

Upgrading Orchestrator Deployments in VMware ESX

This chapter contains the following sections:

- Prerequisites and Guidelines, on page 1
- Upgrading Cisco ACI Multi-Site Orchestrator Using Python, on page 2

Prerequisites and Guidelines

This section describes how to upgrade a 3-node Cisco ACI Multi-Site Orchestrator cluster that was deployed in VMware ESX VMs. If your Orchestrator cluster was deployed in Cisco Application Service Engine, see the #unique_73 chapter. If you deployed a single-node Orchestrator (for example, for testing purposes), the upgrade procedure differs slightly and is described in Installing Single Node Orchestrator chapter instead.

Before you upgrade your Cisco ACI Multi-Site Orchestrator cluster, you must:

- Ensure that you are running at least Cisco ACI Multi-Site Orchestrator, Release 1.2(1). If you are running an earlier release, you must first upgrade it as described in *Upgrading Cisco ACI Multi-Site Orchestrator to Release 1.2(x)*.
- Ensure that your current Cisco ACI Multi-Site Orchestrator installation is running properly and each node in the cluster has at least 19 GB of free disk space.
- Ensure that all Cisco ACI Multi-Site Orchestrator node VMs have been upgraded to any new minimum CPU and RAM requirements that are listed in Prerequisites and Guidelines.

  When upgrading the virtual machines:

  - It is recommended that all virtual machine CPU and RAM changes are done when the VM is powered down, as such we recommend updating the VMs one at a time to ensure that the cluster remains available.
  - Do not change the hard disk size of the Cisco ACI Multi-Site Orchestrator VMs.
- Ensure that you have set up the Python environment as described in Setting Up Python Environment.
- If you are upgrading from a release prior to Release 2.1(1), configure at least 10GHz CPU cycle reservation for each Orchestrator VM.

  Specific steps are described as part of the upgrade procedure.

Multi-Site Orchestrator and Cisco APIC Interoperability Support

Multi-Site Orchestrator (MSO) does not require a specific version of APIC to be running in all sites. The APIC clusters in each site as well as the MSO itself can be upgraded independently of each other and run in mixed operation mode as long as each fabric is running APIC Release 3.2(6) or later. As such, we recommend that you always upgrade to the latest release of the Multi-Site Orchestrator.

However, keep in mind that if you upgrade the MSO before upgrading the APIC clusters in one or more sites, some of the new MSO features may not yet be supported by an earlier APIC release. In that case a check is performed on each template to ensure that every configured option is supported by the target sites.

The check is performed when you save a template or deploy a template. If the template is already assigned to a site, any unsupported configuration options will not be saved; if the template is not yet assigned, you will be able to assign it to a site, but not be able to save or deploy the schema if it contains configuration unsupported by that site.

In case an unsupported configuration is detected, an error message will show, for example: `This APIC site version <site-version> is not supported by MSO. The minimum version required for this <feature> is <required-version> or above.`

The following table lists the features and the minimum required APIC release for each one:

Feature	Minimum APIC Version
ACI Multi-Pod Support	Release 3.2(6)
Service Graphs (L4-L7 Services)	Release 3.2(6)
External EPGs	Release 3.2(6)
ACI Virtual Edge VMM Support	Release 3.2(6)
DHCP Support	Release 3.2(6)
Consistency Checker	Release 3.2(6)
vzAny	Release 3.2(6)
Host Based Routing	Release 4.0(1)
CloudSec Encryption	Release 4.0(1)
Layer 3 Multicast	Release 4.0(1)
MD5 Authentication for OSPF	Release 4.0(1)
EPG Preferred Group	Release 4.0(2)
Intersite L3Out	Release 4.2(1)

Upgrading Cisco ACI Multi-Site Orchestrator Using Python

The following sections describe how to prepare for and upgrade 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

you will need:

- A laptop or server from which you will run the scripts.

  You must not use any of the Multi-Site Orchestrator nodes for this purpose.

- Python 3.4 or later already installed on the system from which you will run the scripts.

Procedure

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.

Depending on your operating system and the versions of Python you have installed previously, the Python 3.x executable may be associated with `python` or `python3` command.

You can use one of the following examples to confirm which executable corresponds to Python 3.4 or later.

- If you have installed Python 2.x first, the `python` command will likely be associated with that version:

  ```
  # python -V
  Python 2.7.18
  ```

  In this case, you may need to use `python3` command instead:

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

- If you have installed only Python 3.x, you can use the `python` command as well:

```
# python -V
Python 3.7.7
```

Note
The following steps use `python` for all commands. If your Python 3.x executable is associated with `python3`, use that command instead.

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

Step 6 Install or update the Python package manager.

```
# python -m ensurepip
```

If the package is already installed, update it to the latest version:

```
# python -m ensurepip --upgrade
```

Step 7 (Optional) Set up Python virtual environment.

We recommend using `virutalenv` to install the packages, so they do not impact the existing packages in the system. The following steps provide a brief overview of how to set up `virutalenv`. For additional information on how to use `virtualenv`, see Installing packages using pip and virtualenv.

a) Install `virtualenv`.

```
# python -m pip install --user virtualenv
```

b) Change into the directory where you want the virtual environment files to be created.

c) Create a virtual environment.

In the following command, provide a name for the virtual environment, for example *mso-deployments*.

```
# python -m venv <env-name>
```

d) Activate the virtual environment.

You need to activate the virtual environment you created before installing the packages required for Orchestrator deployment or upgrade in the next step.

For Windows:

```
# .\<env-name>\Scripts\activate.bat
```

For Linux:

```
# source ./<env-name>/bin/activate
```

Step 8 Install the required packages.

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

```
# 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.

Step 9 If you used virtual Python environment, deactivate it now.

```
# deactivate
```

Sample Upgrade Configuration File

When you upgrade Multi-Site Orchestrator using Python, you can provide all the required information as command line arguments to the upgrade script or you can specify them all in a YAML configuration file and simply provide the configuration file.

This section provides sample `msc_cfg_upgrade.yml` files for two different upgrade scenarios:

- If your Multi-Site Orchestrator cluster was deployed using the Python installation scripts and within the same vCenter, you can use your vCenter login information to automatically find and upgrade all the nodes.

```
vcenter:
  name: 192.168.142.59
  user: administrator@vsphere.local

  # Update script will look for VMs with this prefix and Orchestrator label
  vm_name_prefix: msc


update:
  # Action can be 'upgrade' or 'downgrade'
  action: upgrade

  # Path to the Orchestrator upgrade image file
  msc_tgz_file: ~/tmp/msc-<version>.tar.gz
```

- If your Multi-Site Orchestrator cluster was deployed using the OVA, or in multiple different vCenters, or you simply do not wish to provide the vCenter login information, you can specify each node's IP address explicitly.

```
node1:
  ip: 192.64.136.204

node2:
  ip: 192.64.136.205

node3:
  ip: 192.64.136.206

update:
  # Action can be 'upgrade' or 'downgrade'
  action: upgrade

  # Path to the Orchestrator upgrade image file
  msc_tgz_file: ~/tmp/msc-<version>.tar.gz
```

Upgrading Multi-Site Orchestrator

This section describes how to upgrade Cisco ACI Multi-Site Orchestrator.

Note If you are upgrading from a release prior to Release 2.0(1), due to recent PSIRT updates the Cisco ACI Multi-Site Orchestrator nodes' kernels must be updated during the upgrade to Release 2.1(1) or later. This kernel update requires the nodes to be reloaded prior to performing the Orchestrator software upgrade. The python script performs the necessary update and reload automatically, followed by the Orchestrator software upgrade.

Before you begin

- Ensure that you have completed the prerequisites described in Prerequisites and Guidelines, on page 1
- Set up the Python environment as described in Setting Up Python Environment

Procedure

Step 1 If you are upgrading from a release prior to Release 2.1(1), configure at least 10GHz CPU cycle reservation for each Orchestrator VM.

This release of Multi-Site Orchestrator requires at least 10GHz CPU cycle reservation for each VM. New deployments of Release 2.1(1) or later apply CPU cycle reservation automatically, however if you're upgrading from an earlier release, you must manually update each Orchestrator VM's settings.

a) Log in to the vSphere client.
b) Navigate to the ESX host where your Orchestrator VMs are located.
c) Shut down one of the VMs.
d) Right click the VM and choose **`Edit Settings`**
e) In the **Virtual Hardware** tab, expand the **CPU** category.
f) In the **Reservation** field, enter `10 GHz`.
g) Click **OK** to save the changes.
h) Power on the VM and wait for the Orchestrator cluster to stabilize with all nodes healthy.
i) Repeat the steps for the other Orchestrator VMs.

Step 2 Download the Cisco ACI Multi-Site Orchestrator upgrade 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 Upgrade 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 3 Extract the `tools-msc-<version>.tar.gz` file to the directory from which you want to run the upgrade.

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

Then change into the extracted directory:

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

Step 4 (Optional) Create a configuration file for the upgrade.

We recommend creating an upgrade configuration file with your deployment and upgrade details. This allows you to provide the required information once and then re-use the file for future upgrades.

If you would rather pass all the required information directly to the upgrade script, you can skip this step.

If you choose to use a configuration file, you can copy and rename the provided `msc_cfg_upgrade_example.yml` file, create one using one of the samples provided in Sample Upgrade Configuration File, on page 5, or update one from a previous upgrade with the new image path.

Step 5 If your last upgrade was from a release prior to Release 1.2(x), update the version database.

If your current Multi-Site Orchestrator installation was a fresh install of Release 1.2(1) or later, or you have upgraded to Release 2.0(x) or later in the past, skip this step.

Otherwise, run the following command replacing *1.2.3b* with the currently installed version:

```
# /opt/cisco/msc/builds/msc_1.2.3b/bin/save_msc_version.sh 1.2.3b
```

Step 6 Upgrade the Cisco ACI Multi-Site nodes.

If you created a configuration file for the upgrade as descibed in Step 4, simply run the following command:

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

If you would rather specify all the information on the command line, use the following command:

```
# python msc_vm_util.py -u -f msc-<version>.tar.gz -n1ip <node1-ip> -n2ip <node2-ip> -n3ip <node3-ip>
```

Step 7 Enter the passwords when prompted.

The script creates a backup of the MongoDB before the upgrade. It then copies the upgrade image to each node and executes the upgrade scripts.

It may take several minutes for the upgrade to complete. After the upgrade is complete, you can verify that the upgrade was successful and the Cisco ACI Multi-Site Orchestrator cluster is ready for use by accessing the Orchestrator GUI.

Step 8 If you upgraded from a release prior to Release 2.1(1), log in to your Orchestrator GUI and reset the password.

Due to password requirements change in Release 2.1(1), when you first log in to the Orchestrator GUI after upgrading to Release 2.1(1) or later, you will be prompted to update your 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