Installing Cisco ACI Multi-Site Orchestrator

This chapter contains the following sections:

Deploying Cisco ACI Multi-Site Orchestrator Guidelines

You can deploy Cisco ACI Multi-Site Orchestrator in a number of different ways, such as using an OVA in vCenter, directly in the ESX server without using vCenter, or using a Python script. Cisco recommends using the Python script for deploying Cisco ACI Multi-Site Orchestrator, Release 2.0(1) or later as it automates a number of manual steps and supports remote execution of subsequent Cisco ACI Multi-Site Orchestrator software upgrades.

VMware vSphere Requirements

The following table summarizes the VMware vSphere requirements for Cisco ACI Multi-Site Orchestrator:

Note You must ensure that the following vCPUs, memory, and disk space requirements are reserved for each VM and are not part of a shared resource pool.

Table 1: VMware vSphere Requirements

Cisco ACI Multi-Site Orchestrator Version	VMware vSphere Requirements
Release 2.0(1)	• ESXi 6.0 or later • 6 vCPUs (8 vCPUs recommended) • 24 GB of RAM • 64 GB disk

Deploying Cisco ACI Multi-Site Orchestrator Using Python

The following sections describe how to prepare for and deploy Cisco ACI Multi-Site Orchestrator using Python.

Setting Up Python Environment

This section describes how to set up the Python environment for deploying Cisco ACI Multi-Site Orchestrator using Python. You must set up the Python environment on the laptop or server from which you will run the installation scripts.

Note If you have already set up your python environment, for example for another Multi-Site deployment or upgrade, you can skip this section.

Before you begin

- If you are using Python 2.x, ensure it is version 2.7.14 or later.
- If you are using Python 3.x, ensure it is version 3.4 or later.

Step 1 Download the **ACI Multi-Site Tools** image from Cisco ACI Multi-Site Software Download link.

a) Browse to the Software Download link:

https://software.cisco.com/download/home/285968390/type

b) Click **ACI Multi-Site Software**.

c) Choose the Cisco ACI Multi-Site Orchestrator release version.

d) Download the *ACI Multi-Site Tools Image* file (`tools-msc-<version>.tar.gz`).

Step 2 Extract the files.

```
# tar -xvzf tools-msc-<version>.tar.gz
```

Step 3 Change to the extracted directory.

```
# cd tools-msc-<version>
```

Step 4 Verify that you are running a correct version of Python.

- If you are using Python 2.x, ensure it is version 2.7.14 or later.

```
# python -V
Python 2.7.5
```

- If you are using Python 3.x, ensure it is version 3.4 or later.

```
# python3 -V
Python 3.4.5
```

Step 5 If you plan to use a proxy to access the Internet, make sure to configure the proxy as follows:

```
# export http_proxy=<proxy-ip-address>:<proxy-port>
```

Step 6 Install the Python package installer.

If you are using Python 3.x, replace `python` with `python3` in the following command:

```
# python -m ensurepip
Collecting setuptools
Collecting pip
Installing collected packages: setuptools, pip
Successfully installed pip-9.0.3 setuptools-39.0.1
```

Step 7 Install the required packages.

Cisco recommends using `virutalenv` to install the packages, so they do not impact the existing packages in the system. For more information on how to use `virtualenv`, see Installing packages using pip and virtualenv.

The required packages are listed in the `requirements.txt` file.

If you are using Python 3.x, replace `python` with `python3` in the following command:

```
# python -m pip install -r requirements.txt
```

Note The Python installation must complete successfully. If you encounter any errors, you must address them before proceeding to the next section or the Cisco ACI Multi-Site Orchestrator Python scripts will not work.

Sample `msc_cfg.yml` File

This section provides a sample `msc_cfg.yml` file for deploying Cisco ACI Multi-Site using Python.

In the following sample configuration file all the VMs are created under the same host. The “host” parameter in the configuration file can be given at node level to create the Multi-Site VMs in different hosts.

```
# Vcenter parameters
vcenter:
  name: dev5-vcenter1
  user: administrator@vsphere.local

  # Host under which the MSC VMs need to be created
  host: 192.64.142.55

  # Path to the MSC OVA file
  msc_ova_file: ../images/msc-1.2.1g.ova

  #  Optional. If not given default library name of "msc-content-lib" would be used
  # library: content-library-name

  # Library datastore name
  library_datastore: datastore1

  # Host datastore name
  host_datastore: datastore1

  # MSC VM name prefix. The full name will be of the form vm_name_prefix-node1
  vm_name_prefix: msc-121g

  # Wait Time in seconds for VMs to come up
  vm_wait_time: 120
```

```
# Common parameters for all nodes
common:
  # Network mask
  netmask: 255.255.248.0

  # Gateway' IP address
  gateway: 192.64.136.1

  # Domain Name-Server IP. Leave blank for DHCP
  nameserver: 192.64.136.140

  # Network label of the Management network port-group
  management: "VM Network"


# Node specific parameters
node1:
  # To use static IP, please specify valid IP address for the "ip" attribute
  ip: 192.64.136.204

  # Node specific "netmask" parameter over-rides the comman.netmask
  netmask: 255.255.248.0

node2:
  # To obtain IP via DHCP, please leave the "ip", "gateway" & "nameserver" fields blank
  ip:
  gateway:
  nameserver:

node3:
  # To obtain IP via DHCP, please leave the "ip", "gateway" & "nameserver" fields blank
  ip:
  gateway:
  nameserver:
```

Deploying Multi-Site Orchestrator Using Python

This section describes how to deploy Cisco ACI Multi-Site Orchestrator using Python.

Before you begin

- Ensure that you meet the hardware requirements and compatibility that is listed in the *Cisco ACI Multi-Site Hardware Requirements Guide*.
- Ensure that you meet the requirements and guidelines described in Deploying Cisco ACI Multi-Site Orchestrator Guidelines, on page 1.
- Ensure that the NTP server is configured and reachable from the Orchestrator VMs and that VMware Tools periodic time synchronization is disabled.
- Ensure that the vCenter is reachable from the laptop or server where you will extract the tools and run the installation scripts.
- Ensure that your Python environment is set up as described in Setting Up Python Environment, on page 2.

Step 1 Download the Cisco ACI Multi-Site Orchestrator image and tools.

a) Browse to the Software Download link:

   https://software.cisco.com/download/home/285968390/type

b) Click **ACI Multi-Site Software**.

c) Choose the Cisco ACI Multi-Site Orchestrator release version.

d) Download the *ACI Multi-Site Image* file (`msc-<version>.tar.gz`) for the release.

e) Download the *ACI Multi-Site Tools Image* file (`tools-msc-<version>.tar.gz`) for the release.

Step 2 Extract the `tools-msc-<version>.tar.gz` file to the directory from which you want to run the install scripts.

```
# tar -xvzf tools-msc-<version>.tar.gz
```

Then change into the extracted directory:

```
# cd tools-msc-<version>
```

Step 3 Create a `msc_cfg.yml` configuration file for your install.

You can copy and rename the provided `msc_cfg_example.yml` file or you can create the file using the example provided in Sample msc_cfg.yml File, on page 3.

Step 4 Edit the `msc_cfg.yml` configuration file and fill in all the parameters for your environment.

The parameters that must be filled in are in all caps, for example *<VCENTER_NAME>*. You will also need to update *<MSC_TGZ_FILE_PATH>* with the path to the `msc-<version>.tar.gz` image file you downloaded in Step 1.

For a complete list of available parameters, see the sample `msc_cfg.yml` file is provided in Sample msc_cfg.yml File, on page 3.

Step 5 Execute the script to deploy the Orchestrator VMs and prepare them:

```
# python msc_vm_util.py -c msc_cfg.yml
```

Step 6 Enter vCenter, `node1`, `node2` and `node3` passwords when prompted.

The script creates three Multi-Site Orchestrator VMs and executes the initial deployment scripts. This process may take several minutes to complete. After successful execution, the Multi-Site Orchestrator cluster is ready for use.

It may take several minutes for the deployment to complete.

Step 7 Verify that the cluster was deployed successfully.

a) Log in to any one of the deployed Orchestrator nodes.

b) Verify that all nodes are up and running.

```
# docker node ls
ID                            HOSTNAME        STATUS      AVAILABILITY     [...]
y90ynithc3cejkeazcqlu1uqs *   node1           Ready       Active           [...]
jt67ag14ug2jgaw4r779882xp     node2           Ready       Active           [...]
hoae55eoute615zpqlnxsk8o8     node3           Ready       Active           [...]
```

Confirm the following:

- The `STATUS` field is `Ready` for all nodes.
- The `AVAILABILITY` field is `Active` for all node.
- The `MANAGER STATUS` field is `Leader` for one of the nodes and `Reachable` for the other two.

c) Verify that all replicas are fully up.

```
# docker service ls
ID                  NAME                      MODE            REPLICAS    [...]
p6tw9mflj06u        msc_auditservice          replicated      1/1         [...]
je7s2f7xme6v        msc_authyldapservice      replicated      1/1         [...]
dbd27y76eouq        msc_authytacacsservice    replicated      1/1         [...]
untetoygqn1q        msc_backupservice         global          3/3         [...]
n5eibyw67mbe        msc_cloudsecservice       replicated      1/1         [...]
8inekkof982x        msc_consistencyservice    replicated      1/1         [...]
0qeisrguy7co        msc_endpointservice       replicated      1/1         [...]
e8ji15eni1e0        msc_executionengine       replicated      1/1         [...]
s4gnm2vge0k6        msc_jobschedulerservice   replicated      1/1         [...]
av3bjvb9ukru        msc_kong                  global          3/3         [...]
rqie68m6vf9o        msc_kongdb                replicated      1/1         [...]
51u1g7t6ic33        msc_mongodb1              replicated      1/1         [...]
vrl8xvvx6ky5        msc_mongodb2              replicated      1/1         [...]
0kwk9xw8gu8m        msc_mongodb3              replicated      1/1         [...]
qhejgjn6ctwy        msc_platformservice       global          3/3         [...]
l7co71lneegn        msc_schemaservice         global          3/3         [...]
1t37ew5m7dxi        msc_siteservice           global          3/3         [...]
tu37sw68a1gz        msc_syncengine            global          3/3         [...]
8dr0d7pq6j19        msc_ui                    global          3/3         [...]
swnrzrbcv60h        msc_userservice           global          3/3         [...]
```

d) Log in to the Cisco ACI Multi-Site Orchestrator GUI.

You can access the GUI using any of the 3 nodes' IP addresses.

The default log in is **admin** and the default password is **We1come2msc!**.

When you first log in, you will be prompted to change the password.

What to do next

For more information about Day-0 Operations, see Day 0 Operations of Cisco ACI Multi-Site.

Deploying Cisco ACI Multi-Site Orchestrator Directly in ESX without Using vCenter

This section provides describes how to deploy Cisco ACI Multi-Site Orchestrator directly in ESX without using vCenter.

Step 1 Download the `msc-<version>.ova` from Cisco ACI Multi-Site Software Download link.

a) Go to the Software Download link:

https://software.cisco.com/download/home/285968390/type

b) Click **ACI Multi-Site Software**.

c) Choose the release version image and click the download icon.

Step 2 Untar the OVA file into a new temporary directory:

```
$ mkdir msc_ova
$ cd msc_ova
$ tar xvf ../msc-<version>.ova
esx-msc-<version>.ovf
esx-msc-<version>.mf
esx-msc-<version>.cert
msc-<version>.ovf
msc-<version>.mf
msc-<version>.cert
msc-<version>-disk1.vmdk
```

This creates several files.

Step 3 Use the ESX vSphere client to deploy the OVF.

a) Navigate to **File** > **Deploy OVF Template** > **Browse** and choose the `esx-msc-<version>.ovf` file.

b) Complete rest of the menu options and deploy the VM.

c) Repeat step 3 to create each Cisco ACI Multi-Site Orchestrator node.

Step 4 Configure the hostname for each VM by using the command line interface (CLI) or the text user interface (TUI) tool.

a) Using the CLI:

On the first node, enter the following command:

```
# hostnamectl set-hostname node1
```

On the second node, enter the following command:

```
# hostnamectl set-hostname node2
```

On the third node, enter the following command:

```
# hostnamectl set-hostname node3
```

Using the TUI tool:

Enter the **nmtui** command to configure the hostnames for each VM.

b) You must logout and log back in for each VM.

Step 5 On `node1`, perform the following:

a) Connect to `node1` using SSH.

b) Change to the `/opt/cisco/msc/builds/<build_number>/prodha` directory:

```
[root@node1]# cd /opt/cisco/msc/builds/<build_number>/prodha
```

c) Execute the `msc_cfg_init.py` command:

```
[root@node1 prodha]# ./msc_cfg_init.py
Starting the initialization of the cluster...
.
.
.
Both secrets created successfully.

Join other nodes to the cluster by executing the following on each of the other nodes:
./msc_cfg_join.py \
SWMTKN-1-4pu9zc9d81gxxw6mxec5tuxdt8nbarq1qnmfw9zcme1w1tljZh-7w3iwsddvd97ieza3ym1s5gj5 \
<ip_address_of_the_first_node>
```

d) Take note of the management IP address of the first node, enter the following command:

```
# ifconfig
inet 10.23.230.151  netmask 255.255.255.0  broadcast 192.168.99.255
```

Step 6 On `node2`, perform the following:

a) Connect to `node2` using SSH.

b) Change to the `/opt/cisco/msc/builds/<build_number>/prodha` directory:

```
# cd /opt/cisco/msc/builds/<build_number>/prodha
```

c) Execute the `msc_cfg_join.py` command using the IP address of the first node that was indicated in step 5c and 5d:

Example:

```
# ./msc_cfg_join.py \
SWMTKN-1-4pu9zc9d81gxxw6mxec5tuxdt8nbarq1qnmfw9zcme1w1tljZh-7w3iwsddvd97ieza3ym1s5gj5 \
10.23.230.151
```

Step 7 On `node3`, perform the following:

a) Connect to `node3` using SSH.

b) Change to the `/opt/cisco/msc/builds/<build_number>/prodha` directory:

```
# cd /opt/cisco/msc/builds/<build_number>/prodha
```

c) Execute the `msc_cfg_join.py` command using the IP address of the first node that was indicated in step 5c and 5d:

Example:

```
# ./msc_cfg_join.py \
SWMTKN-1-4pu9zc9d81gxxw6mxec5tuxdt8nbarq1qnmfw9zcme1w1tljZh-7w3iwsddvd97ieza3ym1s5gj5 \
10.23.230.151
```

Step 8 On any node, make sure the nodes are healthy.

Using the following command, verify that the STATUS is `Ready`, the AVAILABILITY is `Active` for each node, and the MANAGER STATUS is `Reachable` except for only one showing `Leader`:

Example:

```
# docker node ls
ID                            HOSTNAME    STATUS    AVAILABILITY    MANAGER STATUS
g3mebdulaed2n0cyywjrtum31     node2       Ready     Active          Reachable
ucgd7mm2e2divnw9kvm4in7r7     node1       Ready     Active          Leader
zjt4dsodu3bff3ipn0dg5h3po *   node3       Ready     Active          Reachable
```

Step 9 On any node, execute the `msc_deploy.py` command:

```
# ./msc_deploy.py
```

Step 10 On any node, make sure that all REPLICAS are up.

Using the following command, verify that all REPLICAS are fully up:

Example:

```
# docker service ls
ID           NAME                 MODE       REPLICAS  IMAGE                          PORTS
1jmn525od7g6 msc_kongdb           replicated 1/1       postgres:9.4
2imn83pd4l38 msc_mongodb3         replicated 1/1       mongo:3.4
2kc6foltcv1p msc_siteservice      global     3/3       msc-siteservice:0.3.0-407
```

```
6673appbs300 msc_schemaservice   global     3/3       msc-schemaservice:0.3.0-407
clqjgftg5ie2 msc_kong            global     3/3       msc-kong:1.1
j49z7kfvmu04 msc_mongodb2        replicated 1/1       mongo:3.4
lt4f2l1yqiw1 msc_mongodb1        replicated 1/1       mongo:3.4
mwsvixcxipse msc_executionengine replicated 1/1       msc-executionengine:0.3.0-407
qnleu9wvw800 msc_syncengine      replicated 1/1       msc-syncengine:0.3.0-407
tfaqq4tkyhtx msc_ui              global     3/3       msc-ui:0.3.0-407
*:80->80/tcp,*:443->443/tcp
ujcmf70r16zw msc_platformservice global     3/3       msc-platformservice:0.3.0-407
uocu9msiarux msc_userservice     global     3/3       msc-userservice:0.3.0-407
```

Step 11 Open the browser and enter any IP address of the 3 nodes to bring up the Cisco ACI Multi-Site Orchestrator GUI.

Example:

https://10.23.230.151

Step 12 Log in to the Cisco ACI Multi-Site Orchestrator GUI, the default log in is **admin** and the password is **We1come2msc!**.

Step 13 Upon initial log in you will be forced to reset the password. Enter the current password and new password.

The new password requirements are:

- At least 12 characters
- At least 1 letter
- At least 1 number
- At least 1 special character (`*` and `space`are not allowed)

For more information about Day 0 Operations, see Day-0 Operations.

Deploying Cisco ACI Multi-Site Orchestrator Using an OVA

This section describes how to deploy Cisco ACI Multi-Site Orchestrator, Release 2.0(x) using an OVA.

Before you begin

- Make sure you meet the hardware requirements. For more information, see the Cisco ACI Multi-Site Hardware Requirements Guide.
- Make sure you meet the VMware vSphere requirements, For more information, see the Deploying Cisco ACI Multi-Site Orchestrator Guidelines, on page 1.

Step 1 Install the virtual machines (VMs):

a) Deploy OVA using the vCenter either the WebGUI or the vSphere Client.

Note The OVA cannot be deployed directly in ESX, it must be deployed using vCenter. If you want to deploy Cisco ACI Multi-Site Orchestrator directly in ESX, see Deploying Cisco ACI Multi-Site Orchestrator Directly in ESX without Using vCenter, on page 6 for instructions on how to extract the OVA and install Multi-Site without vCenter.

In the **Properties** dialog box, enter the appropriate information for each VM:

- In the **Enter password** field, enter the password.
- In the **Confirm password** field, enter the password again.
- In the **Hostname** field, enter the hostnames for each Cisco ACI Multi-Site Orchestrator node. You can use any valid Linux hostname.
- In the **Management Address** (network address) field, enter the network address or leave the field blank to obtain it via DHCP.
- In the **Management Netmask** (network netmask) field, enter the netmask netmask or leave the field blank to obtain it via DHCP.
- In the **Management Gateway** (network gateway) field, enter the network gateway or leave the field blank to obtain it via DHCP.
- In the **Domain Name System Server** (DNS server) field, enter the DNS server or leave the field blank to obtain it via DHCP.
- In the **Time-zone string (Time-zone)** field, enter a valid time-zone string.
- In the **NTP-servers** field, enter Network Time Protocol servers separated by commas or leave the field blank to disable NTP.

  Click **Next**.
- In the **Deployment settings** pane, check all the information you provided is correct.
- Click **Power on after deployment**.
- Click **Finish**.
- Repeat the properties setup for each VM.

b) Ensure that the virtual machines are able to ping each other.

Step 2 On node1, perform the following:

a) Connect to node1 using SSH.

b) Change to the `/opt/cisco/msc/builds/<build_number>/prodha` directory:

```
[root@node1]# cd /opt/cisco/msc/builds/<build_number>/prodha
```

c) Execute the `msc_cfg_init.py` command:

```
[root@node1 prodha]# ./msc_cfg_init.py
Starting the initialization of the cluster...
.
.
.
Both secrets created successfully.

Join other nodes to the cluster by executing the following on each of the other nodes:
./msc_cfg_join.py \
SWMTKN-1-4pu9zc9d81gxxw6mxec5tuxdt8nbarq1qnmfw9zcme1w1tljZh-7w3iwsddvd97ieza3ym1s5gj5 \
<ip_address_of_the_first_node>
```

d) Take note of the management IP address of the first node, enter the following command:

```
[root@node1 prodha]# ifconfig
inet 10.23.230.151  netmask 255.255.255.0  broadcast 192.168.99.255
```

Step 3 On node2, perform the following:

a) Connect to node2 using SSH.

b) Change to the `/opt/cisco/msc/builds/<build_number>/prodha` directory:

```
[root@node2]# cd /opt/cisco/msc/builds/<build_number>/prodha
```

c) Execute the `msc_cfg_join.py` command using the IP address of the first node that was indicated in step 2c and d:

Example:

```
[root@node2 prodha]# ./msc_cfg_join.py \
SWMTKN-1-4pu9zc9d81gxxw6mxec5tuxdt8nbarq1qnmfw9zcme1w1tljZh-7w3iwsddvd97ieza3ym1s5gj5 \
10.23.230.151
```

Step 4 On node3, perform the following:

a) Connect to node3 using SSH.

b) Change to the `/opt/cisco/msc/builds/<build_number>/prodha` directory:

```
[root@node3]# cd /opt/cisco/msc/builds/<build_number>/prodha
```

c) Execute the `msc_cfg_join.py` command using the IP address of the first node that was indicated in step 2c and d:

Example:

```
[root@node3 prodha]# ./msc_cfg_join.py \
SWMTKN-1-4pu9zc9d81gxxw6mxec5tuxdt8nbarq1qnmfw9zcme1w1tljZh-7w3iwsddvd97ieza3ym1s5gj5 \
10.23.230.151
```

Step 5 On any node, make sure the nodes are heathly. Verify that the STATUS is `Ready`, the AVAILABILITY is `Active` for each node, and the MANAGER STATUS is `Reachable` except for only one showing `Leader`:

```
[root@node1 prodha]# docker node ls
```

Sample output:

```
ID                            HOSTNAME    STATUS    AVAILABILITY     MANAGER STATUS
g3mebdulaed2n0cyywjrtum31     node2       Ready     Active           Reachable
ucgd7mm2e2divnw9kvm4in7r7     node1       Ready     Active           Leader
zjt4dsodu3bff3ipn0dg5h3po *   node3       Ready     Active           Reachable
```

Step 6 On any node, execute the `msc_deploy.py` command:

```
[root@node1 prodha]# ./msc_deploy.py
```

Step 7 On any node, make sure that all REPLICAS are up. For example, make sure it states 3/3 (3 out of 3) or 1/1 (1 out of 1).

Example:

```
[root@node1 prodha]# docker service ls
```

Sample output:

```
ID            NAME                MODE        REPLICAS IMAGE                        PORTS
1jmn525od7g6  msc_kongdb          replicated 1/1      postgres:9.4
2imn83pd4138  msc_mongodb3        replicated 1/1      mongo:3.4
2kc6foltcv1p  msc_siteservice     global     3/3      msc-siteservice:0.3.0-407
6673appbs300  msc_schemaservice   global     3/3      msc-schemaservice:0.3.0-407
clqjgftg5ie2  msc_kong            global     3/3      msc-kong:1.1
```

```
j49z7kfvmu04  msc_mongodb2        replicated 1/1      mongo:3.4
lt4f2l1yqiwl  msc_mongodb1        replicated 1/1      mongo:3.4
mwsvixcxipse  msc_executionengine replicated 1/1      msc-executionengine:0.3.0-407
qnleu9wvw800  msc_syncengine      replicated 1/1      msc-syncengine:0.3.0-407
tfaqq4tkyhtx  msc_ui              global     3/3      msc-ui:0.3.0-407
*:80->80/tcp,*:443->443/tcp
ujcmf70r16zw  msc_platformservice global     3/3      msc-platformservice:0.3.0-407
uocu9msiarux  msc_userservice     global     3/3      msc-userservice:0.3.0-407
```

Step 8 Open the browser and enter any IP address of the 3 nodes to bring up the Cisco ACI Multi-Site Orchestrator GUI.

Example:

https://10.23.230.151

Step 9 Log in to the Cisco ACI Multi-Site Orchestrator GUI, the default log in is **admin** and the password is **we1come!**.

Step 10 Upon initial log in you will be forced to reset the password. Enter the current password and new password.

The new password requirements are:

- At least 6 characters
- At least 1 letter
- At least 1 number
- At least 1 special character apart from * and space

For more information about Day 0 Operations, see Day-0 Operations.