



Installing the Cisco MSX Platform and Service Packs

The high-level tasks involved in Cisco Managed Services Accelerator (MSX) installation are:

1. **Installing the MSX platform:** Includes installation of the MSX Base Infrastructure Services, microservices, Service Interface, and various applications and services. See [Installing the Platform](#).
 2. **Installing the Cloud Services Router Management hub.** See [Installing CSR Management Hub](#).
 3. **Installing the Service Packs:** Includes the deployment of Service Packs, Service Integration Framework (SIF), and a Data platform. See [Installing Service Packs](#).
- [Installing the Platform, on page 1](#)
 - [Installing the Cloud Services Router Management Hub , on page 8](#)
 - [Installing Service Packs, on page 10](#)

Installing the Platform

Installing the MSX Platform in Standalone Mode

Before You Begin:

1. Configure Docker. For more information, see [Preparing the Container](#).
2. Configure the main.yml file. For more information, see [Configuring the main.yml Bootstrap file](#).
3. Set the environment variable for the vault password to avoid manually entering the vault password for the ansible playbook installation. For more information, see [Configuring the Vault Password](#).

Step 1 Create target server infrastructure.

- a) Move to the ansible folder and invoke the given playbook to build the target server infrastructure:

```
cd /msx-4.0.0/ansible
export ANSIBLE_VAULT_PASSWORD_FILE=<path to file>
ansible-playbook create-infra.yml (OpenStack)
ansible-playbook create-infra-aws.yml (AWS)
```

Note Use the following command to decrypt and view the installation passwords:

```
ansible-vault --vault-password-file vault view group_vars/all/passwords.yml
```

If any error occurs in executing create-infra.yml, you need to run the ansible-playbook destroy-infra.yml (or destroy-infra-aws.yml) command first, and then rerun ansible-playbook create-infra.yml.

Warning: When you use any destroy playbook, it will delete all content within S3 or MinIO, which includes CockroachDB backups. Ensure that you have relocated those backups before running the destroy playbook if you need to restore your instance.

b) Verify the status after executing create infra playbook:

```
ansible-playbook checks/check-createinfra.yml
```

Step 2 (OpenStack only) After running the create-infra playbook, clear OS_PROJECT_NAME, OS_PROJECT_ID, and set OS_TENANT_NAME, OS_TENANT_ID in your installation environment. On the container, use the env command to get the values, then unset the OS_PROJECT variables and set the OS_TENANT values.

```
env
unset OS_PROJECT_NAME
unset OS_PROJECT_ID
export OS_TENANT_NAME=< os_project_name >
export OS_TENANT_ID=< os_project_id >
```

Step 3 Main.yml contains three proxy environment variables: https_proxy, http_proxy, and no_proxy. Keeping those in mind, you should configure your container's Linux proxy settings appropriately for your environment.

Step 4 Install the Kubernetes cluster. This installation takes around 40 minutes.

a) Invoke the playbook to install the Kubernetes cluster:

```
ansible-playbook isolated.yml
```

b) To verify the status of the Kubernetes cluster, use this command:

```
ansible kube-master -m command -a "kubectl get nodes"
```

Note If you get 'No resources Found' error, verify that you have added OS_TENANT_ID to your OpenStack RC file.

Step 5 (Optional) If you are installing the Datadog monitoring service to provide metrics for your infrastructure, make sure to follow the procedure in [Prerequisites for all Datadog Scenarios](#). Use this command:

```
ansible-playbook upload-datadog-images.yml
```

Step 6 Install base infrastructure services into Kubernetes.

a) Invoke the following playbook to install base infrastructure services in the Docker containers.

```
ansible-playbook deploy-infra-services.yml
```

b) To verify that all nodes are in the running state, use this command:

```
ansible-playbook checks/check-infra-services.yml
```

c) Verify that all infra services are running:

```
ansible kube-master -m command -a "kubectl get pod -n vms -o wide"
```

Step 7 Install the base MSX microservices and portal UI.

- a) Invoke the microservice playbook:

```
ansible-playbook deploy-vms-microservices.yml
```

- b) Invoke the action-orchestrator playbook:

```
ansible-playbook deploy-ao.yml
```

- c) Verify that all microservices are running:

```
ansible-playbook checks/check-vms.yml
```

What to do next

- [Installing CSR Hub](#)
- [Installing Service Packs](#)

Installing the MSX Platform in Dual Data Center Mode

Installing MSX in dual data center mode gives you the ability to switch over from an active data center to a passive data center. This ability to switch over ensures continuous operation with minimum downtime.

Before You Begin

We recommend that you complete the procedures listed in this section before proceeding to dual data center install procedure:

1. Reserve two floating IP addresses from the floating IP addresses pool per data center. For other prerequisites, see [Network Requirements](#).
2. Cisco MSX SD-WAN Service Pack-Specific Requirement: three NAT IP addresses for active data center and three NAT IP addresses for the passive data center will need to be opened for connectivity to vOrchestrator.
3. Configure Docker. For more information, see [Preparing the Container for Cisco MSX Installation](#).
4. Configure main.yml file. Configure the variables file before proceeding with the installation. Use this command to access the main.yml file:

```
vi /msx-4.0.0/ansible/group_vars/all/main.yml
```

The parameters from the main.yml file that must be configured while installing MSX in dual data center mode is given here. For a complete list of parameters with default values, see [Configuring the main.yml Bootstrap File](#).

```
dual_dc: yes
use_existing_ips: yes
dc: active
vms_subdomain: Register your subdomain. Must be unique in your domain.
```

5. Set the environment variable for the vault password to avoid manually entering the vault password for the ansible playbook installation. For more information, see [Configuring the Vault Password](#).

Setting up Dual Data Center Mode



Note A dual data center installation is complex and prone to errors if you do not perform it properly. As such, we do not recommend that you attempt this type of installation without the guidance of the Cisco MSX Operations team.

To set up MSX in dual data center mode:

Step 1

Log in to the installer container and:

- a) Verify if the given variables are updated in `group_vars/all/main.yml`:

```
dual_dc: yes
use_existing_ips: yes
dc: active
vms_subdomain:
# ELASTIC
elasticsearch_volume_size: 100
```

Note The default volume size is set to 100 and you may increase the value, if necessary.

```
csr_vpn_license: ChangeMe
```

For the `csr_vpn_license` variable, provide a license token from your smart account to license the CSRs, that are used to set up a tunnel across the two data centers.

- b) Update the given variables in `group_vars/all/external_addresses.yml`. Provide the reserved floating or elastic IP addresses for CSR VPN.

```
#csrvpn:
active:
local_vpn_ext_ip: Active dc FIP for csrvpn
remote_vpn_ext_ip: Passive dc FIP for csrvpn
local_vpn_ext_ip_2: Active dc FIP for second csrvpn
remote_vpn_ext_ip_2: Passive dc FIP for second csrvpn
```

- c) Source the OpenStack RC file that was copied for the active data center.

```
source vms-backup/infra/openrc-active
```

- d) Change directory to the `/msx-version/ansible` folder inside the container:

```
cd /msx-4.0.0/ansible
```

- e) Export the Ansible Vault password.

```
export ANSIBLE_VAULT_PASSWORD_FILE=<path to file>
```

- f) Create the target server infrastructure using the command:

```
ansible-playbook create-infra.yml --extra-vars '{dc: active}'
```

- g) Verify that the containers are reachable. The host can take some time to boot. To verify the host status, invoke the playbook:

```
ansible -m ping all
```

At this point, it is normal for all hosts to be unreachable except the inception VM that is used as a jumphost.

- h) Install the Kubernetes cluster and its supporting infrastructure. This step takes around 40 minutes.

```
ansible-playbook isolated.yml --extra-vars '{dc: active}'
```

- i) Invoke the playbook:

```
ansible-playbook deploy-dualdc.yml --tags csr-vpn,consul-backup-restore --extra-vars '{dc: active}'
```

Step 2

In the installer container, perform the following steps on the passive data center:

- a) Set the following variables in `group_vars/all/main.yml`:

```
dual_dc: yes
use_existing_ips: yes
dc: passive
vms_subdomain:
```

- b) Ensure that the given variables are updated in `group_vars/all/external_addresses.yml`. The elastic IP addresses are for the CSR VPN virtual machines. The settings given here ensure switching or failing-over to a working VPN tunnel.

```
#csrvpn:
passive:
  local_vpn_ext_ip: Passive dc FIP for csrvpn
  remote_vpn_ext_ip: Active dc FIP for csrvpn
  local_vpn_ext_ip_2: Passive dc FIP for second csrvpn
  remote_vpn_ext_ip_2: Active dc FIP for second csrvpn
# ELASTIC
elasticsearch_volume_size: 100
```

- c) Source the OpenStack RC file that was copied to the passive data center node.

```
source vms-backup/infra/openrc-passive
```

- d) Change directory to the `/msx-version/ansible` folder inside the container:

```
cd /msx-4.0.0/ansible
```

- e) Create the target server infrastructure on the second or the passive node:

```
ansible-playbook --skip-tags password,route53 create-infra.yml --extra-vars '{dc: passive}'
```

- f) Verify that the containers are reachable. The host can take some time to boot. To verify the host status, invoke the playbook:

```
ansible -m ping all
```

At this point, it is normal for all hosts to be unreachable except the inception VM that is used as a jumphost.

- g) Install the Kubernetes cluster and its supporting infrastructure. This step takes around 40 minutes.

```
ansible-playbook isolated.yml --extra-vars '{dc: passive}'
```

- h) Invoke the playbook:

```
ansible-playbook deploy-dualdc.yml --tags csr-vpn,consul-backup-restore --extra-vars '{dc: passive}'
sleep 40
ansible-playbook checks/check-csr-vpn.yml --extra-vars '{dc: passive}'
```

- Step 3** In the installer container, perform the given steps to switch to the active data center.

```
ansible-playbook dualdc-switch-dc.yml --extra-vars '{dc: active}'
source vms-backup/infra/openrc-active
ansible-playbook deploy-infra-services.yml --extra-vars '{dc: active}'
```

- Step 4** In the installer container, perform the given steps to switch to the passive data center.

```
ansible-playbook dualdc-switch-dc.yml --extra-vars '{dc: passive}'
source vms-backup/infra/openrc-active
ansible-playbook deploy-dualdc.yml --tags consul-cluster --extra-vars '{dc: passive}'
ansible-playbook deploy-infra-services.yml --extra-vars '{dc: passive}'
sleep 20
ansible-playbook checks/check-csr-vpn.yml --extra-vars '{dc: passive, with_cassandra: yes }'
```

- Step 5** In the installer container, perform the given steps to switch to the active data center and install all the service packs in the active data center.

```
ansible-playbook dualdc-switch-dc.yml --extra-vars '{dc: active}'
source vms-backup/infra/openrc-active
ansible-playbook deploy-vms-microservices.yml
ansible-playbook deploy-ao.yml
ansible-playbook deploy-csrhub.yml (OpenStack)
ansible-playbook deploy-csrhub-aws.yml (AWS)
ansible-playbook deploy-platform-nso.yml
```

- Step 6** Edit `/msx-4.0.0/ansible/group_vars/all/servicepack_name_variables.yml` for each service pack so that it matches the requirements of your deployment. The `servicepack_name` will be the same as that specified as an extra-vars parameter for the `deploy-service.yml` command.

- Step 7** Install your required service packs.

- a) Install the Managed Device service pack.

```
ansible-playbook deploy-service.yml --extra-vars service_name=manageddevice
```

Note For details about group variables required for the Managed Device installation, see [Installing the Cisco MSX Managed Device Service Pack](#).

- b) Install the SD-Branch service pack.

```
ansible-playbook deploy-service.yml --extra-vars service_name=vbranch
```

Note For details about the group variables required for the SD-Branch installation, see [Installing the Cisco MSX SD-Branch Service Pack](#).

- c) Install the SD-WAN service pack.

```
ansible-playbook deploy-service.yml --extra-vars service_name=sdwan
```

Note: For details about the group variables required for the Cisco MSX SD-WAN installation, see [Installing the Cisco MSX SD-WAN Service Pack](#).

Note Ensure that management hub is installed before deploying Managed Device and SD-Branch service packs. For details about setting up CSR Hub, refer [Installing CSR Hub](#).

Step 8

In the installer container, perform the following steps to switch to the passive data center and install all the service packs in the passive data center.

```
ansible-playbook dualdc-switch-dc.yml --extra-vars '{dc: passive}'
source /infra/openrc-passive
ansible-playbook deploy-vms-microservices.yml --extra-vars '{dc: passive}'
ansible-playbook deploy-ao.yml --extra-vars '{dc: passive}'
ansible-playbook deploy-csrhub.yml (OpenStack)
ansible-playbook deploy-csrhub-aws.yml (AWS)
ansible-playbook deploy-platform-nso.yml
```

Step 9

Edit `/msx-4.0.0/ansible/group_vars/all/servicepack_name_variables.yml` for each service pack so that it matches the requirements of your deployment. The `servicepack_name` will be the same as that specified as an extra-vars parameter for the `deploy-service.yml` command.

Step 10

Install your required service packs.

- a) Install the Managed Device service pack.

```
ansible-playbook deploy-service.yml --extra-vars service_name=manageddevice
```

- b) Install the SD-Branch service pack.

```
ansible-playbook deploy-service.yml --extra-vars service_name=vbranch
```

Note For details about the various group variables required for the SD-Branch installation, see [Installing the Cisco MSX SD-Branch Service Pack](#).

- c) Install the SD-WAN service pack.

```
ansible-playbook deploy-service.yml --extra-vars service_name=sdwan
```

Note For details about the various group variables required for the SD-WAN installation, see [Installing the Cisco MSX SD-WAN Service Pack..](#)

What to do next

Fail over to the passive data center in case of a disaster. To fail over, use the procedure in [Dual Data Center Disaster Recovery](#).

Installing the Cloud Services Router Management Hub

You can use Cisco Cloud Service Router (CSR) as a management hub in Cisco Managed Services Accelerator (MSX).

Before You Begin

- (For OpenStack) Download the CSR1000v (csr1000v-universalk9.16.12.01a.qcow2) image file from [HERE](#).
- (For AWS) Use cisco-CSR-.16.09.02-BYOL-HVM*. This image is in the Amazon marketplace and it is specified in main.yml.
- MSX requires at least 100 MB throughput on the CSR.
- Set the environment variable for the vault password to avoid having to enter the vault password for each playbook installation. See [Configuring the Vault Password](#).
- Install MSX. See [Installing the Platform in Standalone Mode](#).

Installing CSR Management Hub on OpenStack

Step 1 Add the CSR Image to OpenStack using glance:

```
glance image-create --name csr1000v-universalk9.16.12.01a.qcow2 --disk-format=qcow2
--container-format=bare --visibility public --file [qcow2filename] --progress
```

Step 2 Change directory to the /msx-version/ansible folder inside the container:

```
cd /msx-4.0.0/ansible
source <openrc>
export ANSIBLE_VAULT_PASSWORD_FILE=<path to file>
```

Step 3 Verify or update the given variables in group_vars/all/main.yml:

a)

```
openstack:
  # dns_servers and ntp_servers need to be listed
  dns_servers:
    - ChangeMe
  ntp_servers:
    - ChangeMe
  volume_device: /dev/vdb
  flavors:
    csr_hub: small.csr1000v
```

b) Update the CSR Hub license token and platform throughput (if applicable).

```
# Smart lic token for CSR HUB
# Provide lic token from your smart account to license the CSR HUB

csr_hub_license: ChangeMe

#Platform hardware throughput is set to 100MB by default, ensure that smart license account has
licenses to authorize this throughput.
```

```
csr_hub_throughput: 100
```

- c) Ensure that flavor with name ‘small.csr1000v’ and with the given specifications exists in OpenStack.

```
name: small.csr1000v
ram: 4096
vcpus: 2
disk: 0
is_public: true
```

- d) Set `use_existing_ips` to `yes`, if you have preassigned floating IP address in the project for CSR Hub. If its value is not specified, a floating IP address is automatically assigned to CSR hub by the installer.

```
In order to use the same public IPs across installs, set use_existing_ips
# to yes, place external IP addresses to be used in ./external_addresses.yml
# Ensure the external IP addresses you specify are FIPs/EIPs which are
# available in your project/VPC.
#
use_existing_ips: yes
```

- e) Specify the CSR floating IP address in `group_vars/all/external_addresses.yml`, if you are using a pre-defined floating IP address.

```
csrhub_ext_ip: x.x.x.x
```

Step 4 Run the playbook to deploy CSR Hub:

```
ansible-playbook deploy-csrhub.yml
```

Step 5 Use SSH to connect to the CSR Hub to verify connectivity:

```
ssh -F ssh.cfg csr_admin@10.32.1.200
```

Provide `csr_admin` password when prompted. That can be obtained from `passwords.yml`

```
ansible-vault view group_vars/all/passwords.yml
```

Note The IP address 10.32.1.200 applies only for OpenStack deployment. For AWS deployment, the IP address would be different (differs for each deployment).

Step 6 Run the ansible playbook.

```
ansible-playbook checks/check-csrhub.yml
```

Step 7 Deploy the platform Network Service Orchestrator.

```
ansible-playbook deploy-platform-nso.yml
```

Installing CSR Management Hub on AWS

Use the following procedure to install and verify the CSR Management Hub on AWS.

Step 1 Change directory to the /msx-version/ansible folder inside the container. For example: `cd /msx-4.0.0/ansible`.

Step 2 Export the ANSIBLE_VAULT_PASSWORD_FILE main.yml variable to the path of the password file.

```
export ANSIBLE_VAULT_PASSWORD_FILE=<path_to_file>
```

Step 3 Verify that `use_existing_ips` is set to "yes" in main.yml. For example:

```
# In order to use the same public IPs across installs, set use_existing_ips
# to yes, place external IP addresses to be used in ./external_addresses.yml
# Ensure the external IP addresses you specify are FIPs/EIPs which are
# available in your project/VPC.
#
use_existing_ips: yes
```

Step 4 Set the Smart License key for the CSR Hub in main.yml. For example:

```
csr_hub_license: ChangeMe
```

Step 5 Run the playbook to deploy the CSR Management Hub.

```
ansible-playbook deploy-csrhub-aws.yml
```

Step 6 Use SSH to connect to the CSR Management Hub to verify connectivity.

```
ssh -F ssh.cfg csr_admin@csr_mgmt_hub_IP
```

Provide the `csr_admin` password when prompted. You can obtain the password from `passwords.yml`, which is located at: `ansible-vault view group_vars/all/passwords.yml`

Step 7 Run a check of the CSR Management Hub.

```
ansible-playbook checks/check-csrhub.yml
```

Step 8 Deploy the platform Network Service Orchestrator.

```
ansible-playbook deploy-platform-nso.yml
```

What to do next

[Installing Service Packs](#). Use the procedures in this section to install service packs such as Cisco MSX SD-Branch, SD-WAN, and Managed Device.

Installing Service Packs

MSX provides out-of-the-box capabilities, also called as service packs (such as Cisco MSX SD-Branch, SD-WAN, and Managed Device), to orchestrate particular use cases. Depending on your requirements, you can install either a single service pack or a combination of these service packs.



Note Ensure that the CSR Management Hub is installed and available before installing your Service Packs.



Note For the Cisco MSX SD-WAN service pack, installation of Cisco MSX SD-Branch service pack is a pre-requisite. Ensure that you have installed Cisco MSX SD-Branch before you install Cisco MSX SD-WAN.



Note MSX Release 4.0.0 has a dependency on the Managed Device service pack. You must install Managed Device service pack before you install SD-Branch because SD-Branch leverages some of the functionalities of Managed Device.

Service packs must be installed after installing the platform components. For more information, see [Installing Service Packs](#).



Note When more than one service pack is deployed on Cisco MSX, ensure that the PnP management address pool is unique. For example, when the Managed Device service pack is running on a single MSX, the management address pool configuration in NSO CLI should be set as:

```
set resource-pools ip-address-pool managed-device-pool subnet 10.255.0.0 16
```

Here 10.255.0.0 is an example of the subnet from which Tunnel0 management address is allocated and configured on each device. This subnet will be used to push the Day0 and Day1 configurations from NSO to the device.

Installing the Cisco MSX Managed Device Service Pack

Before you begin

- Install Cisco Cloud Services Router CSR1000v and use it as a management hub on MSX. See [Installing CSR Hub](#).
- Set the environment variable for the vault password to avoid having to enter the vault password for each playbook installation. See [Configuring the Vault Password](#).
- Access the container and list the folders in the container. You can skip this step, if you are already inside the container. The command to access the container is:

```
docker exec -it <container-name> bash
ls
```

Step 1

Configure the Managed Device variables file (group_vars/all/manageddevice_variables.yml) to match the requirements of your deployment.

- Network administrators will need to partition their subnets to prevent overlapping of the NSO shards.

```
PNP_MGMT_ADDR_SUBNET_MASK_LIST_MANAGEDDEVICE: ['10.254.0.0/24', '10.254.8.0/24']
```

The number of NSO shards is two by default. For information on adding additional shards, see [Scaling Managed Device NSOs](#).

- Configure the following variables:

`DEV_MGMT_HUB1`: This value is used to add the CSR Hub's public IP into the NSO configuration.

`DEV_MGMT_HUB2`: This value is used to set the CSR Hub's public IP into the NSO configuration for dual data center deployments.

Step 2 Deploy the Cisco MSX Managed Device service pack using the command:

Example:

```
ansible-playbook deploy-service.yml --extra-vars service_name=manageddevice
```

What to do next

After deploying the Cisco MSX Managed Device service pack, you must import the customer configuration template XML files into NSO. These template XMLs are pushed from MSX to customer devices as part of the orchestration of device configuration. For more details, refer to the 'Importing the Template XML File into NSO' section in the *Cisco Managed Services Accelerator (MSX) Managed Device Service Pack User Guide*.

Installing the Cisco MSX SD-Branch Service Pack

Before you begin

- Install Cisco Cloud Services Router CSR1000v and use it as a management hub on MSX. See [Installing CSR Hub](#).
- Set the environment variable for the vault password to avoid having to enter the vault password for each playbook installation. See [Configuring the Vault Password](#).
- Access the container and list the folders available in this container. You can skip this step, if you are already inside the container. Use this command to access the container:

```
docker exec -it <container-name> bash
ls
```

Step 1 Configure the SD-Branch variables file (`group_vars/all/vbranch_variables.yml`) to match the requirements of your deployment.

- **Static Route configurations:** Static routes are required for Management Tunnels and Loopback Addresses to use CSR MGMT Hub as the next hop or the gateway. Update the values for the variables listed below:

`VBRANCH_NFVIS_MGMT_POOL_NET`: This is the subnet that needs the next hop/gateway to be CSRHUB IP.

`VBRANCH_NFVIS_MGMT_POOL_NET_MASK`: The size/mask of the `VBRANCH_NFVIS_MGMT_POOL_NET` network.

- **Single IP Variables:** SD-Branch and NFVIS provide you with the capability to on-board a device in a Single IP mode. This means that a VNF on the device will assume the Single IP. For example, SD-Branch publishes a sample template that can be used to demonstrate a single IP deployment with an ISRv. The ISRv template requires 2 specific day0 files (one IOSXE and one cloud init – ovf env) that is obtained independently. Those day0 files should be

uploaded to an http server that can be accessed by the device. The variables given here are used to specify the URL of the day0 variables.

```
ISRV_SIP_DAY0_IOSXE_URL: <webserver url location of the iosxe day0 config for ISRV single ip>  
ISRV_SIP_DAY0_OVF_URL: <webserver url location of the ovf_env.xml for ISRV single ip>
```

Note The above variables are utilized only by Devnet sample templates that are published by SD-Branch.

Step 2 Navigate to the MSX 4.0.0 folder from the list of folders within the container:

```
cd /msx-4.0.0/ansible
```

Step 3 Deploy the SD-Branch service pack using the command:

```
ansible-playbook deploy-service.yml --extra-vars service_name=vbranch
```

Note After deploying Cisco MSX SD-Branch, you must also install the Template Development Environment (TDE) application. Network engineers can use this desktop application to create Cisco MSX SD-Branch branch templates and network topology diagrams. Currently, TDE is only supported on mac OS and Linux. For more information on how to install TDE, see the Readme document available with the MSX binaries on [cisco.com](https://www.cisco.com).

Setting Up Initial Configuration on the ENCS CPE

Specify the given configuration details (first time use only) on the ENCS CPE:

- PNP server IP address
- PNP server port
- Transport as HTTPS
- Upload the cacert.pem file

To configure these parameters on individual CPEs:

-
- Step 1** Log in to the NFVIS portal for the CPE.
 - Step 2** From the left pane, choose **Host, Plug-n-Play**. The Plug-n-Play screen appears.
 - Step 3** Click **Edit**.
 - Step 4** Enter the PNP server IP address.
 - Step 5** Set the PNP server port to 8443.
 - Step 6** Select HTTPS for the transport.
 - Step 7** To upload the cacert.pem file, click **Choose File** (adjacent to Upload Certificate File) and select the file.
 - Step 8** Click **Save**.
-

Installing the Cisco MSX SD-WAN Service Pack

Before you begin

- Install the SD-Branch service pack before installing the SD-WAN service pack. For more information, see [Installing the Cisco MSX SD-Branch Service Pack](#).
- If you have an SD-WAN deployment with vManage connected, you must copy your external certificates and import them into the centralized MSX keystore. For more information, see [Adding External Certificates to MSX](#).
- Set the environment variable for the vault password to avoid having to enter the vault password for each playbook installation. See [Configuring the Vault Password](#).
- Access the container and list the folders available in the container. You can skip this step, if you are already inside the container. The command to access the container is:

```
docker exec -it <container-name> bash
ls
```

Step 1 Change directory to the /msx-version/ansible folder inside the container:

```
cd /msx-4.0.0/ansible
```

Step 2 Deploy the Cisco MSX SD-WAN service pack using the command:

```
ansible-playbook deploy-service.yml --extra-vars service_name=sdwan
```

Installing Cisco MSX Enterprise Access Service Pack

Before you begin:

- Set the environment variable for the vault password to avoid manually entering the vault password for the ansible playbook installation. For more information, see [Configuring the Vault Password](#) section in the *Cisco MSX 4.0.0 Install and Upgrade Guide*.

Step 1 Access the container and list the folders available in this container. You can skip this step, if you are already inside the container. Use the following command to access the container.

```
docker exec -it <container-name> bash
ls
```

Step 2 Change directory to the Ansible folder.

```
cd /msx-4.0.0/ansible
```

Step 3 Deploy the Cisco MSX Enterprise Access service pack using the following command:

```
ansible-playbook deploy-service.yml --extra-vars service_name=sda
```

Verifying Enterprise Access Installation

Use the following steps to verify the Enterprise Access installation.

Step 1 Navigate to kube-master

```
root@{containerid}:/msx-4.0.0/ansible# ssh -i keys/id_rsa centos@{inception-ip}
[centos@inception-xyz ~]$ ssh -i keys/id_rsa centos@kubernetes-master-xyz-1
[centos@kubernetes-master-xyz-1 ~]$
```

Step 2 Make sure that the Enterprise Access containers are running:

```
[centos@kubernetes-master-xyz-1 ~]$ sudo kubectl get pod | grep sdaservice
sdaservice-7dz6t 2/2 Running 0 3d17h
sdaservice-rxrg4 2/2 Running 0 3d17h
```

Step 3 Make sure that the DNA Center metrics and health collector are running:

```
[centos@kubernetes-master-xyz-1 ~]$ sudo kubectl get pod | grep dnacbeat
dnacbeat-0 2/2 Running 0 6d21h
dnacbeat-1 2/2 Running 0
```

Step 4 Make sure that the Enterprise Access user interface is deployed:

```
[centos@kubernetes-master-xyz-1 ~]$ ls -l /data/vms/skyfallui/services/sda/
total 1632
drwxr-xr-x. 19 root root 4096 Sep 30 13:24 components
drwxr-xr-x. 2 root root 4096 Sep 30 13:24 i18n
drwxr-xr-x. 3 root root 4096 Sep 30 13:24 images
drwxr-xr-x. 5 root root 4096 Sep 30 13:24 modals
drwxr-xr-x. 3 root root 4096 Sep 30 13:24 operator
-rw-r--r--. 1 root root 320 Sep 30 13:24 order.html
-rw-r--r--. 1 root root 29138 Sep 30 13:24 sda.css
-rw-r--r--. 1 root root 32180 Sep 30 13:24 sda.js
-rw-r--r--. 1 root root 1287905 Sep 30 13:24 sda.js.map
```
