

Deploying in Linux KVM

- Prerequisites and guidelines for deploying the Nexus Dashboard cluster in Linux KVM, on page 1
- About the deployment script for QCOW2 images, on page 3
- Deploy Nexus Dashboard in Linux KVM, on page 11

Prerequisites and guidelines for deploying the Nexus Dashboard cluster in Linux KVM

Note Refer to the *Cisco Nexus Dashboard on KVM* document for platform-level guidance for maintaining Nexus Dashboard availability when users patch, upgrade, or otherwise modify the Linux KVM host.

Before you proceed with deploying the Nexus Dashboard cluster in a Linux KVM, the KVM must meet these prerequisites and you must follow these guidelines:

- The KVM form factor must support your scale requirements.

  Scale support and co-hosting vary based on the cluster form factor. You can use the Nexus Dashboard Capacity Planning tool to verify that the virtual form factor satisfies your deployment requirements.

- Review and complete the general prerequisites described in Prerequisites and Guidelines.
- The CPU family used for the Nexus Dashboard VMs must support the AVX instruction set.
- The KVM must have enough system resources, and each node requires a dedicated disk partition. See Understanding system resources, on page 3 for more information.
- The disk must have I/O latency of 20ms or less.

  See Verify the I/O latency of a Linux KVM storage device, on page 2

- Cisco does not support the use of nested virtualization environments. Deploying Nexus Dashboard on a virtual machine that is itself running on a virtualized hypervisor (for example, KVM on ESXi) is not a supported configuration and may result in performance degradation or system instability.
- By default, many RAID controllers are set to `Write Through` mode. In this mode, the controller ensures that the data is physically written to the SSD flash before confirming success to the operating system.

Because of the "handshake" required for every synchronous write, disk I/O latencies can spike more than the recommended value. This is usually happens when Telemetry is enabled on Nexus Dashboard, which requires a high disk I/O.

To optimize disk I/O performance, the RAID controller's onboard RAM must be used as a high-speed buffer. Make sure that the "Write Policy" is set to `Write Back` (WB). This allows the controller to acknowledge writes immediately using its onboard cache.

- KVM deployments are supported for NX-OS and ACI fabrics, as well as for SAN deployments.
- You must deploy in Red Hat Enterprise Linux 8.8, 8.10, or 9.4. In addition, Cisco strongly recommends that all nodes are running on the same Linux version.
- In order for Nexus Dashboard to be up and running on OS reboot scenarios, you must add the UUIDs in the `fstab conf` files of your RHEL host operating system, which is the only way to preserve the Nexus Dashboard upon a reboot of the RHEL operating system.
- You must also configure the following required network bridges at the host level for Nexus Dashboard deployments:
  - Management Network Bridge (mgmt-bridge): The external network to manage Nexus Dashboard.
  - Data Network Bridge (data-bridge): The internal network used to form clustering within Nexus Dashboard.
- We recommend that each Nexus Dashboard node is deployed in a different KVM hypervisor.

Verify the I/O latency of a Linux KVM storage device

When you deploy a Nexus Dashboard cluster in a Linux KVM, the storage device of the KVM must have a latency under 20ms. This is done in the Performance Validation step if you use the `kvm_deployer` deployment script for QCOW2 images, as described in About the deployment script for QCOW2 images, on page 3.

Follow these steps if you want to manually verify the I/O latency of a Linux KVM storage device.

Procedure

Step 1 Create a test directory.

For example, create a directory named `test-data`.

Step 2 Run the Flexible I/O tester (FIO).

```
# fio --rw=write --ioengine=sync --fdatasync=1 --directory=test-data --size=22m --bs=2300 --name=mytest
```

Step 3 After you use the command, confirm that the `99.00th=[value]` in the `fsync/fdatasync/sync_file_range` section is under 20ms.

Understanding system resources

When deploying a Nexus Dashboard cluster in Linux KVM, the KVM must have enough system resources. There are multiple form factors supported with a virtual Nexus Dashboard KVM, and the amount of system resources needed for each node differs based on the form factor.

Table 1: Per node resource requirements

Form factor	Number or vCPUs	RAM size	Disk size
1-node KVM (app)	16	64 GB	550 GB
1-node KVM (data)	32	128 GB	3 TB
3-node KVM (app)	16	64 GB	550 GB
3-node KVM (data)	32	128 GB	3 TB

You will need to know the information above for your form factor when you go through the procedures in Deploy Nexus Dashboard in Linux KVM, on page 11.

About the deployment script for QCOW2 images

Nexus Dashboard supports KVM virtualization across various environments, including RHEL and Nutanix. Nexus Dashboard deployments are typically done using QCOW2 images; however, the diversity of KVM variants and host configurations creates challenges in tracking supported versions and ensuring optimal device emulation for performance.

The deployment script for QCOW2 images addresses these challenges by providing a deployment validation and orchestration script that is packaged with the QCOW2 image. This script is designed to streamline the deployment process, ensure compatibility, and help maintain consistent performance standards across supported platforms.

Key Features

These are the key features for the deployment script for QCOW2 images:

- **Platform Compatibility Check:** Verifies that the underlying host operating system is a supported distribution and version.
- **Resource Validation:** Assesses whether the host meets minimum compute requirements, including CPU, memory, and storage resources.
- **Performance Validation:**
  - **Disk I/O and Network Latency:** Measures key performance indicators such as disk I/O and network I/O latency, ensuring they fall within qualified ranges.
  - **Device Emulation:** Confirms that the correct device emulation (such as network interfaces and virtual disk devices) is available for optimal performance.

- **Deployment Orchestration:** Automates the deployment process for both single-node and multi-node clusters, orchestrating virtual machines with the appropriate device mappings (such as network and storage devices).

Because storage is the most common bottleneck, the script categorizes storage into "Performance Tiers":

- **Disk Type & Placement (SSD/NVMe):** Validates for high-IOPs workloads.
- **RAID vs. JBOD:** Validates write speed.
- **Remote Storage (iSCSI/FC/NFS):** Performs a "Pre-Flight Latency Check":
  - **Sync Latency:** Must be < 10ms for standard apps and < 1ms for high-performance databases.
  - **IOPS Validation:** Uses a temporary `fio` test to ensure the required IOS.

Prerequisites for the deployment script for QCOW2 images

These are the prerequisites before using the deployment script for QCOW2 images.

Requirement	Details
ND qcow2 image	A local path to the Nexus Dashboard `.qcow2` image file
Network info	3 management IPs + a gateway
Go toolchain	Go 1.22+ to build from source (or use a pre-built binary)

Use the deployment script for QCOW2 images

Procedure

Step 1 Locate the tar file with the deployment script.

The tar file should be available at the same location as the QCOW2 image. For example, in the Software Download site for Nexus Dashboard for the 4.3.1 release and later, you should see the tar file with the deployment script for QCOW2 images at the same area as the QCOW2 image itself, such as this:

```
nd-dk9.4.3.1.175.qcow2.tar.gz
```

Step 2 Download the tar file with the deployment script.

Step 3 Extract the tar file.

For example:

```
tar -xzvf nd-dk9.4.3.1.175.qcow2.tar.gz
```

You should see output similar to the following after extracting the tar file and listing the contents in the directory:

```
#ls -lrt
total 24730576
-rwxr-xr-x 1 root root    35991736 May  5 23:35 kvm_deployer
```

```
-rw-r--r-- 1 root root 12651498496 May 17 03:44 nd-dk9.4.3.1.175.qcow2
-rw-r--r-- 1 root root 12636610743 May 18 22:44 nd-dk9.4.3.1.175.qcow2.tar.gz
```

Step 4 Use the **kvm_deployer config** command to create a yaml file to use with the deployment script.

Make the appropriate choices at each stage based on your deployment type.

- For example, to generate a one-node yaml file:

```
# ./kvm_deployer config

Configure Nexus Dashboard deployment:
 Deployment Platform (kvm/nutanix): kvm
 Image Path (qcow2): /home/nd-base/nd-dk9.4.3.1.58.qcow2
 Cluster Size: 1
 VM Type (app/data/large/xlarge): app
 Admin Password:
Configure current node:
 KVM Host:
  Management Network Bridge: nd_mgmt
  Data Network Bridge: nd_data
  Disk Path (device or mount point, absolute path): /home/nd-node1
 Nexus Dashboard:
  VM Name (lowercase, numbers, dashes, max 63 chars): node1
  Management Network (e.g., 192.168.1.100/24): 192.168.1.128/25
  Management Gateway IP: 192.168.1.129
[□] Configuration Summary:
 Deployment Platform        : kvm
 Image Path (qcow2)         : /home/nd-base/nd-dk9.4.3.1.58.qcow2
 Admin Password             : [hidden]
 VM Type                    : app
 VMs:
  VM Name                    : node1
   Disk Path                  : /home/nd-node1
   Management Network Bridge  : nd_mgmt
   Data Network Bridge        : nd_data
   Management Network         : 192.168.1.128/25
   Management Gateway IP      : 192.168.1.129
 Proceed with current config? (y/n): y
 File name to save configuration: kvm_1node_deploy.yaml
```

- To generate a three-node yaml file:

```
# ./kvm_deployer config
Configure Nexus Dashboard deployment:
 Deployment Platform (kvm/nutanix): kvm
 Image Path (qcow2): /home/nd-base/nd-dk9.4.3.1.175.qcow2
 Cluster Size: 3
 VM Type (app/data/large/xlarge): app
 Admin Password:
Configure current node:
 KVM Host:
  Management Network Bridge: nd_mgmt
  Data Network Bridge: nd_data
  Disk Path (device or mount point, absolute path): /home/nd-node1
 Nexus Dashboard:
  VM Name (lowercase, numbers, dashes, max 63 chars): node1
  Management Network (e.g., 192.168.1.100/24): 192.168.1.128/25
  Management Gateway IP: 192.168.1.129
Configure remote to deploy vm (2):
 KVM Host:
  IP Address: 192.168.1.133
  Username (ssh): root
. Password (ssh: root):
```

```
  Management Network Bridge: nd_mgmt
  Data Network Bridge: nd_data
  Disk Path (device or mount point, absolute path): /home/nd-node2
 Nexus Dashboard:
  VM Name (lowercase, numbers, dashes, max 63 chars): node2
  Management Network (e.g., 192.168.1.100/24): 192.168.1.128/25
  Management Gateway IP: 192.168.1.130
Configure remote to deploy vm (3):
 KVM Host:
  IP Address: 192.168.1.133
  Username (ssh): root
. Password (ssh: root):
  Management Network Bridge: nd_mgmt
  Data Network Bridge: nd_data
  Disk Path (device or mount point, absolute path): /home/nd-node3
 Nexus Dashboard:
  VM Name (lowercase, numbers, dashes, max 63 chars): node3
  Management Network (e.g., 192.168.1.100/24): 192.168.1.128/25
  Management Gateway IP: 192.168.1.131
[□] Configuration Summary:
 Deployment Platform        : kvm
 Image Path (qcow2)         : /home/nd-base/nd-dk9.4.3.1.175.qcow2
 Admin Password             : [hidden]
 VM Type                    : app
 VMs:
  VM Name                    : node1
   Disk Path                  : /home/nd-node1
   Management Network Bridge  : nd_mgmt
   Data Network Bridge        : nd_data
   Management Network         : 192.168.1.128/25
   Management Gateway IP      : 192.168.1.129
  Host OS IP Address         : 192.168.1.133
  Host OS Username           : root
  Host OS Password           : [hidden]
  VM Name                    : node2
   Disk Path                  : /home/nd-node2
   Management Network Bridge  : nd_mgmt
   Data Network Bridge        : nd_data
   Management Network         : 192.168.1.128/25
   Management Gateway IP      : 192.168.1.130
  Host OS IP Address         : 192.168.1.133
  Host OS Username           : root
  Host OS Password           : [hidden]
  VM Name                    : node3
   Disk Path                  : /home/nd-node3
   Management Network Bridge  : nd_mgmt
   Data Network Bridge        : nd_data
   Management Network         : 192.168.1.128/25
   Management Gateway IP      : 192.168.1.131
 Proceed with current config? (y/n): y
 File name to save configuration: kvm_3node_deploy.yaml
```

Step 5 (Optional) Use a text editor to verify the information saved in the yaml file, if necessary.

For example, for the three-node example above, the yaml file would have information similar to the following:

```
deployment_type: kvm
image_path: /home/nd-base/nd-dk9.4.3.1.175.qcow2
cluster_size: "3"
vms:
- name: node1
  disk_path: /home/nd-node1
  management_network: 192.168.1.128/25
```

```
  management_gateway: 192.168.1.129
  management_bridge: nd_mgmt
  data_bridge: nd_data
  host_ip: ""
  host_user: ""
  host_password: ""
- name: node2
  disk_path: /home/nd-node2
  management_network: 192.168.1.135/25
  management_gateway: 192.168.1.129
  management_bridge: nd_mgmt
  data_bridge: nd_data
  host_ip: 192.168.1.133
  host_user: root
  host_password: <host_password>
- name: node3
  disk_path: /home/nd-node3
  management_network: 192.168.1.136/25
  management_gateway: 192.168.1.129
  management_bridge: nd_mgmt
  data_bridge: nd_data
  host_ip: 192.168.1.133
  host_user: root
  host_password: <hostpword>
flavor: app
admin_password: <adminpword>
```

Step 6 Run the deployment script, using the yaml file as the input.

```
./kvm_deployer deploy --config kvm_3node_deploy.yaml
```

Output similar to the following displays:

```
Deploying VM node1 using kvm
  Cleanup old images
  [□] Cleaned image /home/nd-node1/boot.img
  [□] Cleaned image /home/nd-node1/disk.img
  [□] Cleaned image /home/nd-node1/config-drive.img
  Creating boot disk, it can take several minutes, pls wait
  [□] Boot disk created successfully
  Creating raw disk, it can take several minutes, pls wait
  [□] Raw disk created successfully
  Creating configuration drive
  [□] Configuration drive created successfully

  Starting VM
  [□] VM started successfully
[□] KVM VM: node1 deployed successfully
Deploying VM node2 using kvm
 Copying file /home/nd-base/nd-dk9.4.3.1.175.qcow2 to /home/nd-node2/nd-dk9.4.3.1.175.qcow2
 [□] File /home/nd-base/nd-dk9.4.3.1.175.qcow2 copied to /home/nd-node2/nd-dk9.4.3.1.175.qcow2
  Cleanup old images
  [□] Cleaned image /home/nd-node2/boot.img
  [□] Cleaned image /home/nd-node2/disk.img
  [□] Cleaned image /home/nd-node2/config-drive.img
  Creating boot disk, it can take several minutes, pls wait
  [□] Boot disk created successfully
  Creating raw disk, it can take several minutes, pls wait
  [□] Raw disk created successfully
  Creating configuration drive
  Copying file /home/nd-node1/node2-bootstrap.json to /home/nd-node2/apic-sn-cfg-mount/bootstrap.json

  [□] File /home/nd-node1/node2-bootstrap.json copied to /home/nd-node2/apic-sn-cfg-mount/bootstrap.json

  [□] Configuration drive created successfully
```

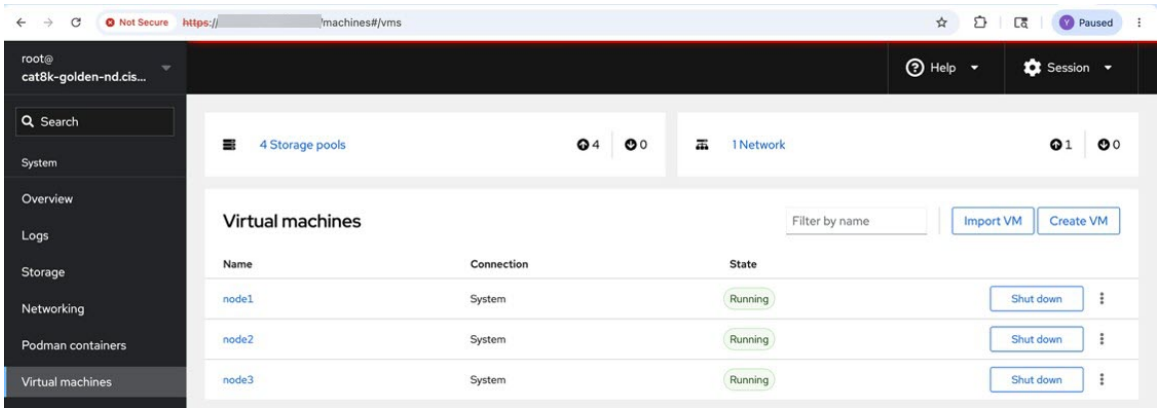
```
  Starting VM
  [□] VM started successfully
[□] KVM VM: node2 deployed successfully
Deploying VM node3 using kvm
 Copying file /home/nd-base/nd-dk9.4.3.1.175.qcow2 to /home/nd-node3/nd-dk9.4.3.1.175.qcow2
 [□] File /home/nd-base/nd-dk9.4.3.1.175.qcow2 copied to /home/nd-node3/nd-dk9.4.3.1.175.qcow2
  Cleanup old images
  [□] Cleaned image /home/nd-node3/boot.img
  [□] Cleaned image /home/nd-node3/disk.img
  [□] Cleaned image /home/nd-node3/config-drive.img
  Creating boot disk, it can take several minutes, pls wait
  [□] Boot disk created successfully
  Creating raw disk, it can take several minutes, pls wait
  [□] Raw disk created successfully
  Creating configuration drive
  Copying file /home/nd-node1/node3-bootstrap.json to /home/nd-node3/apic-sn-cfg-mount/bootstrap.json

  [□] File /home/nd-node1/node3-bootstrap.json copied to /home/nd-node3/apic-sn-cfg-mount/bootstrap.json

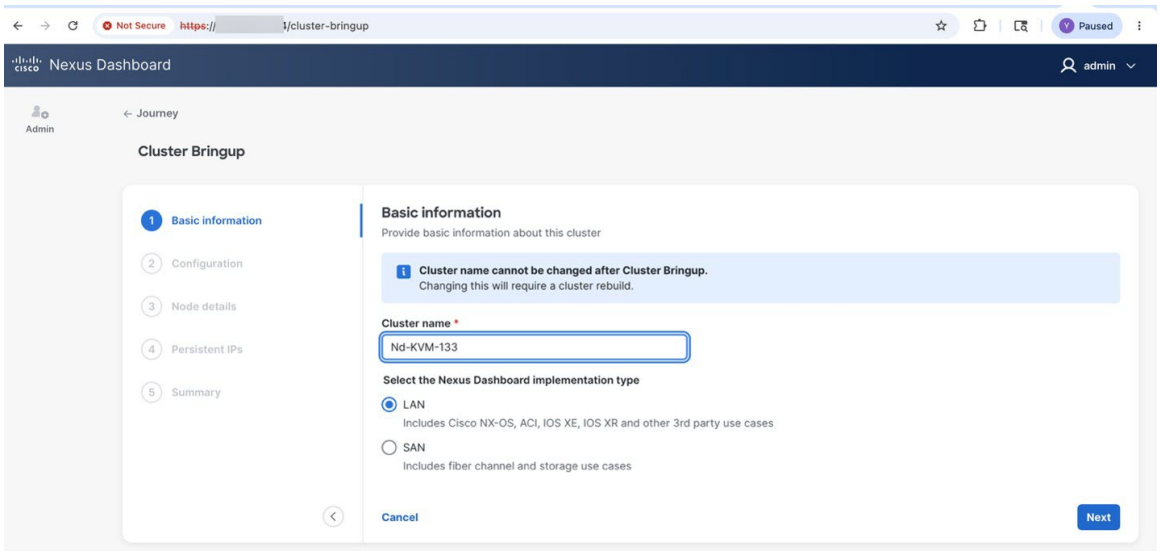
  [□] Configuration drive created successfully
  Starting VM
  [□] VM started successfully
[□] KVM VM: node3 deployed successfully
```

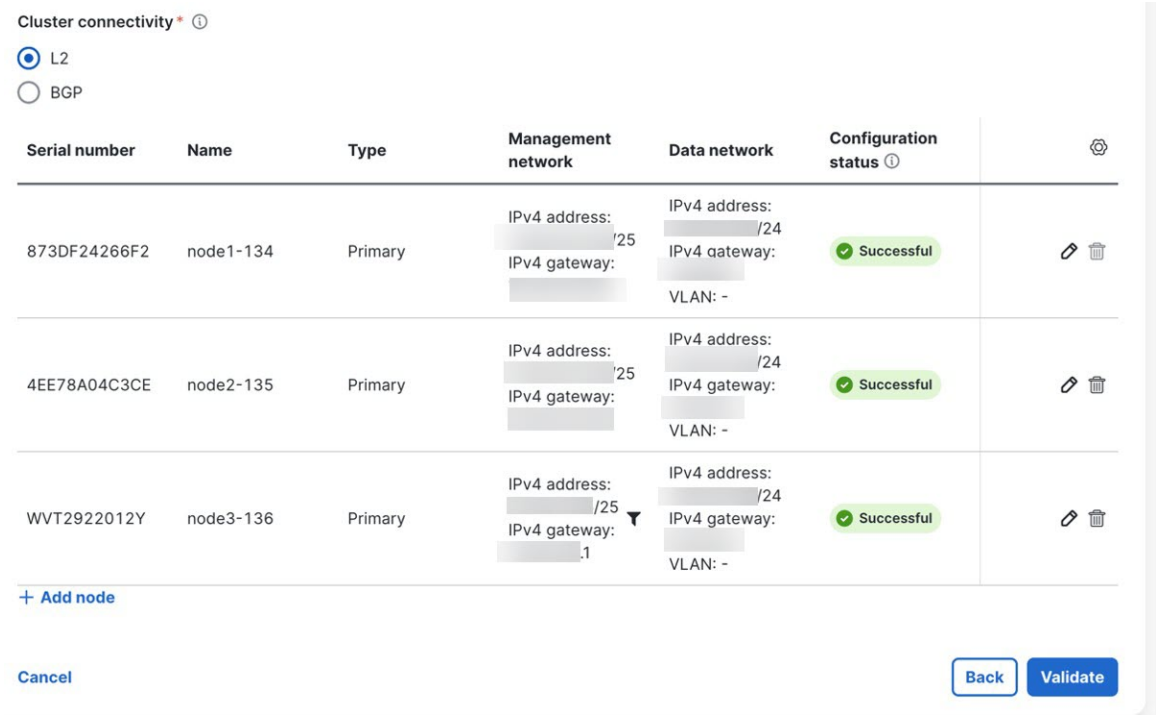
Step 7 Complete the post-deployment steps.

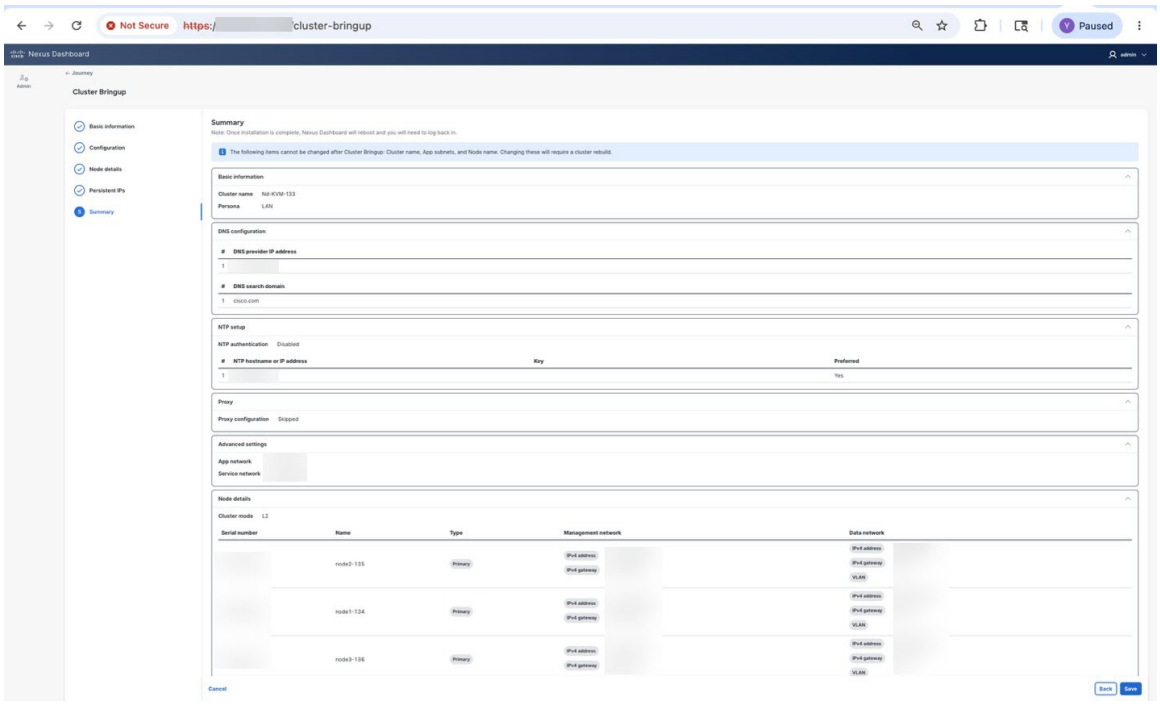
a) **Verify VMs in RHEL host**— Open https://<RHEL_host _ip>:9090 and confirm all three VMs are powered on and healthy.



b) **Cluster formation** — Once all the nodes are running, complete the Nexus Dashboard cluster setup through the Nexus Dashboard UI using the management IP addresses from your config.


Not Secure
https://
/cluster-bringup
Paused
Nexus Dashboard
admin
Admin
Journey
Cluster Bringup
1 Basic information
2 Configuration
3 Node details
4 Persistent IPs
5 Summary
Basic information
Provide basic information about this cluster
Cluster name cannot be changed after Cluster Bringup.
Changing this will require a cluster rebuild.
Cluster name *
Nd-KVM-133
Select the Nexus Dashboard implementation type
LAN
Includes Cisco NX-OS, ACI, IOS XE, IOS XR and other 3rd party use cases
SAN
Includes fiber channel and storage use cases
Cancel
Next


Cluster connectivity *
L2
BGP
Serial number
Name
Type
Management network
Data network
Configuration status
873DF24266F2
node1-134
Primary
IPv4 address:
/25
IPv4 gateway:
IPv4 address:
/24
IPv4 gateway:
VLAN: -
Successful
4EE78A04C3CE
node2-135
Primary
IPv4 address:
/25
IPv4 gateway:
IPv4 address:
/24
IPv4 gateway:
VLAN: -
Successful
WVT2922012Y
node3-136
Primary
IPv4 address:
/25
IPv4 gateway:
.1
IPv4 address:
/24
IPv4 gateway:
VLAN: -
Successful
+ Add node
Cancel
Back
Validate



Optional post-deployment tasks

Troubleshooting

Symptom	Check
Authentication failure	Verify `username` and `password` in the config
VM not getting IP	Check that cloud-init `management_network` uses valid CIDR notation and the gateway is correct

Destroy the deployment

If you want to destroy the deployment, enter this command:

`./kvm_deployer destroy --config <yaml-filename>.yaml`

For example:

```
$ kvm_deployer destroy --config kvm_3node_deploy.yaml
Destroying VM node1 using nutanix
Destroying VM node1 using nutanix
 Looking for VM: node1
 Found VM node1 with UUID: 94fe9518-5196-4807-a6cc-183d1abe760e
 Powering off VM: 94fe9518-5196-4807-a6cc-183d1abe760e
 VM power off request submitted
 Deleting VM: 94fe9518-5196-4807-a6cc-183d1abe760e
 VM deletion task submitted: 8178938b-5014-474b-b168-34c3bf5b1577
 Waiting for task completion: 8178938b-5014-474b-b168-34c3bf5b1577
  Task status: RUNNING, progress: 0%
```

```
  Task status: SUCCEEDED, progress: 100%
 [□] Task completed successfully (status: SUCCEEDED)
 [□] VM deleted successfully
[□] VM node1 destroyed successfully
Destroying VM node2 using nutanix
Destroying VM node2 using nutanix
 Looking for VM: node2
 Found VM node2 with UUID: f1424251-182d-4518-9acc-f06ba89d4883
 Powering off VM: f1424251-182d-4518-9acc-f06ba89d4883
 VM power off request submitted
 Deleting VM: f1424251-182d-4518-9acc-f06ba89d4883
 VM deletion task submitted: a8d41687-3d84-4a38-91cc-5dfee0688351
 Waiting for task completion: a8d41687-3d84-4a38-91cc-5dfee0688351
  Task status: RUNNING, progress: 0%
  Task status: SUCCEEDED, progress: 100%
 [□] Task completed successfully (status: SUCCEEDED)
 [□] VM deleted successfully
[□] VM node2 destroyed successfully
Destroying VM node3 using nutanix
Destroying VM node3 using nutanix
 Looking for VM: node3
 Found VM node3 with UUID: 8cdc2037-fc83-43af-90f6-869c00aa2fdc
 Powering off VM: 8cdc2037-fc83-43af-90f6-869c00aa2fdc
 VM power off request submitted
 Deleting VM: 8cdc2037-fc83-43af-90f6-869c00aa2fdc
 VM deletion task submitted: 710c58a7-0728-4c4e-b63a-f6d90e3b2669
 Waiting for task completion: 710c58a7-0728-4c4e-b63a-f6d90e3b2669
  Task status: RUNNING, progress: 0%
  Task status: SUCCEEDED, progress: 100%
 [□] Task completed successfully (status: SUCCEEDED)
 [□] VM deleted successfully
[□] VM node3 destroyed successfully
$
```

Deploy Nexus Dashboard in Linux KVM

This section describes how to deploy Cisco Nexus Dashboard cluster in Linux KVM.

Before you begin

- Ensure that you meet the requirements and guidelines described in Prerequisites and guidelines for deploying the Nexus Dashboard cluster in Linux KVM, on page 1.

Procedure

Step 1 Download the Cisco Nexus Dashboard image.

a) Browse to the Software Download page.

https://software.cisco.com/download/home/286327743/type/286328258

b) Click **Nexus Dashboard Software**.

c) From the left sidebar, choose the Nexus Dashboard version you want to download.

d) Download the Cisco Nexus Dashboard image for Linux KVM (`nd-dk9.<version>.qcow2`).

Step 2 Copy the image to the Linux KVM servers where you will host the nodes.

You can use `scp` to copy the image, for example:

```
# scp nd-dk9.<version>.qcow2 root@<kvm-host-ip>:/home/nd-base
```

The following steps assume you copied the image into the `/home/nd-base` directory.

Step 3 Make the following configurations on each KVM host:

a) Edit `/etc/libvirt/qemu.conf` and make sure the user and group is correctly configured based on the ownership of the storage that you plan to use for the Nexus Dashboard deployment.

This is only required if you plan to use disk storage paths that are different from the default `libvirtd`.

b) Edit `/etc/libvirt/libvirt.conf` and uncomment `uri_default`.

c) Restart the `libvirtd` service after updating the configuration using the `systemctl restart libvirtd` command from root.

Step 4 Log in to your KVM host as the `root` user and perform the following steps to create the required disk images on each node.

As mentioned in Understanding system resources, on page 3, you will need a total of 550 GB or 3 TB of SSD storage to create two disk images:

- Boot disk based on QCOW2 image that you downloaded
- Data disk

a) Verify that you have a directory with enough space to store the VM disks (for example, `/home/nd-node1`) or mount the storage disk (raw disk or LVM) to directory `/opt/cisco/nd`.

b) Create the following script as `/root/create_vm.sh` under the root directory.



Note If you manually type this information, verify that there are no empty spaces present after any of these lines.

Create the script based on the information provided in Understanding system resources, on page 3 for your form factor.

- For 1-node or 3-node KVM (**app**) form factors:

```
#!/bin/bash -ex

# Configuration
# Name of Nexus Dashboard Virtual machine
name=nd1

# Path of Nexus Dashboard QCOW2 image.
nd_qcow2=/home/nd-base/nd-dk9.4.2.1g.qcow2

# Disk Path to storage Boot and Data Disks.
data_disk=/opt/cisco/nd/data

# Management Network Bridge
mgmt_bridge=mgmt-bridge

# Data Network bridge
data_bridge=data-bridge

# Data Disk Size
data_size=500G
```

```
# CPU Cores
cpus=16

# Memory in units of MB.
memory=65536

# actual script
rm -rf $data_disk/boot.img
/usr/bin/qemu-img convert -f qcow2 -O raw $nd_qcow2 $data_disk/boot.img
rm -rf $data_disk/disk.img
/usr/bin/qemu-img create -f raw $data_disk/disk.img $data_size
virt-install \
--import \
--name $name \
--memory $memory \
--vcpus $cpus \
--os-type generic \
--osinfo detect=on,require=off \
--check path_in_use=off \
--disk path=${data_disk}/boot.img,format=raw,bus=virtio \
--disk path=${data_disk}/disk.img,format=raw,bus=virtio \
--network bridge=$mgmt_bridge,model=virtio \
--network bridge=$data_bridge,model=virtio \
--console pty,target_type=serial \
--noautoconsole \
--autostart
```

- For 1-node or 3-node KVM (**data**) form factors:

```
#!/bin/bash -ex

# Configuration
# Name of Nexus Dashboard Virtual machine
name=nd1

# Path of Nexus Dashboard QCOW2 image.
nd_qcow2=/home/nd-base/nd-dk9.4.2.1g.qcow2

# Disk Path to storage Boot and Data Disks.
data_disk=/opt/cisco/nd/data

# Management Network Bridge
mgmt_bridge=mgmt-bridge

# Data Network bridge
data_bridge=data-bridge

# Data Disk Size
data_size=3072G

# CPU Cores
cpus=32

# Memory in units of MB.
memory=131072

# actual script
rm -rf $data_disk/boot.img
/usr/bin/qemu-img convert -f qcow2 -O raw $nd_qcow2 $data_disk/boot.img
rm -rf $data_disk/disk.img
/usr/bin/qemu-img create -f raw $data_disk/disk.img $data_size
virt-install \
--import \
--name $name \
--memory $memory \
```

```
--vcpus $cpus \
--os-type generic \
--osinfo detect=on,require=off \
--check path_in_use=off \
--disk path=${data_disk}/boot.img,format=raw,bus=virtio \
--disk path=${data_disk}/disk.img,format=raw,bus=virtio \
--network bridge=$mgmt_bridge,model=virtio \
--network bridge=$data_bridge,model=virtio \
--console pty,target_type=serial \
--noautoconsole \
--autostart
```

Note If the automatic PCI assignment does not place the management and data interfaces at `00:03.0` and `00:04.0` respectively, connectivity might fail. In that case, replace these two `--network` lines:

```
--network bridge=$mgmt_bridge,model=virtio \
--network bridge=$data_bridge,model=virtio \
```

with these lines:

```
--network bridge=$mgmt_bridge,model=virtio,address.type=pci,address.bus=0,address.slot=3 \
--network bridge=$data_bridge,model=virtio,address.type=pci,address.bus=0,address.slot=4 \
```

Step 5 Make the `create_vm.sh` script executable and run it using these commands.

```
# chmod +x /root/create_vm.sh
# /root/create_vm.sh
```

Step 6 Repeat previous steps to deploy the second and third nodes, then start all VMs.

Note If you are deploying a single-node cluster, you can skip this step.

Step 7 Open one of the node's console and configure the node's basic information.

a) Press any key to begin initial setup.

You will be prompted to run the first-time setup utility:

```
[ OK ] Started atomix-boot-setup.
       Starting Initial cloud-init job (pre-networking)...
       Starting logrotate...
       Starting logwatch...
       Starting keyhole...
[ OK ] Started keyhole.
[ OK ] Started logrotate.
[ OK ] Started logwatch.

Press any key to run first-boot setup on this console...
```

b) Enter and confirm the `admin` password

This password will be used for the `rescue-user` SSH login as well as the initial GUI password.

Note You must provide the same password for all nodes or the cluster creation will fail.

```
Admin Password:
Reenter Admin Password:
```

c) Enter the management network information.

```
Management Network:
  IP Address/Mask: 192.168.9.172/24
  Gateway: 192.168.9.1
```

d) For the first node only, designate it as the "Cluster Leader".

You will log into the cluster leader node to finish configuration and complete cluster creation.

```
Is this the cluster leader?: y
```

e) Review and confirm the entered information.

You will be asked if you want to change the entered information. If all the fields are correct, choose `n` to proceed. If you want to change any of the entered information, enter `y` to re-start the basic configuration script.

```
Please review the config
Management network:
  Gateway: 192.168.9.1
  IP Address/Mask: 192.168.9.172/24
Cluster leader: yes

Re-enter config? (y/N): n
```

Step 8 Repeat previous step to configure the initial information for the second and third nodes.

You do not need to wait for the first node configuration to complete, you can begin configuring the other two nodes simultaneously.

Note You must provide the same password for all nodes or the cluster creation will fail.

The steps to deploy the second and third nodes are identical with the only exception being that you must indicate that they are not the **Cluster Leader**.

Step 9 Wait for the initial bootstrap process to complete on all nodes.

After you provide and confirm management network information, the initial setup on the first node (`Cluster Leader`) configures the networking and brings up the UI, which you will use to add two other nodes and complete the cluster deployment.

```
Please wait for system to boot: [#########################] 100%
System up, please wait for UI to be online.

System UI online, please login to https://192.168.9.172 to continue.
```

Step 10 Open your browser and navigate to `https://<node-mgmt-ip>` to open the GUI.

The rest of the configuration workflow takes place from one of the node's GUI. You can choose any one of the nodes you deployed to begin the bootstrap process and you do not need to log in to or configure the other two nodes directly.

Enter the password you entered in a previous step and click **Login**

Step 11 After logging in, review the **Meet Nexus Dashboard** welcome screens, then proceed to the **Journey** screen. Click on **Go** under the **Cluster bringup** step to begin the bootstrap process.

Step 12 Enter the requested information in the **Basic Information** page of the **Cluster Bringup** wizard.

a) For **Cluster Name**, enter a name for this Nexus Dashboard cluster.

The cluster name must follow the RFC-1123 requirements.

b) For **Select the Nexus Dashboard Implementation type**, choose either **LAN** or **SAN** then click **Next**.

Step 13 Enter the requested information in the **Configuration** page of the **Cluster Bringup** wizard.

a) (Optional) If you want to enable IPv6 functionality for the cluster, put a check in the **Enable IPv6** checkbox.

b) Click **+Add DNS provider** to add one or more DNS servers, enter the DNS provider IP address, then click the checkmark icon.

c) (Optional) Click **+Add DNS search domain** to add a search domain, enter the DNS search domain IP address, then click the checkmark icon.

d) (Optional) If you want to enable NTP server authentication, put a check in the **NTP Authentication** checkbox.

e) If you enabled NTP authentication, click **+ Add Key**, enter the required information, and click the checkmark icon to save the information.

- **Key**–Enter the NTP authentication key, which is a cryptographic key that is used to authenticate the NTP traffic between the Nexus Dashboard and the NTP servers. You will define the NTP servers in the following step, and multiple NTP servers can use the same NTP authentication key.
- **ID**–Enter a key ID for the NTP host. Each NTP key must be assigned a unique key ID, which is used to identify the appropriate key to use when verifying the NTP packet.
- **Authentication Type**–Choose authentication type for the NTP key.
- Put a check in the **Trusted** checkbox if you want this key to be trusted. Untrusted keys cannot be used for NTP authentication.

For the complete list of NTP authentication requirements and guidelines, see General prerequisites and guidelines.

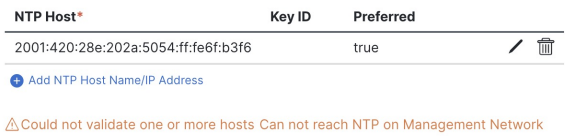
If you want to enter additional NTP keys, click **+ Add Key** again and enter the information.

f) If you enabled NTP authentication, click **+Add NTP Host Name/IP Address**, enter the required information, and click the checkmark icon to save the information.

- **NTP Host**–Enter an IP address; fully qualified domain names (FQDN) are not supported.
- **Key ID**–Enter the key ID of the NTP key you defined in the previous substep.

  If NTP authentication is disabled, this field is grayed out.
- Put a check in the **Preferred** checkbox if you want this host to be preferred.

Note If the node into which you are logged in is configured with only an IPv4 address, but you have checked **Enable IPv6** in a previous step and entered an IPv6 address for an NTP server, you will get the following validation error:



This is because the node does not have an IPv6 address yet and is unable to connect to an IPv6 address of the NTP server. You will enter IPv6 address in the next step. In this case, enter the other required information as described in the following steps and click **Next** to proceed to the next page where you will enter IPv6 addresses for the nodes.

If you want to enter additional NTP servers, click **+Add NTP Host Name/IP Address** again and enter the information.

g) For **Proxy Server**, enter the URL or IP address of a proxy server.

For clusters that do not have direct connectivity to Cisco cloud, we recommend configuring a proxy server to establish the connectivity. This allows you to mitigate risk from exposure to non-conformant hardware and software in your fabrics.

You can click **+Add Ignore Host** to enter one or more destination IP addresses for which traffic will skip using the proxy.

The proxy server must permit these URLs:

```
svc.intersight.com
svc-static1.intersight.com
svc-static1.ucs-connect.com
```

If you do not want to configure a proxy, click **Skip Proxy** then click **Confirm**.

h) (Optional) If your proxy server requires authentication, put a check in the **Authentication required for Proxy** checkbox and enter the login credentials.

i) (Optional) Expand the **Advanced Settings** category and change the settings if required.

Under advanced settings, you can configure these settings:

- **App Network**–The address space used by the application's services running in the Nexus Dashboard. Enter the IP address and netmask.
- **Service Network**–An internal network used by Nexus Dashboard and its processes. Enter the IP address and netmask.
- **App Network IPv6**–If you put a check in the **Enable IPv6** checkbox earlier, enter the IPv6 subnet for the app network.
- **Service Network IPv6**–If you put a check in the **Enable IPv6** checkbox earlier, enter the IPv6 subnet for the service network.

For more information about the application and service networks, see General prerequisites and guidelines.

j) Click **Next**.

Step 14 In the **Node Details** page, update the first node's information.

You have defined the Management network and IP address for the node into which you are currently logged in during the initial node configuration in earlier steps, but you must also enter the Data network information for the node before you can proceed with adding the other `primary` nodes and creating the cluster.

a) For **Cluster Connectivity**, if your cluster is deployed in L3 mode, choose **BGP**. Otherwise, choose **L2**.

BGP configuration is required for the persistent IP addresses feature used by telemetry. This feature is described in more detail in the BGP configuration and persistent IP addresses and Nexus Dashboard persistent IP addresses sections.



Note You can enable BGP at this time or in the Nexus Dashboard GUI after the cluster is deployed. All remaining nodes need to configure BGP if it is configured. You must enable BGP now if the data network of nodes have different subnets.

b) Click the **Edit** button next to the first node.

The node's **Serial Number**, **Management Network** information, and **Type** are automatically populated, but you must enter the other information.

c) For **Name**, enter a name for the node.

The node's **Name** will be set as its hostname, so it must follow the RFC-1123 requirements.



Note If you need to change the name but the **Name** field is not editable, run the CIMC validation again to fix this issue.

d) For **Type**, choose **Primary**.

The first nodes of the cluster must be set to **Primary**. You will add the secondary nodes in a later step if required for higher scale.

e) In the **Data Network** area, enter the node's data network information.

Enter the data network IP address, netmask, and gateway. Optionally, you can also enter the VLAN ID for the network. Leave the VLAN ID field blank if your configuration does not require VLAN. If you chose **BGP** for **Cluster Connectivity**, enter the ASN.

If you enabled IPv6 functionality in a previous page, you must also enter the IPv6 address, netmask, and gateway.



Note If you want to enter IPv6 information, you must do so during the cluster bootstrap process. To change the IP address configuration later, you would need to redeploy the cluster.

All nodes in the cluster must be configured with either only IPv4, only IPv6, or dual stack IPv4/IPv6.

f) If you chose **BGP** for **Cluster Connectivity**, then in the **BGP peer details** area, enter the peer's IPv4 address and ASN.

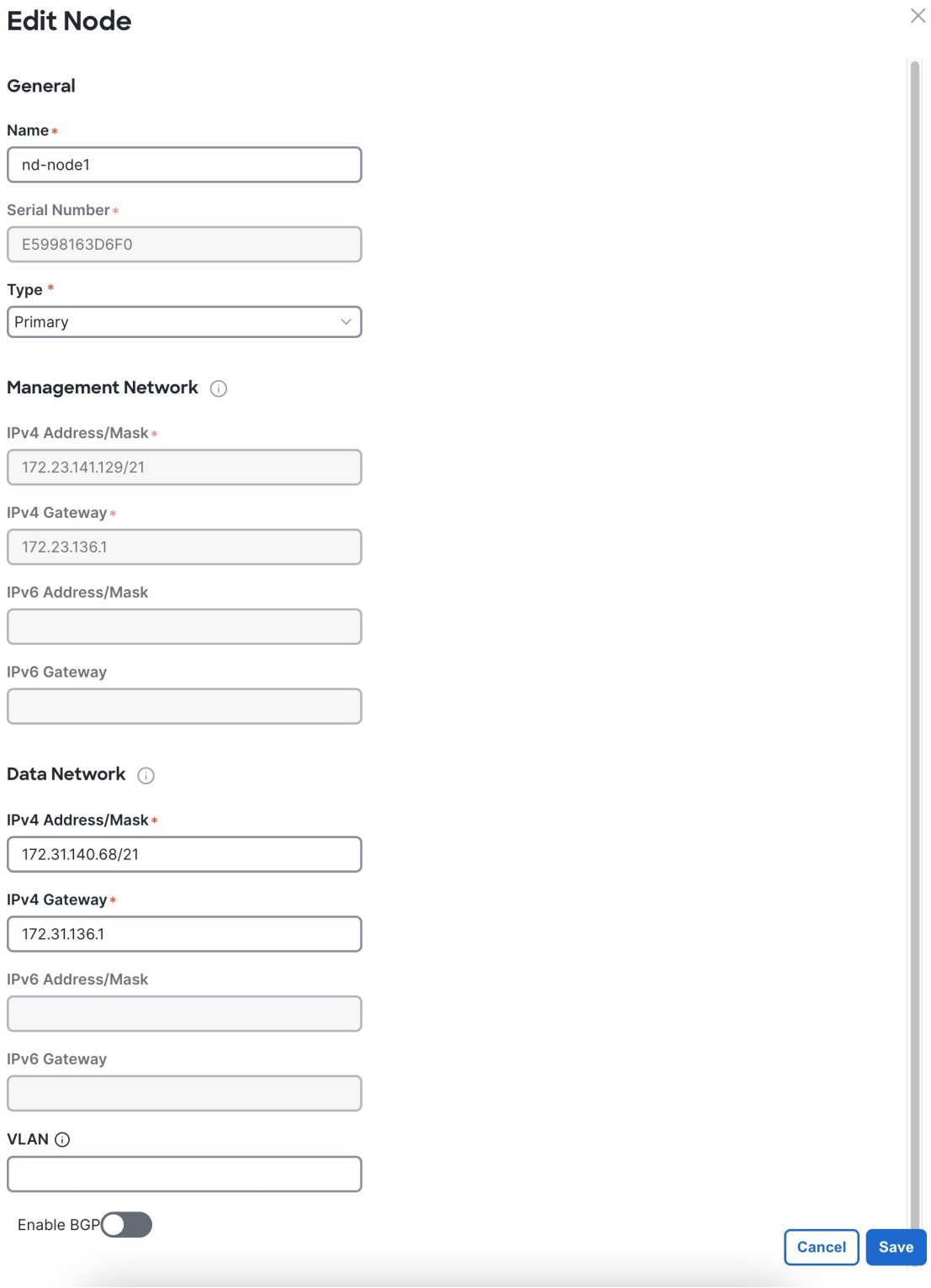
You can click **+ Add IPv4 BGP peer** to add addition peers.

If you enabled IPv6 functionality in a previous page, you must also enter the peer's IPv6 address and ASN.

g) Click **Save** to save the changes.

Step 15 In the **Node Details** screen, click **Add Node** to add the second node to the cluster.

If you are deploying a single-node cluster, skip this step.



a) In the **Deployment Details** area, provide the **Management IP Address** and **Password** for the second node

You defined the management network information and the password during the initial node configuration steps.

b) Click **Validate** to verify connectivity to the node.

The node's **Serial Number** and the **Management Network** information are automatically populated after connectivity is validated.

c) Provide the **Name** for the node.

d) From the **Type** dropdown, select `Primary`.

The first 3 nodes of the cluster must be set to `Primary`. You will add the secondary nodes in a later step if required for higher scale.

e) In the **Data Network** area, provide the node's **Data Network** information.

You must provide the data network IP address, netmask, and gateway. Optionally, you can also provide the VLAN ID for the network. For most deployments, you can leave the VLAN ID field blank.

If you had enabled IPv6 functionality in a previous screen, you must also provide the IPv6 address, netmask, and gateway.

Note If you want to provide IPv6 information, you must do it during cluster bootstrap process. To change IP configuration later, you would need to redeploy the cluster.

All nodes in the cluster must be configured with either only IPv4, only IPv6, or dual stack IPv4/IPv6.

f) (Optional) If your cluster is deployed in L3 mode, **Enable BGP** for the data network.

BGP configuration is required for the persistent IP addresses feature. This feature is described in more detail in BGP configuration and persistent IP addresses and the "Persistent IP Addresses" sections of the *Cisco Nexus Dashboard User Guide*.

Note You can enable BGP at this time or in the Nexus Dashboard GUI after the cluster is deployed.

If you choose to enable BGP, you must also provide the following information:

- **ASN** (BGP Autonomous System Number) of this node.

  You can configure the same ASN for all nodes or a different ASN per node.

- For IPv6-only, the **Router ID** of this node.

  The router ID must be an IPv4 address, for example `1.1.1.1`

- **BGP Peer Details**, which includes the peer's IPv4 or IPv6 address and peer's ASN.

g) Click **Save** to save the changes.

h) Repeat this step for the final (third) primary node of the cluster.

Step 16 In the **Node Details** page, verify the information that you entered, then click **Next**.

Step 17 Choose the **Deployment Mode** for the cluster.

a) Click **Add Persistent Service IPs/Pools** to provide the required persistent IP addresses.

For more information about persistent IP addresses, see the Nexus Dashboard persistent IP addresses section.

b) Click **Next** to proceed.

Step 18 In the **Summary** screen, review and verify the configuration information and click **Save** to build the cluster.

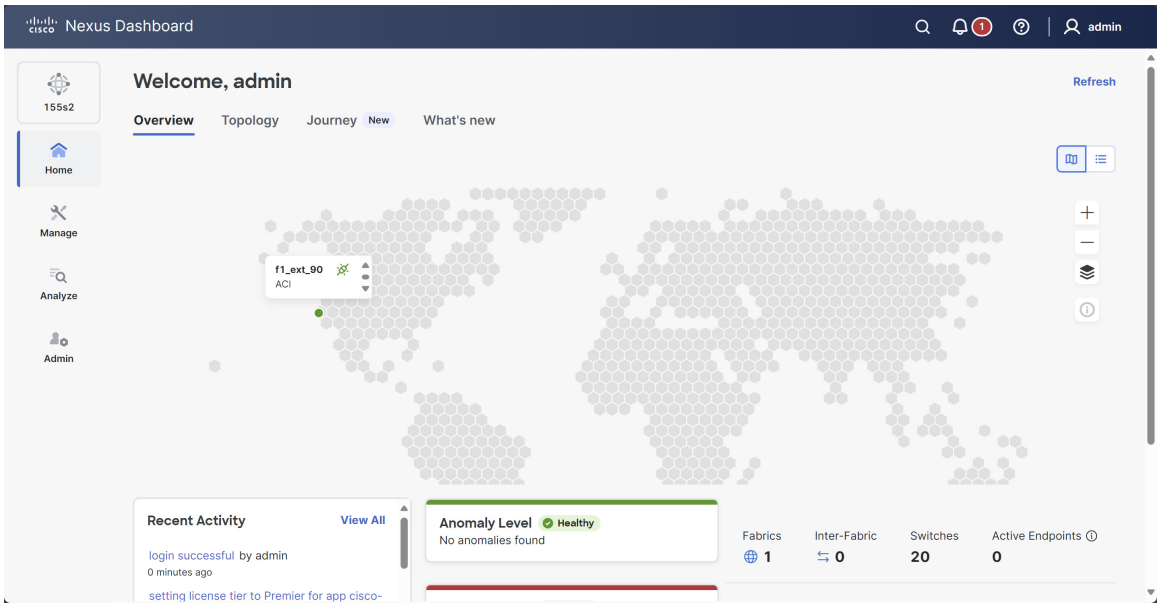
During the node bootstrap and cluster bring-up, the overall progress as well as each node's individual progress will be displayed in the UI. If you do not see the bootstrap progress advance, manually refresh the page in your browser to update the status.

It may take up to 30 minutes for the cluster to form and all the services to start. When cluster configuration is complete, the page will reload to the Nexus Dashboard GUI.

Step 19 Verify that the cluster is healthy.

After the cluster becomes available, you can access it by browsing to any one of your nodes' management IP addresses. The default password for the `admin` user is the same as the `rescue-user` password you chose for the first node. During this time, the UI will display a banner at the top stating "Service Installation is in progress, Nexus Dashboard configuration tasks are currently disabled".

After all the cluster is deployed and all services are started, you can look at the **Anomaly Level** on the **Home** > **Overview** page to ensure the cluster is healthy:



Alternatively, you can log in to any one node using SSH as the `rescue-user` using the password you entered during node deployment and using the `acs health` command to see the status:

- While the cluster is converging, you may see the following output:

```
$ acs health
k8s install is in-progress

$ acs health
k8s services not in desired state - [...]

$ acs health
k8s: Etcd cluster is not ready
```

- When the cluster is up and running, the following output will be displayed:

```
$ acs health
All components are healthy
```

Note In some situations, you might power cycle a node (power it off and then back on) and find it stuck in this stage:

```
deploy base system services
```

This is due to an issue with `etcd` on the node after a reboot of the physical Nexus Dashboard cluster.

To resolve the issue, enter the `acs reboot clean` command on the affected node.

Step 20 (Optional) Connect your Cisco Nexus Dashboard cluster to Cisco Intersight for added visibility and benefits. Refer to *Working with Cisco Intersight* for detailed steps.

Step 21 After you have deployed Nexus Dashboard, see the collections page for this release for configuration information.

What to do next

The next task is to create the fabrics and fabric groups. See the *Creating Fabrics and Fabric Groups* article for this release on the Cisco Nexus Dashboard collections page.