-clients

This endpoint retrieves a list of available API clients.

23.44 User Sources

User Sources can be configured for each Tenant. Supported types include LDAP, JumpCloud, Active Directory, and others.
23.44.1 Get All User Sources

```
curl "https://api.gomorpheus.com/api/user-sources"
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
{
  "userSources": [ 
    {
      "id": 17,
      "name": "jump cloud",
      "description": "our jump cloud users",
      "code": "KsUGxwhTU",
      "type": "jumpCloud",
      "active": true,
      "deleted": false,
      "autoSyncOnLogin": true,
      "account": { 
        "id": 59,
        "name": "acme"
      },
      "defaultAccountRole": { 
        "id": 19,
        "authority": "Basic User"
      },
      "roleMappings": [ 
      ],
      "subdomain": "acme",
      "dateCreated": "2018-03-22T01:57:12+0000",
      "lastUpdated": "2018-03-22T01:57:12+0000"
    } 
  ],
  "meta": { 
    "size": 1,
    "total": 1,
    "max": 25,
    "offset": 0
  }
}
```

This endpoint retrieves all user sources.

**HTTP Request**

GET https://api.gomorpheus.com/api/user-sources

**Query Parameters**

23.44.2 Get a Specific User Source

```
curl "https://api.gomorpheus.com/api/user-sources/2" \
-H "Authorization: BEARER access_token"
```

23.44. User Sources
The above command returns JSON structured like this:

```json
{
    "userSource": {
        "id": 17,
        "name": "jump cloud",
        "description": "our jump cloud users",
        "code": "KsUGxwhTU",
        "type": "jumpCloud",
        "active": true,
        "deleted": false,
        "autoSyncOnLogin": true,
        "account": {
            "id": 59,
            "name": "acme"
        },
        "defaultAccountRole": {
            "id": 19,
            "authority": "Basic User"
        },
        "config": {
            "organizationId": "34a927g43e21be3786b2343b",
            "bindingPassword": "************",
            "bindingUsername": "jumpadmin",
            "requiredRole": "MorpheusTag"
        },
        "roleMappings": [],
        "subdomain": "acme",
        "loginURL": "https://app.gomorpheusdata.com/login/account/acme",
        "dateCreated": "2018-03-22T01:57:12+0000",
        "lastUpdated": "2018-03-22T01:57:12+0000"
    }
}
```

This endpoint retrieves a specific user source.

**HTTP Request**

GET https://api.gomorpheus.com/api/user-sources/:id

**URL Parameters**

23.44.3 Create a User Source

```bash
curl -XPOST "https://api.gomorpheus.com/api/accounts/60/user-sources" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d "{"userSource": {
    "type": "activeDirectory",
    "name": "Ninja AD",
    "config": {
        "url": "10.30.10.155",
        "domain": "ad.morpheusdata.ninja",
```
"useSSL": "on",
"bindingUsername": "adadmin",
"bindingPassword": "goodadpassword",
"requiredGroup": "MorpheusUsers",
"searchMemberGroups": "off"
},
"defaultAccountRole": {
"id": 19
}
]'

The above command returns JSON structured like getting a single user source:

**HTTP Request**

POST https://api.gomorpheus.com/api/accounts/:accountId/user-sources

**URL Parameters**

**JSON Parameters**

**JSON Parameters for LDAP**

**JSON Parameters for jumpCloud**

**JSON Parameters for activeDirectory**

**JSON Parameters for okta**

**JSON Parameters for oneLogin**

**JSON Parameters for saml**

**JSON Parameters for customExternal**

**JSON Parameters for customApi**

### 23.44.4 Updating a User Source

```
curl -XPUT "https://api.gomorpheus.com/api/user-sources/3" \ 
-H "Authorization: BEARER access_token" \ 
-H "Content-Type: application/json" \ 
-d '{"userSource": {
  "type": "activeDirectory",
  "name": "Ninja AD",
  "config": {
    "url": "10.30.10.155",
    "domain": "ad.morpheusdata.ninja",
    "useSSL": "on",
    "bindingUsername": "adadmin",
```
"bindingPassword": "goodadpassword",
"requiredGroup": "MorpheusUsers",
"searchMemberGroups": "off"
},
"defaultAccountRole": {
"id": 19
}
]
}
]

The above command returns JSON structured like getting a single user source:

HTTP Request

PUT https://api.gomorpheus.com/api/user-sources/:id

URL Parameters

JSON Parameters

23.44.5 Updating Subdomain for a User Source

curl -XPUT "https://api.gomorpheus.com/api/user-sources/3/subdomain" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"subdomain": "ninjas"}"

The above command returns JSON Structured like this:

{
  "success": true
}

HTTP Request

PUT https://api.gomorpheus.com/api/user-sources/:id/subdomain

URL Parameters

JSON Parameters

This endpoint updates the subdomain for the account associated with the user source.

23.44.6 Delete a User Source

curl -XDELETE "https://api.gomorpheus.com/api/user-sources/3" \
-H "Authorization: BEARER access_token"

The above command returns JSON Structured like this:
HTTP Request

DELETE https://api.gomorpheus.com/api/user-sources/:id

URL Parameters

Will delete a user source from the system and make it no longer usable.

23.45 Users

Managing users via the API is always scoped to a specific account. Most of the API’s regarding user management require that the account Id of the user also be known

23.45.1 Get All Users for an Account

curl "https://api.gomorpheus.com/api/accounts/1/users"
   -H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```json
{
   "users": [
   {
    "id": 1,
    "accountId": 1,
    "username": "davydotcom",
    "displayName": "David Estes",
    "email": "destes@bcap.com",
    "firstName": "David",
    "lastName": "Estes",
    "dateCreated": "2015-11-10T18:58:55+0000",
    "lastUpdated": "2015-11-10T18:58:55+0000",
    "enabled": true,
    "accountExpired": false,
    "accountLocked": false,
    "passwordExpired": false,
    "role": {
     "id": 1,
     "authority": "System Admin",
     "description": "Super User"
    },
    "account": {
     "id": 1,
     "name": "Root Account"
    },
    "instanceLimits": null
   }
} (continues on next page)```
This endpoint retrieves all accounts.

**HTTP Request**

GET https://api.gomorpheus.com/api/accounts/:accountId/users

**Query Parameters**

23.45.2 Get a Specific User

curl "https://api.gomorpheus.com/api/accounts/1/users/1" \
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```
{
  "user": {
    "id": 1,
    "accountId": 1,
    "username": "davydotcom",
    "displayName": "David Estes",
    "email": "destes@bcap.com",
    "firstName": "David",
    "lastName": "Estes",
    "dateCreated": "2015-11-10T18:58:55+0000",
    "lastUpdated": "2015-11-10T18:58:55+0000",
    "enabled": true,
    "accountExpired": false,
    "accountLocked": false,
    "passwordExpired": false,
    "role": {
      "id": 1,
      "authority": "System Admin",
      "description": "Super User"
    },
    "account": {
      "id": 1,
      "name": "Root Account"
    },
    "instanceLimits": null
  }
}
```

This endpoint will retrieve a specific user by id if the user has permission to access the user.
HTTP Request

GET https://api.gomorpheus.com/api/accounts/:accountId/users/:id

23.45.3 Create a User

```bash
curl -X POST "https://api.gomorpheus.com/api/accounts/1/users" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"user":{
    "username": "testuser",
    "email": "testuser@yourcompany.com",
    "firstName": "Test",
    "lastName": "User",
    "password": "aStrongpassword123!",
    "role": {"id": 1}
}}'
```

The above command returns JSON structured like getting a single user:

HTTP Request

POST https://api.gomorpheus.com/api/accounts/:accountId/users

JSON User Parameters

23.45.4 Updating a User

```bash
curl -X PUT "https://api.gomorpheus.com/api/accounts/1/users/2" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"user":{
    "username": "testUser",
    "firstName": "Jane",
    "lastName": "Doe",
    "password": "abc123",
    "role": {"id": 1},
    "instanceLimits": {
        "maxCpu": 0,
        "maxMemory": 0,
        "maxStorage": 0
    }
}}'
```

The above command returns JSON structured like getting a single user:

HTTP Request

PUT https://api.gomorpheus.com/api/accounts/:accountId/users/:id
JSON User Parameters

23.45.5 Delete a User

```
curl -XDELETE "https://api.gomorpheus.com/api/accounts/1/users/2" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON Structured like this:

```
{
   "success": true
}
```

This will disassociate the user from any instances they have previously provisioned.

HTTP Request

DELETE https://api.gomorpheus.com/api/accounts/:accountId/users/:id

23.46 Virtual Images

Virtual Images can be managed via the API.

23.46.1 Get List of Virtual Images

```
curl "https://api.gomorpheus.com/api/virtual-images" \
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```
|
 | "virtualImages": [
 | ]
```

(continues on next page)
This endpoint retrieves a list of virtual images for the specified filter.

**HTTP Request**

GET https://api.gomorpheus.com/api/virtual-images

**Query Parameters**

**23.46.2 Get a Specific Virtual Image**

curl "https://api.gomorpheus.com/api/virtual-images/764" \
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

(continues on next page)
"id": 764,
"name": "testimage",
"description": null,
"ownerId": 1,
"imageType": "vmware",
"userUploaded": true,
"userDefined": false,
"systemImage": false,
"isCloudInit": true,
"sshUsername": "root",
"sshPassword": "****",
"sshKey": null,
"osType": {
   "id": 9,
   "name": "ubuntu 64-bit",
   "description": null,
   "vendor": "canonical",
   "category": "ubuntu",
   "osFamily": "debian",
   "osVersion": "all",
   "bitCount": 64,
   "platform": "linux"
},
"minDisk": null,
"minRam": null,
"rawSize": 56077536,
"trialVersion": false,
"virtioSupported": true,
"isAutoJoinDomain": false,
"vmToolsInstalled": true,
"isForceCustomization": false,
"isSysprep": false,
"userData": null,
"storageProvider": {
   "id": 2,
   "name": "testdrive2"
},
"externalId": null,
"visibility": "private",
"accounts": [
   { "id": 1,
     "name": "root"
   }
],
"cloudFiles": [
   { "name": "testimage.vmdk",
     "size": 1034592
   },
   { "name": "testimage.ovf",
     "size": 28038768
   }
]
This endpoint retrieves a specific virtual image and its files.

**HTTP Request**

```plaintext
GET https://api.gomorpheus.com/api/virtual-images/:id
```

**URL Parameters**

**23.46.3 Create a Virtual Image**

```
curl -X POST "https://api.gomorpheus.com/api/virtual-images" \
-H "Authorization: BEARER access_token" \
-H "Content-Type: application/json" \
-d '{"virtualImage":{ 
  "name": "testimage2",
  "imageType": "vmware",
  "isCloudInit": true,
  "installAgent": true,
  "sshUsername": "root",
  "sshPassword": "mygoodpassword123",
  "sshKey": null,
  "osType": {
    "id": 9
  },
  "virtioSupported": true,
  "vmToolsInstalled": true,
}}'
```

The above command returns JSON structured like getting a single virtual image.

This endpoint creates a new virtual image, without any files yet.

**HTTP Request**

```plaintext
POST https://api.gomorpheus.com/api/virtual-images
```

**JSON Virtual Image Parameters**

**23.46.4 Upload Virtual Image File**

```
curl -X POST "https://api.gomorpheus.com/api/virtual-images/765/upload?filename=disk-0.vmdk" \
-H "Authorization: BEARER access_token" \
--data-binary '@/path/to/file'
```

The above command returns JSON structure like this:

```
{
  "success": true
}
```

This will upload the file and associate it to the Virtual Image.
HTTP Request

POST https://api.gomorpheus.com/api/virtual-images/:id/upload

URL Parameters

Query Parameters

23.46.5 Remove Virtual Image File

curl -XDELETE "https://api.gomorpheus.com/api/virtual-images/765/files?filename=testimage.ovf" \
-H "Authorization: BEARER access_token"

The above command returns JSON structure like this:

{
   "success": true
}

HTTP Request

DELETE https://api.gomorpheus.com/api/virtual-images/:id/files?filename=

URL Parameters

Query Parameters

23.46.6 Delete a Virtual Image

curl -XDELETE "https://api.gomorpheus.com/api/virtual-images/765" \
-H "Authorization: BEARER access_token"

The above command returns JSON structure like this:

{
   "success": true
}

Will delete a virtual image and any associated files.

HTTP Request

DELETE https://api.gomorpheus.com/api/virtual-images/:id

23.47 Whoami

Provides API to retrieve information about yourself, including your roles and permissions.
curl "https://api.gomorpheus.com/api/whoami" \
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```json
{
  "user": {
    "id": 1,
    "accountId": 1,
    "username": "admin",
    "displayName": "Admin",
    "email": "admin@morpheustestdata.com",
    "firstName": "Admin",
    "lastName": "",
    "dateCreated": "2016-08-28T03:28:09+0000",
    "lastUpdated": "2018-11-20T05:11:50+0000",
    "enabled": true,
    "accountExpired": false,
    "accountLocked": false,
    "passwordExpired": false,
    "roles": [
      {
        "id": 1,
        "authority": "System Admin",
        "description": "Super User"
      }
    ],
    "account": {
      "id": 1,
      "name": "root"
    }
  },
  "isMasterAccount": true,
  "permissions": {
    "ComputeSite": "full",
    "ComputeZone": "full",
    "InstanceType": "full",
    "account-usage": "full",
    "admin-accounts-users": "full",
    "admin-accounts": "full",
    "admin-appliance": "full",
    "admin-backupSettings": "full",
    "admin-certificates": "full",
    "admin-cm": "full",
    "admin-containers": "full",
    "admin-environments": "full",
    "admin-global-policies": "full",
    "admin-groups": "full",
    "admin-identity-sources": "full",
    "admin-keypairs": "full",
    "admin-licenses": "full",
    "admin-logSettings": "full",
    "admin-monitorSettings": "full",
    "admin-policies": "full",
    "admin-provisioningSettings": "full",
    "admin-roles": "full",
    "admin-servers": "full"
  }
}
```
"admin-servicePlans": "full",
"admin-users": "full",
"admin-utilities": "full",
"admin-whitelabel": "full",
"admin-zones": "full",
"app-templates": "full",
"apps": "full",
"arm-template": "full",
"automation-services": "full",
"backup-services": "full",
"backups": "full",
"billing": "full",
"cloudFormation-template": "full",
"dashboard": "read",
"deployment-services": "full",
"deployments": "full",
"guidance": "full",
"infrastructure-boot": "full",
"infrastructure-loadbalancer": "full",
"infrastructure-networks": "full",
"infrastructure-securityGroups": "full",
"infrastructure-state": "full",
"infrastructure-storage-browser": "full",
"infrastructure-storage": "full",
"logs": "full",
"migrations": "full",
"monitoring": "full",
"operations-approvals": "full",
"operations-health": "read",
"provisioning-admin": "full",
"provisioning-force-delete": "full",
"provisioning": "full",
"reports-analytics": "full",
"reports": "full",
"scheduling-execute": "full",
"scheduling-power": "full",
"services-archives": "full",
"services-cypher": "full",
"services-image-builder": "full",
"services-kubernetes": "full",
"services-network-registry": "full",
"support-menu": "read",
"task-scripts": "full",
"tasks": "full",
"terminal-access": "yes",
"terminal": "full",
"terraform-template": "full",
"thresholds": "full",
"trust-services": "full",
"virtual-images": "full"
},
"appliance": {
  "buildVersion": "3.5.3"
}
23.47.1 HTTP Request

GET https://api.gomorpheus.com/api/whoami

This endpoint retrieves your user information, roles and permissions. The appliance build version is also returned.

23.48 Workflows

Provides API interfaces for managing the creation and modification of automation workflows. Workflows, also called Task Sets, are a collection of tasks that are organized in phases. A task phase determines if/when each task runs.

23.48.1 Get All Workflows

curl "https://api.gomorpheus.com/api/task-sets"
-H "Authorization: BEARER access_token"

The above command returns JSON structured like this:

```json
{
  "taskSets": [
    {
      "id": 13,
      "name": "my workflow",
      "description": null,
      "dateCreated": "2017-06-26T15:36:19+0000",
      "lastUpdated": "2017-06-26T15:44:38+0000",
      "accountId": 1,
      "tasks": [
        8
      ],
      "taskSetTasks": [
        {
          "id": 51,
          "taskPhase": "provision",
          "taskOrder": 2,
          "task": {
            "id": 8,
            "name": "my task",
            "taskType": {
              "id": 1,
              "code": "script",
              "name": "Shell Script"
            },
            "taskOptions": {
              "script": "echo \"hello\"
            }
          }
        }
      ],
      "meta": {
        "offset": 0,
        "max": 25
      }
    }
  ],
  "meta": {
    "offset": 0,
    "max": 25,
  }
}```

(continues on next page)
This endpoint retrieves all workflows.

**HTTP Request**

GET https://api.gomorpheus.com/api/task-sets

**Query Parameters**

23.48.2 Get a Specific Workflow

```bash
curl "https://api.gomorpheus.com/api/task-sets/1" \ 
-H "Authorization: BEARER access_token"
```

The above command returns JSON structured like this:

```json
{
    "taskSet": {
        "id": 8,
        "name": "uname",
        "description": "",
        "dateCreated": "2017-05-24T20:24:02+0000",
        "lastUpdated": "2017-05-24T20:24:02+0000",
        "accountId": 1,
        "tasks": [
            10
        ],
        "taskSetTasks": [
            {
                "id": 33,
                "taskPhase": "postProvision",
                "taskOrder": 0,
                "task": {
                    "id": 10,
                    "name": "uname",
                    "taskType": {
                        "id": 1,
                        "code": "script",
                        "name": "Shell Script"
                    },
                    "taskOptions": {
                        "script": "echo `uname a`"
                    }
                }
            }
        ]
    }
}
```

This endpoint will retrieve a specific workflow by id
HTTP Request

GET https://api.gomorpheus.com/api/task-sets/:id

URL Parameters

23.48.3 Create a Workflow

curl -XPOST "https://api.gomorpheus.com/api/task-sets" \ 
-H "Authorization: BEARER access_token" \ 
-H "Content-Type: application/json" \ 
-d '{"taskSet": { \ 
    "name": "my workflow", \ 
    "tasks": [ \ 
        { \ 
            "taskId": 3 \ 
        }, \ 
        { \ 
            "taskId": 8 \ 
        }, \ 
        { \ 
            "taskId": 9, \ 
            "taskPhase": "postProvision" \ 
        } \ 
    ] \ 
}}'

The above command returns JSON structured like getting a single workflow:

HTTP Request

POST https://api.gomorpheus.com/api/task-sets

JSON Parameters

23.48.4 Updating a Workflow

curl -XPUT "https://api.gomorpheus.com/api/task-sets/5" \ 
-H "Authorization: BEARER access_token" \ 
-H "Content-Type: application/json" \ 
-d '{"taskSet": { \ 
    "tasks": [ \ 
        { \ 
            "taskId": 3 \ 
        } \ 
    ] \ 
}}'

The above command returns JSON structured like getting a single workflow:

HTTP Request

PUT https://api.gomorpheus.com/api/task-sets/:id
23.48.5 Delete a Workflow

curl -XDELETE "https://api.gomorpheus.com/api/task-sets/1" \
-H "Authorization: BEARER access_token"

The above command returns JSON Structured like this:

```json
{
  "success": true
}
```

HTTP Request

DELETE https://api.gomorpheus.com/api/task-sets/:id

URL Parameters