



# Cisco ACI Multi-Site Installation

This chapter contains the following sections:

- [Deploying Cisco ACI Multi-Site Guidelines, on page 1](#)
- [Deploy Cisco ACI Multi-Site Using Python, on page 2](#)
- [Deploying Cisco ACI Multi-Site Directly in ESX without Using vCenter, on page 6](#)
- [Deploying Cisco ACI Multi-Site Release 1.2\(x\) Using an OVA, on page 7](#)
- [Deploying Cisco ACI Multi-Site Release 1.1\(x\) Using an OVA, on page 9](#)
- [Deploying Cisco ACI Multi-Site Release 1.0\(x\) Using an OVA, on page 12](#)

## Deploying Cisco ACI Multi-Site Guidelines

### VMware vSphere Requirements

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



**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 Version	VMware vSphere Requirements
Release 1.2(x)	<ul style="list-style-type: none"><li>• ESXi 6.0 or later</li><li>• 6 vCPUs (8 vCPUs recommended)</li><li>• 24 GB of RAM</li><li>• 64 GB disk</li></ul>

Cisco ACI Multi-Site Version	VMware vSphere Requirements
Release 1.1(x)	<ul style="list-style-type: none"> <li>• ESXi 6.0 or later</li> <li>• 4 vCPUs</li> <li>• 8 GB of RAM</li> <li>• 64 GB disk</li> </ul>
Release 1.0(x)	<ul style="list-style-type: none"> <li>• ESXi 5.5 or later</li> <li>• 4 vCPUs</li> <li>• 8 GB of RAM</li> <li>• 64 GB disk</li> </ul>

## Deploy Cisco ACI Multi-Site Using Python

After you fulfill the preinstallation prerequisites, you can use Python to deploy Cisco ACI Multi-Site.

### Setting Up the Python Environment for Deploying Cisco ACI Multi-Site

This section describes how to set up the Python environment for deploying Cisco ACI Multi-Site 1.2(1) or later.

#### Before you begin

- Make sure that you have Python 2.7.14+ or Python 3.4+.

#### Procedure

**Step 1** Download the **ACI Multi-Site Tools image** 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 **ACI Multi-Site Tools image** release version and click the download icon.

**Step 2** Untar and extract the files:

```
$ tar xvf tools-msc-<build_number>.tar.gz
```

```
msc_cfg_example.yml
msc_lib.py
msc_vm_clean.py
msc_vm_util.py
Node.py
python
```

```
README
requirements.txt
```

**Step 3** Change to the `tools-msc-<build_number>` directory:

```
$ cd tools-msc-<build_number>
```

**Step 4** Verify that you are running either Python 2.7.14 or later or Python 3.4 or later.

```
$ python -V
Python 2.7.15
```

**Step 5** Ensure you have permission to install python packages. For example, change shell to become super-user:

```
$ sudo bash
```

**Step 6** If you plan to use a proxy to access the Internet, make sure to configure the proxy as follows:

**Example:**

```
$ export http_proxy=your_proxy_ip:your_proxy_port
$ export https_proxy=your_proxy_ip:your_proxy_port
```

**Step 7** Install the python package installer:

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

**Step 8** Install the packages in `requirements.txt`:

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

**Step 9** Exit the shell:

```
# exit
$
```

Once you have completed all the steps, proceed to [Deploying Cisco ACI Multi-Site Using Python, on page 3](#).

If there is any errors, address them. You must complete the above steps or the Multi-Site python scripts will not work.

---

## Deploying Cisco ACI Multi-Site Using Python

This section describes how to deploy Cisco ACI Multi-Site 1.2(1) or later using Python.

### Before you begin

- Make sure that you meet the hardware requirements and compatibility listed in the [Cisco ACI Multi-Site Hardware Requirements Guide](#).
- Set up the Python environment as described in [Setting Up the Python Environment for Deploying Cisco ACI Multi-Site, on page 2](#)
- Make sure that the vCenter is reachable from the server where the tools are being executed.

## Procedure

---

- Step 1** Copy the `msc_cfg_example.yml` file and rename it to `msc_cfg.yml`.
- ```
$ cp msc_cfg_example.yml msc_cfg.yml
```
- a) Edit the `msc_cfg.yml` configuration file and fill in all the parameters for your environment.
- All the parameters that need to be filled in are in all caps, for example: `<VCENTER_NAME>`.
- For a sample `msc_cfg.yml` file, see [Sample msc\\_cfg.yml File, on page 4](#).
- Step 2** Execute the script to deploy the MSC VMs and prepare them:
- ```
$ python msc_vm_util.py
```
- To see the full options supported, enter:
- ```
$ python msc_vm_util.py -h
```
- a) Enter vCenter, node1, node2 and node3 passwords when prompted.
- You have completed the deployment.
- Step 3** The script creates three Multi-Site VMs and execute the initial deployment scripts. It will take several minutes to create the VMs and execute the deployment scripts. After successful execution the Multi-Site cluster is ready for use. You can verify by accessing the Multi-Site GUI.
- 

## Sample msc\_cfg.yml File

This is a sample `msc_cfg.yml` file:

```
#
# 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
#
# Example: /home/user/image/msc-1.2.1b.ova
#
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-nodel
#
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 maske
#
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" field blank
#
ip: 192.64.136.206
```



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

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

This section describes how to deploy Cisco ACI Multi-Site 1.2(1) or later directly in ESX without using vCenter.

### Procedure

- 
- Step 1** Download the `msc-<version>.ova` from Cisco ACI Multi-Site Software Download link.
- Go to the Software Download link:  
<https://software.cisco.com/download/home/285968390/type>
  - Click **ACI Multi-Site Software**.
  - 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.
- Navigate to **File > Deploy OVF Template > Browse** and choose the `esx-msc-<version>.ovf` file.
  - Complete rest of the menu options and deploy the VM.
  - Repeat step 3 to create each Multi-Site node.
- Step 4** Follow the procedure in [Deploying Cisco ACI Multi-Site Release 1.0\(x\) Using an OVA, on page 12](#) to manually configure each of the nodes and bring up the Multi-Site node cluster.
-

# Deploying Cisco ACI Multi-Site Release 1.2(x) Using an OVA

This section describes how to deploy Cisco ACI Multi-Site Release 1.2(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 Guidelines, on page 1](#).

## Procedure

### Step 1

Install the virtual machines (VMs):

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

**Note** The Multi-Site OVA cannot be directly deployed in ESX. Multi-Site OVA must be deployed using 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 first node as node1, the second node as node2, and third node as node3. The given hostnames must be node1, node2, and node3.

**Note** Any deviation from using the given hostnames ("node1", "node2", "node3") causes the setup to fail.

- In the **Management Address** (network address) field, enter the network address.
- In the **Management Netmask** (network netmask) field, enter the netmask netmask.
- In the **Management Gateway** (network gateway) field, enter the network gateway.
- In the **Domain Name System Server** (DNS server) field, enter the DNS server.
- 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-1-4pu9zc9d81gxxw6mxec5tuxdt8nbarq1qnmfw9zcmelw1t1jZh-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-1-4pu9zc9d81gxxw6mxec5tuxdt8nbarq1qnmfw9zcmelw1t1jZh-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-1-4pu9zc9d81gxxw6mxec5tuxdt8nbarq1qnmfw9zcmelw1t1jZh-7w3iwsddvd97ieza3ym1s5gj5 \
10.23.230.151
```

### Step 5

On any node, make sure the nodes are healthy. 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
g3mebdulaed2n0cywjrjum31	node2	Ready	Active	Reachable



```
ucgd7mm2e2divnw9kvm4in7r7    node1    Ready    Active    Leader
zjt4dsodu3bfff3ipn0dg5h3po *  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	
lt4f21lyqiwl	msc_mongodb1	replicated	1/1	mongo:3.4	
mwsvixcxipte	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 Multi-Site GUI.

**Example:**

**https://10.23.230.151**

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

**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 Overview](#).

## Deploying Cisco ACI Multi-Site Release 1.1(x) Using an OVA

This section describes how to deploy Cisco ACI Multi-Site Release 1.1(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 Guidelines, on page 1](#).

**Procedure****Step 1**

Install the virtual machines (VMs):

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

**Note** In Release 1.1(x), the new OVF properties have been added to Multi-Site OVA, the Multi-Site OVA cannot be directly deployed in ESX. Multi-Site OVA must be deployed using 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 first node as node1, the second node as node2, and third node as node3. The given hostnames must be node1, node2, and node3.

**Note** Any deviation from using the given hostnames ("node1", "node2", "node3") causes the setup to fail.

- In the **Management Address** (network address) field, enter the network address.
- In the **Management Netmask** (network netmask) field, enter the netmask netmask.
- In the **Management Gateway** (network gateway) field, enter the network gateway.
- In the **Domain Name System Server** (DNS server) field, enter the DNS server.
- 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-1-4pu9zc9d81gxxw6mxec5tuxdt8nbarq1qnmfw9zcmelw1t1jZh-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:

- Connect to node2 using SSH.
- Change to the `/opt/cisco/msc/builds/<build_number>/prodha` directory:

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

- 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-1-4pu9zc9d81gxxw6mxec5tuxdt8nbarq1qnmfw9zcmelw1t1jZh-7w3iwsddvd97ieza3ym1s5gj5 \
10.23.230.151
```

### Step 4

On node3, perform the following:

- Connect to node3 using SSH.
- Change to the `/opt/cisco/msc/builds/<build_number>/prodha` directory:

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

- 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-1-4pu9zc9d81gxxw6mxec5tuxdt8nbarq1qnmfw9zcmelw1t1jZh-7w3iwsddvd97ieza3ym1s5gj5 \
10.23.230.151
```

### Step 5

On any node, make sure the nodes are healthy. 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	
lt4f21lyqiwl	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 Multi-Site GUI.

**Example:**

<https://10.23.230.151>

**Step 9** Log in to the Multi-Site 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 Overview](#).

## Deploying Cisco ACI Multi-Site Release 1.0(x) Using an OVA

This section describes how to deploy Cisco ACI Multi-Site Release 1.0(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 Guidelines, on page 1](#).

## Procedure

---

### Step 1

Install the virtual machines (VMs):

- a) Deploy OVA to the vSphere.
- b) Clone the VM two more times.
- c) Power on each VM.
- d) Use the vSphere console to log in to the VM:
  - Log in using the default root password `cisco`.
  - Upon first log in, it forces you to change your passwords.  
If you see the following error on initial login password reset:  
`Authentication token manipulation error`  
Ensure you are re-entering the current password `cisco`.
  - Specify the IP address for `eth0` using the `nmtui` command or you can use another method.  
If using the `nmtui` command, you must deactivate and activate the `eth0` NIC to ensure the changes apply.
  - Repeat step 1d for the other two VMs.
- e) Ensure that the virtual machines are able to ping each other.

### Step 2

Configure the hostname for each VM by using the command line interface (CLI) or the text user interface (TUI) tool. The given hostnames must be `node1`, `node2`, and `node3`.

**Note** Any deviation from using the given hostnames ("`node1`", "`node2`", "`node3`") causes the setup to fail.

- 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 3

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-1-4pu9zc9d81gxxw6mxec5tuxdt8nbarq1qnmfw9zcmelw1tljZh-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 4** On node2, perform the following:

- Connect to node2 using SSH.
- Change to the `/opt/cisco/msc/builds/<build_number>/prodha` directory:

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

- Execute the `msc_cfg_join.py` command using the IP address of the first node that was indicated in step 3c and d:

**Example:**

```
[root@node2 prodha]# ./msc_cfg_join.py \
SWMTKN-1-4pu9zc9d81gxxw6mxec5tuxdt8nbarq1qnmfw9zcmelw1tljZh-7w3iwsddvd97ieza3ym1s5gj5 \
10.23.230.151
```

**Step 5** On node3, perform the following:

- Connect to node3 using SSH.
- Change to the `/opt/cisco/msc/builds/<build_number>/prodha` directory:

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

- Execute the `msc_cfg_join.py` command using the IP address of the first node that was indicated in step 3c and d:

**Example:**

```
[root@node3 prodha]# ./msc_cfg_join.py \
SWMTKN-1-4pu9zc9d81gxxw6mxec5tuxdt8nbarq1qnmfw9zcmelw1tljZh-7w3iwsddvd97ieza3ym1s5gj5 \
10.23.230.151
```

**Step 6** On any node, make sure the nodes are healthy. 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 7** On any node, execute the `msc_deploy.py` command:

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

**Step 8** 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	
2imn83pd4l38	msc_mongodb3	replicated	1/1	mongo:3.4	
2kc6foltcvlp	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	
lt4f211yqiwl	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 9** Open the browser and enter any IP address of the 3 nodes to bring up the Multi-Site GUI.

**Example:**

<https://10.23.230.151>

**Step 10** Log in to the Multi-Site GUI, the default log in is **admin** and the password is **welcome!**.

**Step 11** 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 Overview](#).

