Cisco Optical Network Controller Installation Guide, Releases 26.x.x

Cisco Optical Network Controller
Updated May 7, 2026

Topics included

1 Install and deploy standalone Cisco Optical Network Controller

Topics:

Explains standalone Cisco Optical Network Controller installation using VMware vSphere, including environment preparation, SSH key generation, OVF deployment, access verification, service packs, upgrades, and reverts.

Install Cisco Optical Network Controller in a standalone mode

Use this procedure to install Cisco Optical Network Controller on VMware vSphere, including installation requirements, network and ESXi preparation, SSH key generation, OVF deployment, credentials, NTP, and Web UI access.

Install Cisco Optical Network Controller on VMware vSphere by completing the deployment, verification, and access subtasks in order.

Procedure

1. Start the VMware deployment by completing Select the OVF template and prepare the VMware deployment.
2. Customize the template and finish the deployment by completing Customize the template and finish the VMware deployment.
3. Power on the VM and connect through SSH by completing Power on the VM and connect through SSH.
4. Verify the system and configure the initial credentials by completing Verify the system and configure initial credentials.
5. Configure NTP and access the Web UI by completing Configure NTP and access the Cisco Optical Network Controller Web UI.

Cisco Optical Network Controller is installed on VMware vSphere and ready for use.

Installation requirements

This section provides details about the prerequisites, system requirements, network requirements, and port assignments for Cisco Optical Network Controller installation.

This topic lists the requirements that you must review before you install Cisco Optical Network Controller.

VMware platform requirements

The VMware platform requirements are:

- VMware vCenter Server version 7.0 or later.
- VMware vSphere server and client version 7.0 or later.
- ESXi host version 7.0 or later on the servers that host the virtual machines.

> **Attention**
>
> Upgrade to VMware vCenter Server 8.0 U2 if you are using VMware vCenter Server 8.0.2 or VMware vCenter Server 8.0.1.

Cisco Optical Network Controller is deployed on rack or blade servers within vSphere.

Note

For more details on VMware vSphere, see VMware vSphere.

Supporting service requirements

The supporting service requirements are:

- DNS server: The DNS server can be an internal DNS server if the Cisco Optical Network Controller instance is not exposed to the internet.
- NTP server or NTP pool: Use the same NTP source on Cisco Optical Network Controller, the client PC or VM, and the ESXi host.

Network requirements

The network requirements are:

- Before installation, three networks must be created.
- **Control Plane Network**: Used for internal communication between deployed virtual machines within a cluster. In a standalone deployment, this network can be a private network. In a high availability cluster, create this network between the servers where each HA node is deployed.
- **VM Network or Northbound Network**: Used for communication between users and the cluster. This network carries public traffic and hosts the UI.
- **Eastbound Network**: Used for internal communication between deployed virtual machines within a cluster. In a standalone deployment, this network can be a private network.

Preparation task

ESXi hosts use a self-signed certificate by default. Before deployment, ensure that browser certificate warnings are accepted and the required VMware networks are prepared.

For the procedure, see Prepare VMware networking and ESXi access.

Hardware requirements and sizing notes

This table lists the minimum hardware requirements for the base profile with daily retention only.

Table 1: Standalone mode hardware requirements for base profile with daily retention

Sizing	CPU	Memory	Solid State Drive (SSD)
Extra Small (XS)	16 vCPU	64 GB	1 TB
Small (S)	32 vCPU	128 GB	2.5 TB
Medium (M)	48 vCPU	256 GB	6 TB

This table lists the minimum hardware requirements for the extended profile with weekly and monthly retention.

Table 2: Standalone mode hardware requirements for extended profile with weekly and monthly retention

Sizing	CPU	Memory	Solid State Drive (SSD)
Extra Small (XS)	16 vCPU	64 GB	1.5 TB
Small (S)	32 vCPU	128 GB	5 TB
Medium (M)	48 vCPU	256 GB	12 TB

Use SSDs that meet the disk write latency requirement of 100 ms or lower.

> **Attention**
>
> Cisco Optical Network Controller supports only SSDs for storage.

The hardware sizing notes are:

- Configure vCPU and memory according to the VM profile before you power on the VM in vCenter. XS uses 16 vCPU and 64 GB. S uses 32 vCPU and 128 GB.
- The supported vCPU-to-physical-core ratio is 2:1 if hyperthreading is enabled and supported by the hardware. Otherwise, the ratio is 1:1.

Deployment-specific requirements

This table lists the deployment-specific requirements.

Table 3: Deployment requirements

Deployment type	Requirements
Standalone (SA)	Control Plane Network: Can be a private network. Requires one IP address. Gateway required. DNS server required. Northbound Network (VM Network): Must be a public network. Requires one public IP address. Gateway required. DNS server required. Eastbound Network: Can be a private network. Requires one private IP address. Gateway required. DNS server required.
Highly Available (HA)	**Control Plane:** Three IP addresses for the individual nodes. This can be a private network. **VM Network:** Four IP addresses. This must be a public network, with three IP addresses for node management and one virtual IP for northbound communication and UI access.

> **Note**
>
> For a high availability deployment, nodes on different ESXi hosts should have a minimum link bandwidth of 10G between them to ensure efficient data communication and synchronization.

Communications matrix

This table lists the default port assignments.

Table 4: Communications matrix

Traffic type	Port	Description
Inbound	TCP 22	SSH remote management
	TCP 8443	HTTPS for UI access
Outbound	TCP 22	NETCONF to routers
	TCP 389	LDAP if using Active Directory
	TCP 636	LDAPS if using Active Directory
	Customer specific	HTTP for access to an SDN controller
	User specific	HTTPS for access to an SDN controller
	TCP 3082, 3083, 2361, 6251	TL1 to optical devices
Eastbound	TCP 10443	Supercluster join requests
	UDP 8472	VxLAN

Prepare VMware networking and ESXi access

Use this procedure to accept the ESXi certificate warning and prepare the VMware networking required for Cisco Optical Network Controller installation.

Prepare the VMware environment for installation by accepting ESXi browser certificate warnings and creating the required VMware networks.

Complete this task before you start the Cisco Optical Network Controller deployment on VMware vSphere.

Procedure

1. Access the ESXi host by using a web browser.
2. If the browser displays a security warning that the connection is not private or the certificate is not trusted, accept the risk or bypass the warning so that you can continue.

   ESXi hosts use a self-signed certificate by default. Accepting the certificate allows you to continue with template customization, VM deployment, and login during the Cisco Optical Network Controller installation process.
3. From the vSphere client, select the datacenter where you want to add the ESXi host.

4. Right-click the server from the vCenter inventory and click **Add Networking**.
5. To create a private network for the Control Plane and Eastbound Networks, follow the Standard Switch wizard for each network.
   a) In **Select connection type**, choose **Virtual Machine Port Group for a Standard Switch** and click **Next**.
   b) In **Select target device**, select **New Standard Switch (MTU 1500)** and click **Next**.
   c) In **Create a Standard Switch**, click **Next** and confirm that there are no active physical network adapters for the switch.
   d) In **Connection settings**, choose a network label such as Control Plane or Eastbound, select VLAN ID as None (0), and click **Next**.
   e) In **Ready to complete**, review the configuration and click **Finish**.
6. Create the Control Plane, Northbound, and Eastbound Networks before deployment.

The ESXi host is accessible, and the required VMware networks are prepared for the Cisco Optical Network Controller deployment.

Installation considerations

Outlines installation considerations, including guidance and constraints to review before Cisco Optical Network Controller installation, upgrade, deployment, or recovery workflows.

Based on your choice and availability of disk space, select the data source.

OVA template supports pem key authentication.

The initiator node is the initiator parameter that needs to be set to true on only the primary node, which initializes the creation of a cluster as a primary node while the other nodes can join the network as secondary or tertiary.

In the HA environment, Cisco Optical Network Controller creates two NETCONF sessions, one after another immediately. Since by default NCS 1010 only allows one session per second, for the HA onboarding to be successful for NCS 1010 the CLI **ssh server rate-limit 600** must be added in the NCS 1010 configuration.

Generate an SSH ed25519 key

Use this procedure to generate an SSH ed25519 key. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Generate an SSH ed25519 key pair for Cisco Optical Network Controller access.

Cisco Optical Network Controller requires an ed25519 key for SSH access. The ed25519 key is different from an RSA key.

Before you begin

Ensure that the system where you generate the key has the **ssh-keygen** utility available.

Follow these steps to generate an SSH ed25519 key.

Procedure

---

1. Run the **ssh-keygen -t ed25519** command.

   Example

   ```
   ssh-keygen -t ed25519
   ```

2. When prompted, enter the filename for the key and the passphrase.

   Example

   ```
   Enter file in which to save the key (/Users/xyz/.ssh/id_ed25519):
   ./<file-name-of-your-key>.pem
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   ```

3. Verify that the private key and public key files are created.

   Example

   ```
   Your identification has been saved in ./<file-name-of-your-key>.pem
   Your public key has been saved in ./<file-name-of-your-key>.pem.pub
   ```

4. Display the public key and copy it for use in the deployment template.

   Example

   ```
   cat <file-name-of-your-key>.pem.pub
   ```

   Use the displayed public key in the **SSH Public Key** field during deployment.

The SSH ed25519 key pair is generated, and the public key is ready to use in the deployment template.

Select the OVF template and prepare the VMware deployment

Use this procedure to select the OVF template and prepare the VMware deployment. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Start the OVF deployment in VMware vSphere and select the initial deployment resources.

Before you begin

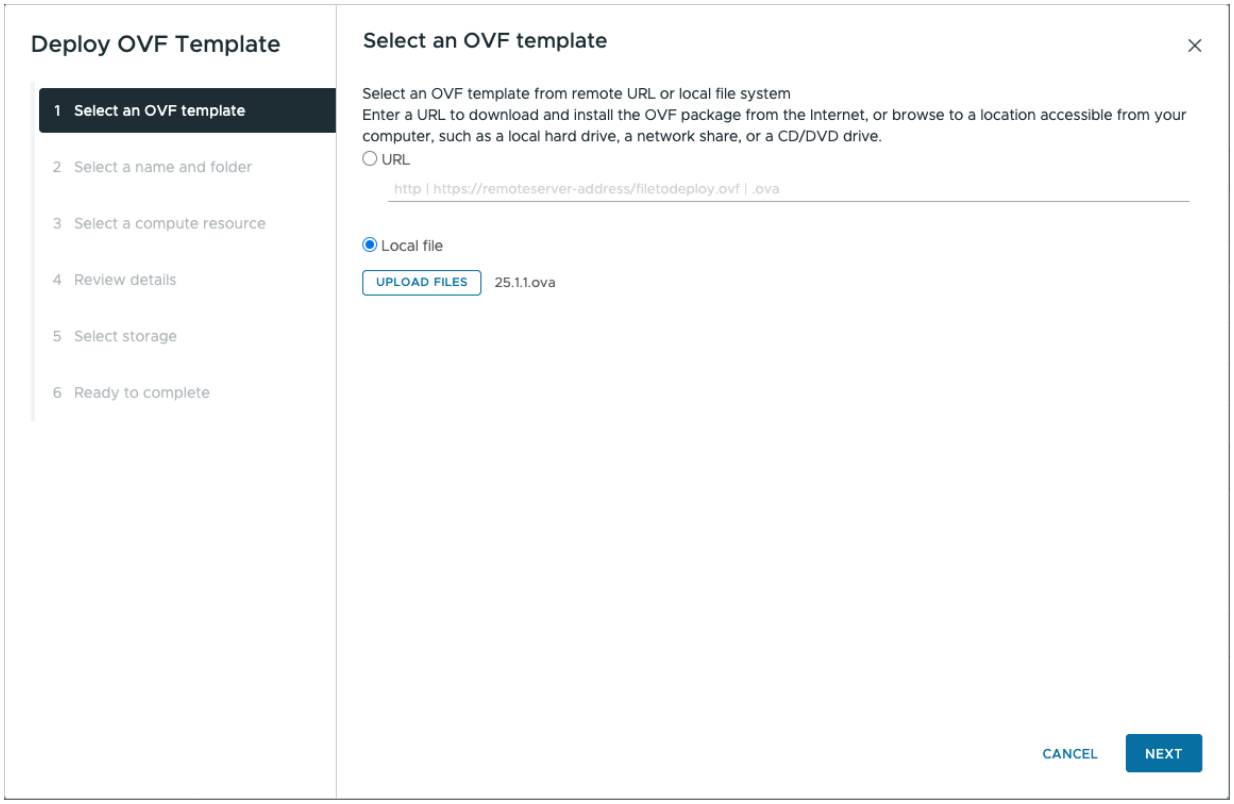
> **Note**
>
> During the OVF deployment, the deployment gets aborted if there is an internet disconnection.

Procedure

---

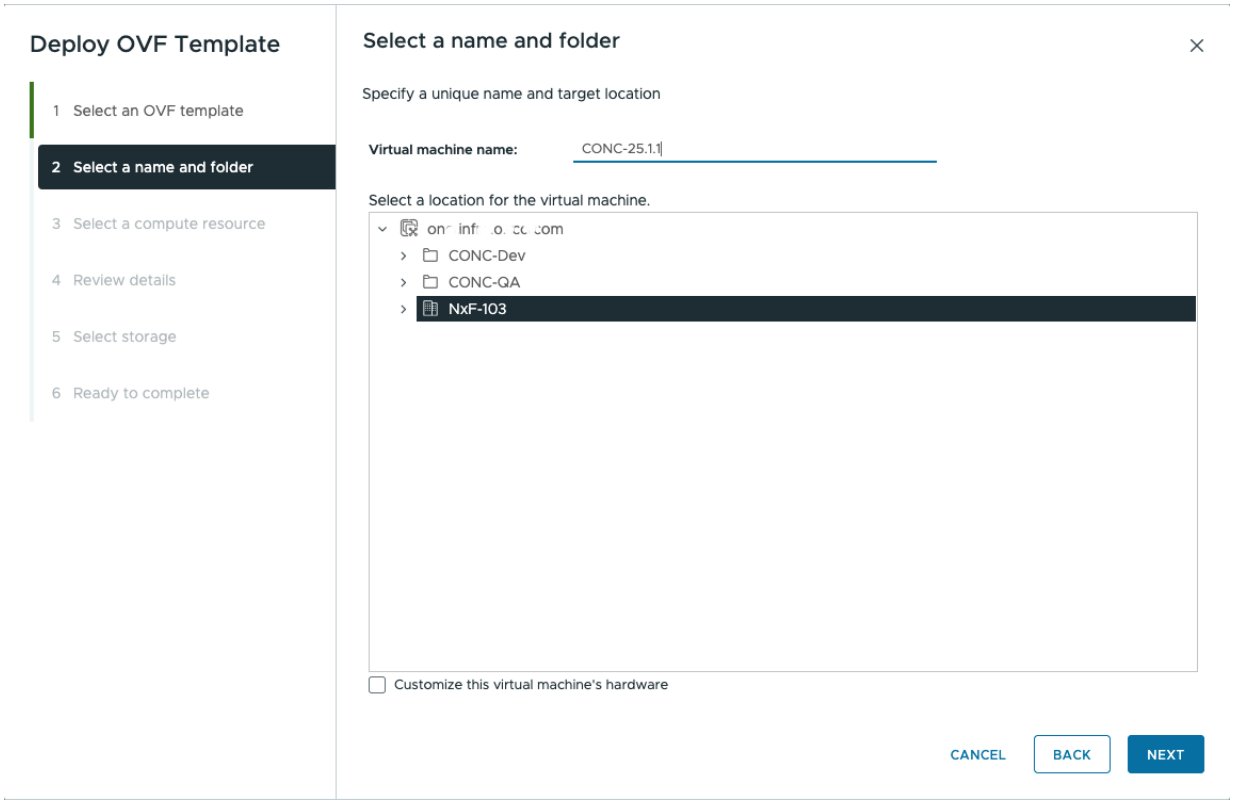
1. Right-click the ESXi host in the vSphere client screen and click **Deploy OVF Template**.
2. In the **Select an OVF template** screen, select the **URL** radio button for specifying the URL to download and install the OVF package from the Internet or select the **Local file** radio button to upload the downloaded OVA files from your local system, and click **Next**.

   Figure 1: Select an OVF Template



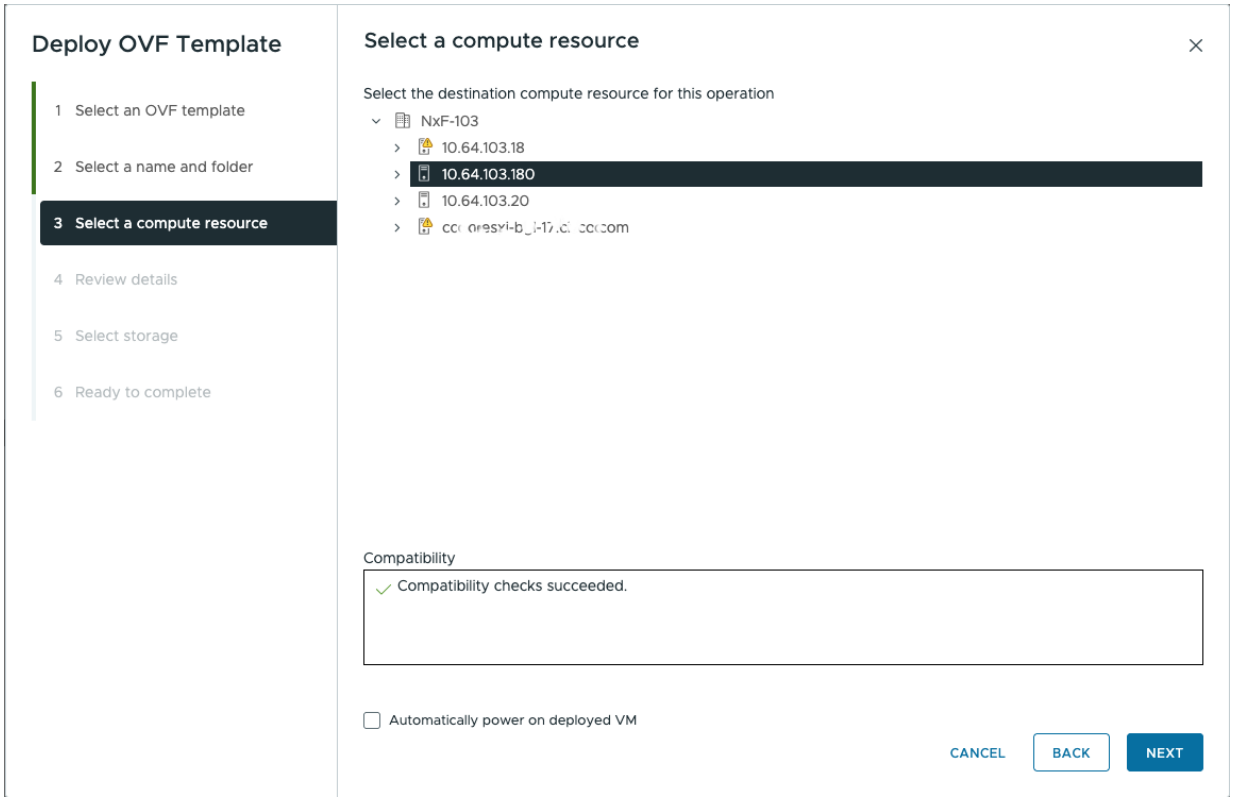
3. In the **Select a name and folder** screen, specify a unique name for the virtual machine instance. Cisco Optical Network Controller can be deployed as standalone or high availability. Select the location of the VM from the list of options for standalone or high availability (primary, secondary, or tertiary) and click **Next**.

Figure 2: Select a name and folder



4. In the **Select a compute resource** screen, select the destination compute resource on which you want to deploy the VM and click **Next**.

Figure 3: Select a Compute Resource



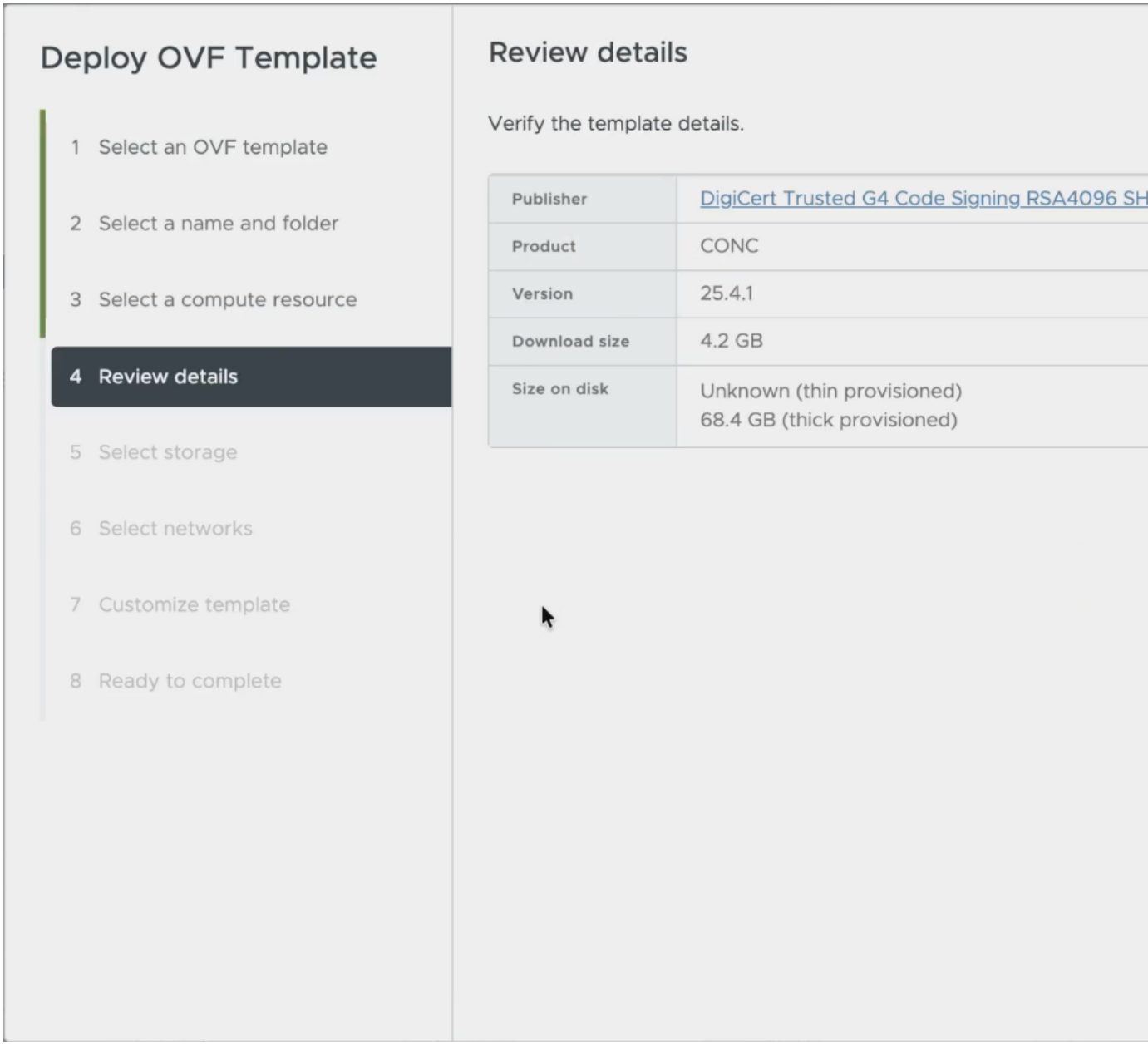
Note

While selecting the compute resource the compatibility check proceeds till it completes successfully.

5. In the **Review details** screen, verify the template details and click **Next**.

Figure 4: Review Details

Figure 5: Review Details



The **Publisher** field shows the trusted publisher information. For example, *DigiCert Trusted G4 Code Signing RSA4096 SHA384 2021 CA1 (Trusted certificate)*.

The initial OVF deployment setup is complete, and you can continue with storage, network, and template customization.

Customize the template and finish the VMware deployment

Use this procedure to customize the template and finish the VMware deployment. It helps you complete the related installation workflow and confirm the deployment state before continuing.

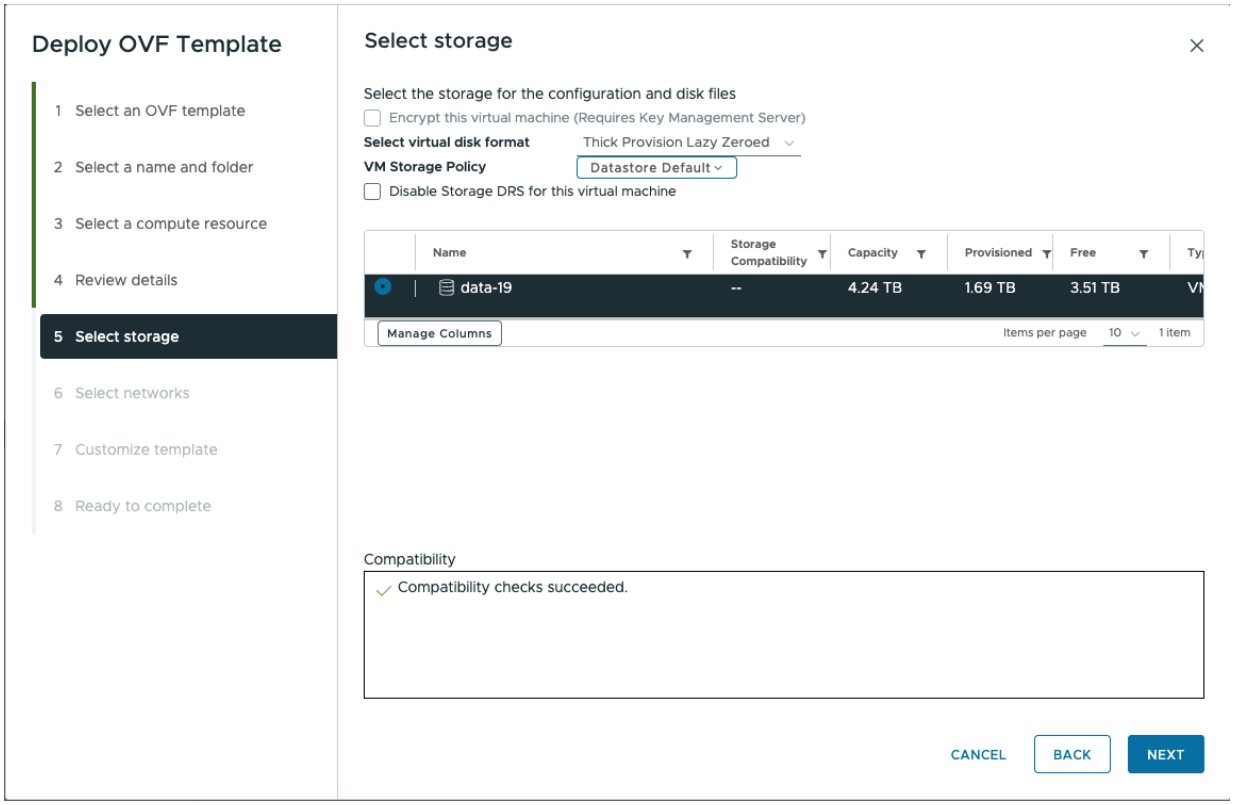
Complete the remaining VMware deployment configuration, customize the template values, and finish the OVF deployment.

Procedure

1. In the **Select storage** screen, select the virtual disk format based on provision type requirement, set **VM Storage Policy** as *Datastore Default*, select the **virtual disk format** as *Thin Provision*, and click **Next**.

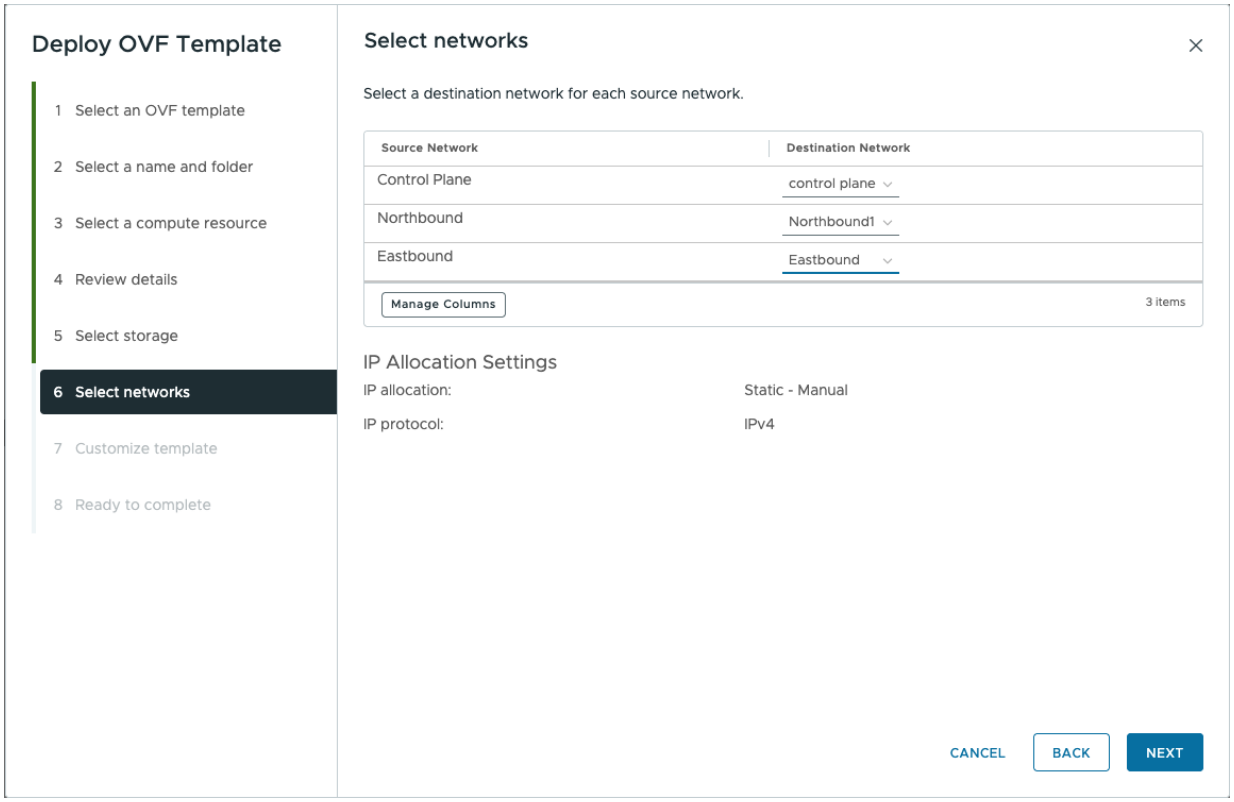
   You must select "Thin provision" as the virtual disk format.

   Figure 6: Select Storage



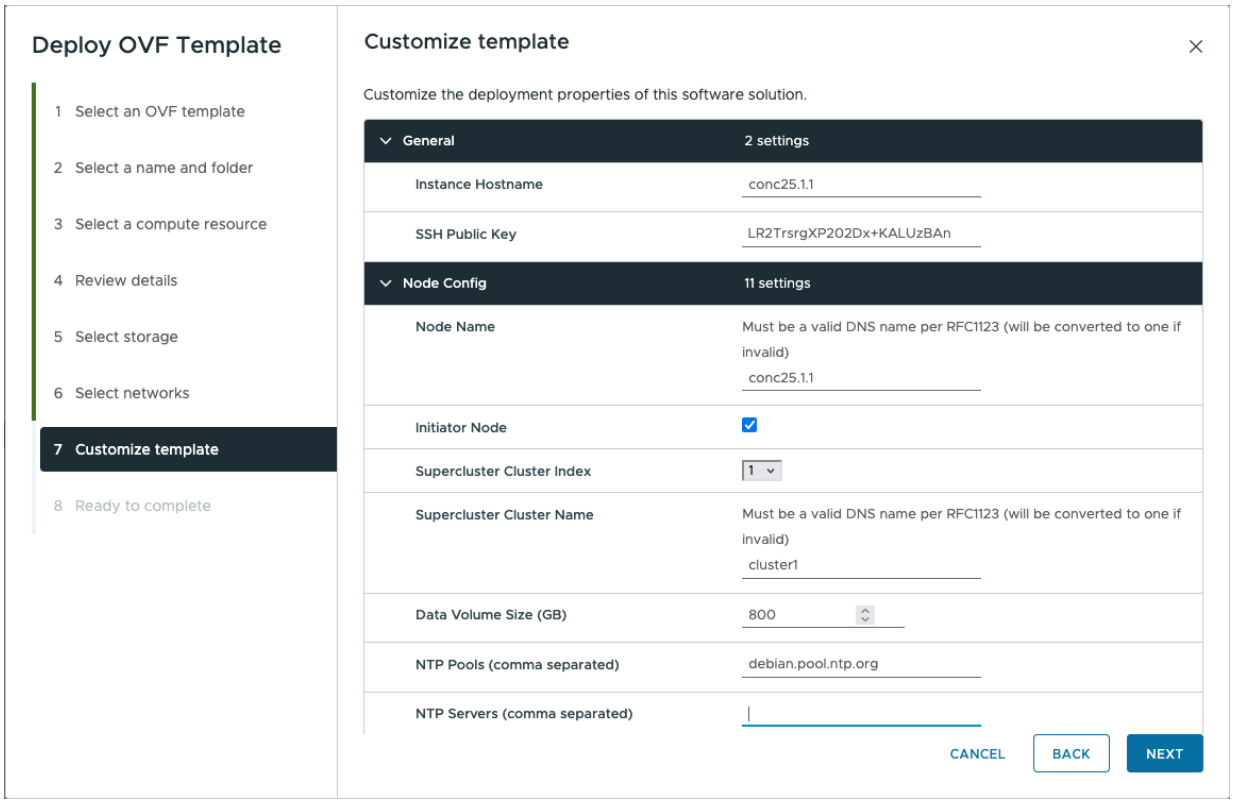
2. In the **Select networks** screen, select the control and management networks as **Control Plane**, **Eastbound**, and **Northbound** from the networks created earlier and click **Next**.

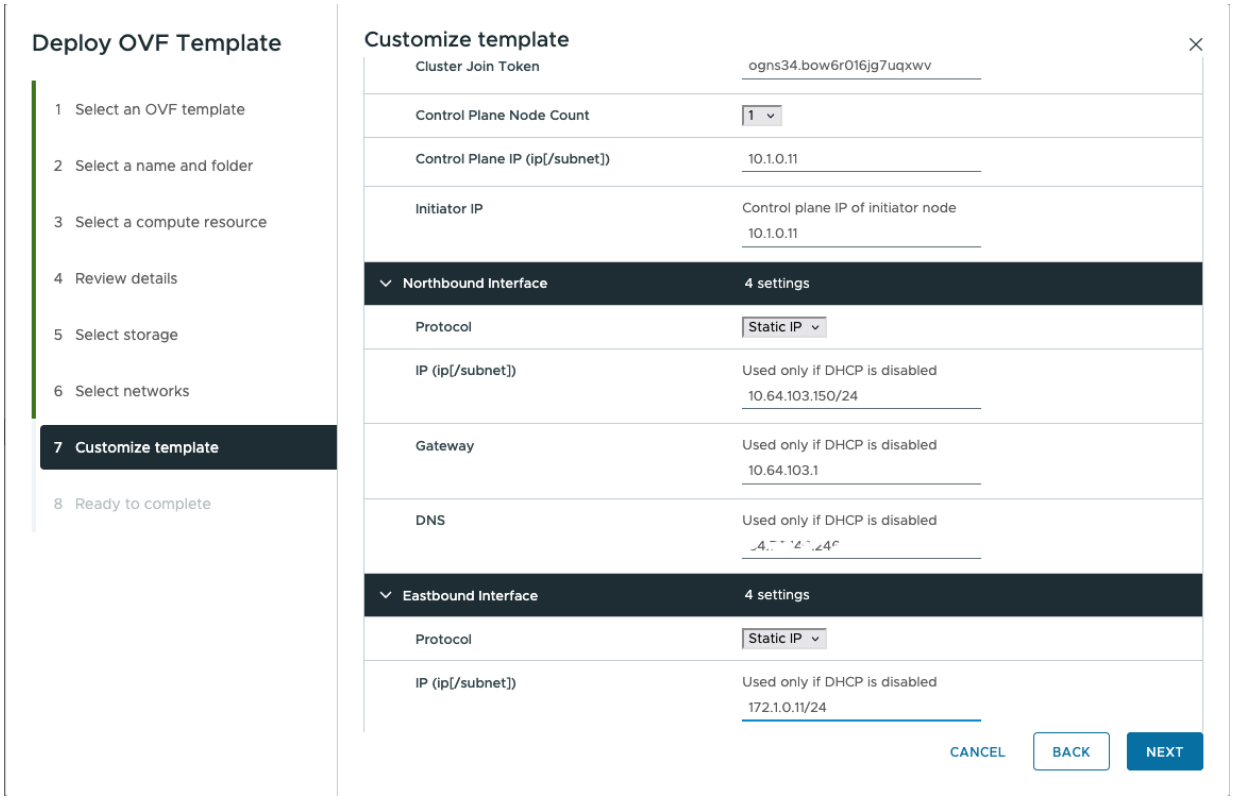
   Figure 7: Select Networks



3. In the **Customize template** screen, set the values using the Table 5: Customize template parameters on page 16 table as a guideline for deployment.

Figure 8: Customize Template





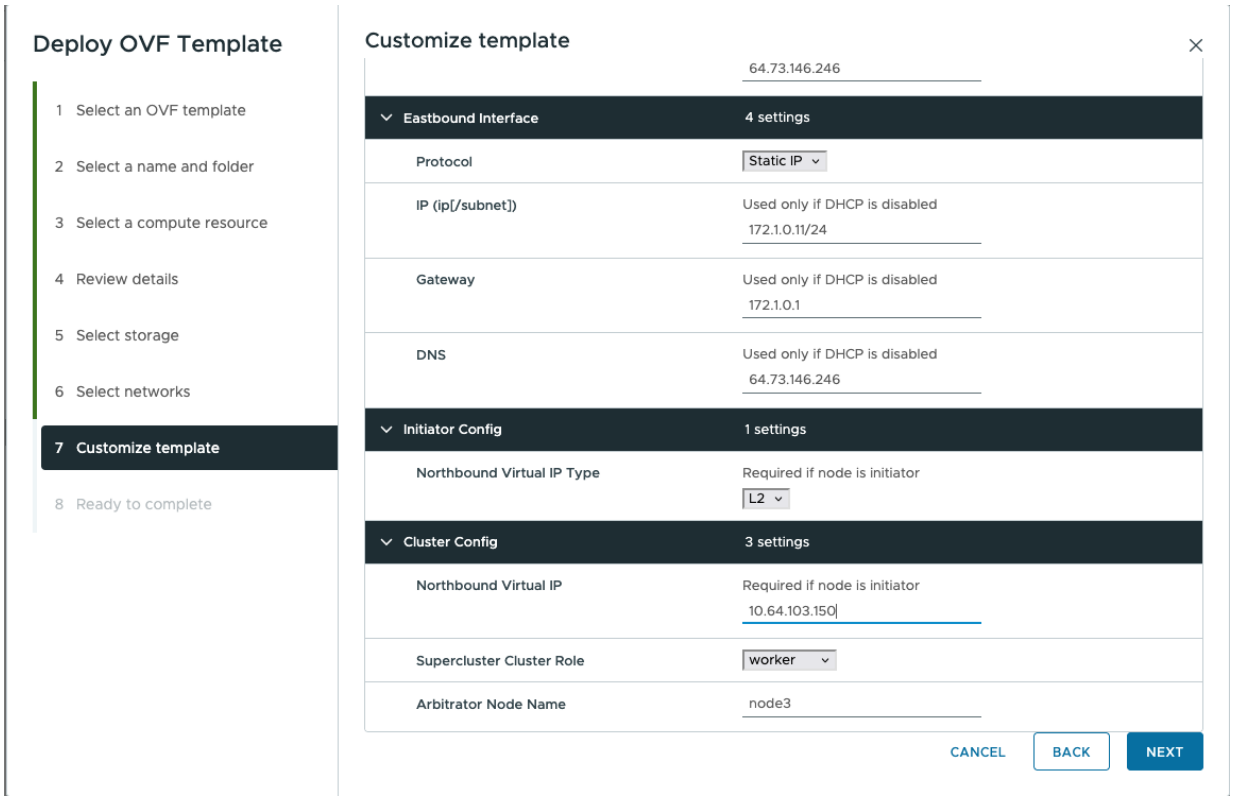


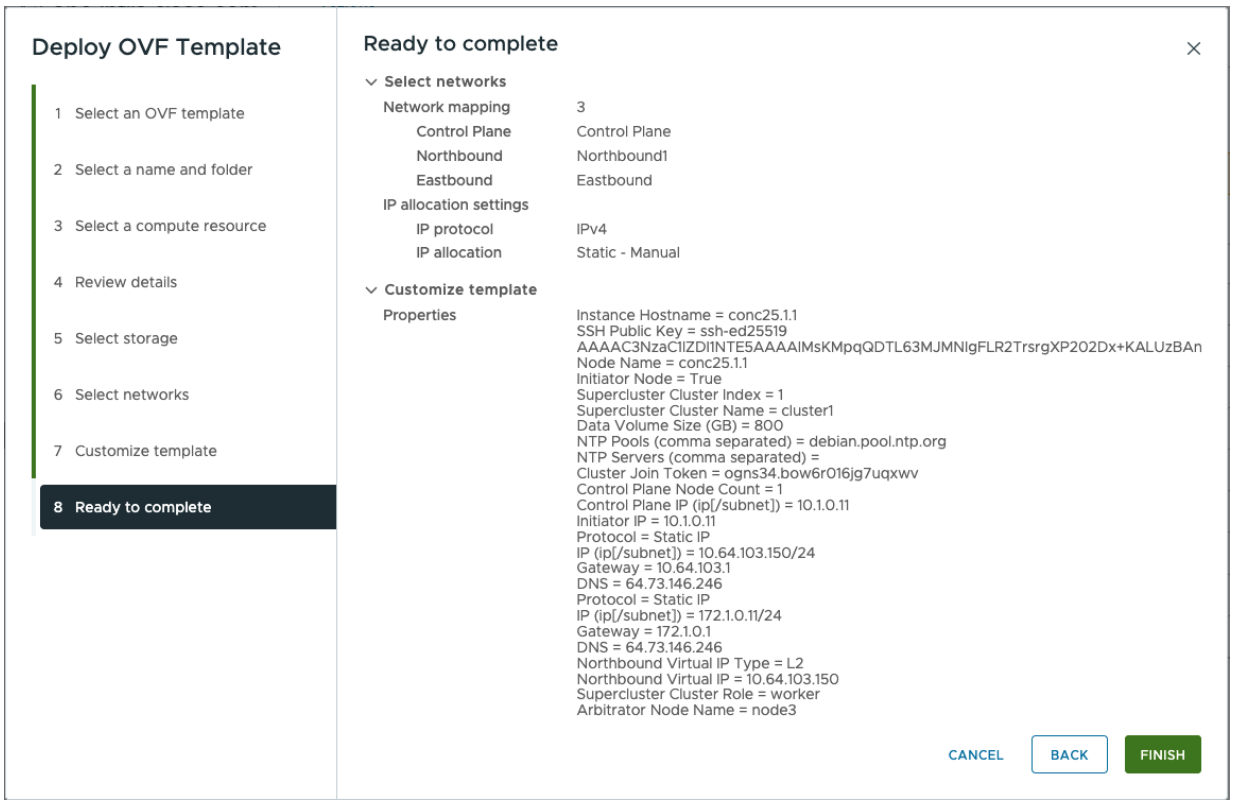
Table 5: Customize template parameters

Key	Values
Instance Hostname	<instance hostname>

Key	Values
SSH Public Key	Paste the public key generated using the ed25519 algorithm
Node Name	<node-name> Must start with an alphanumeric character. Can contain alphanumeric characters and hyphens. End with an alphanumeric character. Standalone: primary **Note** If you have other instances of Cisco Optical Network Controller, ensure that the node name is unique across instances.
Data Volume Size (GB)	<recommended-size> Configure data volume according to the VM profile. 800 GB and 1.5 TB for XS and S profiles respectively.
NTP Pools (comma separated)	(Optional) A comma-separated list of the NTP pools. For example, debian.pool.ntp.org
NTP Servers (comma separated)	(Optional) A comma-separated list of the NTP servers.
Cluster Join Token	Can be left with the default value
Control Plane Node Count	1
Control Plane IP (ip[/subnet])	<Private IP for the Instance> Control Plane Network
Initiator IP	<Same IP as Control Plane> Control Plane Network
Protocol	Static IP
IP (ip[/subnet]) - if not using DHCP	<Public IP for the Instance> Northbound Network
Gateway - if not using DHCP	<Gateway IP for the Instance> Northbound Network
DNS	DNS Server IP
Protocol	Static IP

4. In **Review the details**, review all your selections and click **Finish**. To check or change any properties before clicking **Finish**, click **Back** to return to **Customize template**.

Figure 9: Ready to Complete



5. Using the steps above from step 1 to 8, you can create one VM for standalone and three VMs for high availability. In case of high availability, it is recommended to create all three VMs (primary, secondary, and tertiary) before they are turned **ON**.

The VMware deployment is complete, and the required virtual machines are ready to be powered on.

Power on the VM and connect through SSH

Use this procedure to power on the VM and connect through SSH. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Procedure

---

1. After creating the VM, power it on and verify that the IP addresses appear in vSphere.
2. Connect to the VM using the pem key generated earlier.

> **Attention**
>
> - For more details on the pem key, see Generate an SSH ed25519 key.
> - Use the private key that is generated along with the public key during customizing the public key options.
> - Upon activation of the virtual machine (VM), it is designed not to respond to ping requests. However, you can log in using SSH if the installation has been completed successfully.

3. Log in to the VM using the private key.

> **Note**
>
> - After the nodes are deployed, the deployment of OVA progress can be checked in the Tasks console of vSphere Client. After successful deployment Cisco Optical Network Controller takes around 30 minutes to boot.
> - By default, use the username `admin` to log in to the Web UI. Only the password must be configured. To access the device through SSH, use the username `nxf`.

4. Set read permissions on the *.pem* file.

```
chmod 400 <file-name-of-your-key>.pem
```

5. SSH to the node and execute the following CLI command.

Example

```
ssh -i [ed25519 Private key] nxf@<northbound-vip>
Enter passphrase for key '<file-name-of-your-key>.pem':
```

> **Note**
>
> Private key is created as part of the key generation with just the **.pem** extension, and it must be set with the least permission level before using it.

The VM is powered on, and you are connected to the node through SSH.

Verify the system and configure initial credentials

Use this procedure to verify the system and configure initial credentials. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Procedure

1. After you SSH into the node, use the **sedo system status** command to check the status of all the pods.

Example

```
sedo system status
```

> **Note**
>
> - The different pods along with their statuses including active and standby modes are all displayed in the different terminal sessions for each pod.
> - All the services with owner *onc* must display the status as *Running*.

2. (Optional) Change the password policy settings using the **sedo security password-policy set** command.

   Example

   ```
   sedo security password-policy set --expiration-days <number> --reuse-limit
   <number> --min-complexity-score <number>
   ```

   Table 6: Password policy parameters

Parameter	Description
expiration-days	Default password expiration used when creating new users, in days. Default: 180 days
min-complexity-score	Minimum password complexity score required. Default: 3
reuse-limit	Number of historical passwords retained and blocked from reuse when changing a password. Default: 12

3. You can check the current version using the **sedo version** command.

   Example

   ```
   sedo version
   ```

4. Use the following CLI command to check and verify which node is active.

   Example

   ```
   kubectl describe project onc | head
   ```

5. Use the following CLI command to check and verify whether all nodes have joined the cluster.

   Example

   ```
   kubectl get nodes
   ```

> **Note**
>
> These cluster verification steps apply to high availability installation.

6. SSH to the node and set the initial UI password for the admin user.

Example

```
sedo security user set admin --password
```

> **Note**
>
> The password policy for the system includes both configurable settings and non-configurable hard requirements to ensure security.
>
> **Password requirements**
>
> - The password must contain at least:
>   - 1 uppercase letter
>   - 1 lowercase letter
>   - 1 number
>   - 1 special character
> - Must have a minimum length of 8 characters
>
> **Configurable requirements**
>
> You can change the password policy settings using the **sedo security password-policy set** command. Specify the desired parameters to adjust the configuration.

The system status and credentials are verified and configured.

Configure NTP and access the Cisco Optical Network Controller Web UI

Use this procedure to configure NTP, verify access details, and sign in to the Cisco Optical Network Controller Web UI.

Procedure

---

1. To check the default admin user ID, use the command **sedo security user list**. To change the default password, use the command **sedo security user admin set --password** on the CLI console of the VM or through the Web UI.
2. Use a web browser to access *https://<virtual ip>:8443/* to access the Cisco Optical Network Controller Web UI.

   Use the admin ID and the password you set to log in to Cisco Optical Network Controller.

> **Note**
>
> Access the Web UI only after all the `onc` services are running. Use the **sedo system status** command to verify that all services are running.

Time synchronization and Web UI access are configured, and Cisco Optical Network Controller is ready for use.

Upgrade a standalone deployment of Cisco Optical Network Controller to a new version

Use this procedure to upgrade a standalone Cisco Optical Network Controller deployment from an older release to a new release.

Upgrade a standalone deployment of Cisco Optical Network Controller to a new version while ensuring seamless communication between nodes.

This section provides instructions for upgrading a standalone deployment of Cisco Optical Network Controller and configuring the necessary networks to ensure seamless communication between nodes in a geo-redundant supercluster.

> **Note**
>
> If you are using CONC 24.3.1, you must first upgrade to CONC 25.1.2 as an intermediate release before upgrading to CONC 26.1.1.

Restriction

- Cisco Optical Network Controller does not support direct downgrades to older releases.
- To revert to a previous version, you must first create a database backup using the SWIMU application before upgrading. Then, install the desired older version using its OVA file, and finally, restore the database.
- For details about backup and restore database, see Backup and Restore Database.

Before you begin

Before you begin, ensure that you have:

- **Backup creation**: Ensure that a full system backup is created using the command `sedo backup create full` and exported for recovery if needed.

  Example:

```
sedo backup create full
sedo backup list
cd /data
```

```
sedo backup download base_0000000E000000010000009E
scp /data/nxf-backup-3.0-1736872559.tar.gz <remote location>
```

- Use `/data` for all file operations such as collecting, staging, copying, extracting, uploading, or downloading logs, backups, system-pack files, service-pack files, and software images. Do not use `/home/nxf` as a landing or staging directory. The `/home/nxf` directory is on the root file system and has limited capacity. Adding data to this directory can cause disk pressure and upgrade failure.
- Before starting the upgrade, verify that `/home/nxf` does not contain user-generated logs, backups, image files, tar files, or extracted bundles. Move required files to `/data` or external storage and remove only unneeded user-generated files from `/home/nxf`.
- **Network configuration**: Before the Cisco Optical Network Controller upgrade, three networks must be created.
  - **Control plane network**: The control plane network helps in the internal communication between the deployed VMs within a cluster.
  - **VM network or northbound network**: The VM network is used for communication between the user and the cluster. It handles all the traffic to and from the VMs running on your ESXi hosts. This network is your public network through which the UI is hosted. Cisco Optical Network Controller uses this network to connect to Cisco Optical Site Manager devices using Netconf/gRPC.
  - **Eastbound network**: The Eastbound Network helps in the internal communication between the deployed VMs within a cluster. If you are setting up a standalone system, this can refer to any private network.
- **VMware setup**: Ensure that the vCenter has the required networks configured and attached correctly. Verify that physical adapters are correctly mapped for Northbound and Eastbound networks.
- **Access and permissions**: Ensure you have the necessary permissions to execute commands and modify network settings on the nodes.
- Verify a system pack image package before use. Each package contains all required files for verification. For detailed steps, see Verify a signed qcow2 or system pack image on page 108.

Follow these steps to upgrade a standalone deployment.

Procedure

1. Log in to the standalone node CLI using the private key.

   Example

   ```
   ssh -i <private-key_file> nxf@<node_ip>
   ```

2. Download or copy the release system pack `system-pack-file.tar.gz` to the NxF system and place it in the `/tmp` directory using `curl` or `scp`.

   Example

   ```
   scp user@remote_server:/path/to/system-pack-file.tar.gz /tmp/
   ```

   Or use curl:

   ```
   curl -o /tmp/system-pack-file.tar.gz
   http://example.com/path/to/system-pack-file.tar.gz
   ```

3. Upgrade the SA VM using the sedo system upgrade commands.

   Note

   When you run the **sedo system upgrade pull** command, the pulled image may take a few moments to appear in the output of **sedo system upgrade list** command.

   Example

   ```
   sedo system upgrade upload /tmp/system-pack-file.tar.gz
   sedo system upgrade apply
   reboot
   ```

   The system reboots and upgrades. The system takes approximately 30 minutes to complete this.

4. After the system reboots, verify the NxF version and system status.

   Use the **sedo version** and **sedo system status** commands.

   Example

   ```
   sedo version
   ```

   ```
   sedo system status
   ```

5. Verify onboarded sites and services by accessing the Cisco Optical Network Controller UI.

   Example

   Use a web browser to access *https://<virtual ip>:8443/* to access the Cisco Optical Network Controller Web UI.

The standalone deployment of Cisco Optical Network Controller is successfully upgraded. All services are running, and you can access the Web UI.

Update time zone configuration in a standalone deployment

Use this procedure to update time zone configuration in a standalone deployment. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Update the timezone configuration for each VM in a standalone deployment and restart the VM to ensure a seamless change into the new timezone configuration.

From Cisco Optical Network Controller Release 25.1.2, you can update the timezone configuration. Previously, only the UTC timezone was supported. Now you can configure Cisco Optical Network Controller in your preferred timezone.

For standalone deployments, you must use the command to update the timezone in the CLI for each VM and then restart the VM according to the steps in this procedure to ensure a seamless change into the new timezone configuration.

Limitations

- Alarms and logs are saved in UTC in the database, which minimizes impact during time zone transitions, although during the transition period, for example, during a switchover, you might briefly see alarms with different time zone stamps in the UI before the system converges to the final setting.

- Do not make timezone changes frequently as they might cause inconsistencies and require reboots of VMs/services.
- When cross-launching from Cisco Optical Network Controller, the time zone offset will remain the same, but the IANA time zone name displayed in the cross-launched application might differ from the one configured in Cisco Optical Network Controller. This discrepancy occurs because the same timezone offset can have multiple IANA timezone names.
- TAPI data and notifications continue to use UTC +0000.
- SNMP traps use epoch time without any time zone offset calculated on the epoch.
- Developer logs and techdump data uses UTC.

Before you begin

You must perform these pre-checks on each VM before changing the timezone.

- Make sure all the pods are running by running the **sedo system status -w** command.

  This example shows a sample output where all pods are running. Verify status of every pod is `Running`.

```
root@vm1-cluster1-node1:~# sedo system status -w
┌──────────────────────────────────────────────────────────────────────────────────────────────────────────────┐
│ System Status (Thu, 20 Nov 2025 09:55:35 UTC)                                                                │
└──────────────────────────────────────────────────────────────────────────────────────────────────────────────┘
┌─────────┬──────────────────────────┬──────────────────┬─────────┬──────────┬──────────────────────┐
│ OWNER   │ NAME                     │ NODE             │ STATUS  │ RESTARTS │ STARTED              │
├─────────┼──────────────────────────┼──────────────────┼─────────┼──────────┼──────────────────────┤
│ onc     │ monitoring               │ concgha2-clb-vm1 │ Running │ 0        │ 5 days ago           │
│ onc     │ onc-alarm-service        │ concgha2-clb-vm1 │ Running │ 3        │ (Latest 5 days ago) | 1 week ago  │
│ onc     │ onc-apps-ui-service      │ concgha2-clb-vm1 │ Running │ 3        │ (Latest 5 days ago) | 1 week ago  │
│ onc     │ onc-circuit-service      │ concgha2-clb-vm1 │ Running │ 3        │ (Latest 5 days ago) | 1 week ago  │
│ onc     │ onc-collector-service    │ concgha2-clb-vm1 │ Running │ 3        │ (Latest 5 days ago) | 1 week ago  │
│ onc     │ onc-config-service       │ concgha2-clb-vm1 │ Running │ 3        │ (Latest 5 days ago) | 1 week ago  │
│ onc     │ onc-devicemanager-service│ concgha2-clb-vm1 │ Running │ 3        │ (Latest 5 days ago) | 1 week ago  │
│ onc     │ onc-inventory-service    │ concgha2-clb-vm1 │ Running │ 3        │ (Latest 5 days ago) | 1 week ago  │
│ onc     │ onc-nbi-service          │ concgha2-clb-vm1 │ Running │ 3        │ (Latest 5 days ago) | 1 week ago  │
│ onc     │ onc-netconfcollector-service │ concgha2-clb-vm1 │ Running │ 0    │ 5 days ago           │
│ onc     │ onc-osapi-gw-service     │ concgha2-clb-vm1 │ Running │ 3        │ (Latest 5 days ago) | 1 week ago  │
│ onc     │ onc-pce-service          │ concgha2-clb-vm1 │ Running │ 3        │ (Latest 5 days ago) | 1 week ago  │
│ onc     │ onc-pm-service           │ concgha2-clb-vm1 │ Running │ 3        │ (Latest 5 days ago) | 1 week ago  │
│ onc     │ onc-pmcollector-service  │ concgha2-clb-vm1 │ Running │ 0        │ 5 days ago           │
│ onc     │ onc-topology-service     │ concgha2-clb-vm1 │ Running │ 3        │ (Latest 5 days ago) | 1 week ago  │
│ onc     │ onc-torch-service        │ concgha2-clb-vm1 │ Running │ 3        │ (Latest 5 days ago) | 1 week ago  │
│ system  │ authenticator            │ concgha2-clb-vm1 │ Running │ 0        │ 5 days ago           │
```

```
| system | bgp                              | concgha2-clb-vm1 | Running | 0
                 | 5 days ago  |
| system | controller                       | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 2 weeks ago |
| system | flannel                          | concgha2-clb-vm1 | Running | 1
(Latest 5 days ago) | 2 weeks ago |
| system | ingress-proxy                    | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 1 week ago  |
| system | kafka                            | concgha2-clb-vm1 | Running | 1
(Latest 5 days ago) | 2 weeks ago |
| system | loki                             | concgha2-clb-vm1 | Running | 4
(Latest 5 days ago) | 2 weeks ago |
| system | metrics                          | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 2 weeks ago |
| system | minio                            | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 2 weeks ago |
| system | postgres                         | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 2 weeks ago |
| system | promtail-vff8n                   | concgha2-clb-vm1 | Running | 1
(Latest 5 days ago) | 2 weeks ago |
└────────┴──────────────────────────────┴──────────────────┴─────────┴──────────────────┴──────────┘
```

Procedure

1. SSH into the VM and run the **sudo timedatectl set-timezone** command with your preferred timezone name.

   Command syntax: **sudo timedatectl set-timezone** *timezone-name*

   Example

   In the following example, we set the timezone to JST.

   ```
   root@vm1-cluster1-node1:~# sudo timedatectl set-timezone Asia/Tokyo

   root@vm1-cluster1-node1:~# timedatectl

                  Local time: Mon 2025-06-09 15:01:26 JST

              Universal time: Mon 2025-06-09 06:01:26 UTC

                    RTC time: Mon 2025-06-09 06:01:26

                   Time zone: Japan (JST, +0900)

   System clock synchronized: yes

                 NTP service: active

             RTC in local TZ: no
   ```

   A few valid timezones are:

   ```
   Asia/Kolkata
   Asia/Dubai
   Europe/Amsterdam
   Africa/Bujumbura
   ```

2. Reboot the node using the **sudo reboot** command.
3. Verify the node is up and running using the **sedo system status -w** command.

Verify the timezone in one of the pods using these commands. See the offset after the time.

```
root@vm1-cluster1-node1:~# sedo shell onc-torch-service
Entering into following pod:
```

Name	onc-torch-service-0
Node	achitrad-nxf
IP	10.241.0.186
Namespace	onc
Labels	statefulset.kubernetes.io/pod-name="onc-torch-service-0" app="onc-torch-service" apps.kubernetes.io/pod-index="0" controller-revision-hash="onc-torch-service-74947495fc" nxf="customer" profile="ActiveStandby" role="Active"

```
/ $ date -R
Fri, 30 Jan 2026 06:21:50 +0000
/ $
```

Timezone configuration has been updated and Cisco Optical Network Controller webUI now displays time in the newly configured timezone.

The following screenshots show the difference between the behaviour in 25.1.1 and 25.1.2. Note that the timestamps are displayed differently with the timezone name and offset included in the timestamp in Release 25.1.2.

Figure 10: PM History in Release 25.1.2

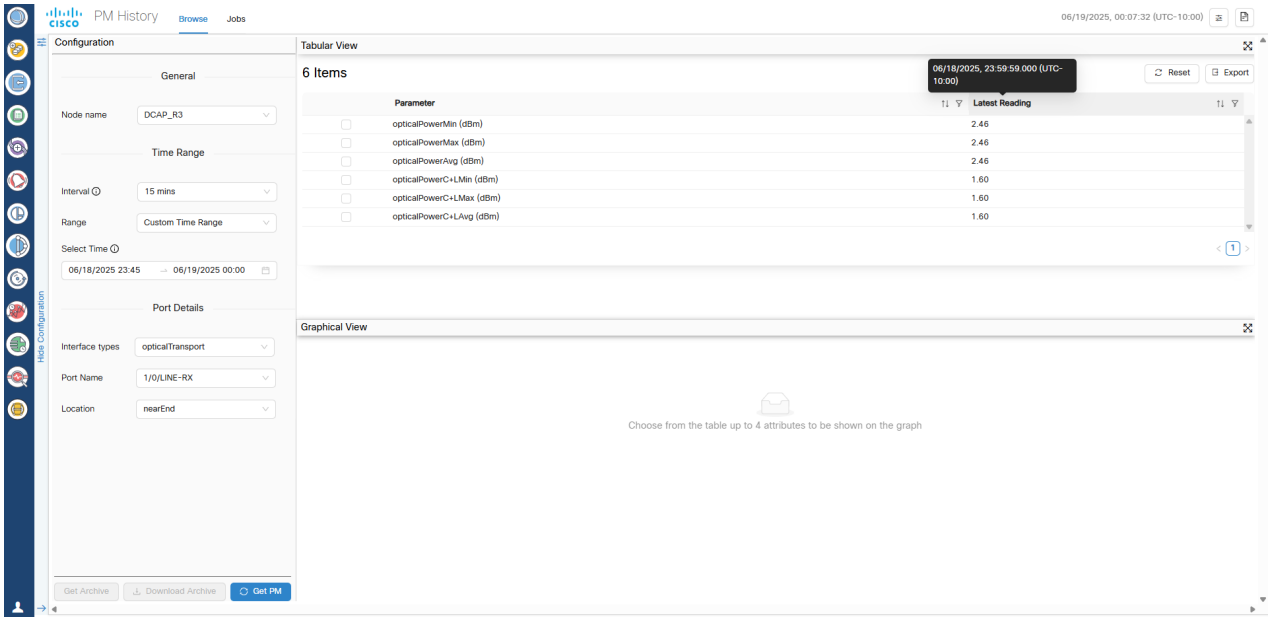


Figure 11: PM History in Release 25.1.1

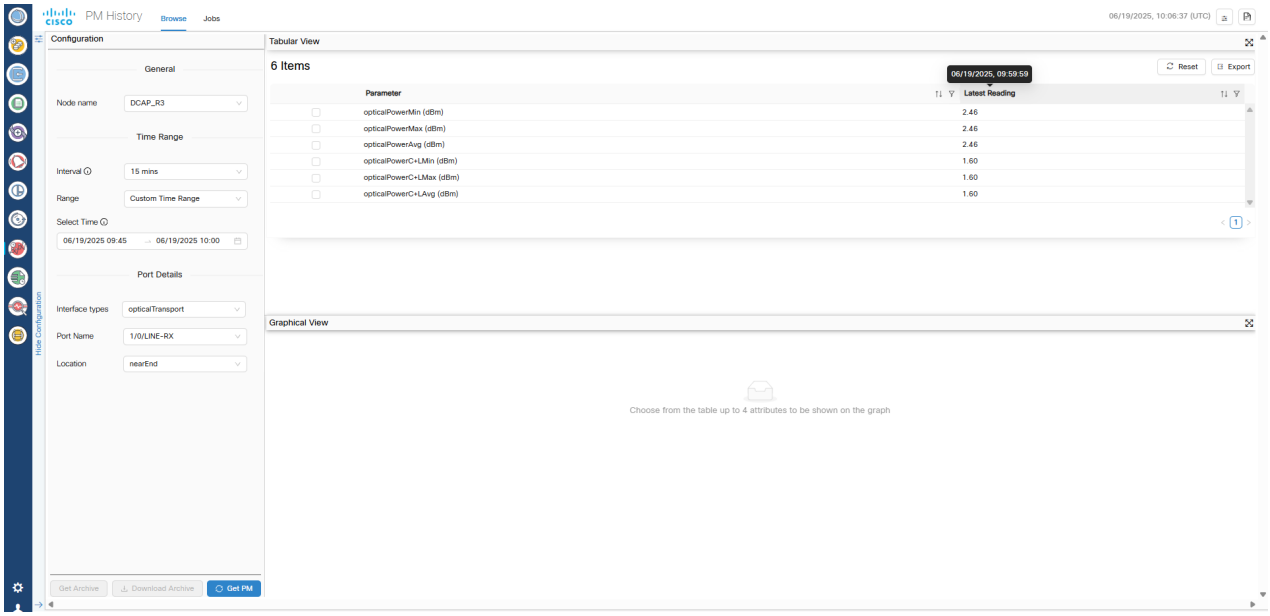


Figure 12: Nodes in Release 25.1.2

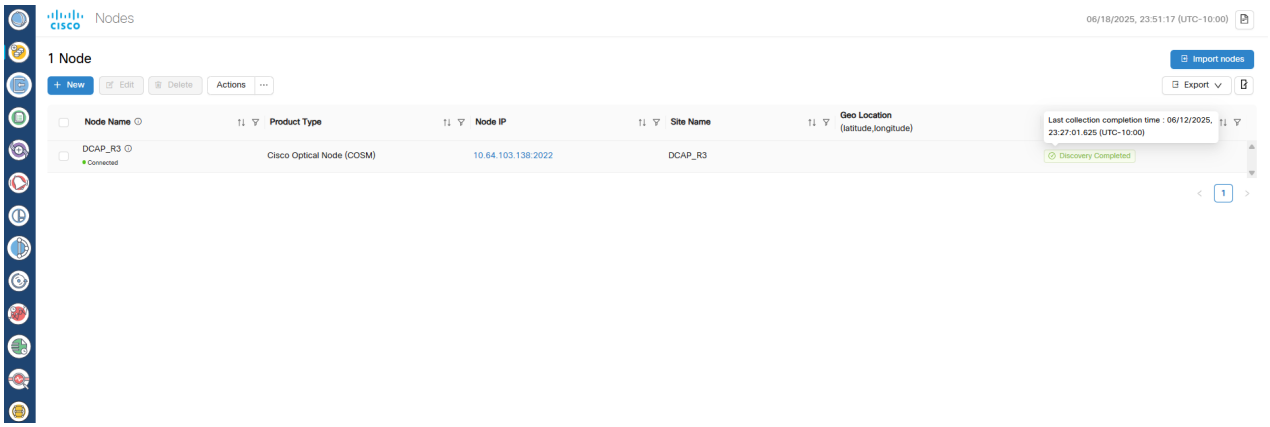
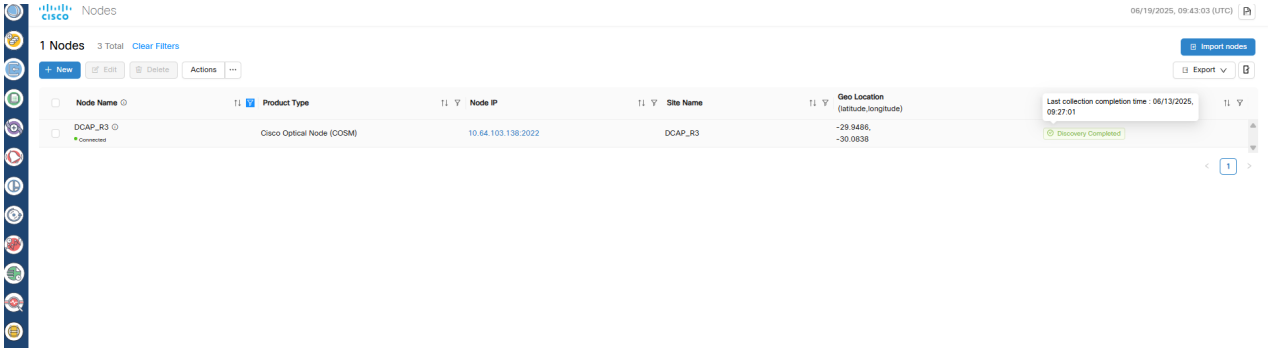


Figure 13: Nodes in Release 25.1.1



Revert to a previous version of Cisco Optical Network Controller

Use this procedure to revert to a previous version of Cisco Optical Network Controller. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Revert Cisco Optical Network Controller to the earlier version by reinstalling the previous release and restoring the backup.

This section describes how to revert to the previous version of Cisco Optical Network Controller after installation for standalone and geo-redundant deployments. This is a manual process. Automatic rollback is not supported, and you cannot perform the revert from within Cisco Optical Network Controller.

Restriction

- Cisco Optical Network Controller does not support direct downgrades to older releases.
- To revert to a previous version, you must first create a database backup using the SWIMU application before upgrading. Then install the desired older version by using its OVA file, and finally restore the database.
- For detailed instructions, see Backup and Restore Database.

Before you begin

Before you begin, ensure that you have:

- A backup created from the earlier Cisco Optical Network Controller version.
- The installation package for the version you want to restore.
- Access to the database restore procedure documented in Backup and Restore Database.

Follow these steps to revert to a previous version.

Procedure

1. For standalone deployments:
   a) Reinstall the previous version of Cisco Optical Network Controller, which is the version from which you created the backup. See Install Cisco Optical Network Controller using VMware vSphere.
   b) Perform the database restore by using Backup and Restore Database.
2. For geo-redundant deployments:
   a) Reinstall the previous version of Cisco Optical Network Controller, which is the version from which you created the backup.
   b) Perform the database restore by using Backup and Restore Database.

Cisco Optical Network Controller runs the earlier software version and the system state is restored from the backup.

Install a Cisco Optical Network Controller service pack

Use this procedure to install a Cisco Optical Network Controller service pack and verify that the updated image is active.

Install a service pack to update your Cisco Optical Network Controller with bug fixes and enhancements. Transfer the Service Pack file, install the service pack, and verify that the new image is active.

Apply service packs to keep the controller up to date. In GeoHA deployments, install the service pack only on the active node.

Before you begin

Follow these steps to install a Cisco Optical Network Controller Service Pack:

Download the *.tar.gz* file for your service pack from the Cisco Software Download page.

Procedure

1. Use the **scp** command to copy the file to the */data* directory on the active node.

   Example

   ```
   scp 26.1.1-5-MSMU-1-SP.tar.gz -i <pem_file> nxf@<CONC_HOST_IP>:/data
   ```

   Note

   You can download the SHA-256 checksum from the Cisco Software Download page and compare the checksum with the service pack file to verify the integrity of the file. Use the following command to get the checksum of the downloaded file.

   ```
   openssl sha256 <service-pack-file>.tar.gz
   ```

2. Use the **sedo supercluster status** command to identify the active Cisco Optical Network Controller instance.

   Look for the rows labeled **Cluster Name** and **Current Active**.

   Example

   This sample output shows the **Cluster Name** and **Current Active** as the active Cisco Optical Network Controller instance in bold.

   ```
   concgha2-clb-vm1:~# sedo supercluster status
   ┌──────────────────────────────────────────────────────────────────────────────┐
   │ Supercluster Status                                                          │
   ├──────────────────┬───────────────────────────────────────────────────────────┤
   │ Cluster ID       │ INVetOV3DZKTSR5kmTFrE1PmDj954yyoBO6F4GbjlM8               │
   │ Cluster Name     │ cluster2                                                  │
   │ Cluster Role     │ worker                                                    │
   │ Peers            │ cluster1 (worker,
   ej60l6x1PZtRdmAj9sEd07Vk201duytRUaPttJnXwk8)    │
   │                  │ cluster3 (arbitrator,
   lSK7K7xavBTjaNDou6mTVGKR15bRZeOPRmC6fwgX8) │
   │ Mode             │ Running                                                   │
   │ Current Active   │ cluster2                                                  │
   │ Previous Active  │ cluster1                                                  │
   │ Standby Clusters │ cluster1                                                  │
   │ Last Switchover  │ 2025-11-13 14:32:40.801 +0000 UTC                         │
   │ Last Failover    │ 2025-11-14 10:22:02.158 +0000 UTC                         │
   │ Last Seen        │ controller-0.cluster2: 2025-11-20 10:28:26.17 +0000 UTC
   │                  │ controller-0.cluster3: 2025-11-20 10:28:26.17 +0000 UTC
   ```

```
          |
|                | controller-0.cluster1: 2025-11-20 10:28:26.165 +0000 UTC
          |
| Last Peer Error  |
          |
| Server Error     |
          |
| DB Replication   | streaming
          |
| DB Lag           | 0 bytes
          |
└─────────────────┴──────────────────────────────────────────────────────────┘
```

3. Use the **sedo service install** command to install the Service Pack on the active node.

   Example

   ```
   root@superman-c1-n1:/data# sedo service install 26.1.1-5-MSMU-1-SP.tar.gz
   Importing images...
   ```

   The Cisco Optical Network Controller service is restarted with the new image. Images are automatically synced to the standby node, and services restart on standby.

4. Use the **sedo system status -w** command to verify that all services are running.

   In a GeoHA setup, verify the system status on both the active and standby virtual machines.

   Example

   This sample output shows all the services as *Running* under the **STATUS** column.

```
concgha2-clb-vm1:~# sedo system status -w
┌──────────────────────────────────────────────────────────────────────────────┐
| System Status (Thu, 20 Nov 2025 09:55:35 UTC)
                                  |
├──────┬──────────────────────────┬──────────────┬─────────┬──────────┬─────────┤
| OWNER  | NAME                          | NODE           | STATUS   | RESTARTS
          | STARTED      |
├──────┼──────────────────────────┼──────────────┼─────────┼──────────┼─────────┤
| onc    | monitoring                    | concgha2-clb-vm1 | Running | 0
               | 5 days ago  |
| onc    | onc-alarm-service             | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 1 week ago  |
| onc    | onc-apps-ui-service           | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 1 week ago  |
| onc    | onc-circuit-service           | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 1 week ago  |
| onc    | onc-collector-service         | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 1 week ago  |
| onc    | onc-config-service            | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 1 week ago  |
| onc    | onc-devicemanager-service     | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 1 week ago  |
| onc    | onc-inventory-service         | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 1 week ago  |
| onc    | onc-nbi-service               | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 1 week ago  |
| onc    | onc-netconfcollector-service | concgha2-clb-vm1 | Running | 0
               | 5 days ago  |
| onc    | onc-osapi-gw-service          | concgha2-clb-vm1 | Running | 3
```

```
(Latest 5 days ago) | 1 week ago  |
| onc    | onc-pce-service             | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 1 week ago  |
| onc    | onc-pm-service              | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 1 week ago  |
| onc    | onc-pmcollector-service     | concgha2-clb-vm1 | Running | 0
                | 5 days ago  |
| onc    | onc-topology-service        | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 1 week ago  |
| onc    | onc-torch-service           | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 1 week ago  |
| system | authenticator               | concgha2-clb-vm1 | Running | 0
                | 5 days ago  |
| system | bgp                         | concgha2-clb-vm1 | Running | 0
                | 5 days ago  |
| system | controller                  | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 2 weeks ago |
| system | flannel                     | concgha2-clb-vm1 | Running | 1
(Latest 5 days ago) | 2 weeks ago |
| system | ingress-proxy               | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 1 week ago  |
| system | kafka                       | concgha2-clb-vm1 | Running | 1
(Latest 5 days ago) | 2 weeks ago |
| system | loki                        | concgha2-clb-vm1 | Running | 4
(Latest 5 days ago) | 2 weeks ago |
| system | metrics                     | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 2 weeks ago |
| system | minio                       | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 2 weeks ago |
| system | postgres                    | concgha2-clb-vm1 | Running | 3
(Latest 5 days ago) | 2 weeks ago |
| system | promtail-vff8n              | concgha2-clb-vm1 | Running | 1
(Latest 5 days ago) | 2 weeks ago |
```

5. Use the **sedo version** command to verify the Service Pack image is installed on both active and standby nodes.

Example

This sample output shows the *26.1.1-5-MSMU-1* as the installed service pack.

```
concgha2-clb-vm1:~# sedo version
Installer: 26.1.1

NODE NAME           | OS VERSION
     | KERNEL VERSION            |
vc39-es33-sa-free-85 | NxFOS 4.1-784
(a35508a8b352d52d56ec3af970f4eceb431da43b) | 6.1.0-39-cisco-cloud-amd64 |

IMAGE NAME
 | VERSION          | NODES                |
registry.nxf-system.svc:8443/cisco-onc-docker/dev/alarmservice
 | 26.1.1-5         | vc39-es33-sa-free-85 |
registry.nxf-system.svc:8443/cisco-onc-docker/dev/alarmservice
 | 26.1.1-5-MSMU-1-3 | vc39-es33-sa-free-85 |
registry.nxf-system.svc:8443/cisco-onc-docker/dev/circuit-service
```

```
 | 26.1.1-5           | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/circuit-service
 | 26.1.1-5-MSMU-1-3 | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/conc-mcp-server
 | 26.1.1-5           | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/devicemanager-service
 | 26.1.1-5-MSMU-1-3 | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/devicemanager-service
 | 26.1.1-5           | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/inventory-service
 | 26.1.1-5-MSMU-1-1 | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/inventory-service
 | 26.1.1-5           | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/monitoring
 | 26.1.1-5           | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/nbi-service
 | 26.1.1-5-MSMU-1-2 | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/nbi-service
 | 26.1.1-5           | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/netconfcollector-service
 | 26.1.1-5           | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/netconfcollector-service
 | 26.1.1-5-MSMU-1-1 | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/onc-apps-ui-service
 | 26.1.1-5-MSMU-1-3 | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/onc-apps-ui-service
 | 26.1.1-5           | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/osapi-gw-service
 | 26.1.1-5           | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/osapi-gw-service
 | 26.1.1-5-MSMU-1-1 | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/pce_service
 | 26.1.1-5-MSMU-1-1 | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/pce_service
 | 26.1.1-5           | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/pm-service
 | 26.1.1-5           | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/pm-service
 | 26.1.1-5-MSMU-1-2 | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/pmcollector-service
 | 26.1.1-5           | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/topology-service
 | 26.1.1-5           | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/topology-service
 | 26.1.1-5-MSMU-1-2 | vc39-es33-sa-free-85 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/torch
 | 26.1.1-5           | vc39-es33-sa-free-85 |
| registry.sedona.ciscolabs.com/rel/nxf/alpine
 | 4.1-713            | vc39-es33-sa-free-85 |
| registry.sedona.ciscolabs.com/rel/nxf/authenticator
 | 4.1-708            | vc39-es33-sa-free-85 |
| registry.sedona.ciscolabs.com/rel/nxf/bgp
 | 4.1-713            | vc39-es33-sa-free-85 |
| registry.sedona.ciscolabs.com/rel/nxf/controller
 | 4.1-739            | vc39-es33-sa-free-85 |
| registry.sedona.ciscolabs.com/rel/nxf/firewalld
 | 4.1-713            | vc39-es33-sa-free-85 |
| registry.sedona.ciscolabs.com/rel/nxf/flannel
 | 4.1-713            | vc39-es33-sa-free-85 |
| registry.sedona.ciscolabs.com/rel/nxf/ingress-proxy
 | 4.1-715            | vc39-es33-sa-free-85 |
| registry.sedona.ciscolabs.com/rel/nxf/iptables
```

```
 | 4.1-715           | vc39-es33-sa-free-85 |
| registry.sedona.ciscolabs.com/rel/nxf/kafka
 | 4.1-713           | vc39-es33-sa-free-85 |
| registry.sedona.ciscolabs.com/rel/nxf/kubernetes
 | 4.1-713           | vc39-es33-sa-free-85 |
| registry.sedona.ciscolabs.com/rel/nxf/loki
 | 4.1-713           | vc39-es33-sa-free-85 |
| registry.sedona.ciscolabs.com/rel/nxf/metrics-exporter
 | 4.1-713           | vc39-es33-sa-free-85 |
| registry.sedona.ciscolabs.com/rel/nxf/minio
 | 4.1-713           | vc39-es33-sa-free-85 |
| registry.sedona.ciscolabs.com/rel/nxf/service-proxy
 | 4.1-715           | vc39-es33-sa-free-85 |
| registry.sedona.ciscolabs.com/rel/nxf/syslog-forwarder
 | 4.1-704           | vc39-es33-sa-free-85 |
| registry.sedona.ciscolabs.com/rel/nxf/timescale
 | 4.1-725           | vc39-es33-sa-free-85 |
└──────────────────────────────────────────────┴──────────────┴──────────────────┘
```

All Cisco Optical Network Controller services display the installer service pack number in the **Version** column. For example *26.1.1-5*.

6. Use the **sedo service list-installed** command to identify the service packs currently installed on the VM.

```
nxf@vc39-es33-sa-free-85:~$ sedo service list-installed
┌──────────────┬──────────────────────┬──────────────────────┬──────────────────────
│ INSTALLED BY │ APPLY TIME           | SERVICE PACK VERSION | PLATFORM VERSION
                                    | FILE NAME
    |
├──────────────┼──────────────────────┼──────────────────────┼──────────────────────
| sedo         | 2026-07-23T11:08:10Z | 26.1.1-5             |
4.1-702+2c420b5a5383cd076ffb299d3a7fb8c663df26e3 |
/config/service-packs/000_26.1.1-SP-5.tar.gz |
| sedo         | 2026-07-23T11:54:04Z | 26.1.1-5-MSMU-1      |
4.1-702+2c420b5a5383cd076ffb299d3a7fb8c663df26e3 | 26.1.1-5-MSMU-1-SP.tar.gz
                    |
└──────┴──────────┴──────────┴──────────────────────┴──────────────────────
nxf@vc39-es33-sa-free-85:~$
```

The Cisco Optical Network Controller Service Pack is installed, and all services are running the updated version.

KVM deployment for Cisco Optical Network Controller standalone mode

Explains KVM-based standalone deployment, including KVM requirements, network configuration, cloud-init files, virtual machine creation, and access verification for Cisco Optical Network Controller.

A KVM-based standalone installation deploys Cisco Optical Network Controller as a single KVM virtual machine by using a QCOW2 image and a cloud-init ISO that contains the node configuration.

- The deployment uses one VM with control-plane, northbound, and eastbound network interfaces.
- The cloud-init ISO provides the `meta-data`, `network-config`, and `user-data` files that initialize the system.
- The standalone deployment does not require Geo-HA node joining or BGP-based multi-site routing.

Although the control-plane network is not used for standalone service traffic, it must still be defined during KVM deployment.

What to prepare before you start

Before you begin the standalone installation workflow, gather the values and files required to create the VM configuration.

- QCOW2 image, VM sizing values, deployment directory, and KVM bridge names for the standalone VM.
- Host name, control-plane, northbound, and eastbound IP addresses, DNS values, and the northbound virtual IP.
- SSH key pair and the cloud-init values required for the standalone node definition and local users.

Use the standalone installation topics to create the cloud-init files, deploy the VM, verify system status, secure the admin account, and continue with post-installation setup.

KVM deployment requirements

This section provides the software requirements, hardware requirements, and network information required to deploy Cisco Optical Network Controller on a KVM host.

This section lists the software requirements, hardware requirements, and network information for deploying Cisco Optical Network Controller on a KVM host.

Software requirements

This table lists the software and access requirements for the KVM deployment.

Table 7: KVM software requirements

Requirement	Description
`libvirt-client` and `qemu-kvm`	Install these packages on the KVM host.
`cloud-utils`	Use this package to create the cloud-init ISO.
`wget` or `curl`	Use one of these utilities to download the OS image.
SSH key pair	Use the SSH key pair to access the virtual machine.

Hardware requirements

This table lists the host infrastructure requirements for the KVM deployment.

Table 8: KVM hardware requirements

Requirement	Description
KVM host	Use RHEL 8.x or later with KVM and libvirt installed.
UEFI support	Install the OVMF firmware packages on the host.
Storage	Provide at least 100 GB of available disk space.
Memory	Provide at least 16 GB of available RAM.
CPU	Provide at least 8 vCPUs.

Configure networks for standalone deployment

Learn how to configure the eastbound, control-plane, and northbound networks for the control plane, eastbound traffic, and northbound traffic on the KVM host before you deploy virtual machines.

Create the libvirt network definitions required Cisco Optical Network Controller standalone deployment on a KVM host.

Before you begin

- Log in to the KVM host with privileges to create libvirt network definitions.
- Choose a working directory to store the network XML files.

In a standalone deployment, one eastbound network and one control-plane network are required.

Procedure

1. Create the control-plane network definition in a file.

   The control plane network can be a private network.

   Example
   Example for *control.xml*

   ```
   <network>
   <name>control</name>
   <forward mode='none'/>
   <bridge name='virbr-control'/>
   <ip address='192.168.1.1' netmask='255.255.255.0'/>
   </network>
   ```

2. Create the eastbound network definitions in file named *eastbound.xml*.

   The eastbound network can be a private network.

   Example
   Example for *eastbound1.xml*.

   ```
   <network>
     <name>eastbound1</name>
     <forward mode='none'/>
     <bridge name='virbr-east1'/>
     <ip address='172.10.10.1' netmask='255.255.255.0'/>
   </network>
   ```

3. Specify the default northbound interface using the <NORTHBOUND_BRIDGE> parameter for northbound connectivity when running the **virt-install** command.

   For example, use *bridge0* as the value for <NORTHBOUND_BRIDGE> to assign the northbound interface during deployment.

4. Define, start, and enable each network at boot.

   Example

   ```
   virsh net-define <network-file.xml>
   virsh net-start <network-name>
   virsh net-autostart <network-name>
   ```

5. Verify that all required networks are available.

   Example

   ```
   virsh net-list --all
   ```

   The network list shows the defined networks and their current state.

The control-plane and eastbound networks are configured and available for KVM deployment.

Install Cisco Optical Network Controller using KVM in standalone mode

Use this procedure to deploy Cisco Optical Network Controller on KVM in standalone mode by preparing the node configuration, deploying the virtual machine, verifying the installation, and continuing with post-installation tasks.

Use this supertask to complete the full KVM-based standalone installation workflow for Cisco Optical Network Controller.

Before you begin

Prepare KVM configuration files for standalone deployment.

Procedure

---

1. Create the KVM virtual machine.

   See Create a KVM virtual machine for standalone deployment.

2. Verify the deployment and access the web UI.

   See Verify and access the standalone deployment.

Cisco Optical Network Controller is installed on KVM in standalone mode.

Prepare KVM configuration files for standalone deployment

Use this procedure to create the cloud-init files for the standalone node and package them into an ISO for KVM deployment.

Prepare the cloud-init configuration used to deploy Cisco Optical Network Controller on KVM as a standalone instance.

Before you begin

- Perform this procedure for one VM.
- Have the host name, IP addresses, virtual IP, DNS values, and SSH public key available before you create the files.

Procedure

---

1. Create the SSH public-key file by following Create the meta-data file for a standalone node.
2. Create the network definition by following Create the network-config file for a standalone node.
3. Create the cluster and user settings by following Create the user-data file for a standalone node.

The *cidata.iso* file is created and is ready to be attached when you deploy the standalone Cisco Optical Network Controller VM.

What to do next

Create a KVM virtual machine for standalone deployment.

Create the meta-data file for a standalone node

Use this procedure to create the meta-data file for a standalone node. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Create the *meta-data* cloud-init file for a standalone Cisco Optical Network Controller node.

Procedure

1. Create a file named *meta-data*.
2. Add the instance ID, local host name, and SSH public key.

   Meta-data file syntax:

```
instance-id: <instance_id>
local-hostname: <node_name>
public-keys:
  - <ssh_public_key>
```

The *meta-data* file is ready to be included in the cloud-init ISO.

Meta-data file example

```
instance-id: iid-conc-sa-kvm-1
local-hostname: conc-sa-kvm-1
public-keys:
  - ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIKAn33NZjrMWyMJpf7QvrD4vCvEAg4PPdpb3UFXXXXXX
```

Create the network-config file for a standalone node

Use this procedure to create the `network-config` file that defines the control-plane, northbound, and eastbound interfaces for the standalone node.

Create the *network-config* file that defines the standalone node IP addresses, default gateway, and DNS settings.

Procedure

1. Create a file named *network-config*.
2. Define the control-plane, northbound, and eastbound interfaces.

   network-config file syntax.

```
version: 2
ethernets:
  enp1s0:
    dhcp4: <false-or-true>
    addresses:
      - <control_plane_ip/subnet_mask>
  enp2s0:
    dhcp4: <false-or-true>
    addresses:
      - <northbound_ip/subnet_mask>
```

```
    gateway4: <gateway_ip>
    nameservers:
      addresses:
        - <dns_server_ip>
  enp3s0:
    dhcp4: <false-or-true>
    addresses:
      - <eastbound_ip/subnet_mask>
```

The *network-config* file is ready for use by the standalone node.

***network-config* file example**

```
version: 2
ethernets:
  enp1s0:
    dhcp4: false
    dhcp6: false
    addresses:
      - 192.168.119.10/24
  enp2s0:
    dhcp4: false
    dhcp6: false
    addresses:
      - 10.58.231.119/22
    gateway4: 10.58.228.1
    nameservers:
      addresses:
        - 144.254.71.184
  enp3s0:
    dhcp4: false
    dhcp6: false
    addresses:
      - 172.10.10.10/24
```

Create the user-data file for a standalone node

Use this procedure to create the `user-data` file that defines the standalone node settings, local users, storage mount, and optional NTP servers.

Create the *user-data* cloud-init file for a standalone Cisco Optical Network Controller node.

Procedure

1. Create a file named *user-data*.
2. Add the base cloud-init content for the data disk, optional NTP servers, and the standalone cluster configuration for the node.

   user-data file syntax

```
fs_setup:
  - label: data
    device: /dev/vdb
    filesystem: ext4
mounts:
  - ["/dev/vdb", "/data"]
ntp:
```

```
  enabled: true
  ntp_client: chrony
  servers:
    - <ntp_server>
nxf:
  minControlPlaneCount: 1
  clusterIndex: <cluster_index>
  clusterName: <cluster_name>
  node:
    name: <node_name>
    initiator: <control_plane_ip>
    joinToken: <join_token>
    controlPlaneInterface: enp1s0
    eastboundInterface: enp3s0
    vip:
      northbound:
        interface: enp2s0
  initiator:
    vip:
      northbound:
        ip: <vip>
        type: L3
    kafka:
      enabled: true
    joinToken: <join_token>
    arbitratorNode: my-node3
    clusterRole: <worker-or-arbitrator>
    security:
      localUsers:
        - username: admin
          displayName: NxF Admin
          description: NextFusion Default Administrator
          locked: true
          mustChangePassword: false
          expiresInDays: 0
          access:
            - permission/admin
```

Use a valid DNS name for both *clusterName* and *node_name*.

The *user-data* file is ready to be included in the cloud-init ISO for the standalone node.

***user-data* file example**

```
#cloud-config
#NxF User Data
fs_setup:
- label: data
  device: /dev/vdb
  filesystem: ext4

mounts:
- ["/dev/vdb", "/data"]
ntp:
  enabled: true
  ntp_client: chrony
  servers:
  - ntp.esl.cisco.com,10.58.228.1

nxf:
```

```
minControlPlaneCount: 1
clusterIndex: 1
clusterName: cluster1
node:
  name: node1
  #Should be same as control plane
  initiator: 192.168.119.10
  joinToken: z9wfwl.ye6pmu6pr27aogjk
  controlPlaneInterface: enp1s0
  eastboundInterface: enp3s0
  vip:
    northbound:
      interface: enp2s0
initiator:
  vip:
    northbound:
      ip: 10.58.231.119
      type: L3
  kafka:
    enabled: true
  joinToken: z9wfwl.ye6pmu6pr27aogjk
  arbitratorNode: node3
  #should be arbitrator for cluster3
  clusterRole: worker
  security:
    localUsers:
    - username: admin
      displayName: NxF Admin
      description: NextFusion Default Administrator
      locked: true
      mustChangePassword: false
      expiresInDays: 0
      access:
      - permission/admin
```

Create a KVM virtual machine for standalone deployment

Use this procedure to create a KVM virtual machine for standalone deployment. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Deploy a KVM virtual machine that hosts a standalone Cisco Optical Network Controller instance.

Run this task on the KVM host after you create the *cidata.iso* file for the standalone node.

Before you begin

- Complete Prepare KVM configuration files for standalone deployment.
- Verify the qcow image package before use. Each package contains all required files for verification. For detailed steps, see Verify a signed qcow2 or system pack image on page 108.

Use a unique QCOW2 filename for the VM.

Procedure

1. Copy the three configuration files to a designated folder and generate the ISO.

Example

```
cp network-config user-data meta-data
<PATH_TO_DEPLOY_DIR>/vmConfig/cloud-config/
mkisofs -o "<PATH_TO_DEPLOY_DIR>/vmConfig/cidata.iso" -r -J -V cidata
"<PATH_TO_DEPLOY_DIR>/vmConfig/cloud-config/"
```

2. Deploy the standalone node by running the **virt-install** command with the values for that VM.

   For the standalone and geo-ha deployment, specify `bridge0` as the <NORTHBOUND_BRIDGE> bridge interface to connect to the northbound interface using the north bridge interface.

   virt-install command syntax

```
virt-install \
--name <HOST_NAME> \
--vcpus <CPUs> \
--memory <MEMORY> \
--disk path=<PATH_TO_DEPLOY_DIR>/<QCOW2_FILE_NAME>.qcow2,format=qcow2 \
--disk
path=<PATH_TO_DEPLOY_DIR>/<QCOW2_FILE_NAME>-data.qcow2,size=<SIZE>,device=disk,bus=virtio,format=qcow2
 \
--disk path=<PATH_TO_DEPLOY_DIR>/vmConfig/cidata.iso,device=cdrom \
--osinfo debian12 \
--network bridge=<CONTROL_PLANE_BRIDGE>,model=virtio \
--network bridge=<NORTHBOUND_BRIDGE>,model=virtio \
--network bridge=<EASTBOUND_BRIDGE>,model=virtio \
--boot
loader=/usr/share/edk2/ovmf/OVMF_CODE.fd,loader.readonly=yes,loader_secure=no,nvram.template=/usr/share/edk2/ovmf/OVMF_VARS.fd,hd,uefi
 \
--serial pty \
--console pty,target_type=serial \
--noautoconsole
```

   Wait for the VM to finish booting. SSH access is available after the VM is started.

The standalone VM is created with the QCOW2 system disk, the data disk, the cloud-init ISO, and the required control-plane, northbound, and eastbound interfaces.

virt-install command example

```
virt-install   \
  --name conc-sa-kvm-1   \
  --vcpus 32   \
  --memory 131072   \
  --disk
path=/var/lib/libvirt/images/VMs/conc-sa-kvm-1/CONC-26.1.1.qcow2,format=qcow2
   \
  --disk
path=/var/lib/libvirt/images/VMs/conc-sa-kvm-1/CONC-26.1.1-data.qcow2,size=2500,device=disk,bus=virtio,format=qcow2
   \
  --disk path=/var/lib/libvirt/images/VMs/conc-sa-kvm-1/cidata.iso,device=cdrom
   \
  --osinfo debian12   \
  --network bridge=virbr-control,model=virtio   \
  --network bridge=bridge0,model=virtio   \
  --network bridge=virbr-east1,model=virtio   \
  --boot
loader=/usr/share/edk2/ovmf/OVMF_CODE.fd,loader.readonly=yes,loader_secure=no,nvram.template=/usr/share/edk2/ovmf/OVMF_VARS.fd,hd,uefi
   \
```

```
--serial pty   \
--console pty,target_type=serial   \
--autostart \
--noautoconsole
```

What to do next

Verify and access the standalone deployment.

Verify and access the standalone deployment

Use this procedure to connect to the standalone virtual machine, verify the system status, secure access, and open the web UI.

Confirm that the standalone Cisco Optical Network Controller VM is operational and complete the initial access and security tasks.

Before you begin

Deploy the VM by following Create a KVM virtual machine for standalone deployment.

Procedure

1. Verify SSH access to the standalone VM.

   Example

   ```
   ssh -i <private-key_file> nxf@<hco_management_ip>
   ```

   If you are not prompted for a password, there is probably a problem with the key. If the command times out, verify the IP setting.

2. Check the system status.

   Example

   ```
   sedo system status
   ```

3. Change the default password and access string of the admin user.

   Example

   ```
   sedo security user set --access role/admin --password admin
   ```

   You are prompted to enter the new password.

4. Configure additional local users if required.

   For more information, see Add local users to Cisco Optical Network Controller.

5. Open the Cisco Optical Network Controller web UI by browsing to the northbound virtual IP on port 8443.

   Example

   ```
   https://<northbound-virtual-ip>:8443/
   ```

Cisco Optical Network Controller is installed on KVM as a standalone instance.

Restore DB replication on standby cluster

Use this procedure to restore DB replication on standby cluster. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Restore DB replication on the standby cluster when replication does not automatically recover after a cluster failover and switchover.

When a cluster restarts after a prolonged outage or failover, DB replication may remain `disconnected`. If replication does not automatically recover within 10 minutes after boot, use this manual recovery procedure.

This procedure is not supported for restart scenarios. Use this procedure only after a failover.

Before you begin

Ensure that you perform this procedure during an appropriate maintenance window and that you create a full backup before you modify the standby cluster database.

Follow these steps to restore DB replication.

Procedure

1. Use the **sedo supercluster status** to verify the supercluster status on the active cluster.

Example

```
root@vm1-cluster1-node1:~# sedo supercluster status
┌──────────────────────────────────────────────────────────────────────────────────────────┐
│ Supercluster Status                                                                      │
├──────────────────┬───────────────────────────────────────────────────────────────────────┤
│ Cluster ID       │ k_Z8em3rUhM5AYDBlnZM2R2MSzQBI0AoJCMQYamjI                             │
│ Cluster Name     │ cluster1                                                              │
│ Cluster Role     │ worker                                                                │
│ Peers            │ cluster2 (worker,
QYCMqsFrx5flQi5kg0bgocuD8ZglAqi0VWHvMwP4zPo)     │
│                  │ cluster3 (arbitrator,
zW6eASGfeyMXUUzKJ7dLTF8Zx7bh3MIUtXy1Yjf8pF4) │
│ Mode             │ Running                                                               │
│ Current Active   │ cluster1                                                              │
│ Previous Active  │ cluster2                                                              │
│ Standby Clusters │ cluster2                                                              │
│ Last Switchover  │ 2026-04-20 13:20:44.506 +0000 UTC                                     │
│ Last Failover    │                                                                       │
│ Last Seen        │ controller-0.cluster1: 2026-03-20 13:59:49.403 +0000 UTC              │
│                  │ controller-0.cluster2: 2026-03-20 13:59:49.403 +0000 UTC              │
│                  │ controller-0.cluster3: 2026-03-20 13:59:49.403 +0000 UTC              │
```

```
           |
| Last Peer Error  |
           |
| Server Error     |
           |
| DB Replication   | disconnected
           |
| DB Lag           |
           |
```

2. Use the **sedo system status** command to verify that all services are running on the active cluster.

Example

```
sedo system status
┌──────────────────────────────────────────────────────────────────────────────┐
│ System Status (Fri, 20 Sep 2024 08:21:27 UTC)                                │
└──────┬──────────────────────────────┬──────┬─────────┬──────────┬─────────────┘
| OWNER  | NAME                         | NODE  | STATUS  | RESTARTS | STARTED
      |
├──────┼──────────────────────────────┼──────┼─────────┼──────────┼─────────────┤
| onc    | monitoring                   | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-alarm-service            | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-apps-ui-service          | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-circuit-service          | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-collector-service        | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-config-service           | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-devicemanager-service    | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-inventory-service        | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-nbi-service              | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-netconfcollector-service | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-osapi-gw-service         | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-pce-service              | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-pm-service               | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-pmcollector-service      | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-topology-service         | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-torch-service            | node1 | Running | 0        | 3 hours
 ago  |
| system | authenticator                | node1 | Running | 0        | 12
hours ago |
| system | controller                   | node1 | Running | 0        | 12
hours ago |
| system | flannel                      | node1 | Running | 0        | 12
hours ago |
| system | ingress-proxy                | node1 | Running | 0        | 12
```

```
hours ago |
| system | kafka                          | node1 | Running | 0        | 12
hours ago |
| system | loki                           | node1 | Running | 0        | 12
hours ago |
| system | metrics                        | node1 | Running | 0        | 12
hours ago |
| system | minio                          | node1 | Running | 0        | 12
hours ago |
| system | postgres                       | node1 | Running | 0        | 12
hours ago |
| system | promtail-cltmk                 | node1 | Running | 0        | 12
hours ago |
| system | vip-add                        | node1 | Running | 0        | 12
hours ago |
└────────┴────────────────────────────────┴───────┴─────────┴──────────┴──────────────┘
```

3. Select an appropriate maintenance window.
4. Create a full backup of the CONC database. To create a backup:
   - Follow the steps outlined at this topic Backup and Restore Database.
   - Use the `sedo backup create full` command and export the backup for recovery if needed.

5. To clean up the DB on the standby cluster and trigger a resync, perform these steps in order.
   a) Scale down the PostgreSQL stateful set on the standby cluster.

      Example

      ```
      sudo kubectl --kubeconfig /etc/kubernetes/admin.conf -n nxf-system scale
      --replicas 0 statefulset/postgres
      ```

   b) Delete the PostgreSQL persistent volumes and persistent volume claims.

      Example

      ```
      sudo kubectl --kubeconfig /etc/kubernetes/admin.conf -n nxf-system get
      pv,pvc | grep postgres | awk '{print $1}' | xargs sudo kubectl --kubeconfig
       /etc/kubernetes/admin.conf -n nxf-system delete
      ```

   c) Delete the PostgreSQL config maps.

      Example

      ```
      sudo kubectl --kubeconfig /etc/kubernetes/admin.conf -n nxf-system delete
       configmap -l app=postgres -l cluster-name=postgres
      ```

   d) Retrieve the remote worker cluster name.

      Example

      ```
      remote=$(kubectl -n nxf-system get cm nxf-supercluster-config -o
      jsonpath='{.data.config\.json}' | yq '.peers[] | select(.role == "worker")
       | .clusterName')
      ```

   e) Patch the PostgreSQL stateful set with the standby cluster target.

Example

```
kubectl -n nxf-system patch statefulset postgres -p
'{"spec":{"template":{"spec":{"containers":[{"name":"postgres","env":[{"name":"STANDBY_CLUSTER","value":"postgres-0.patroni.nxf-system.svc.${node}.local"}]}]}}}}'
```

f) Scale the PostgreSQL stateful set back up.

Example

```
sudo kubectl --kubeconfig /etc/kubernetes/admin.conf -n nxf-system scale
--replicas 1 statefulset/postgres
```

The standby cluster starts a full PostgreSQL resynchronization and DB replication is restored.

6. Use the **sedo supercluster status** to verify the supercluster status on the active cluster is *streaming*.

```
root@vm1-cluster1-node1:~# sedo supercluster status
┌──────────────────────────────────────────────────────────────────────────────┐
│ Supercluster Status                                                          │
├──────────────┬───────────────────────────────────────────────────────────────┤
│ Cluster ID   │ k_Z8em3rUhM5AYDBlnZM2R2MSzQBI0AoJCMQYamjI                     │
│ Cluster Name │ cluster1                                                      │
│ Cluster Role │ worker                                                        │
│ Peers        │ cluster2 (worker,
QYCMqsFrx5flQi5kg0bgocuD8ZglAqi0VWHvMwP4zPo)      │
│              │ cluster3 (arbitrator,
zW6eASGfeyMXUUzKJ7dLTF8Zx7bh3MIUtXy1Yjf8pF4) │
│ Mode         │ Running                                                       │
│ Current Active  │ cluster1                                                   │
│ Previous Active │ cluster2                                                   │
│ Standby Clusters │ cluster2                                                  │
│ Last Switchover  │ 2026-04-20 13:20:44.506 +0000 UTC                         │
│ Last Failover    │                                                           │
│ Last Seen        │ controller-0.cluster1: 2026-03-20 13:59:49.403 +0000 UTC  │
│                  │ controller-0.cluster2: 2026-03-20 13:59:49.403 +0000 UTC  │
│                  │ controller-0.cluster3: 2026-03-20 13:59:49.403 +0000 UTC  │
│ Last Peer Error  │                                                           │
│ Server Error     │                                                           │
│ DB Replication   │ streaming                                                 │
│ DB Lag           │                                                           │
└──────────────┴───────────────────────────────────────────────────────────────┘
```

DB replication is restored on the standby cluster.

2 Install and deploy geo-redundant Cisco Optical Network Controller

Topics:

Explains geo-redundant Cisco Optical Network Controller installation, including OVA deployment, supercluster setup, Web UI access, routing, switchovers, upgrades, KVM deployment, and image verification.

Geo-redundant deployment

Explains geo-redundant deployment, including the deployment context, dependencies, and operational behavior that affect Cisco Optical Network Controller installation or upgrade planning.

Geo-redundant deployment is a high-availability deployment model in which Cisco Optical Network Controller is deployed across geographically separate data centers to maintain service continuity during regional outages.

- Uses multiple Kubernetes clusters connected as a single geo supercluster
- Supports asynchronous replication between active and standby regions
- Maintains service availability through automated failover mechanisms

Geo redundancy protects against large-scale failures such as natural disasters, power outages, or data center loss.

How geo-redundant deployment works

In a geo-redundant deployment, Cisco Optical Network Controller clusters are grouped into a geo supercluster that enables coordinated service operation across regions.

Key architectural components include:

- **Active node**: Hosts operational Cisco Optical Network Controller services
- **Standby node**: Maintains synchronized state and assumes control during failover
- **Arbitrator node**: Participates in active node selection using the RAFT algorithm

The standard geo-redundant configuration is:

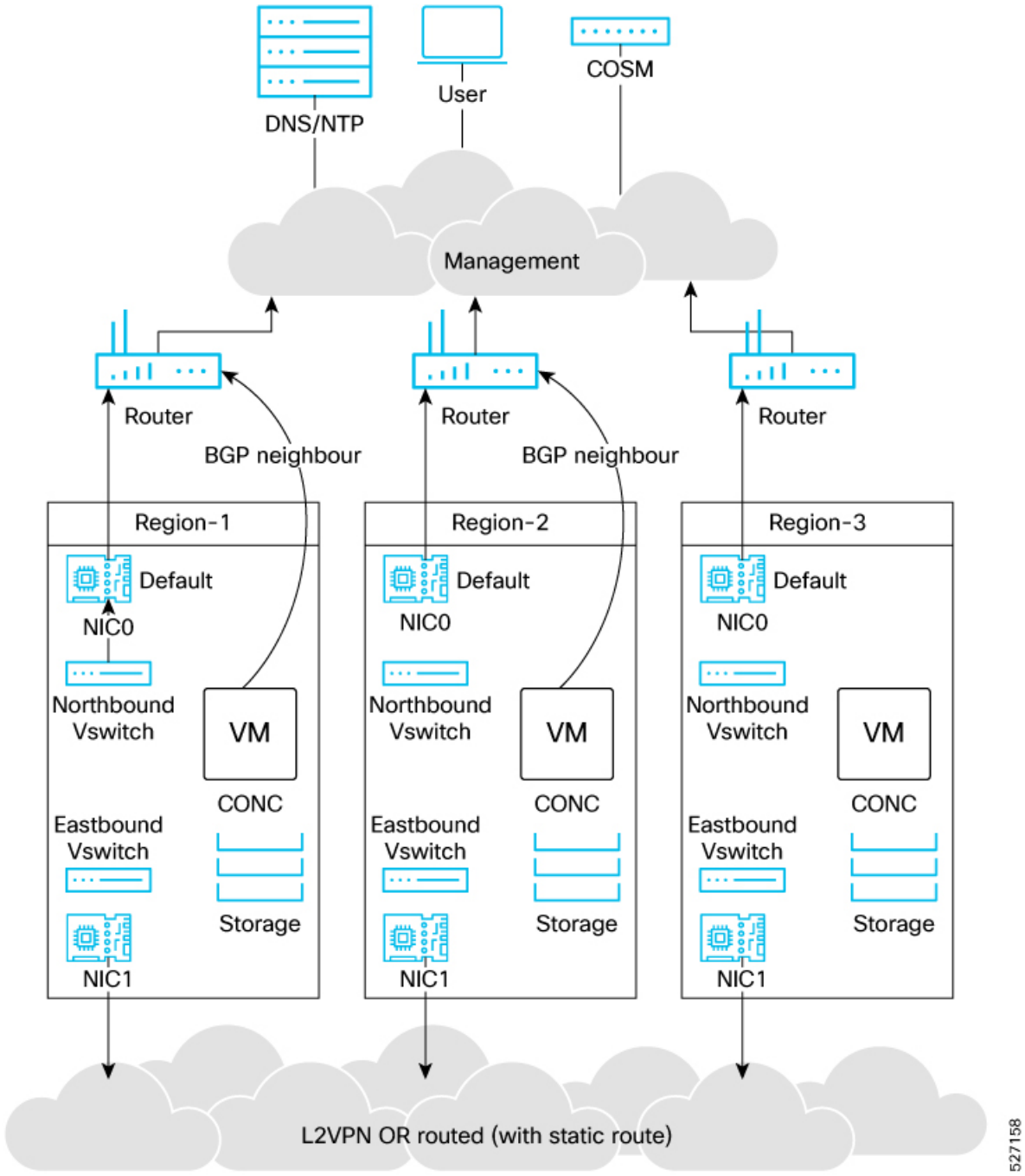
1. One active single-node worker
2. One standby single-node worker
3. One arbitrator node

Each region operates as an independent Kubernetes cluster while participating in the supercluster.

> **Note**
>
> The arbitrator node runs only the operating system and system services. Cisco Optical Network Controller microservices do not run on the arbitrator.

The following figure illustrates a typical geo-redundant deployment:

Figure 14: Cisco Optical Network Controller geo-redundant deployment



Use the following information to plan your geo-redundant deployment environment.

Infrastructure requirements include:

- **Platform**: VMware ESXi 7.0 and later, and vCenter 7.0 and later.

> **Attention**
>
> Upgrade to VMware vCenter Server 8.0 U2 if you are using VMware vCenter Server 8.0.2 or VMware vCenter Server 8.0.1.

- **Virtual machines**: Deploy three VMs (one per cluster) for a 1+1+1 supercluster.
  - One worker VM (active)
  - One worker VM (standby)
  - One arbitrator VM (witness)
- **Geographic separation**: Cisco recommends placing the three VMs in three different zones or regions to avoid a single point of failure. At least two of the three VMs must be reachable for service continuity.

VM sizing profiles:

Choose a profile based on your scale requirements.

This table lists the minimum hardware requirements for the base profile in HA mode with daily retention only.

Table 9: HA mode hardware requirements for base profile with daily retention

Profile	Worker Node CPU (in cores)	Arbitrator Node CPU (in cores)	Worker Node Memory (GB)	Arbitrator Node Memory (GB)	Solid State Drive (SSDs) (TB)
XS	16 vCPU	8	64	32	1
S	32 vCPU	8	128	32	2.5
M	48 vCPU	8	256	32	6

This table lists the minimum hardware requirements for the extended profile in HA mode with weekly and monthly retention.

Table 10: HA mode hardware requirements for extended profile with weekly and monthly retention

Profile	Worker Node CPU (in cores)	Arbitrator Node CPU (in cores)	Worker Node Memory (GB)	Arbitrator Node Memory (GB)	Solid State Drive (SSDs) (TB)
XS	16 vCPU	8	64	32	1.5
S	32 vCPU	8	128	32	5
M	48 vCPU	8	256	32	12

> **Attention**
>
> Cisco Optical Network Controller supports only SSDs for storage.

vCPU to physical CPU core ratio: A ratio of 2:1 is supported when hyperthreading is enabled and supported by the hardware. Otherwise, use 1:1.

Network requirements:

A geo-redundant deployment uses three networks.

- **Control plane network**: Internal communication within a cluster. In geo-redundant deployments, the control plane network is used only within a single cluster. You can use a dummy vSwitch for a cluster and apply the same configuration to each cluster.
- **Northbound (VM) network**: Traffic between users and the cluster, including web UI access. Cisco Optical Network Controller uses this network to connect to Cisco Optical Site Manager devices using NETCONF/gRPC.

  Bandwidth and latency requirements for the northbound network:

  - Web UI: 1 Gbps
  - Connection to optical nodes: 100 Mbps
  - Latency: less than 100 ms

- **Eastbound network**: Internal communication across regions within the supercluster. Active and standby nodes use this network to replicate databases. Postgres is replicated between active and standby nodes. MinIO is replicated on the arbitrator.

  Bandwidth and latency requirements for the eastbound network: 1 Gbps bandwidth and latency less than 100 ms.

  You can configure the eastbound network as a flat Layer 2 network or an L2VPN where eastbound IP addresses are in the same subnet. If eastbound IP addresses are in different subnets, configure static routing between nodes for eastbound connectivity.

Restriction

Do not configure the control plane, northbound, and eastbound networks in the same subnet or VLAN segment. Use separate subnets and VLAN segments.

Virtual IP routing: BGP is used to route traffic to the virtual IP from multiple locations. Configure the BGP router and add the nodes as neighbors. Coordinate with your network administrator to configure BGP.

Storage requirements: Use SSD storage that meets the disk write latency requirement of ≤ 100 ms.

Deployment constraints include:

- You need three separate VMs with separate eastbound, northbound, and control plane network connectivity.
- You cannot remove nodes from a cluster or change cluster roles after a cluster joins a supercluster.

Default port assignments:

This table lists the default port assignments.

Table 11: Communications matrix

Traffic type	Port	Description
Inbound	TCP 22	SSH remote management
	TCP 8443	HTTPS for UI access
Outbound	TCP 830	NETCONF to Cisco Optical Site Manager devices
	TCP 389	LDAP if using Active Directory
	TCP 636	LDAPS if using Active Directory
	Customer specific	HTTP access to an SDN controller
	User specific	HTTPS access to an SDN controller
	TCP 3082, 3083, 2361, 6251	TL1 to optical devices
Eastbound	TCP 10443	Supercluster join requests
	UDP 8472	VXLAN
Syslog	User specific	TCP/UDP
Control plane ports (internal, not exposed)	TCP 443	Kubernetes
	TCP 6443	Kubernetes
	TCP 10250	Kubernetes
	TCP 2379	etcd
	TCP 2380	etcd
	UDP 8472	VXLAN
	ICMP	Ping between nodes (optional)

Installation files:

Cisco Optical Network Controller is released as a single VMware OVA distribution. The OVA includes an OVF descriptor and virtual disk files that contain the operating system and Cisco Optical Network Controller installation files. You can deploy the OVA using vCenter on ESXi hosts for standalone or supercluster deployments.

> **Note**
>
> During OVF deployment, the deployment is aborted if there is an internet disconnection.

Geo-redundant deployment limitations and behavior

Provides details about geo-redundant deployment limitations and behavior, including requirements, limitations, values, or configuration data that support Cisco Optical Network Controller installation and deployment decisions.

Use this reference to understand operational limitations and system behavior during switchover and failover events in geo-redundant deployments.

- **Switchover:** A planned, manual transition of services from the active node to a standby node.
- **Failover:** An automatic transition that occurs when the active node becomes unavailable or unreachable.

Geo-redundant deployments have these limitations and behavioral characteristics:

- **Replication lag**: Geo redundancy uses asynchronous replication. If a switchover or failover occurs during an ongoing operation, there is a small risk of data loss when network latency is high.

  The newly active node might not have information about an in-progress operation because the database transaction was not fully replicated.

  For example, if a node or circuit delete operation completes on the active node but a switchover occurs before replication finishes, the new active node might still display the deleted object. Retry the operation to resolve the issue.

- **Double failures**: For Cisco Optical Network Controller releases 25.1.2 and earlier, if two out of three nodes are down or unreachable, the remaining node transitions to a standby state.

  During recovery, one virtual machine (VM) is designated as active, but no failover alarm is generated. Cisco Optical Network Controller cannot be accessed using the virtual IP address until at least one additional node becomes available. The active VM is decided dynamically based on election.

- **Consistent role state enforcement**: In the GeoHA design, if a role transition request is received while another role transition is still being processed, Cisco Optical Network Controller automatically restarts the network service.

  This behavior ensures a clean and accurate view of the active role and prevents inconsistent role status across nodes.

- **Northbound notification loss**: During a switchover or failover, the northbound virtual IP interface is temporarily unreachable.

  During this interruption, event notifications sent to hierarchical or external controllers are lost. Cisco Optical Network Controller releases 24.x.x and 25.x.x do not support notification replay.

- **Performance monitoring (PM) data loss**: The 15-minute and 1-day PM buckets collected during a switchover or failover event are lost are lost and cannot be recovered.

  PM data collection resumes normally with the next bucket after the switchover or failover alarm clears.

- **SWIM job failures**: Any SWIMU ad hoc device configuration backup jobs that are running during a switchover or failover transition to the *Failed* state.

  Recreate the job to trigger the backup again. Scheduled SWIM jobs that are in progress also fail, but future scheduled executions continue according to the configured schedule.

- **Data corruption during restore operations**: Cisco Optical Network Controller supports database restore operations only on the active node.

If a switchover or failover occurs while a restore operation is in progress, database corruption can occur. In this case, controller services might not return to the ready state.

Perform the restore operation again to recover the cluster.

- **Switchover and failover duration**: Before triggering a manual switchover, verify that all microservices on both active and standby nodes are in the ready state by running the `sedo system status` command.

  A switchover or failover requires approximately 4 minutes to complete. Do not initiate another switchover during this period.

  After a node failover, the failed node requires approximately 15 to 20 minutes to become ready for a subsequent switchover or failover. Triggering another event before the node is ready can result in a double failure.

  When TAPI is enabled, switchover time can exceed 4 minutes depending on the number of devices and circuits.

- **Web UI unavailability during failover**: During a failover event, the Cisco Optical Network Controller web UI is unavailable until the failover process completes.

  This unavailability typically lasts approximately 4 minutes. After the failover completes, refresh the browser to regain access. To confirm a failover, review the switchover alarm in the Alarm History.

- **Incomplete circuit configurations**: Incomplete circuit configurations: If a circuit is partially provisioned and a switchover or failover occurs before database replication completes, the system can create incomplete or disconnected configurations.

  Manually clean up these configurations in the Cross-Connect tab in Cisco Optical Site Manager.

Install and deploy geo-redundant Cisco Optical Network Controller

Use this procedure to install and deploy geo-redundant Cisco Optical Network Controller. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Deploy the Cisco Optical Network Controller OVA for each supercluster node. Deploy a separate OVA for every node in vCenter.

For background and deployment considerations, see Geo-redundant deployment overview.

Use the same template values across all nodes and adjust only node-specific settings. Assign consistent names and configure network mappings for every node.

Before you begin

Review prerequisites in Geo-redundant deployment prerequisites.

Follow these steps to deploy the OVA for each supercluster node.

Procedure

1. Right-click the ESXi host in the vSphere client screen and click **Deploy OVF Template**.
2. In the **Select an OVF template** screen, choose the **URL** button to download and install the OVF package from the Internet.

   Alternatively, select the **Local file** radio button to upload the OVA files from your local system, then click **Next**.

   Figure 15: Select an OVF Template

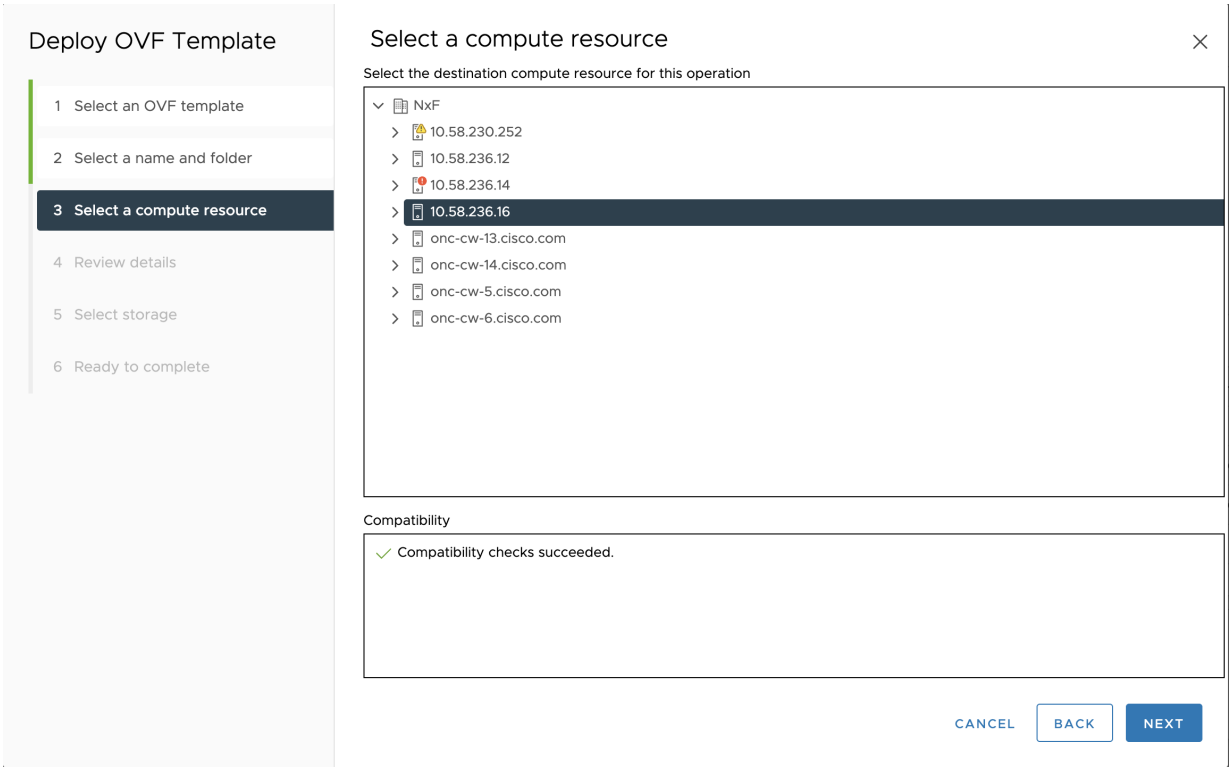
3. In the **Select a name and folder** screen, specify a unique name for the virtual machine instance. Select the VM location and click **Next.**

Note

Choose the data center and location for each virtual machine based on your deployment requirements. Compute resources in the subsequent step will appear according to your selection.

4. In the **Select a compute resource** screen, select the destination for the VM. In the **Review details** screen, verify the template details and click **Next**.

Figure 16: Select a Compute Resource

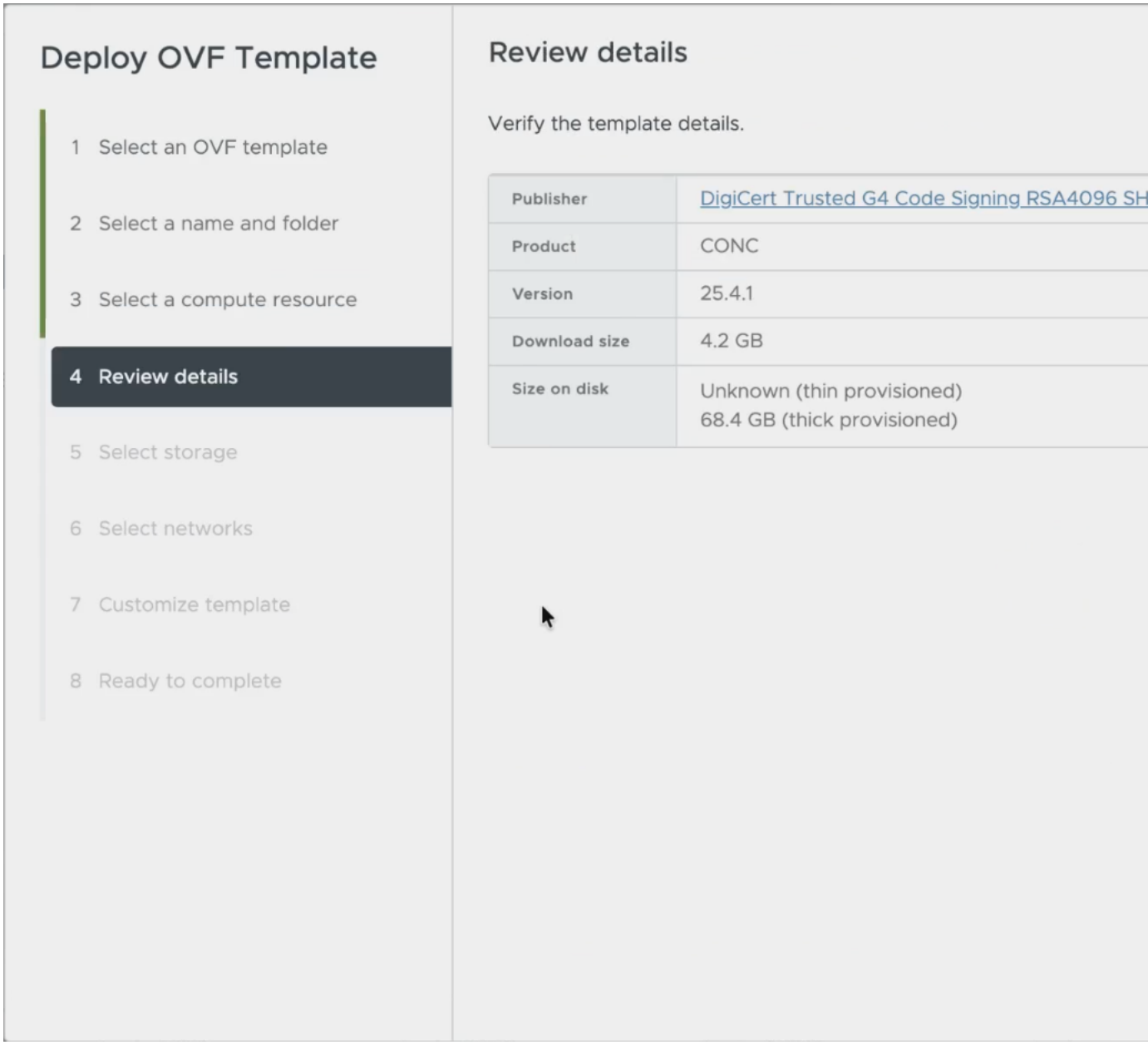


Note

The compatibility check proceeds until it completes successfully.

Figure 17: Review Details

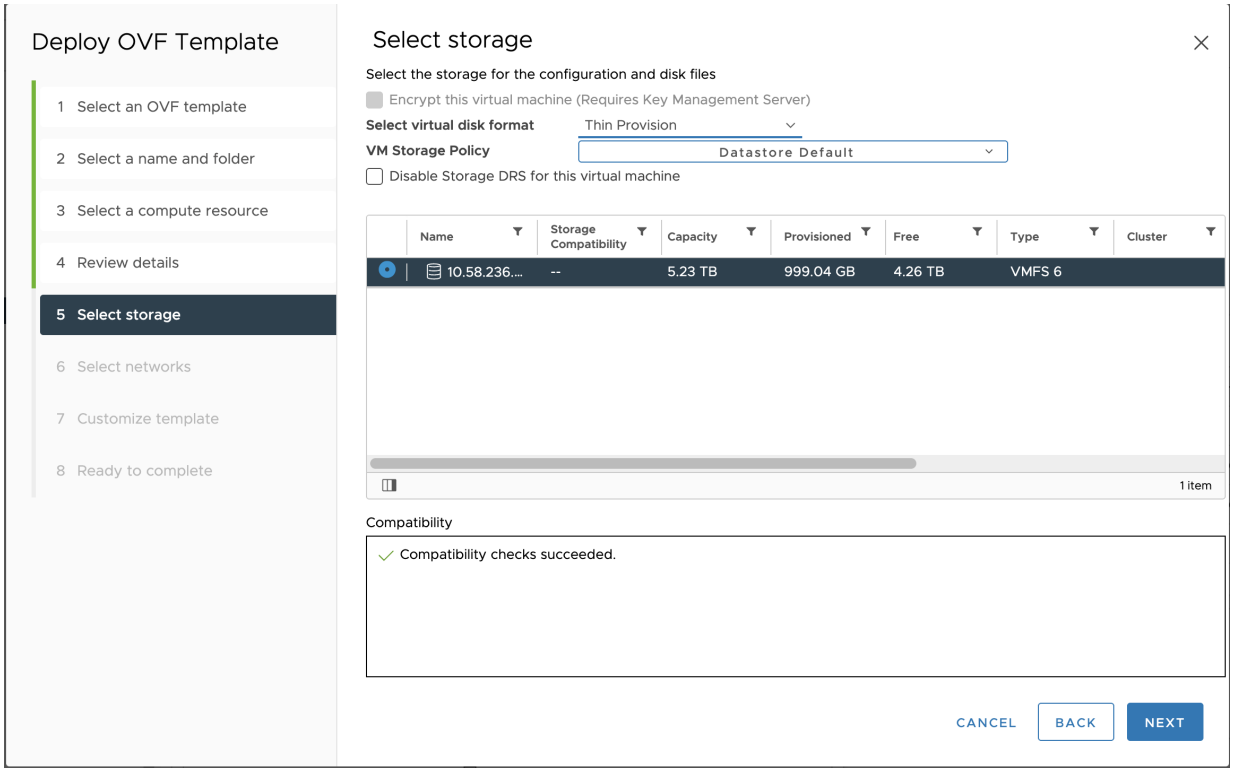
Figure 18: Review Details



The **Publisher** field shows the trusted publisher information. For example, *DigiCert Trusted G4 Code Signing RSA4096 SHA384 2021 CA1 (Trusted certificate)*.

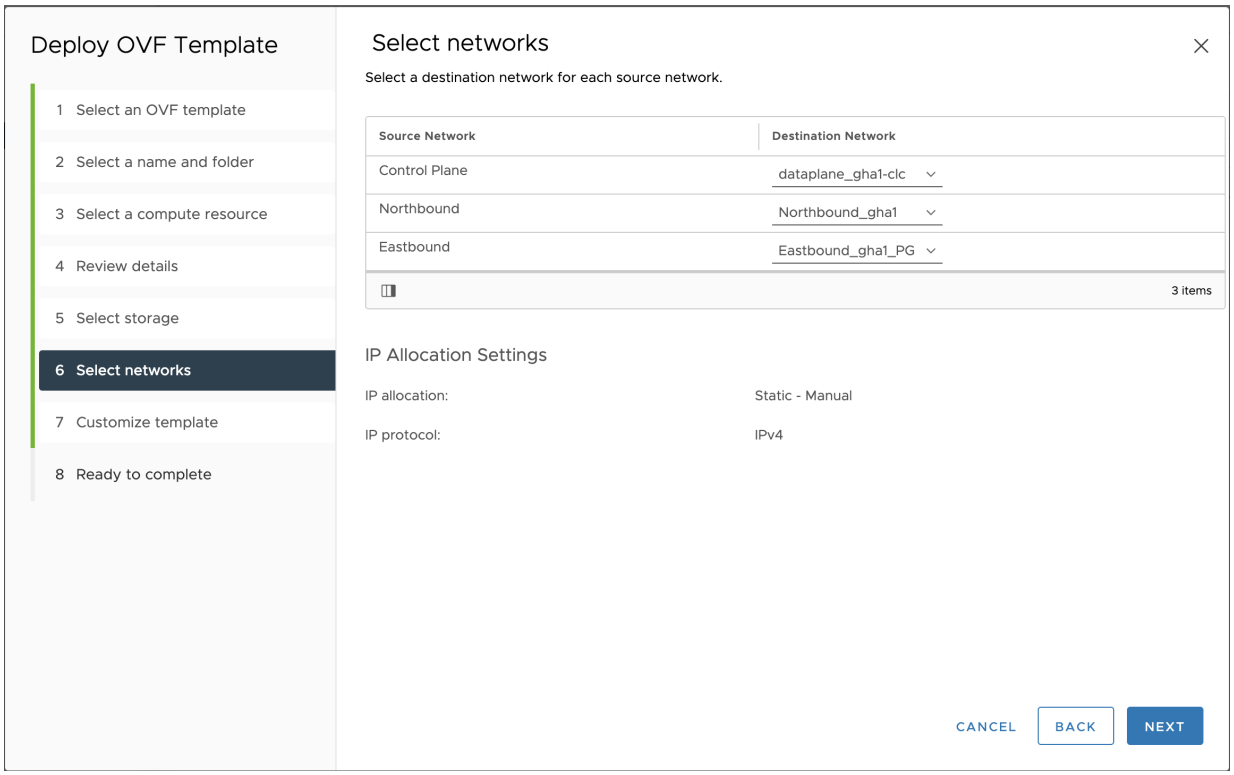
5. In the **Select storage** screen, select the virtual disk format according to your requirements. **VM Storage Policy** is set as *Datastore Default* and click **Next**. Select the **virtual disk format** as *Thin Provision*.

Figure 19: Select Storage



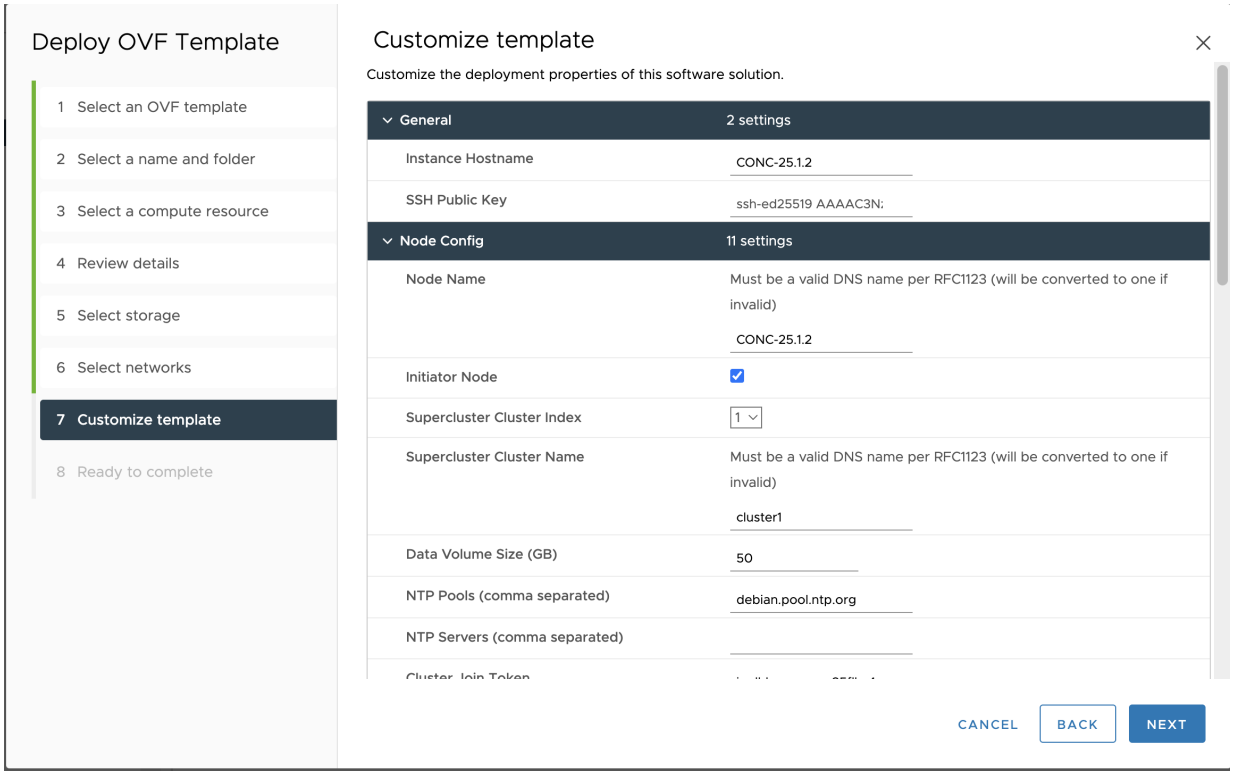
6. In the **Select networks** screen, select the Control Plane, Eastbound, and Northbound networks you created for each VM and **click Next**.

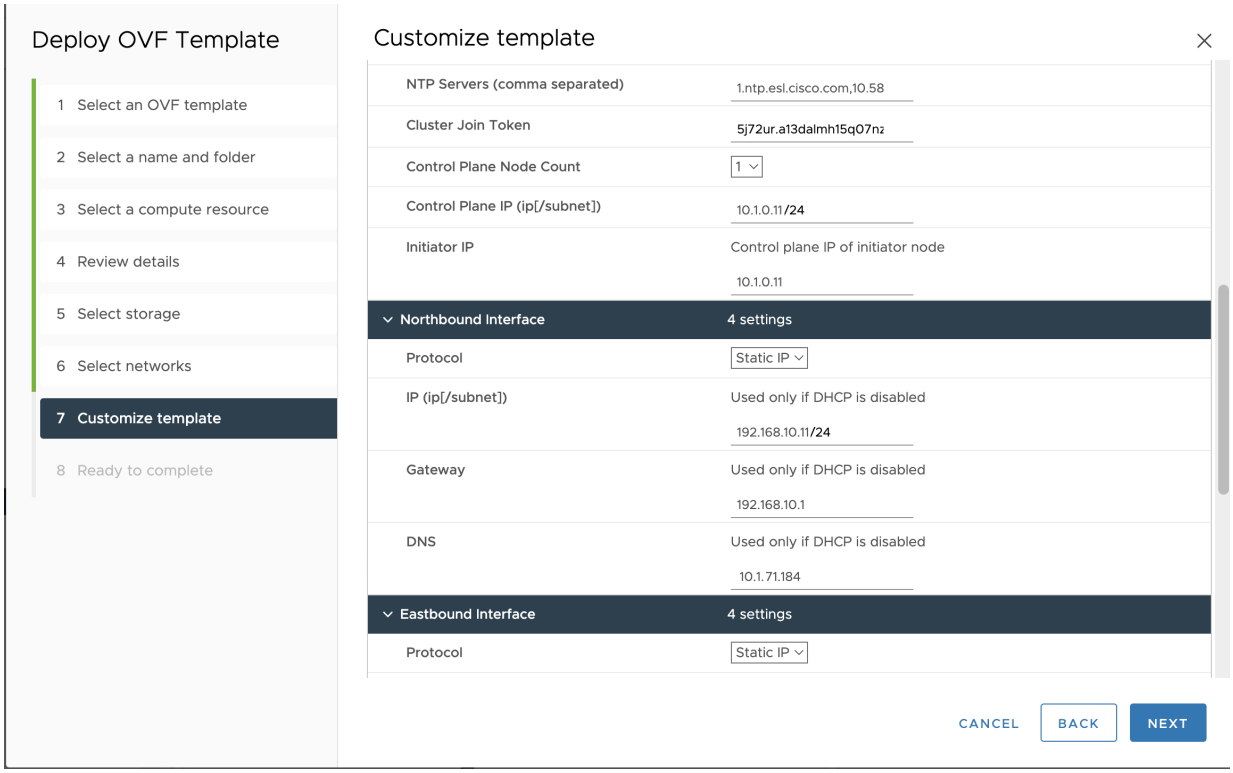
Figure 20: Select Networks

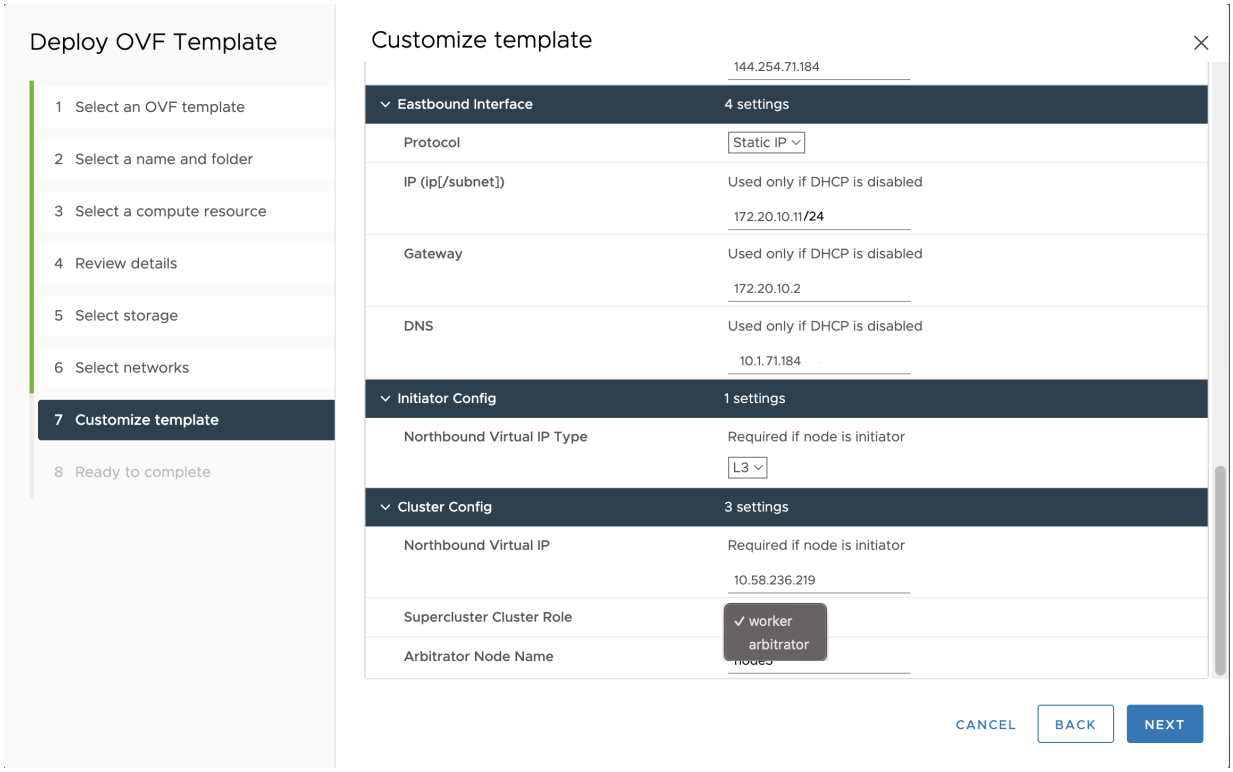


7. In the **Customize template** screen, set the values using the following table as a guideline.

Figure 21: Customize Template


Deploy OVF Template
1 Select an OVF template
2 Select a name and folder
3 Select a compute resource
4 Review details
5 Select storage
6 Select networks
7 Customize template
8 Ready to complete
Customize template
Customize the deployment properties of this software solution.
General
2 settings
Instance Hostname
CONC-25.1.2
SSH Public Key
ssh-ed25519 AAAAC3N:
Node Config
11 settings
Node Name
Must be a valid DNS name per RFC1123 (will be converted to one if invalid)
CONC-25.1.2
Initiator Node
Supercluster Cluster Index
1
Supercluster Cluster Name
Must be a valid DNS name per RFC1123 (will be converted to one if invalid)
cluster1
Data Volume Size (GB)
50
NTP Pools (comma separated)
debian.pool.ntp.org
NTP Servers (comma separated)
CANCEL
BACK
NEXT


Deploy OVF Template
1 Select an OVF template
2 Select a name and folder
3 Select a compute resource
4 Review details
5 Select storage
6 Select networks
7 Customize template
8 Ready to complete
Customize template
NTP Servers (comma separated)
1.ntp.esl.cisco.com,10.58
Cluster Join Token
5j72ur.a13dalmh15q07nz
Control Plane Node Count
1
Control Plane IP (ip[/subnet])
10.1.0.11/24
Initiator IP
Control plane IP of initiator node
10.1.0.11
Northbound Interface
4 settings
Protocol
Static IP
IP (ip[/subnet])
Used only if DHCP is disabled
192.168.10.11/24
Gateway
Used only if DHCP is disabled
192.168.10.1
DNS
Used only if DHCP is disabled
10.1.71.184
Eastbound Interface
4 settings
Protocol
Static IP
CANCEL
BACK
NEXT



For the arbitrator node, choose *arbitrator* as the **Supercluster Cluster Role**.

Table 12: Customize Template

Key	Values
General	
Instance Hostname	*<instance-name>* Must be a valid DNS name per RFC1123.1.2.4. ▪ Contain at most 63 characters. ▪ Contain only lowercase alphanumeric characters or '-' ▪ Start with an alphanumeric character. ▪ End with an alphanumeric character.
SSH Public Key	*<ssh-public-key>*. Used for SSH access that allows you to connect to the instances securely without the need to manage credentials for multiple instances. SSH public key must be a ed25519 key. See SSH Key Generation.
Node Config	
Node Name	Use the same name as *Instance Hostname*
Initiator Node	Select the check box
Supercluster Cluster Index	Set to **1** (active cluster), **2** (standby cluster), or **3** (arbitrator).

Key	Values
Supercluster Cluster Name	Set to **cluster1** (active cluster), **cluster2** (standby cluster), or **cluster3** (arbitrator).
Data Volume Size (GB)	Configure data volume according to the VM profile.
NTP Pools (comma separated)	(Optional) A comma-separated list of the NTP pools. For example, debian.pool.ntp.org
NTP Servers (comma separated)	A comma-separated list of the NTP servers.
Cluster Join Token	Autogenerated value. Leave as is.
Control Plane Node Count	1
Control Plane IP (ip[/subnet])	<Private IP for the Instance> Control Plane Network
Initiator IP	<Same IP as Control Plane> Control Plane Network
Northbound Interface	
Protocol	Static IP
IP (ip[/subnet]) - if not using DHCP	<Public IP for the Instance> Northbound Network
Gateway - if not using DHCP	<Gateway IP for the Instance> Northbound Network
DNS	DNS Server IP
Eastbound Interface	
Protocol	Static IP
IP (ip[/subnet]) - if not using DHCP	< IP for the Instance> Eastbound Network
Gateway - if not using DHCP	<Gateway IP for the Network> Eastbound Network
DNS	DNS Server IP
Initiator Config	
Northbound Virtual IP Type	L3
Cluster Config	
Northbound Virtual IP	Virtual IP for the SuperCluster
Supercluster Cluster Role	*worker* for primary and secondary nodes *arbitrator* for arbitrator node

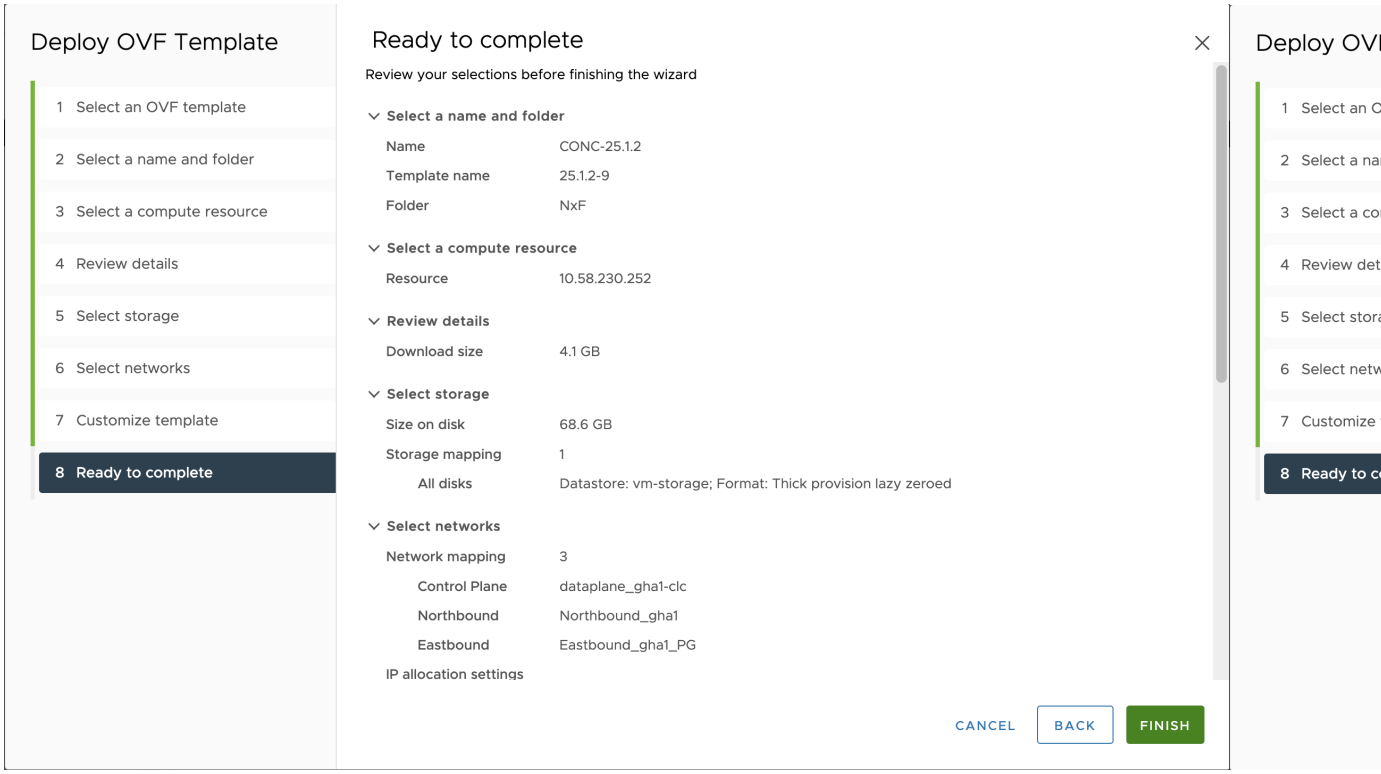
Key	Values
Arbitrator Node Name	a unique node name. **Attention** • The arbitrator node name must not the same as any node in the supercluster. This field must not be the same as the node name of the arbitrator node either. • The arbitrator node name must be the same across all nodes in the supercluster.

Restriction

Do not configure the Northbound and Eastbound networks in the same subnet or VLAN segment. Use separate subnets and VLAN segments for these networks.

8. In **Review the details** screen, review all your selections and click **Finish**. To check or change any properties from the review screen, before clicking **Finish**, click **BACK** to return to the **Customize template** screen.

Figure 22: Ready to Complete



9. Repeat the step 8 three times to create two worker node VMs (active and standby) and one arbitrator node VM.

Attention

- You can create the other nodes at a different data center, host, or vCenter instance as needed. Ensure Eastbound and Northbound network connectivity between the nodes.
- Upon activation of the VM, it does not respond to ping requests. However, you can log in using SSH if the installation is successful.

The OVA deployment is completed for the supercluster nodes.

What to do next

Set up the supercluster on page 65

Set up the supercluster

Use this procedure to set up the supercluster, including SSH access, static eastbound routes, cluster joining, connectivity checks, BGP routing, and service validation.

Set up the supercluster by completing the required configuration and validation tasks.

Complete the subtasks in the order listed to prepare networking, routing, and cluster membership before starting the supercluster.

Before you begin

Before you begin, you must have created three VMs for geo-redundant deployment of Cisco Optical Network Controller. For more details, see Install and deploy geo-redundant Cisco Optical Network Controller

Procedure

1. Complete Connect to the supercluster virtual machines.
2. Complete Configure eastbound routes between supercluster nodes.
3. Complete Join clusters into a supercluster.
4. Complete Verify connectivity and start the supercluster.
5. Complete Configure Border Gateway Protocol for supercluster routing.
6. Complete Validate supercluster services and version.

The supercluster setup tasks are complete.

Connect to supercluster VM using SSH keys

Use this procedure to connect to supercluster VM using SSH keys. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Connect to each supercluster VM using SSH keys for secure access.

Use the PEM key generated during SSH key setup to access each node.

Before you begin

- Confirm that three VMs have been created for geo-redundant deployment of Cisco Optical Network Controller. For more details, see Install and deploy geo-redundant Cisco Optical Network Controller on page 57.
- Verify that each VM is powered on. Wait for the IP addresses for the VMs on vSphere to appear.

Follow these steps to connect to the supercluster virtual machines.

Procedure

1. Connect to each VM using the PEM key generated during SSH Key Generation.
2. Log in to each VM using the private key.

```
# ssh -i <private-key_file> nxf@<node_ip>
```

> **Note**
>
> - If you are prompted for a password, there might be a problem with the key. If your SSH key has a passphrase, the system prompts you for the passphrase. If you are prompted for a password even after entering your SSH key passphrase, your PEM key might be wrong or corrupted.
> - If the command times out, check your network settings and make sure the node is reachable.
> - After the nodes are deployed, check the OVA deployment progress in the Tasks console of vSphere Client. Upon successful deployment, Cisco Optical Network Controller can take about 20 minutes to boot.
> - The default user ID is *admin*. Set the password using the `sedo security user set admin --password` command.

You are connected to each supercluster VM.

What to do next

Configure static eastbound routes between supercluster nodes on page 66

Configure static eastbound routes between supercluster nodes

Use this procedure to configure static eastbound routes between supercluster nodes. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Configure static routes to allow eastbound traffic to pass between supercluster nodes. If peer node eastbound IPs are in different subnets, create static routes for eastbound traffic between the nodes.

Before you begin

Connect to supercluster VM using SSH keys on page 65

Follow these steps to configure eastbound routes.

Procedure

1. Navigate to the configuration directory.

```
cd /etc/systemd/network/
```

2. Identify the network configuration file for the eastbound interface *ens256*. For example, it may be named *10-cloud-init-ens256.network*.
3. Open the configuration file with administrative privileges. Update the **[Route]** section by adding the static routes using this template.

> **Note**
>
> Replace all placeholders with the actual IP addresses and gateway information.

```
[Match]
Name=ens256

[Network]
DHCP=no
DNS=<dns-server-ip>

[Address]
Address=<cluster1-eastbound-ip>/<subnet-mask>

[Route]
Destination=<eastbound-subnet-of-cluster2>/<subnet-mask>
Gateway=<gateway-ip>

[Route]
Destination=<eastbound-subnet-of-cluster3>/<subnet-mask>
Gateway=<gateway-ip>
```

4. Save the file. Exit the editor.

Example

```
# Example
[Match]
Name=ens256

[Network]
DHCP=no
DNS=10.10.128.236

[Address]
Address=172.10.10.11/24

[Route]
Destination=172.10.20.0/24
Gateway=172.30.10.2

[Route]
Destination=172.10.30.0/24
Gateway=172.30.10.2
```

> **Note**
>
> - Verify that the `Name` in the `[Match]` section mateches the correct network interface.
> - Verify that the DNS and gateway IPs are correctly assigned for your network.

5. Use the **ping** command to verify connectivity between the nodes.
6. Restart the `systemd-networkd` service to apply the changes.

Example

```
sudo systemctl restart systemd-networkd
```

7. Verify that the routes are created.

```
ip route
```

The eastbound routes are configured. Connectivity has been verified.

Join clusters into a supercluster

Use this procedure to join clusters into a supercluster. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Join three clusters into a single supercluster.

Before you begin

- Complete the Border Gateway Protocol (BGP) and eastbound routing configuration before you join clusters.
- Ensure you can run commands on each cluster node.

Follow these steps to join clusters into a supercluster.

Procedure

---

1. Use the **sedo supercluster status** command on each node to retrieve the cluster ID.

Example

```
sedo supercluster status
# Sample output
┌──────────────────────────────────────────────────────────────────┐
│ Supercluster Status                                              │
├──────────────┬───────────────────────────────────────────────────┤
│ Cluster ID   │ vk0uFBSwM1vX4_mC1BAabDxAKXYUTv1KH5dcCDawZw4       │
│ Cluster Name │ cluster1                                          │
│ Cluster Role │ worker                                            │
│ Peers        │ <No Peers>                                        │
│ Initialized  │ No                                                │
└──────────────┴───────────────────────────────────────────────────┘
```

> **Note**
>
> You need the cluster ID for each node for the next steps.

2. Connect *cluster1* to *cluster2*.
   a) Initiate the connection on cluster1.

   Example

   ```
   # Sample output
   sudo sedo supercluster wait-for -b 172.20.2.89:10443
   uUD21AaV4cQ8CzZQf0E0YrGmALi0vHASpZI07YzcsQ
   Listening for join requests on 172.20.2.89:10443...
   Please run the following on peer node:
   $ sudo /usr/bin/sedo supercluster join
   Lh9Gv3FwSUsx7Gu_7EJoIMe4r5YE6ApyHqOEt83fko
   https://172.20.2.89:10443/join/g4jKVulJo74ptz82lMvngQ
   ```

   b) Run the join command generated by cluster1 on cluster2.

   Example

   ```
   sudo /usr/bin/sedo supercluster join
   Lh9Gv3FwSUsx7Gu_7EJoIMe4r5YE6ApyHqOEt83fko
   https://172.20.2.89:10443/join/g4jKVulJo74ptz82lMvngQ
   ```

3. Connect *cluster1* to *cluster3*.
   a) Initiate the connection on cluster1.

   ```
   sudo sedo supercluster wait-for -b <cluster1_node_eastbound_ip>:10443
   <cluster3_node_cluster_id>
   ```

   b) Run the join command generated by cluster1 on cluster3.
4. Connect *cluster2* to *cluster3*.
   a) Initiate the connection on cluster2.

   ```
   sudo sedo supercluster wait-for -b <cluster2_node_eastbound_ip>:10443
   <cluster3_node_cluster_id>
   ```

   b) Run the join command generated by cluster2 on cluster3.

Clusters are joined and ready for connectivity validation.

Verify cluster connectivity and start the supercluster

Use this procedure to verify cluster connectivity and start the supercluster. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Verify connectivity, start the supercluster, and confirm its operational status.

Before you begin

Ensure that clusters are joined and reachable.

Follow these steps to verify connectivity and start the supercluster.

Procedure

1. Use the **sedo supercluster connectivity** command to verify connectivity between clusters.

   Note

   Wait until all connections are successful. Clusters typically establish connectivity within 5 minutes.

   Example

```
sudo sedo supercluster connectivity
```

Supercluster Connectivity			
FROM	TO	RTT	RESULT
cluster2/controller-0	cluster1/controller-0	14ms	Success
cluster2/controller-0	cluster3/controller-0	15ms	Success
cluster1/controller-0	cluster3/controller-0	12ms	Success
cluster1/controller-0	cluster2/controller-0	12ms	Success
cluster3/controller-0	cluster2/controller-0	13ms	Success
cluster3/controller-0	cluster1/controller-0	13ms	Success

2. Use the **sedo supercluster start** command to start the supercluster.

   Note

   The node where you execute this command becomes the active node. The other worker node becomes the standby node.

   Example

```
sudo sedo supercluster start

Checking Supercluster connectivity...Passed
Initiating Supercluster...Done
```

3. Use the **sedo supercluster status** to verify the supercluster status.

Example

This sample output shows the result of the status command on the standby node. When **DB replication** is `streaming` and **DB Lag** is `0 bytes`, the geo-redundant deployment is running.

```
sedo supercluster status
┌──────────────────┬──────────────────────────────────────────────────────────────────────────────┐
│ Supercluster Status                                                                             │
├──────────────────┼──────────────────────────────────────────────────────────────────────────────┤
│ Cluster ID       │ QgQV2uXgP1udqshlIssyTwf3LZzEyRh6I3z5MH8almA                                  │
│ Cluster Name     │ cluster1                                                                     │
│ Cluster Role     │ worker                                                                       │
│ Peers            │ cluster2 (worker,                                                            │
jaWeN9BdXUUTxvofwt6Hukt6OQXIUaqo4NxN6zHYDc)                                                       │
│                  │ cluster3 (arbitrator,                                                        │
SUCrwqQjXToG5GKBwckcg_CtzgHstQigaEM1X0988E)                                                       │
│ Mode             │ Running                                                                      │
│ Current Active   │ cluster1                                                                     │
│ Previous Active  │                                                                              │
│ Standby Clusters │ cluster2                                                                     │
│ Last Switchover  │                                                                              │
│ Last Failover    │                                                                              │
│ Last Seen        │ controller-0.cluster2: 2025-03-19 11:16:57.051 +0000 UTC                     │
│                  │ controller-0.cluster3: 2025-03-19 11:16:57.047 +0000 UTC                     │
│                  │ controller-0.cluster1: 2025-03-19 11:16:57.051 +0000 UTC                     │
│ Last Peer Error  │                                                                              │
│ Server Error     │                                                                              │
│ DB Replication   │ streaming                                                                    │
│ DB Lag           │ 0 bytes                                                                      │
└──────────────────┴──────────────────────────────────────────────────────────────────────────────┘
```

The supercluster is running and connectivity is confirmed.

Configure BGP for supercluster routing

Use this procedure to configure BGP for supercluster routing. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Configure BGP so the supercluster can advertise the virtual IP route.

Before you begin

- Obtain the router IP address, autonomous system number, and password from your network administrator before you begin.
- Confirm that each node can reach the northbound network.

Follow these steps to configure BGP for supercluster routing.

Procedure

1. Initialize BGP on cluster 1 and 2 nodes.

```
sedo ha bgp init <current_node_name> <current_node_northbound_ip>
<current_node_as> --nexthop <current_node_northbound_ip>
```

2. Add a BGP router to each node.

   sedo ha bgp router add *<current_node_name> <bgp_router_ip> <bgp_router_as> <bgp_password>* **--ttl-min** *255*

   > **Note**
   >
   > Collect the BGP router IP address, router autonomous system number, and BGP password from your network administrator. The BGP password must match the neighbor configuration on the router.

   Example

```
sedo ha bgp router add conc2512-2 192.168.125.1 65534 password --ttl-min 255
```

BGP is initialized, and the router is added to each node.

Validate supercluster service status and installed version

Use this procedure to validate supercluster service status and installed version. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Validate the service status and confirm the installed version.

Run these checks after the supercluster has started.

Before you begin

Follow these steps to validate services and version.

Procedure

1. Check the status of all pods.

```
sedo system status
┌──────────────────────────────────────────────────────────────────────────────┐
│ System Status (Fri, 20 Sep 2024 08:21:27 UTC)                                │
├─────────┬──────────────────────────────┬──────┬─────────┬──────────┬──────────┤
│ OWNER   │ NAME                         │ NODE │ STATUS  │ RESTARTS │ STARTED  │
├─────────┼──────────────────────────────┼──────┼─────────┼──────────┼──────────┤
│ onc     │ monitoring                   │ node1 │ Running │ 0        │ 3 hours ago │
│ onc     │ onc-alarm-service            │ node1 │ Running │ 0        │ 3 hours ago │
│ onc     │ onc-apps-ui-service          │ node1 │ Running │ 0        │ 3 hours ago │
```

```
| onc    | onc-circuit-service          | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-collector-service        | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-config-service           | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-devicemanager-service    | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-inventory-service        | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-nbi-service              | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-netconfcollector-service | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-osapi-gw-service         | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-pce-service              | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-pm-service               | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-pmcollector-service      | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-topology-service         | node1 | Running | 0        | 3 hours
 ago  |
| onc    | onc-torch-service            | node1 | Running | 0        | 3 hours
 ago  |
| system | authenticator                | node1 | Running | 0        | 12
hours ago |
| system | controller                   | node1 | Running | 0        | 12
hours ago |
| system | flannel                      | node1 | Running | 0        | 12
hours ago |
| system | ingress-proxy                | node1 | Running | 0        | 12
hours ago |
| system | kafka                        | node1 | Running | 0        | 12
hours ago |
| system | loki                         | node1 | Running | 0        | 12
hours ago |
| system | metrics                      | node1 | Running | 0        | 12
hours ago |
| system | minio                        | node1 | Running | 0        | 12
hours ago |
| system | postgres                     | node1 | Running | 0        | 12
hours ago |
| system | promtail-cltmk               | node1 | Running | 0        | 12
hours ago |
| system | vip-add                      | node1 | Running | 0        | 12
hours ago |
└────────┴──────────────────────────────┴───────┴─────────┴──────────┴──────────────┘
```

> **Note**
>
> - The pod statuses appear in separate terminal sessions for each node.
> - The status of all services must be *Running*.

2. Check the current version.

```
sedo version
┌──────────────────────────────────────────────────────────────────────────────────────────────┐
│Installer: 24.3.2                │                                                            │
├──────────────┬──────────────────────────────────────────────┬──────────────┬─────────────────┤
│ NODE NAME    │ OS VERSION                                   │ KERNEL VERSION │               │
├──────────────┼──────────────────────────────────────────────┼──────────────┼─────────────────┤
│ node1-c1-sa1 │ NxFOS 3.2-555 (93358ad257a6cf1e3da439144e3d2e8343b53008) │
6.1.0-31-amd64 │
└──────────────┴──────────────────────────────────────────────┴──────────────┴─────────────────┘

┌──────────────────────────────────┬──────────────────────────────┬──────────┐
│ IMAGE NAME                       │ VERSION                      │ NODES    │
├──────────────────────────────────┼──────────────────────────────┼──────────┤
...
└──────────────────────────────────┴──────────────────────────────┴──────────┘
```

The service status and version information are confirmed.

Set Up Web UI Access to Cisco Optical Network Controller

Use this procedure to set Up Web UI Access to Cisco Optical Network Controller. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Before you begin

Ensure you have administrator access to the Cisco Optical Network Controller VM.

Follow these steps to set up web user interface access to Cisco Optical Network Controller.

Enable web user interface access to Cisco Optical Network Controller.

Procedure

1. Use the **sedo security user set admin --password** command to set the initial UI password for the admin user.

   Example

   ```
   sedo security user set admin --password
   ```

   The password must include at least

   - one uppercase letter,
   - one lowercase letter,

- one number,
- one special character,
- must have a minimum length of eight characters.

> **Note**
>
> The password policy for the system includes both configurable settings and nonconfigurable hard requirements to ensure security.

2. (Optional) Change the password policy settings using the **sedo security password-policy set** command.

```
sedo security password-policy set --expiration-days <number> --reuse-limit
<number> --min-complexity-score <number>
```

Parameter	Description
`expiration-days`	Default password expiration used when creating new users, in days. **Default value:** 180
`min-complexity-score`	The password strength forced for local users can be enabled or disabled and can be set in scores of one to five (weak to strong). The password is checked against several dictionaries and common passwords lists, to ensure its complexity according to the selected score. **Default value:** 3
`reuse-limit`	This specifies how many historical passwords are retained and blocked from reuse when you change your password. **Default value:** 12

3. Use the **sedo security user list** to check the default admin user ID.
4. Use the **sedo security user admin set --password** to change the default password.
5. Open this URL to access the Cisco Optical Network Controller Web UI.

```
https://<virtual IP>:8443/
```

> **Note**
>
> Access the web UI only after all the `onc` services are running. Use the **sedo system status** command to verify that all services are running.

6. Log in to Cisco Optical Network Controller with the *admin* user ID and the password set in step 1.

Perform a switchover in a geo-redundant Cisco Optical Network Controller deployment

Use this procedure to perform a switchover in a geo-redundant Cisco Optical Network Controller deployment. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Before you begin

- Verify that a geo-redundant Cisco Optical Network Controller deployment is configured.
- Verify that the **DB replication** is *streaming* and **DB Lag** is *0 bytes* using the **sedo supercluster status** command.

```
sedo supercluster status
┌──────────────────────────────────────────────────────────────────────────────────────────┐
│ Supercluster Status          │                                                           │
├──────────────────┬───────────────────────────────────────────────────────────────────────┤
│ Cluster ID       │ QgQV2uXgP1udqshlIssyTwf3LZzEyRh6I3z5MH8almA                            │
│                  │                                                                       │
│ Cluster Name     │ cluster1  │                                                           │
│ Cluster Role     │ worker    │                                                           │
│ Peers            │ cluster2 (worker,
jaWeN9BdXUUTxvofwt6Hukt6OQXIUaqo4NxN6zHYDc)     │
│                  │ cluster3 (arbitrator,
SUCrwqQjXToG5GKBwckcg_CtzgHstQigaEM1X0988E) │
│ Mode             │ Running   │                                                           │
│ Current Active   │ cluster1  │                                                           │
│ Previous Active  │           │                                                           │
│ Standby Clusters │ cluster2  │                                                           │
│ Last Switchover  │           │                                                           │
│ Last Failover    │           │                                                           │
│ Last Seen        │ controller-0.cluster2: 2025-03-19 11:16:57.051 +0000 UTC
                   │                                                                       │
│                  │ controller-0.cluster3: 2025-03-19 11:16:57.047 +0000 UTC
                   │                                                                       │
│                  │ controller-0.cluster1: 2025-03-19 11:16:57.051 +0000 UTC
                   │                                                                       │
│ Last Peer Error  │           │                                                           │
│ Server Error     │           │                                                           │
│ DB Replication   │ disconnected │                                                        │
│ DB Lag           │ 0 bytes   │                                                           │
└──────────────────┴───────────────────────────────────────────────────────────────────────┘
```

Follow these steps to perform a switchover in a geo-redundant deployment.

Switch active and standby roles in a geo-redundant deployment.

Use this procedure when you need to move the active role to another cluster.

Procedure

1. Run the **sedo supercluster switchover** *<target-active-cluster-name>* command and confirm when prompted.

> **Note**
>
> When you perform a dynamic switchover of the active cluster using the **sedo supercluster switchover** command, the Cisco Optical Network Controller UI may display an *HTTP 500 Internal Server Error* for up to four minutes while the system stabilizes again.

Example

```
nxf@node:~$ sudo sedo supercluster switchover cluster2
Are you sure you want to initiate supercluster switchover to cluster
"cluster2"? [y/n]y
```

The switchover takes place and WebUI displays a message that says *Switchover happened. Please refresh the page.*, and the WebUI update takes about 20 seconds.

2. Use the **sedo supercluster status** command to SSH in to the new active node and view the supercluster status.

```
sedo supercluster status
┌──────────────────────────────────────────────────────────────────────────────┐
│ Supercluster Status          │                                               │
├──────────────────────────────┼───────────────────────────────────────────────┤
│ Cluster ID                   │ jaWeN9BdXUUTxvofwt6Hukt6OQXIUaqo4NxN6zHYDc    │
│ Cluster Name                 │ cluster2                                      │
│ Cluster Role                 │ worker                                        │
│ Peers                        │ cluster1 (worker,
QgQV2uXgP1udqshlIssyTwf3LZzEyRh6I3z5MH8almA)                                    │
│                              │ cluster3 (arbitrator,
SUCrwqQjXToG5GKBwckcg_CtzgHstQigaEM1X0988E)                                      │
│ Mode                         │ Running                                       │
│ Current Active               │ cluster2                                      │
│ Previous Active              │ cluster1                                      │
│ Standby Clusters             │ cluster1                                      │
│ Last Switchover              │ 2025-03-19 11:20:49.705 +0000 UTC             │
│ Last Failover                │                                               │
│ Last Seen                    │ controller-0.cluster1: 2025-03-19 11:24:07.056 +0000 UTC
│                              │ controller-0.cluster2: 2025-03-19 11:24:07.058 +0000 UTC
│                              │ controller-0.cluster3: 2025-03-19 11:24:07.058 +0000 UTC
│ Last Peer Error              │                                               │
│ Server Error                 │                                               │
│ DB Replication               │ streaming                                     │
│ DB Lag                       │ 0 bytes                                       │
└──────────────────────────────┴───────────────────────────────────────────────┘
```

The DB replication status changes from `disconnected` to `streaming` as the switchover process progresses. Database replication is complete when the **DB Replication** status is streaming and **DB Lag** is 0 bytes.

Note

A switchover alarm is raised by Cisco Optical Network Controller during the switchover process. The alarm is cleared after the switchover. You can see the alarm details under Alarm History in the alarms app.

3. Optional: Use the raft API to get the supercluster status.

Example

```
nxf@node:~$ kubectl exec -it onc-devicemanager-service-0 -- curl -X GET
http://controller.nxf-system.svc.cluster.local/api/v1/raft/status
```

The API response gives you the information from the **sedo supercluster status** command.

Restriction

- Do not perform a switchover until the **DB replication** status is Streaming and **DB Lag** is 0 bytes after the previous switchover. This typically takes five minutes. Performing a switchover before replication is fully synchronized can result in data loss or data corruption.
- If you perform a switchover while a delete operation was in progress, you must repeat the deleted operation on the new active after the switchover. This restriction applies to node and circuit delete operations.
- If the active cluster goes down for some reason, a failover takes place. During a failover, the web UI becomes unavailable for up to a minute, and the system raises the switchover alarm.

The active role switches to the target cluster and status reflects the change.

Upgrade a standalone or high-availability Cisco Optical Network Controller deployment to a geo-redundant deployment

Use this procedure to upgrade a standalone or high-availability deployment to geo-redundant mode, including eastbound and northbound network setup and worker and arbitrator node bringup.

Before you begin

- **Backup Creation:** Verify that a full system backup is created. For details about creating a backup, see Backup and Restore Database or use the `sedo backup create full` command and export the backup for recovery if needed. Use this backup to revert to the older version if your upgrade fails.

  Example:

```
root@conc-1:~# sedo backup  create full
Creating backup, this may take a while...
Done creating backup
root@conc-1:~# sedo backup  list
┌──────────────────────────────────────┬──────────────┬──────────────────────────┐
│ NAME                                 │ TIME         │ SIZE
```

```
│ TYPE │ HOSTNAME    │ POSTGRES VERSION │
├──────────────────────────────────────────────────────────┤
│ base_0000000E000000010000009E │ 2025-03-11 04:11:47.733980894 +0000 UTC │
87 MB (838 MB Uncompressed) │ full │ postgres-0 │ 150008          │
└──────────────────────────────────────────────────────────┘

root@conc-1:~# cd /data
root@conc-1:/data# sedo backup download base_0000000E000000010000009E
Downloading Backup      ...  [.....<#>...............] [63.03MB in 9.200973s]
Finished downloading backup to "/data/nxf-backup-3.2-1741666307.tar.gz"

root@conc-1:/data# scp /data/nxf-backup-3.0-1736872559.tar.gz <remote location>
```

- Use `/data` for all file operations such as collecting, staging, copying, extracting, uploading, or downloading logs, backups, system-pack files, service-pack files, and software images. Do not use `/home/nxf` as a landing or staging directory. The `/home/nxf` directory is on the root file system and has limited capacity. Adding data to this directory can cause disk pressure and upgrade failure.
- Before starting the upgrade, verify that `/home/nxf` does not contain user-generated logs, backups, image files, tar files, or extracted bundles. Move required files to `/data` or external storage and remove only unneeded user-generated files from `/home/nxf`.
- **Network Configuration:** Before installing Cisco Optical Network Controller, create the required networks.
  - **Control Plane network:** The control plane network helps in the internal communication between the deployed VMs within a cluster.
  - **VM network or Northbound network:** The VM network is used for communication between the user and the cluster. It handles all the traffic to and from the VMs running on your ESXi hosts.

    This network is your public network through which the UI is hosted. Cisco Optical Network Controller uses this network to connect to Cisco Optical Site Manager devices using Netconf/gRPC.
  - **Eastbound network:** The Eastbound network helps in the internal communication between the deployed VMs within a supercluster. The active and standby nodes use this network to synchronize their databases. The Postgres database is replicated across both active and standby nodes. MinIO is also replicated on the arbitrator.

    > **Note**
    >
    > **Bandwidth requirement:** The Eastbound network should provide 1 Gbps (1,000 Mbps) bandwidth and maintain latency below 100 ms (milliseconds).
    >
    > You can configure the Eastbound network to be a flat Layer 2 network or an L2VPN, where the Eastbound IP addresses of all nodes are in the same subnet. If your Eastbound IPs are in different subnets, you must configure static routing between your nodes for the eastbound network.

- **BGP Router Configuration:** Obtain the BGP router IP, Router autonomous system number, and BGP password from network administrators for configuration.
- **VMware Setup:** Ensure that the vCenter has the required networks configured and attached correctly. Verify that physical adapters are correctly mapped for Northbound and Eastbound networks.
- **Access and Permissions:** Ensure you have the necessary permissions to execute commands and modify network settings on the nodes.

- Verify a system pack image package before use. Each package contains all required files for verification. For detailed steps, see Verify a signed qcow2 or system pack image on page 108.

For more details about creating networks, see Installation Requirements.

Follow these steps to upload the system pack for a standalone or high-availability deployment.

Upgrade a standalone deployment or a high-availability deployment to a geo-redundant deployment. Cisco Optical Network Controller supports upgrades to a new release from previous releases. The required upgrade path depends on your current version.

For more details about the upgrade path, see Supported Upgrade Paths.

These instructions explain how to upgrade a standalone deployment from an older release as well as configuring necessary networks for geo-redundant supercluster communication.

Restriction

- Cisco Optical Network Controller does not support direct downgrades to older releases.
- To revert to a previous version, you must first create a database backup using the SWIMU application before upgrading. Then, install the desired older version using its OVA file, and finally, restore the database.
- Refer to the Backup and Restore Database documentation for detailed instructions.

Procedure

1. Perform any of these tasks based on the standalone or Geo HA setup.
   - For standalone deployment, log in to the standalone node using the private key.
   - For Geo-redundant deployment, log in to the active node using the private key.

   Example

   ```
   ssh -i <private-key_file> nxf@<node_ip>
   ```

2. Download or copy the system pack `system-pack-file.tar.gz` to the NxF SA system running 25.1.1 and place it in the `/tmp` directory using `curl` or `scp`.

   Example

   ```
   scp user@remote_server:/path/to/system-pack-file.tar.gz /tmp/
   ```

   ```
   curl -o /tmp/system-pack-file.tar.gz
   http://example.com/path/to/system-pack-file.tar.gz
   ```

3. Check the system pack status using the **sedo system upgrade list** command.

   Example

   ```
   sedo system upgrade list
   ```

4. Follow these steps to upload system pack for standalone or HA deployment.

For standalone	For HA

Upload the system pack using **the sedo system upgrade upload** command.
`sedo system upgrade upload /tmp/system-pack-file.tar.gz`
	a. Upload the system pack using the **sedo system upgrade upload** command.
	`sedo system upgrade upload /tmp/system-pack-file.tar.gz`
	b. Check the system pack status using the **sedo system upgrade pull** command.
	`sedo system upgrade pull /tmp/system-pack-file.tar.gz`

5. Apply the system pack using the **sedo system upgrade apply** command.

   Example

   ```
   sedo system upgrade apply /tmp/system-pack-file.tar.gz
   ```

   The upgrade process takes approximately 30 minutes to complete.
6. Reboot the system using the **reboot** command.

   Example

   ```
   reboot
   ```

7. After the system reboots, verify the NxF version and system status by using the `sedo version` and `sedo system status` commands.

   Example

```
sedo version
┌──────────────────────────────────────────────────────────────────────────────────────┐
│ Installer: 24.3.2                │                                                   │
├──────────┬──────────────────────────────────────────────────────────┬────────────────┤
│ NODE NAME    │ OS VERSION                        │ KERNEL VERSION │
├──────────┼──────────────────────────────────────────────────────────┼────────────────┤
│ node1-c1-sc2 │ NxFOS 3.2-555 (93358ad257a6cf1e3da439144e3d2e8343b53008) │
6.1.0-31-amd64 │
└──────────┴──────────────────────────────────────────────────────────┴────────────────┘

┌──────────────────────────────────────────┬──────────────────┬──────────────┐
│ IMAGE NAME          │ VERSION                │ NODES        │
├──────────────────────────────────────────┼──────────────────┼──────────────┤
│ docker.io/rancher/local-path-provisioner       │ v0.0.30               │
node1-c1-sc2 │
│ dockerhub.cisco.com/cisco-onc-docker/dev/monitoring
 │ dev_latest             │ node1-c1-sc2 │
│ quay.io/coreos/etcd│ v3.5.15               │ node1-c1-sc2 │
│ registry.nxf-system.svc:8443/cisco-onc-docker/dev/alarmservice
 │ 24.3.2-5               │ node1-c1-sc2 │
│ registry.nxf-system.svc:8443/cisco-onc-docker/dev/circuit-service
 │ 24.3.2-5               │ node1-c1-sc2 │
│ registry.nxf-system.svc:8443/cisco-onc-docker/dev/collector-service
 │ 24.3.2-5               │ node1-c1-sc2 │
```

```
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/config-service
 | 24.3.2-5              | node1-c1-sc2 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/devicemanager-service
 | 24.3.2-5              | node1-c1-sc2 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/inventory-service
 | 24.3.2-5              | node1-c1-sc2 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/monitoring
 | 24.3.2-5              | node1-c1-sc2 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/nbi-service
 | 24.3.2-5              | node1-c1-sc2 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/netconfcollector-service
 | 24.3.2-5              | node1-c1-sc2 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/onc-apps-ui-service
 | 24.3.2-5              | node1-c1-sc2 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/onc-kafkarecap-service
 | 0.1.PR93-26c53efb0cf6ebc1f0c4a2aa226a0ab3751b9101 | node1-c1-sc2 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/osapi-gw-service
 | 24.3.2-5              | node1-c1-sc2 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/pce_service
 | 24.3.2-5              | node1-c1-sc2 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/pm-service
 | 24.3.2-5              | node1-c1-sc2 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/pmcollector-service
 | 24.3.2-5              | node1-c1-sc2 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/topology-service
 | 24.3.2-5              | node1-c1-sc2 |
| registry.nxf-system.svc:8443/cisco-onc-docker/dev/torch
 | 24.3.2-5              | node1-c1-sc2 |
| registry.sedona.ciscolabs.com/nxf/authenticator| 3.2-508              |
node1-c1-sc2 |
| registry.sedona.ciscolabs.com/nxf/bgp          | 3.2-505              |
node1-c1-sc2 |
| registry.sedona.ciscolabs.com/nxf/controller   | 3.2-533              |
node1-c1-sc2 |
| registry.sedona.ciscolabs.com/nxf/firewalld    | 3.2-505              |
node1-c1-sc2 |
| registry.sedona.ciscolabs.com/nxf/flannel      | 3.2-505              |
node1-c1-sc2 |
| registry.sedona.ciscolabs.com/nxf/ingress-proxy| 3.2-508              |
node1-c1-sc2 |
| registry.sedona.ciscolabs.com/nxf/kafka        | 3.2-505              |
node1-c1-sc2 |
| registry.sedona.ciscolabs.com/nxf/kubernetes   | 3.2-505              |
node1-c1-sc2 |
| registry.sedona.ciscolabs.com/nxf/loki         | 3.2-505              |
node1-c1-sc2 |
| registry.sedona.ciscolabs.com/nxf/metrics-exporter
 | 3.2-505               | node1-c1-sc2 |
| registry.sedona.ciscolabs.com/nxf/minio        | 3.2-505              |
node1-c1-sc2 |
| registry.sedona.ciscolabs.com/nxf/service-proxy| 3.2-508              |
node1-c1-sc2 |
| registry.sedona.ciscolabs.com/nxf/timescale    | 3.2-515              |
node1-c1-sc2 |
| registry.sedona.ciscolabs.com/nxf/timescale    | 3.2-514              |
node1-c1-sc2 |
└──────────────────────────────────────────┴──────────────────────────┴──────────┘
```

```
sedo system status
┌──────────────────────────────────────────────────────────────────────────────┐
| System Status (Fri, 20 Sep 2024 08:21:27 UTC)         |
```

```
┌─────────┬──────────────────────────────────┬─────────┬─────────┬──────────┬──────────────┐
│ OWNER   │ NAME                             │ NODE    │ STATUS  │ RESTARTS │ STARTED      │
├─────────┼──────────────────────────────────┼─────────┼─────────┼──────────┼──────────────┤
│ onc     │ monitoring                       │ node1   │ Running │ 0        │ 3 hours ago  │
│ onc     │ onc-alarm-service                │ node1   │ Running │ 0        │ 3 hours ago  │
│ onc     │ onc-apps-ui-service              │ node1   │ Running │ 0        │ 3 hours ago  │
│ onc     │ onc-circuit-service              │ node1   │ Running │ 0        │ 3 hours ago  │
│ onc     │ onc-collector-service            │ node1   │ Running │ 0        │ 3 hours ago  │
│ onc     │ onc-config-service               │ node1   │ Running │ 0        │ 3 hours ago  │
│ onc     │ onc-devicemanager-service        │ node1   │ Running │ 0        │ 3 hours ago  │
│ onc     │ onc-inventory-service            │ node1   │ Running │ 0        │ 3 hours ago  │
│ onc     │ onc-nbi-service                  │ node1   │ Running │ 0        │ 3 hours ago  │
│ onc     │ onc-netconfcollector-service     │ node1   │ Running │ 0        │ 3 hours ago  │
│ onc     │ onc-osapi-gw-service             │ node1   │ Running │ 0        │ 3 hours ago  │
│ onc     │ onc-pce-service                  │ node1   │ Running │ 0        │ 3 hours ago  │
│ onc     │ onc-pm-service                   │ node1   │ Running │ 0        │ 3 hours ago  │
│ onc     │ onc-pmcollector-service          │ node1   │ Running │ 0        │ 3 hours ago  │
│ onc     │ onc-topology-service             │ node1   │ Running │ 0        │ 3 hours ago  │
│ onc     │ onc-torch-service                │ node1   │ Running │ 0        │ 3 hours ago  │
│ system  │ authenticator                    │ node1   │ Running │ 0        │ 12 hours ago │
│ system  │ controller                       │ node1   │ Running │ 0        │ 12 hours ago │
│ system  │ flannel                          │ node1   │ Running │ 0        │ 12 hours ago │
│ system  │ ingress-proxy                    │ node1   │ Running │ 0        │ 12 hours ago │
│ system  │ kafka                            │ node1   │ Running │ 0        │ 12 hours ago │
│ system  │ loki                             │ node1   │ Running │ 0        │ 12 hours ago │
│ system  │ metrics                          │ node1   │ Running │ 0        │ 12 hours ago │
│ system  │ minio                            │ node1   │ Running │ 0        │ 12 hours ago │
│ system  │ postgres                         │ node1   │ Running │ 0        │ 12 hours ago │
│ system  │ promtail-cltmk                   │ node1   │ Running │ 0        │ 12 hours ago │
│ system  │ vip-add                          │ node1   │ Running │ 0        │ 12 hours ago │
└─────────┴──────────────────────────────────┴─────────┴─────────┴──────────┴──────────────┘
```

8. Verify onboarded sites and services by accessing the Cisco Optical Network Controller UI.

Example

Use a web browser to open *https://<virtual ip>:8443/* and access the Cisco Optical Network Controller Web UI.

What to do next

Configure eastbound and northbound networks on page 84

Configure eastbound and northbound networks

Use this procedure to configure eastbound and northbound networks. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Update the interface configuration files. Restart the network services. Verify the network settings in vCenter. Set the eastbound interface to ensure correct connectivity.

Prepare the standalone node so it can communicate over the designated eastbound and northbound interfaces.

Before you begin

Ensure you know the required IP addresses and DNS values for eastbound and northbound interfaces.

Follow these steps to set up eastbound and northbound networks.

Procedure

---

1. Verify the Eastbound (ens256) and Northbound (ens224) interfaces using the `ip address` command.

Example

```
ip address

3: ens224: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
 default qlen 1000
    link/ether 00:50:56:9c:16:fb brd ff:ff:ff:ff:ff:ff
    altname enp19s0
    inet 192.168.10.11/24 brd 192.168.10.255 scope global ens224
       valid_lft forever preferred_lft forever
    inet 10.64.103.73/32 scope global ens224
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe9c:16fb/64 scope link
       valid_lft forever preferred_lft forever
4: ens256: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
 default qlen 1000
    link/ether 00:50:56:9c:e1:fc brd ff:ff:ff:ff:ff:ff
    altname enp27s0
    inet 172.10.10.11/24 brd 172.10.10.255 scope global ens256
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe9c:e1fc/64 scope link
       valid_lft forever preferred_lft forever
```

> **Note**
>
> This sample output shows only the relevant part of the command output.

2. Update the IP address for the northbound interface (ens224) by modifying the configuration file located at `/etc/systemd/network/10-cloud-init-ens224.network`.

   Example

```
[Address]
Address=<northbound-node1-ip-address>/<subnet>

[Match]
Name=ens224

[Network]
DHCP=no
DNS=<northbound-node1-dns>

[Route]
Destination=0.0.0.0/0
Gateway=<northbound-node1-gateway>
```

3. Update the IP address of the Eastbound interface (ens256) by editing the interface file located at `/etc/systemd/network/10-cloud-init-ens256.network`.

   Example

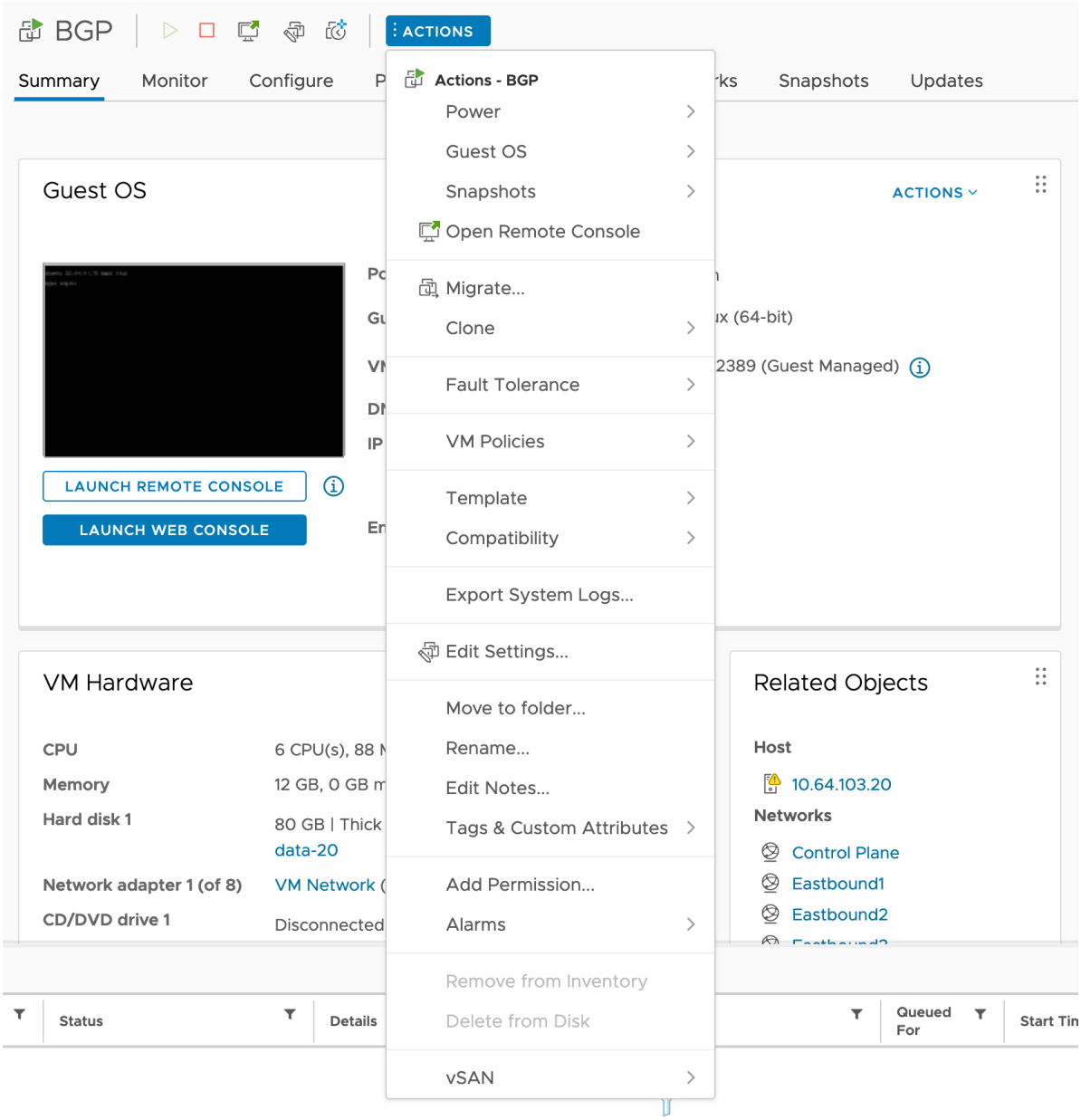
```
[Address]
Address=<eastbound-node1-ip-address>/<subnet>

[Match]
Name=ens256

[Network]
DHCP=no
DNS=<eastbound-node1-dns>

# Optional - when static route is needed for eastbound network
[Route]
Destination=<network address need to be routed>/<subnet>
Gateway=<eastbound network gateway>
```

4. Restart the network service to apply the changes.

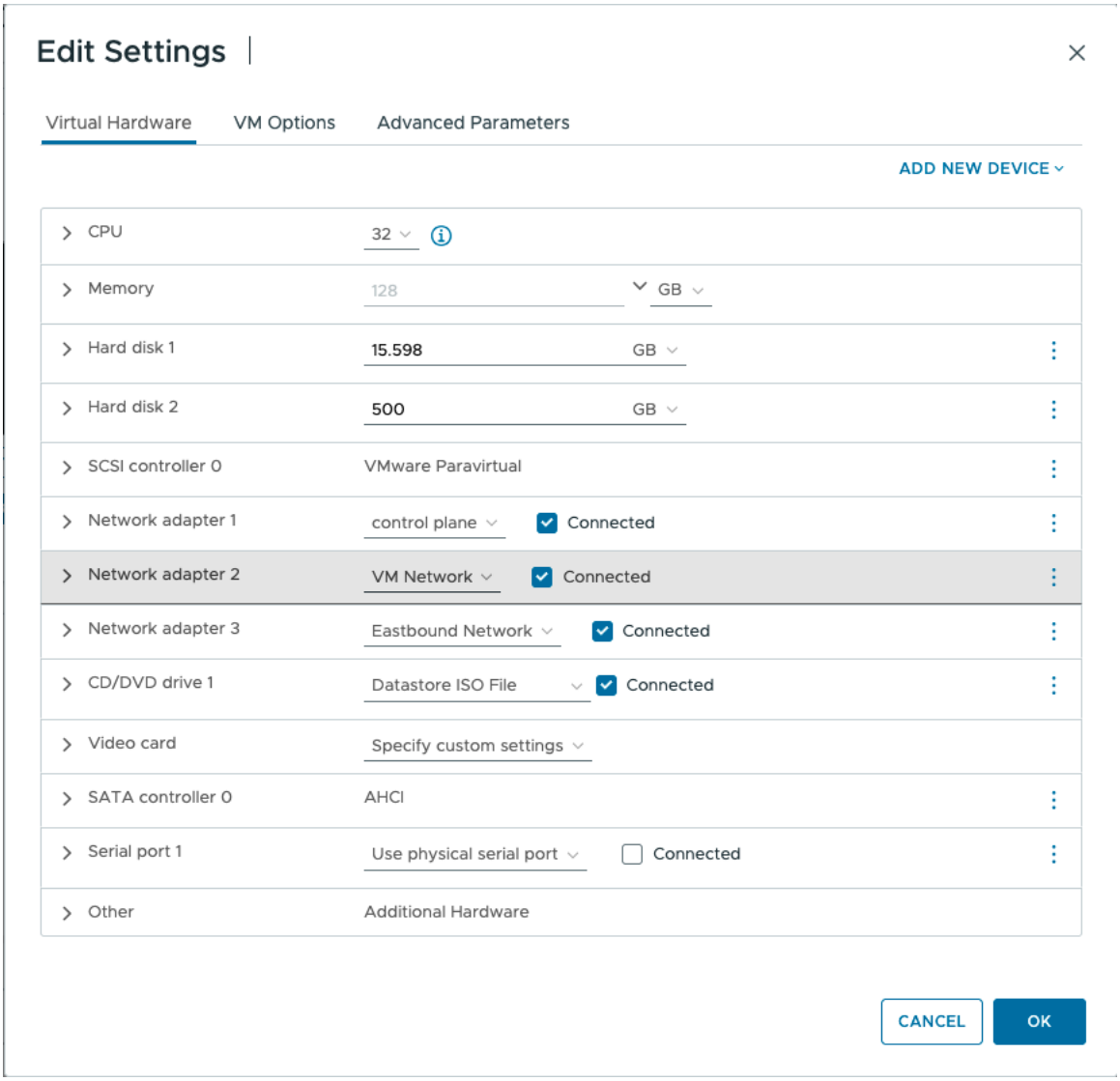
   Example

```
sudo systemctl restart systemd-networkd
```

5. Use **vCenter** to verify and update the northbound and eastbound network settings for the node.
   a) In vCenter, click **ACTIONS** in the node screen.



b) Click **Edit Settings** in the drop-down list.
c) Update the northbound and eastbound networks that you created for the supercluster.



6. Use SSH to access the upgraded node with the new northbound IP address. Set the eastbound interface.

   sedo system set-eastbound *eastbound-interface*

   Example

   ```
   sedo system set-eastbound ens256
   ```

You have configured the eastbound and northbound networks for the standalone node.

What to do next

Bring up a Worker Node and an Arbitrator Node.

Create worker and arbitrator nodes

Use this procedure to create worker and arbitrator nodes. It helps you complete the related installation workflow and confirm the deployment state before continuing.

This task is necessary when setting up georedundancy. It also prepares the nodes for supercluster configuration.

Before you begin

Configure eastbound and northbound networks on page 84

Follow these steps to bring up a worker node and an arbitrator node. Create two more Cisco Optical Network Controller nodes for georedundancy.

Use this task to create worker and arbitrator nodes as part of a georedundant Cisco Optical Network Controller deployment.

Procedure

1. Create two more Cisco Optical Network Controller nodes for Georedundancy.

   For more details about the instructions, see Install and Deploy Geo Redundant Cisco Optical Network Controller.

2. Optional: Edit the interface file located */etc/systemd/network/10-cloud-init-ens256.network* to create static routes between the nodes for the eastbound network if the eastbound interfaces for the nodes are in different subnets.

   Example

   ```
   # Optional - when static route is needed for eastbound network
   [Route]
   Destination=<network address need to be routed>/<subnet>
   Gateway=<eastbound network gateway>
   ```

   Add the configuration section with the required IP addresses to set up static routes.

3. Optional: Restart the network service to apply the changes using:

   Example

   ```
   sudo systemctl restart systemd-networkd
   ```

The worker and arbitrator nodes are created and prepared for supercluster setup.

What to do next

Set up the supercluster on page 65

Update time zone configuration in a geo-redundant deployment

Use this procedure to update time zone configuration in a geo-redundant deployment. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Before you begin

- Verify status of every pod is `Running` by using the kubectl get pods -A | grep onc command. This example shows a sample output where all pods are running.

  ```
  root@vm1-cluster1-node1:~# kubectl get pods -A | grep onc
  onc                  monitoring-0        2/2     Running   0             21m
  ```

```
onc                  onc-alarm-service-0 2/2     Running   3 (51m ago)
3h6m
onc                  onc-apps-ui-service-6f95dfbc7c-60w87ne          2/2
 Running   3 (51m ago)    3h6m
onc                  onc-circuit-service-0                           2/2
 Running   3 (51m ago)    3h6m
onc                  onc-collector-service-0                         2/2
 Running   3 (51m ago)    3h6m
onc                  onc-config-service-02/2     Running   3 (51m ago)
3h6m
onc                  onc-devicemanager-service-0                     2/2
 Running   3 (51m ago)    3h6m
onc                  onc-inventory-service-0                         2/2
 Running   3 (51m ago)    3h6m
onc                  onc-nbi-service-0   2/2     Running   3 (51m ago)
3h6m
onc                  onc-netconfcollector-service-85bd7c89bf-qc8pf   2/2
 Running   0             21m
```

- Verify that any previous switchover or failover has finished. Confirm that data replication across the active and standby nodes is complete. Use the **sedo supercluster status** to see the supercluster status and confirm that the **DB replication** status is *streaming* and **DB Lag** is *0*.

```
sedo supercluster status
┌──────────────────────────────────────────────────────────────────────────────┐
│ Supercluster Status          │
├──────────────┬───────────────────────────────────────────────────────────────┤
│ Cluster ID       │ QCTdDdt_rlRd9lgzRM15vSeb0r1tkLMkfCK4DoAy1aw
          │
│ Cluster Name     │ cluster1   │
│ Cluster Role     │ worker     │
│ Peers            │ cluster2 (worker,
rabSbdhIWtq1qzhW1lZTm0Hu5_tIxOFZgDyWr5pac90)     │
│                  │ cluster3 (arbitrator,
XxHjr5wMmDyiYW6jbvaCcGZW8VIasb4sBv8x0B15DYk) │
│ Mode             │ Running    │
│ Current Active   │ cluster1   │
│ Previous Active  │ cluster2   │
│ Standby Clusters │ cluster2   │
│ Last Switchover  │ 2025-06-09 00:34:46.826 -0500 CDT      │
│ Last Failover    │            │
│ Last Seen        │ controller-0.cluster3: 2025-06-09 00:58:23.636 -0500 CDT
          │
│                  │ controller-0.cluster2: 2025-06-09 00:58:23.641 -0500 CDT
          │
│                  │ controller-0.cluster1: 2025-06-09 00:58:23.641 -0500 CDT
          │
│ Last Peer Error  │            │
│ Server Error     │            │
│ DB Replication   │ streaming  │
│ DB Lag           │ 0 bytes    │
└──────────────────┴───────────────────────────────────────────────────────────┘
```

Follow these steps to configure time zone configuration in a geo-redundant deployment.

From R25.1.2, you can update the timezone configuration. Previously, only the UTC time zone was supported. You can now configure Cisco Optical Network Controller in your preferred time zone.

For geo-redundant deployments, use the CLI command to update the timezone on each VM. Restart each VM using the steps described in this procedure to ensure a seamless change to the new timezone configuration. If the time zone configuration differs between VMs, a discrepancy can occur during failover or switchover.

> **Note**
>
> Change the timezone only when necessary. Each timezone change requires a reboot of the VMs and services and can cause inconsistencies.

Apply the same time zone on all three VMs. Be sure to review these limitations before making changes.

Procedure

1. Use SSH to access the three VMs and run this command.

   sudo timedatectl set-timezone *timezone-name*

   Example

   In this example, the time zone is set to JST.

```
root@vm1-cluster1-node1:~# sudo timedatectl set-timezone Asia/Tokyo
root@vm1-cluster1-node1:~# timedatectl
               Local time: Mon 2025-06-09 15:01:26 JST
           Universal time: Mon 2025-06-09 06:01:26 UTC
                 RTC time: Mon 2025-06-09 06:01:26
                Time zone: Japan (JST, +0900)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
```

   A few valid time zones are:

```
Asia/Kolkata
Asia/Dubai
Europe/Amsterdam
Africa/Bujumbura
```

2. Reboot the standby cluster using the **sudo reboot** command.
3. Verify the standby is up and running using these commands.
   - **kubectl get pods -A | grep onc**
   - **sedo supercluster status**

Verify the time zone in one of the pods using these commands. See the offset after the time.

```
root@vm1-cluster1-node1:~# kubectl exec -ti onc-torch-service-0 -n onc --
bash

onc-torch-service-0:/$ date -R

Mon, 09 Jun 2025 15:22:42 +0900
```

4. Perform a manual switchover using the **sedo supercluster switchover** *cluster* command. Wait for the switchover and data replication to complete.

> **Note**
>
> When you perform a dynamic switchover of the active cluster using the **sedo supercluster switchover** command, the Cisco Optical Network Controller UI may display an *HTTP 500 Internal Server Error* for up to four minutes while the system stabilizes again.

```
root@vm1-cluster1-node1:~# sedo supercluster switchover cluster2

Are you sure you want to initiate supercluster switchover to cluster
"cluster2"? [y/n] y
```

Make sure DB replication status is streaming and DB Lag is 0.

```
root@vm1-cluster1-node1:~# sedo supercluster status

┌──────────────────────────────────────────────────────────────────────────────┐
│ Supercluster Status                                                          │
├──────────────────┬───────────────────────────────────────────────────────────┤
│ Cluster ID       │ QCTdDdt_rlRd9lgzRM15vSeb0r1tkLMkfCK4DoAy1aw               │
│ Cluster Name     │ cluster1                                                  │
│ Cluster Role     │ worker                                                    │
│ Peers            │ cluster2 (worker,                                         │
rabSbdhIWtq1qzhW1lZTm0Hu5_tIxOFZgDyWr5pac90)     │
│                  │ cluster3 (arbitrator,                                     │
XxHjr5wMmDyiYW6jbvaCcGZW8VIasb4sBv8x0B15DYk) │
│ Mode             │ Running                                                   │
│ Current Active   │ cluster2                                                  │
│ Previous Active  │ cluster1                                                  │
│ Standby Clusters │ cluster1                                                  │
│ Last Switchover  │ 2025-06-09 15:23:29.686 +0900 JST                         │
│ Last Failover    │                                                           │
│ Last Seen        │ controller-0.cluster3: 2025-06-09 15:23:34.277 +0900 JST
│                  │ controller-0.cluster2: 2025-06-09 15:23:34.418 +0900 JST
│                  │ controller-0.cluster1: 2025-06-09 15:23:34.418 +0900 JST
│ Last Peer Error  │                                                           │
│ Server Error     │                                                           │
│ DB Replication   │ streaming                                                 │
│ DB Lag           │ 0 bytes                                                   │
```

```
└──────────────────┴──────────────────────────────────────────────────────────────┘
```

```
root@vm109-cluster2-node1:~# kubectl get pods -A | grep onc

onc                  monitoring-0        2/2     Running   0              50m

onc                   onc-alarm-service-0 2/2     Running   16 (65m ago)
4h23m
onc                   onc-apps-ui-service-6c474df87d-6aq3bqd          2/2
 Running   15 (65m ago)   4h23m
onc                   onc-circuit-service-0                          2/2
 Running   15 (65m ago)   4h23m
onc                   onc-collector-service-0                        2/2
 Running   15 (65m ago)   4h23m
onc                   onc-config-service-02/2     Running   15 (65m ago)
4h23m
onc                   onc-devicemanager-service-0                    2/2
 Running   17 (65m ago)   4h23m
onc                   onc-inventory-service-0                        2/2
 Running   15 (65m ago)   4h23m
onc                   onc-nbi-service-0   2/2     Running   15 (65m ago)
4h23m
onc                   onc-netconfcollector-service-59b855956b-hrbbb   2/2
 Running   0              3m18s
onc                   onc-osapi-gw-service-0                         2/2
 Running   15 (65m ago)   4h23m
onc                   onc-pce-service-0   2/2     Running   15 (65m ago)
4h23m
onc                   onc-pm-service-0    2/2     Running   13 (65m ago)
3h34m
onc                   onc-pmcollector-service-785669f8b7-7ndn4        2/2
 Running   0              50m
onc                   onc-topology-service-0                         2/2
 Running   15 (65m ago)   4h23m
onc                   onc-torch-service-0 2/2     Running   16 (65m ago)
4h23m
```

5. Repeat steps 2 and 3 for the new standby VM and the arbitrator VM.
6. If you want to make the original VM active, repeat Step 4.

Time zone configuration has been updated and Cisco Optical Network Controller web UI now displays time in the newly configured time zone.

This table shows screenshots highlighting the behavioral differences between Releases 25.1.1 and 25.1.2. In Release 25.1.2, the timestamp includes the time zone name and offset.

Release 25.1.2	Release 25.1.1
Figure 23: Alarms	**Figure 24: Alarms**

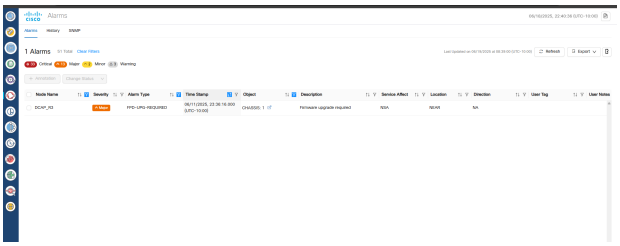
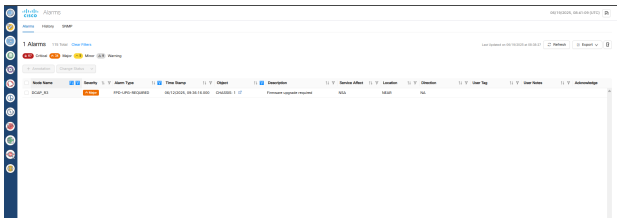

	

Figure 25: PM History	**Figure 26: PM History**

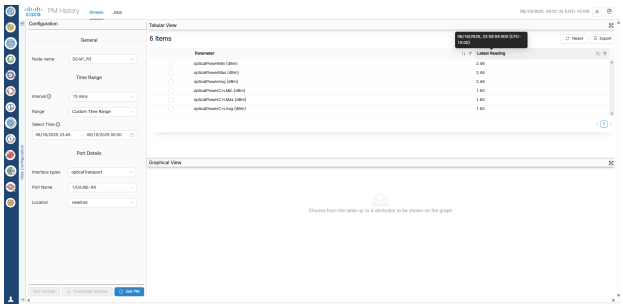
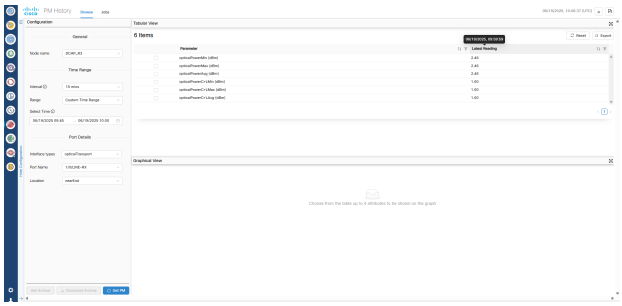
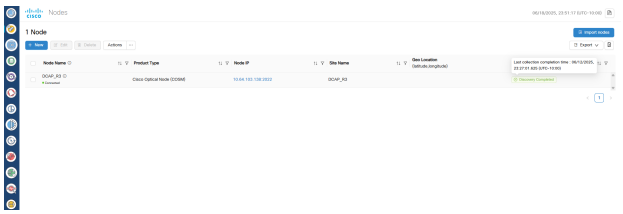
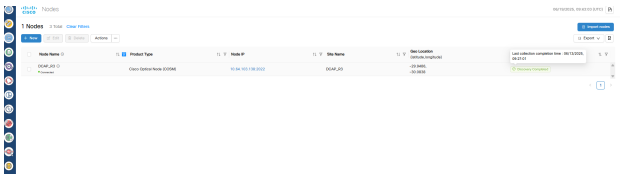

	

Figure 27: Nodes	**Figure 28: Nodes**
	

This table summarizes how different system components handle time zones and describes related limitations.

Component	Time Zone Behavior	Notes
Database (Alarms & Logs)	Stored in UTC	During time zone transitions (for example, switchover), the UI might temporarily show alarms with different time zone stamps until the system converges.
Cross-launch from Cisco Optical Network Controller	Offset preserved and IANA name may differ	Multiple IANA names can map to the same offset (for example, Asia/Colombo and Asia/Kolkata are both UTC +05:30).
TAPI Data and Notifications	UTC (+00:00)	Always uses UTC regardless of system time zone.
SNMP Traps	Epoch time	Time zone offset is not applied to epoch timestamps.
Developer Logs and Techdump	UTC	No time zone conversion applied.

Revert to a previous version of Cisco Optical Network Controller

Use this procedure to revert to a previous version of Cisco Optical Network Controller. It helps you complete the related installation workflow and confirm the deployment state before continuing.

This is a manual process. Automatic rollback is not supported. You cannot perform a revert from within Cisco Optical Network Controller.

Restriction

- Cisco Optical Network Controller does not support direct downgrades to older releases.
- To revert to a previous version, you must first create a database backup using the SWIMU application before upgrading. Then, install the desired older version using its OVA file, and finally, restore the database.
- Refer to the Backup and Restore Database documentation for detailed instructions.

Before you begin

Create a backup of the Cisco Optical Network Controller database. For details on creating a database backup, see Backup and Restore Database.

Follow these steps to revert to a previous version of Cisco Optical Network Controller.

You can revert Cisco Optical Network Controller to a previous version by reinstalling the software and restoring the database from a backup.

Procedure

1. For stand-alone deployments:
   a) Reinstall the previous version of Cisco Optical Network Controller, which is the version used for the backup. See Install Cisco Optical Network Controller Using VMware vSphere.
   b) Follow the procedure to perform database restore from a backup. See Backup and Restore Database.
2. For georedundant deployments:
   a) Reinstall the previous version of Cisco Optical Network Controller, which is the version used for the backup. See Install and Deploy Geo Redundant Cisco Optical Network Controller.
   b) Follow the procedure to perform database restore from a backup. See Backup and Restore Database.

KVM deployment for Cisco Optical Network Controller in Geo-HA mode

This section explains what a KVM-based Geo-HA installation includes, why the deployment uses three nodes, and what information you must prepare before installation.

A KVM-based Geo-HA installation deploys Cisco Optical Network Controller as three KVM virtual machines that operate as an active worker node, a standby worker node, and a witness node.

- The deployment uses a QCOW2 image for each VM and a cloud-init ISO that provides node-specific configuration.
- Each node uses control plane, northbound, and eastbound network interfaces.
- The worker nodes use BGP so that traffic can be routed to the northbound virtual IP from different locations.

In the Geo-HA 1+1+1 topology, the witness node participates in cluster coordination while the active and standby worker nodes provide service continuity across sites.

What to prepare before you start

Before starting the Geo-HA installation workflow, gather the values and files required to create the configuration for each node.

- You need the QCOW2 image, VM sizing values, deployment directory, and bridge names for the three KVM VMs.
- You should gather the host names, control-plane, northbound, and eastbound IP addresses, DNS values, and the shared northbound virtual IP.

Use the Geo-HA installation topics to create the cloud-init files, deploy the VMs, configure BGP, join the nodes, and verify the installation.

KVM deployment requirements

This section provides the software requirements, hardware requirements, and network information required to deploy Cisco Optical Network Controller on a KVM host.

This section lists the software requirements, hardware requirements, and network information for deploying Cisco Optical Network Controller on a KVM host.

Software requirements

This table lists the software and access requirements for the KVM deployment.

Table 13: KVM software requirements

Requirement	Description
`libvirt-client` and `qemu-kvm`	Install these packages on the KVM host.
`cloud-utils`	Use this package to create the cloud-init ISO.
`wget` or `curl`	Use one of these utilities to download the OS image.
SSH key pair	Use the SSH key pair to access the virtual machine.

Hardware requirements

This table lists the host infrastructure requirements for the KVM deployment.

Table 14: KVM hardware requirements

Requirement	Description
KVM host	Use RHEL 8.x or later with KVM and libvirt installed.
UEFI support	Install the OVMF firmware packages on the host.
Storage	Provide at least 100 GB of available disk space.
Memory	Provide at least 16 GB of available RAM.
CPU	Provide at least 8 vCPUs.

Configure networks for Geo-HA deployment

Learn how to configure the eastbound, control-plane, and northbound networks for the control plane, eastbound traffic, and northbound traffic on the KVM host before you deploy virtual machines.

Create the libvirt network definitions required Cisco Optical Network Controller deployment on a KVM host.

Before you begin

- Log in to the KVM host with privileges to create libvirt network definitions.
- Choose a working directory to store the network XML files.

In a standalone deployment, one eastbound network and one control-plane network are sufficient. In a Geo-HA deployment, use separate networks for the required UCS servers.

Procedure

1. Create the control-plane network definition in a file.

   The control plane network can be a private network.

   Example
   Example for *control.xml*

   ```
   <network>
   <name>control</name>
   <forward mode='none'/>
   <bridge name='virbr-control'/>
   <ip address='192.168.1.1' netmask='255.255.255.0'/>
   </network>
   ```

2. Create the eastbound network definitions in files named *eastbound1.xml*, *eastbound2.xml*, and *eastbound3.xml* on the respective UCS servers.

   The eastbound networks can be public networks.

   Example
   Example for *eastbound1.xml*.

   ```
   <network>
     <name>eastbound1</name>
     <forward mode='none'/>
     <bridge name='virbr-east1'/>
     <ip address='172.10.10.1' netmask='255.255.255.0'/>
   </network>
   ```

   Example for *eastbound2.xml*.

   ```
   <network>
     <name>eastbound1</name>
     <forward mode='none'/>
     <bridge name='virbr-east1'/>
     <ip address='172.10.10.2' netmask='255.255.255.0'/>
   </network>
   ```

Example for *eastbound3.xml*.

```
<network>
  <name>eastbound1</name>
  <forward mode='none'/>
  <bridge name='virbr-east1'/>
  <ip address='172.10.10.3' netmask='255.255.255.0'/>
</network>
```

3. Specify the default northbound interface using the <NORTHBOUND_BRIDGE> parameter for northbound connectivity when running the **virt-install** command.

   For example, use *bridge0* as the value for <NORTHBOUND_BRIDGE> to assign the northbound interface during deployment. The northbound network can be a public network.

4. Define, start, and enable each network at boot.

   Example

   ```
   virsh net-define <network-file.xml>
   virsh net-start <network-name>
   virsh net-autostart <network-name>
   ```

5. Verify that all required networks are available.

   Example

   ```
   virsh net-list --all
   ```

   The network list shows the defined networks and their current state.

Install Cisco Optical Network Controller using KVM in Geo-HA mode

Use this procedure to deploy Cisco Optical Network Controller on KVM in Geo-HA mode by preparing the node configuration, deploying the virtual machines, configuring BGP, joining the nodes, and verifying the installation.

Use this supertask to complete the full KVM-based Geo-HA installation workflow for Cisco Optical Network Controller.

Before you begin

Before you begin, Prepare KVM configuration files for Geo-HA.

Procedure

1. Create the KVM virtual machines.

   See Create KVM virtual machines for Geo-HA nodes.

2. Configure BGP on the worker nodes.

   See Configure BGP on the Geo-HA nodes.

3. Join the active worker, standby worker, and witness nodes.

   See Join the Geo-HA nodes.

4. Activate the deployment and verify access.

See Activate and verify Geo-HA.

5. Review the installed applications if required.

   See View installed applications.

6. Continue with network onboarding tasks.

   See Add network adapters and discover network devices.

Cisco Optical Network Controller is installed on KVM in Geo-HA mode.

Prepare KVM configuration files for Geo-HA deployment

Use this procedure to create the cloud-init files for each Geo-HA node and package them into an ISO for KVM deployment.

Prepare the cloud-init configuration used to deploy Cisco Optical Network Controller on KVM in a Geo-HA 1+1+1 topology.

Before you begin

- Perform this procedure for three VMs: the active worker node, the standby worker node, and the arbitrator node.
- Have the node names, IP addresses, virtual IP, join token, DNS values, and BGP settings available before you create the files.

Procedure

---

1. Create the SSH public-key file by following the steps at Create the meta-data file for a Geo-HA node.
2. Create the network configuration file by following the steps at Create the network-config file for a Geo-HA node.
3. Create the cluster and user settings file by following the steps at Create the user-data file for a Geo-HA node.

The *cidata.iso* file is created and is ready to be attached when you deploy the Cisco Optical Network Controller KVM virtual machines.

What to do next

Create KVM virtual machines for Geo-HA nodes.

Create the meta-data file for a Geo-HA node

Use this procedure to create the meta-data file for a Geo-HA node. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Create the *meta-data* cloud-init file for a Cisco Optical Network Controller Geo-HA node.

Procedure

---

1. Create a file named *meta-data*.
2. Add the instance ID and the SSH public key for the node.

   Meta-data file syntax:

```
instance-id: <instance_id>
local-hostname: <node_name>
public-keys:
  - <ssh_public_key>
```

Use the public key that matches the private key you plan to use for SSH access after the VM boots.

The *meta-data* file is ready to be included in the cloud-init ISO.

Meta-data file example

```
instance-id: iid-nxf-kvm-01
local-hostname: nxf-kvm-01
public-keys:
  - ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIKAn33NZjrMWyMJpf7QvrD4vCvEAg4PPdpb3UFXXXXXX
```

Create the network-config file for a Geo-HA node

Use this procedure to create the `network-config` file that defines the control-plane, northbound, and eastbound interfaces for a Geo-HA node.

Create the *network-config* file that defines the node IP addresses, default gateway, and DNS settings.

Procedure

---

1. Create a file named *network-config*.
2. Define the control-plane, northbound, and eastbound interfaces.

   network-config file syntax.

```
version: 2
ethernets:
  enp1s0:
    dhcp4: <false-or-true>
    addresses:
      - <control_plane_ip/subnet_mask>
  enp2s0:
    dhcp4: <false-or-true>
    addresses:
      - <northbound_ip/subnet_mask>
    gateway4: <gateway_ip>
    nameservers:
      addresses:
        - <dns_server_ip>
  enp3s0:
    dhcp4: <false-or-true>
    addresses:
      - <eastbound_ip/subnet_mask>
```

The *network-config* file is ready for use by the Geo-HA node.

***network-config* file syntax**

```
version: 2
ethernets:
  enp1s0:
    dhcp4: false
    dhcp6: false
    addresses:
      - 192.168.119.10/24
```

```
enp2s0:
  dhcp4: false
  dhcp6: false
  addresses:
    - 10.58.231.119/22
  gateway4: 10.58.228.1
  nameservers:
    addresses:
      - 144.254.71.184
enp3s0:
  dhcp4: false
  dhcp6: false
  addresses:
    - 172.10.10.10/24
```

Create the user-data file for a Geo-HA node

Use this procedure to create the `user-data` file that defines the node role, cluster settings, local users, storage mount, and optional NTP servers for a Geo-HA node.

Create the *user-data* cloud-init file for a Cisco Optical Network Controller Geo-HA node.

Procedure

1. Create a file named *user-data*.
2. Add the base cloud-init content for the data disk, the optional NTP servers and the Geo-HA cluster configuration for the node.

   user-data file syntax

```
fs_setup:
  - label: data
    device: /dev/vdb
    filesystem: ext4
mounts:
  - ["/dev/vdb", "/data"]
ntp:
  enabled: true
  ntp_client: chrony
  servers:
    - <ntp_server>
nxf:
  minControlPlaneCount: 1
  clusterIndex: <cluster_index>
  clusterName: <cluster_name>
  node:
    name: <node_name>
    initiator: <control_plane_ip>
    joinToken: <join_token>
    controlPlaneInterface: enp1s0
    eastboundInterface: enp3s0
    vip:
      northbound:
        interface: enp2s0
  initiator:
    vip:
      northbound:
        ip: <vip>
        type: L3
    kafka:
```

```
      enabled: true
    joinToken: <join_token>
    arbitratorNode: my-node3
    clusterRole: <worker-or-arbitrator>
    security:
      localUsers:
        - username: admin
          displayName: NxF Admin
          description: NextFusion Default Administrator
          locked: true
          mustChangePassword: false
          expiresInDays: 0
          access:
            - permission/admin
```

Set *clusterIndex* to `1` for the active worker node, `2` for the standby worker node, and `3` for the witness node. Set *clusterRole* to `worker` for the worker nodes and `arbitrator` for the witness node.

The *user-data* file is ready to be included in the cloud-init ISO for the node.

***user-data* file example**

```
#cloud-config
#NxF User Data
fs_setup:
- label: data
  device: /dev/vdb
  filesystem: ext4

mounts:
- ["/dev/vdb", "/data"]
ntp:
  enabled: true
  ntp_client: chrony
  servers:
  - ntp.esl.cisco.com,10.58.228.1

nxf:
  minControlPlaneCount: 1
  clusterIndex: 1
  clusterName: cluster1
  node:
    name: node1
    #Should be same as control plane
    initiator: 192.168.119.10
    joinToken: z9wfwl.ye6pmu6pr27aogjk
    controlPlaneInterface: enp1s0
    eastboundInterface: enp3s0
    vip:
      northbound:
        interface: enp2s0
  initiator:
    vip:
      northbound:
        ip: 10.58.231.119
        type: L3
    kafka:
      enabled: true
    joinToken: z9wfwl.ye6pmu6pr27aogjk
    arbitratorNode: node3
```

```
    #should be arbitrator for cluster3
    clusterRole: worker
    security:
      localUsers:
      - username: admin
        displayName: NxF Admin
        description: NextFusion Default Administrator
        locked: true
        mustChangePassword: false
        expiresInDays: 0
        access:
        - permission/admin
```

Create KVM virtual machines for Geo-HA nodes

Use this procedure to provision the active, standby, and witness virtual machines for a Cisco Optical Network Controller Geo-HA deployment on KVM.

Deploy the three KVM virtual machines that host the Geo-HA nodes for Cisco Optical Network Controller.

Run this task on the KVM host after you create the *cidata.iso* file for the node that you are deploying.

Before you begin

- Complete Prepare KVM configuration files for Geo-HA.
- Verify a qcow2 image package before use. Each package contains all required files for verification. For detailed steps, see Verify a signed qcow2 or system pack image on page 108.

Ensure that each VM uses a unique QCOW2 filename.

Procedure

1. Copy the three configuration files to a designated folder and generate the ISO.

   Use a deployment directory such as *<PATH_TO_DEPLOY_DIR>/vmConfig/cloud-config/*.

   Example

   ```
   cp network-config user-data meta-data
   <PATH_TO_DEPLOY_DIR>/vmConfig/cloud-config/
   mkisofs -o "<PATH_TO_DEPLOY_DIR>/vmConfig/cidata.iso" -r -J -V cidata
   "<PATH_TO_DEPLOY_DIR>/vmConfig/cloud-config/"
   ```

2. Login to each VM and deploy the nodes by running the **virt-install** command with the values for that VM.

   The control-plane network is mandatory even though it is not used for east-west Geo-HA traffic in the 1+1+1 deployment.

   This sample shows the syntax of the **virt-install** command.

   ```
   virt-install \
   --name <HOST_NAME> \
   --vcpus <CPUs> \
   --memory <MEMORY> \
   --disk path=<PATH_TO_DEPLOY_DIR>/<QCOW2_FILE_NAME>.qcow2,format=qcow2 \
   --disk
   path=<PATH_TO_DEPLOY_DIR>/<QCOW2_FILE_NAME>-data.qcow2,size=<SIZE>,device=disk,bus=virtio,format=qcow2
    \
   --disk path=<PATH_TO_DEPLOY_DIR>/vmConfig/cidata.iso,device=cdrom \
   ```

```
--osinfo debian12 \
--network bridge=<CONTROL_PLANE_BRIDGE>,model=virtio \
--network bridge=<NORTHBOUND_BRIDGE>,model=virtio \
--network bridge=<EASTBOUND_BRIDGE>,model=virtio \
--boot
loader=/usr/share/edk2/ovmf/OVMF_CODE.fd,loader.readonly=yes,loader_secure=no,nvram.template=/usr/share/edk2/ovmf/OVMF_VARS.fd,hd,uefi
 \
--serial pty \
--console pty,target_type=serial \
--noautoconsole
```

Wait for the VM to finish booting. SSH access is available after the VM is started.

The Geo-HA node VM is created with the QCOW2 system disk, the data disk, the cloud-init ISO, and the required control-plane, northbound, and eastbound interfaces.

virt-install command example

```
virt-install   \
  --name nxf-kvm-01   \
  --vcpus 32   \
  --memory 131072   \
  --disk
path=/var/lib/libvirt/images/VMs/nxf-kvm-01/CONC-26.1.1.qcow2,format=qcow2
\
  --disk
path=/var/lib/libvirt/images/VMs/nxf-kvm-01/CONC-26.1.1-data.qcow2,size=2500,device=disk,bus=virtio,format=qcow2
   \
  --disk path=/var/lib/libvirt/images/VMs/nxf-kvm-01/cidata.iso,device=cdrom
  \
  --osinfo debian12   \
  --network bridge=virbr-control,model=virtio   \
  --network bridge=bridge0,model=virtio   \
  --network bridge=virbr-east1,model=virtio   \
  --boot
loader=/usr/share/edk2/ovmf/OVMF_CODE.fd,loader.readonly=yes,loader_secure=no,nvram.template=/usr/share/edk2/ovmf/OVMF_VARS.fd,hd,uefi
   \
  --serial pty   \
  --console pty,target_type=serial   \
  --autostart \
  --noautoconsole
```

What to do next

Configure BGP on the Geo-HA nodes.

Configure BGP on the Geo-HA nodes

Use this procedure to connect to the worker nodes and configure BGP routing for the northbound virtual IP in a Geo-HA deployment.

Initialize and verify BGP on the Region 1 and Region 2 worker nodes so that traffic can be routed to the Cisco Optical Network Controller northbound virtual IP from the different sites in the Geo-HA topology.

Before you begin

- Deploy the VMs by following Create KVM virtual machines for Geo-HA nodes.
- Have the SSH private key that corresponds to the public key defined in the *meta-data* file.

Procedure

1. Verify SSH access to each worker node.

   Example

   ```
   ssh -i <private-key-file> nxf@<hco_management_ip>
   ```

   If you are prompted for a password, verify the key configuration. If the command times out, verify the IP configuration.

2. Initialize BGP and add the BGP router on the Region 1 worker node.

   Example

   ```
   sedo ha bgp init <region1_node_name> <region1_northbound_ip> <BGP_AS_1>
   sedo ha bgp router add <region1_node_name> <region1_northbound_gateway>
   <BGP_AS> <BGP_session_password>
   ```

3. Initialize BGP and add the BGP router on the Region 2 worker node, and then verify the router entries on both worker nodes.

   Example

   ```
   sedo ha bgp init <region2_node_name> <region2_northbound_ip> <BGP_AS_2>
   sedo ha bgp router add <region2_node_name> <region2_northbound_gateway>
   <BGP_AS> <BGP_session_password>
   sedo ha bgp router list <region1_node_name>
   sedo ha bgp router list <region2_node_name>
   ```

SSH access is confirmed and BGP is configured on the Region 1 and Region 2 worker nodes.

What to do next

Join the Geo-HA nodes.

Join the Geo-HA nodes

Use this procedure to join the Geo-HA nodes. It helps you complete the related installation workflow and confirm the deployment state before continuing.

Join the Region 1 worker, Region 2 worker, and witness nodes into a Geo-HA 1+1+1 topology.

Before you begin

Verify that BGP is configured on the worker nodes. For details, see Configure BGP on the Geo-HA nodes.

Procedure

1. Run the following command on each node and record the `CLUSTER_ID` value.

   Example

   ```
   sudo sedo supercluster status
   ```

2. On the Region 1 node, generate the join command for the Region 2 node.

Example

```
sudo sedo supercluster wait-for -b <region1_node_eastboundIP>:10443
<region2_node_CLUSTER_ID>
```

3. On the Region 2 node, run the command generated in the previous step.
4. On the Region 1 node, generate the join command for the witness node.

Example

```
sudo sedo supercluster wait-for -b <region1_node_eastboundIP>:10443
<region3_node_CLUSTER_ID>
```

5. On the witness node, run the command generated in the previous step.
6. On the Region 2 node, generate the join command for the witness node.

Example

```
sudo sedo supercluster wait-for -b <region2_node_eastboundIP>:10443
<region3_node_CLUSTER_ID>
```

7. On the witness node, run the command generated in the previous step.
8. Run the following command on each node to view the peers.

Example

```
sudo sedo supercluster status
```

9. Run the following command to verify Geo-HA connectivity.

Example

```
sudo sedo supercluster connectivity
```

The active worker, standby worker, and witness nodes are joined into the Geo-HA topology.

What to do next

Activate and verify Geo-HA.

Activate and verify Geo-HA

Use this procedure to start the Geo-HA deployment, confirm its status, secure access to the system, and open the web UI.

Activate the Geo-HA topology after all three nodes have joined, verify the deployment state, and complete the initial security and access tasks.

Before you begin

Verify that the three nodes have joined successfully. For details, see Join the Geo-HA nodes.

Procedure

1. In Region 1, start the Geo-HA deployment.

   This operation can take a few minutes.

   Example

```
sudo sedo supercluster start
Checking Supercluster connectivity...Passed
Initiating Supercluster...Done
```

2. After the connectivity check succeeds, verify that the Geo-HA deployment is started.

   Example

```
sudo sedo supercluster status
┌──────────────────────────────────────────────────────────────────────────────────────┐
│ Supercluster Status                                                                  │
├──────────────────┬───────────────────────────────────────────────────────────────────┤
│ Cluster ID       │ qYfNdRnsPIQWFFhz7KMFLiKZV3OD6rGrfmrUtbMGKk                        │
│ Cluster Name     │ cluster2                                                          │
│ Cluster Role     │ worker                                                            │
│ Peers            │ cluster1 (worker,
sgHMamJznknIncWnHp5zda7rPkrt1DepMjqWToNviCc)     │
│                  │ cluster3 (arbitrator,
Ma9nBKkW0GOXzWc3gVjqOjNBNC8oL29SvowBifu36QA) │
│ Mode             │ Running                                                           │
│ Current Active   │ cluster1                                                          │
│ Previous Active  │                                                                   │
│ Standby Clusters │ cluster2                                                          │
│ Last Switchover  │                                                                   │
│ Last Failover    │                                                                   │
│ Last Seen        │ controller-0.cluster3: 2024-09-21 12:35:03.073 +0000 UTC          │
│                  │ controller-0.cluster2: 2024-09-21 12:35:03.109 +0000 UTC          │
│ Last Peer Error  │                                                                   │
│ DB Replication   │ connecting                                                        │
│ DB Lag           │                                                                   │
└──────────────────┴───────────────────────────────────────────────────────────────────┘
```

3. Review the BGP state.

Example

```
sedo ha bgp show
```

4. Change the default password and access string of the admin user.

   Example

   ```
   sedo security user set --access role/admin --password admin
   ```

   You are prompted to enter the new password.

5. Configure additional local users if required.

   For more information, see Add local users to Cisco Optical Network Controller.

6. Open the Cisco Optical Network Controller web UI by browsing to the northbound virtual IP on port 8443.

   Example

   ```
   https://<northbound-virtual-ip>:8443/
   ```

Cisco Optical Network Controller is installed on KVM and the Geo-HA deployment is active.

Image verification for Cisco Optical Network Controller

This section explains how to verify Cisco Optical Network Controller images by validating certificates and signatures across OVA, qcow2, and system pack formats before installation or upgrade.

Image verification for qcow2 and system pack packages is the process of validating the certificate and signature that are provided with a Cisco Optical Network Controller image before installation or upgrade.

- qcow2 and system pack packages include the image file, certificate, signature, verification script, and README so that you can verify the package manually.
- A successful verification confirms that the downloaded image is signed by Cisco and can be used for installation or upgrade.

Cisco Optical Network Controller image types and verification methods

This topic lists the supported Cisco Optical Network Controller image types, the verification method for each type, and the files that are included in manually verified packages.

This reference lists the image types supported by Cisco Optical Network Controller. It describes the verification method for each delivery format. It also specifies the files provided in signed qcow2 and system pack packages.

Table 15: Supported image types and verification methods

Image type	Delivery format	Verification method	When to verify
VMware image	OVA	Verify the publisher information during the vSphere **Deploy OVF Template** workflow.	Before you complete the OVA deployment.
KVM image	Signed qcow2 package	Extract the package and run the Python verification script.	Before you copy or use the qcow2 image.

Image type	Delivery format	Verification method	When to verify
Upgrade image	Signed system pack package	Extract the package and run the Python verification script.	Before you copy the system pack file to the Cisco Optical Network Controller node.

Table 16: Files in a signed qcow2 or system pack package

File	Typical extension or name	Purpose	Usage
Image file	`.qcow2` or `.tar.gz`	Contains the software image that you deploy or upgrade.	Use this file only after verification succeeds.
Certificate file	`.pem`	Contains the public certificate that is used during verification.	Pass this file to the verification script.
Signature file	`.signature`	Contains the signature for the image file.	Pass this file to the verification script.
Verification script	`cisco_x509_verify_release.py3`	Runs certificate-chain and signature validation.	Execute the script from the extracted package directory.
README file	`CONC.readme`	Lists package requirements, command syntax, and example output.	Review this file before you run the verification command.

Verify a signed qcow2 or system pack image

Learn how to verify a signed qcow2 or system pack image before using it for Cisco Optical Network Controller installation or upgrade. This procedure ensures your installation files are authentic and safe.

Confirm the authenticity and integrity of signed qcow2 and system pack packages before deploying or upgrading Cisco Optical Network Controller.

Signed qcow2 and system pack packages include these files you can use to validate the image before deployment or upgrade:

- a certificate,
- a signature file,
- a verification script, and
- a README.

Before you begin

- Ensure that Python 3.4 or later is installed. The system must support running `openssl` and `grep` command-line calls.
- Download the signed qcow2 or system pack package from Software Download and extract it.
- Review the requirements and command syntax listed in the `CONC.readme` file.

Follow these steps to verify a signed qcow2 or system pack image.

Procedure

1. Change to the directory that contains the extracted image verification files.
2. Open the `CONC.readme` file and review the requirements and command syntax.
3. Verify that the verification script has execute permission.
4. Run the verification script on the extracted image file.

   Use the certificate, image, and signature files that were included in the extracted package.

   ```
   ./cisco_x509_verify_release.py3 -e <certificate-file>.pem -i <image-file> -s
    <signature-file>.signature -v dgst --algo sha512 --failExpiredCerts
   ```

5. Review the output and confirm that the verification completed successfully.

   A successful run verifies the certificate chain and ends with a message that states that the signature was verified successfully.

The qcow2 or system pack image is verified and ready to use.

This example command shows how to verify a qcow2 image.

```
./cisco_x509_verify_release.py3 -e 25.4.1-2.pem -i 25.4.1-2.tar.gz -s
25.4.1-2.signature -v dgst --algo sha256 --failExpiredCerts
25.4.1-2.pem expiration date is Jan 10 18:46:37 2099 GMT

CA chain genecc384 chosen based on finding
'//www.cisco.com/security/pki/certs/cgsiscap384.cer' string in eecert
Using cert chain 'genecc384' (csircap384.cer and cgsiscap384.cer)
Retrieving rootCA certificate from
https://www.cisco.com/security/pki/certs/csircap384.cer ...
Success in downloading https://www.cisco.com/security/pki/certs/csircap384.cer
Using downloaded rootCA cert
/var/folders/nc/70t5lkk50mlfd95t3l2vq6_00000gn/T/tmp1pfoov5u/csircap384.cer
Retrieving subCA certificate from
https://www.cisco.com/security/pki/certs/cgsiscap384.cer ...
Success in downloading https://www.cisco.com/security/pki/certs/cgsiscap384.cer
Using downloaded subCA cert
/var/folders/nc/70t5lkk50mlfd95t3l2vq6_00000gn/T/tmp1pfoov5u/cgsiscap384.cer
csircap384.cer expiration date is Jan 10 18:46:39 2099 GMT

cgsiscap384.cer expiration date is Jan 10 18:46:38 2099 GMT

Successfully verified root, subca and end-entity certificate chain.
Decoded Base64 signature 25.4.1-2.signature → 25.4.1-2.signature.der
Successfully fetched a public key from 25.4.1-2.pem
Successfully verified the signature of 25.4.1-2.tar.gz using 25.4.1-2.pem
```

What to do next

Table 17: Next action after verification

If...	Then...
You verified a qcow2 image	Copy or place the qcow2 file where it is needed for the KVM installation workflow.

If...	Then...
You verified a system pack image	Copy the extracted system pack file to the Cisco Optical Network Controller node and continue with the upgrade procedure.

A Platform and software compatibility matrix

This section provides platform and software compatibility details for Cisco Optical Network Controller. Use it to verify supported combinations before you plan deployment or upgrade activities.

Use this reference to understand platforms software compatibility matrix.

Details about platforms software compatibility matrix.

Table 18: Details

Cisco ONC version	Platform (XR) Software Release	NCS 1010		NCS 1014		NCS 1004		NCS 1001		NCS 2000	
		Support	CSM Host	Support	CSM Host	Support	CSM Host	Support	CSM Host	Support	CSM Host
24.3.1 and 24.3.2	7.10.1	–	–	–	–	–	–	✓	–	–	–
	24.3.1	All C band cards and passives	✓	All cards except 1.2T	✓	All cards except 1.2T and QXP	–	–	–	–	–
25.1.1	7.10.1	–		–		–	–	✓	–	–	–
	24.3.1	All C band cards and passives	✓	All cards except 1.2T	✓	✓	–	–	–	–	–
	25.1.1	All C band cards and passives	✓	✓	✓	–	–	–	–	✓	✓
25.1.2	7.10.1	–		–		–	–	✓	–	–	–
	24.3.1	All C band cards and passives	✓	All cards except 1.2T	✓	✓	–	–	–	–	–
	25.1.1	All C band cards and passives	✓	✓	✓	–	–	–	–	✓	✓
	25.2.1	–	–	✓	✓	–	–	–	–	–	–

CONC version	IOS XR Software Release	NCS 1010		NCS 1014		NCS 1004		NCS 1001		NCS 2000	
		Support	CSM Host	Support	CSM Host	Support	CSM Host	Support	CSM Host	Support	CSM Host
26.1.1	7.10.1	–	–	–	–	–	–	✓	✓	–	–
	24.3.1	All C band cards and passives	✓	All cards	✓	All cards	×	–	–	–	–
	25.1.1	All C band cards and passives	✓	All cards	✓	–	–	–	–	✓	✓
	25.2.1	–	×	All cards	✓	–	–	–	–	–	–
	25.4.1	–	–	–	–	–	–	✓	✓	–	–
	26.1.1	All C band cards and passives	✓	All cards	✓	All cards	✓	–	–	✓	✓