

Linux KVMでの展開

- Linux KVM で Nexus Dashboard クラスタを展開するための前提条件と注意事項 （1 ページ）
- QCOW2 イメージのデプロイメント スクリプトについて （3 ページ）
- Linux KVM での Nexus ダッシュボードの展開 （12 ページ）

Linux KVM で Nexus Dashboard クラスタを展開するための前提条件と注意事項

（注） ユーザーがLinux KVMホストにパッチを適用、アップグレード、または変更するときにNexus Dashboard の可用性を維持するためのプラットフォーム レベルのガイダンスについては、『*Cisco Nexus Dashboard on KVM*』 ドキュメントを参照してください。

Linux KVM で Nexus Dashboard クラスタの展開に進む前に、KVM が次の前提条件を満たしている必要があり、次の注意事項に従う必要があります。

- KVM フォーム ファクタが拡張性要件をサポートする必要があります。

  クラスタ フォーム ファクタに基づいて、拡張性サポートおよび共同ホストは変わります。Nexus ダッシュボード キャパシティ プランニング ツールを使用して、仮想フォーム ファクタが展開要件を満たすことを確認できます。

- 前提条件とガイドライン に記載されている一般的な前提条件を確認して完了します。
- Nexus Dashboard VM に使用される CPU ファミリが AVX 命令セットをサポートしている必要があります。
- KVM には十分なシステムリソースが必要です。また、各ノードには専用のディスクパーティションが必要です。詳細については、「システム リソースを理解する （3ページ）」を参照してください。
- ディスクの I/O 遅延は 20 ミリ秒以下である必要があります。

「Linux KVM ストレージ デバイスの I/O 遅延の確認 （2 ページ）」を参照してください。

- Cisco は、ネストされた仮想化環境の使用をサポートしていません。仮想化ハイパーバイザ（例：ESXi上のKVM）上で動作している仮想マシンにNexus Dashboardを展開することは、サポート対象外の構成であり、パフォーマンスの低下やシステムの不安定化を招く可能性があります。
- デフォルトでは、多くの RAID コントローラが ライト スルー モードに設定されています。このモードでは、オペレーティング システムの成功を確認する前に、コントローラはデータが SSD フラッシュに物理的に書き込まれることを確認します。

  すべての同期書き込みに必要な「ハンドシェイク」により、ディスク I/O 遅延は推奨値を超えてスパイク可能性があります。これは通常、 Nexus Dashboard でテレメトリが有効になっている場合に発生し、高いディスク I/O を必要とします。

  ディスク I/O のパフォーマンスを最適化するには、 RAIDコントローラのオンボードRAMを高速バッファとして使用する必要があります。「書き込みポリシー」が ライトバック (WB) に設定されていることを確認します。これにより、コントローラはオンボードキャッシュを使用して書き込みをすぐに確認できます。
- KVM の導入は、NX- OS および ACI ファブリックのみならず、SAN の展開でサポートされています。
- Red Hat Enterprise Linux 8.8、8.10、または 9.4 に展開する必要があります。また、すべてのノードが同じ Linux バージョンで実行されていることを強く推奨します。
- OS のリブート時にNexus Dashboardを稼働させるには、RHEL ホスト オペレーティング システムの `fstab設定` ファイルにUUIDを追加する必要があります。これが、RHEL オペレーティング システムの再起動時にNexus Dashboard を保持できる唯一の方法です。
- また、 Nexus Dashboad の展開に必要な次のネットワーク ブリッジをホスト レベルで構成する必要があります。
  - 管理網ブリッジ（mgmt-bridge）： Nexus Dashboad を管理するための外部ネットワーク。
  - データ網ブリッジ（data-bridge）： Nexus Dashboad 内でクラスタリングを形成するために使用される内部ネットワーク。
- 各 Nexus Dashboard ノードは異なる KVM ハイパーバイザに展開することを推奨します。

Linux KVM ストレージ デバイスの I/O 遅延の確認

Linux KVM に Nexus Dashboard クラスタを展開する場合、KVM のストレージ デバイスの遅延は 20 ミリ秒未満である必要があります。QCOW2 イメージに `kvm_deployer` デプロイメント スクリプトを使用する場合は、「QCOW2 イメージのデプロイメント スクリプトについて （3 ページ）」で説明されているように、パフォーマンス検証手順でこれを実行します。

手動で Linux KVM ストレージ デバイスの I/O 遅延を確認したい場合は、次の手順を実行します。

手順

ステップ 1 テスト ディレクトリを作成します。

たとえば、 `test-data`という名前のディレクトリを作成します。

ステップ 2 フレキシブル I/O テスター（FIO）を実行します。

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

ステップ 3 コマンドの実行後に、`fsync/fdatasync/sync_file_range` セクションの `99.00th=[<value>]` が 20 ミリ秒未満であることを確認します。

システム リソースを理解する

Linux KVM に Nexus ダッシュボード クラスターを展開する場合、KVM には十分なシステム リソースが必要です。仮想 Nexus Dashboard KVM では複数のフォーム ファクタがサポートされており、各ノードに必要なシステム リソースの量はフォーム ファクタによって異なります。

表 *1:* ノード当たりのリソース要件

フォーム ファクタ	vCPU の数	RAM サイズ	ディスク サイズ
1 ノードKVM（アプリ）	16	64 GB	550 GB
1 ノードKVM（データ）	32	128 GB	3 TB
3 ノードKVM（アプリ）	16	64 GB	550 GB
3 ノードKVM（データ）	32	128 GB	3 TB

Linux KVM での Nexus ダッシュボードの展開 （12 ページ）の手順を実行するときに、フォーム ファクタに関する上記の情報を知っておく必要があります

QCOW2 イメージのデプロイメント スクリプトについて

Nexus Dashboard は、RHEL や Nutanix など、さまざまな環境での KVM 仮想化をサポートしています。Nexus Dashboard の展開は通常、QCOW2 イメージを使用して行われます。しかし、

KVM のバリアントとホスト構成が多様であるため、サポートされているバージョンの追跡と最適なデバイスエミュレーションのパフォーマンス確保に課題が生じています。

QCOW2 イメージのデプロイメント スクリプトは、QCOW2 イメージにパッケージ化されたデプロイメントの検証およびオーケストレーション スクリプトを提供することで、これらの課題に対処します。このスクリプトは、デプロイメント プロセスを合理化、互換性を確保し、サポートされているプラットフォーム間で一貫したパフォーマンス標準を維持するように設計されています。

主な機能

QCOW2 イメージのデプロイメント スクリプトの主な機能は次のとおりです。

- **プラットフォーム互換性チェック**：基盤となるホストのオペレーティング システムがサポートされているディストリビューションとバージョンであることを確認します。
- **リソース検証**： ホストが CPU、メモリ、ストレージ リソースなどの最小コンピューティング要件を満たしているかどうかを評価します。
- **パフォーマンス検証**：
  - **ディスク I/O と網の遅延**： ディスク I/O やネットワーク I/O の遅延などの主要なパフォーマンス インジケータを測定し、適格な範囲内にあることを確認します。
  - **デバイス エミュレーション:** 適切なデバイス エミュレーション（ネットワーク インターフェイスやリモート対応ディスク デバイスなど）が最適なパフォーマンスのために使用可能であることを確認します。
- **展開のオーケストレーション**： 単一ノードとマルチノードの両方のクラスタのデプロイメント プロセスを自動化し、適切なデバイス マッピング（ネットワークやストレージ デバイスなど）でリモート対応マシンをオーケストレーションします。

ストレージは最も一般的なボトルネックであるため、このスクリプトではストレージを「パフォーマンス層」に分類します。

- **ディスクのタイプと配置（SSD/NVMe）**： 高 IOP ワークロードを検証します。
- **RAID 対 JBOD**： 書き込み速度を検証します。
- **リモート ストレージ（iSCSI/ FC/NFS）**： 「Pre-Flight Latency Check」を実行します。
  - **同期遅延**： 標準規格アプリケーションの場合は 10 ミリ秒未満、高性能データベースの場合は 1 ミリ秒未満である必要があります。
  - **IOPS 検証**： 一時的な `FIO` テストを使用して、必要な IOS を確認します。

QCOW2 イメージのデプロイメント スクリプトの前提条件

QCOW2 イメージのデプロイメント スクリプトの前提条件

要件	詳細
ND qcow2 image	Nexus Dashboard `.qcow2` イメージ ファイルへのローカル パス
ネットワーク情報	3 管理 IP + ゲートウェイ
Go toolchain	送信元から構築するには、Go 1.22+ を使用します（または、事前作成のバイナリを使用します。）

QCOW2 イメージのデプロイメント スクリプトを使用します

手順

ステップ 1 デプロイメント スクリプトを含む tar ファイルを見つけます。

tar ファイルは、QCOW2 イメージと同じ場所に用意する必要があります。たとえば、4.3.1 リリース以降の Nexus Dashboard のソフトウェア ダウンロード サイト には、次のように QCOW2 イメージのデプロイメントスクリプトを含む tar ファイルが QCOW2 イメージ自体と同じエリアに表示されます。

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

ステップ 2 デプロイメント スクリプトを含む tar ファイルをダウンロードします。

ステップ 3 qcow2 tar ファイルを解凍します。

例：

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

tar ファイルを抽出し、ディレクトリの内容をリストした後、次のような出力が表示されます。

```
#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
```

ステップ 4 **kvm_deployer config** コマンドを使用して、デプロイメント スクリプトで使用する yaml ファイルを作成します。

デプロイメントのタイプに基づいて、各段階で適切な選択をしてください。

- たとえば、1 ノードの yaml ファイルを生成するには：

```
# ./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
```

- 3 ノードの yaml ファイルを生成する場合：

```
# ./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
```

ステップ 5 （オプション）必要に応じて、テキストエディタを使用して yaml ファイルに保存された情報を検証します。

たとえば、前述の 3 ノードの例では、yaml ファイルには次のような情報が含まれます。

```
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>
```

ステップ 6 yaml ファイルを入力として使用して、デプロイメント スクリプトを実行します。

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

次のような出力が表示されます。

```
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

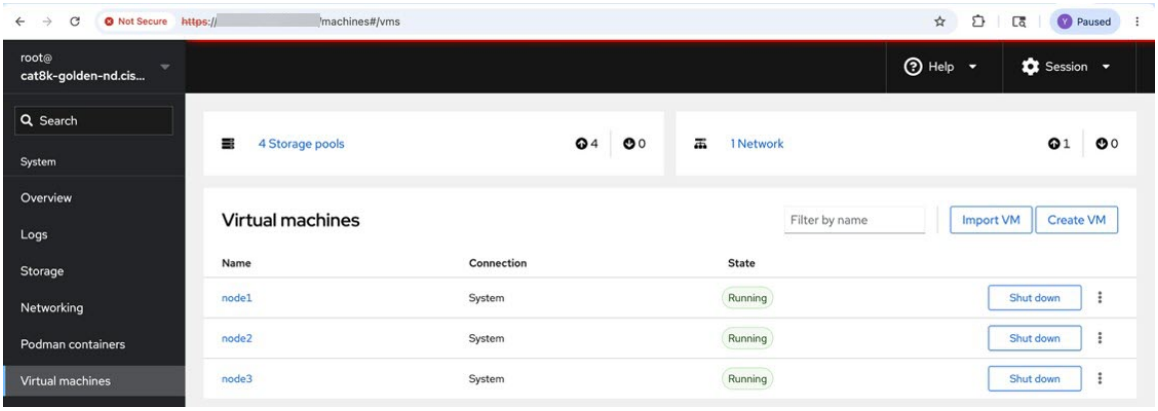
  [□] File /home/nd-node1/node2-bootstrap.json copied to
/home/nd-node2/apic-sn-cfg-mount/bootstrap.json
  [□] Configuration drive created successfully
  Starting VM
  [□] VM started successfully
[□] KVM VM: node2 deployed successfully
Deploying VM node3 using kvm
 Copying file /home/nd-base/nd-dk9.4.3.1.175.qcow2 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
```

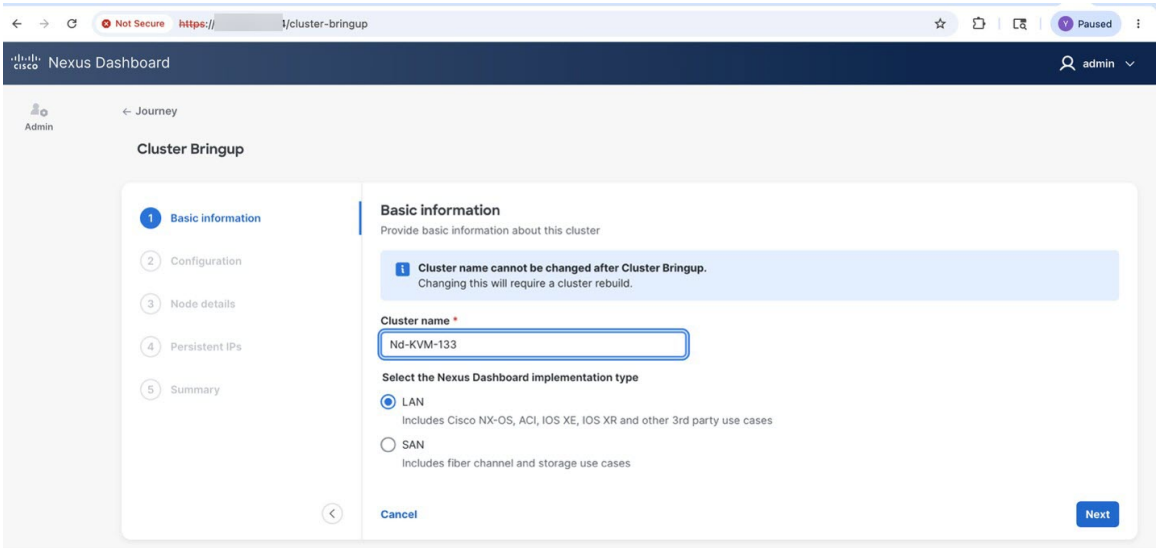
```
/home/nd-node3/apic-sn-cfg-mount/bootstrap.json
  [□] Configuration drive created successfully
  Starting VM
  [□] VM started successfully
[□] KVM VM: node3 deployed successfully
```

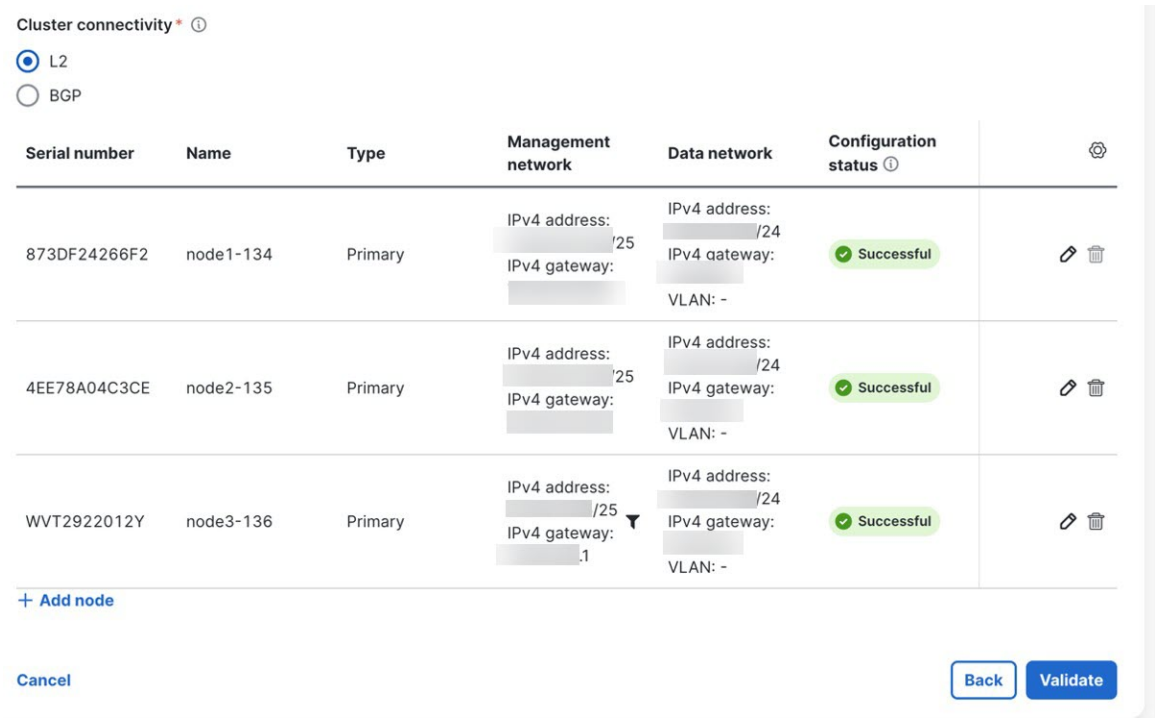
ステップ 7 展開後の手順を完了します。

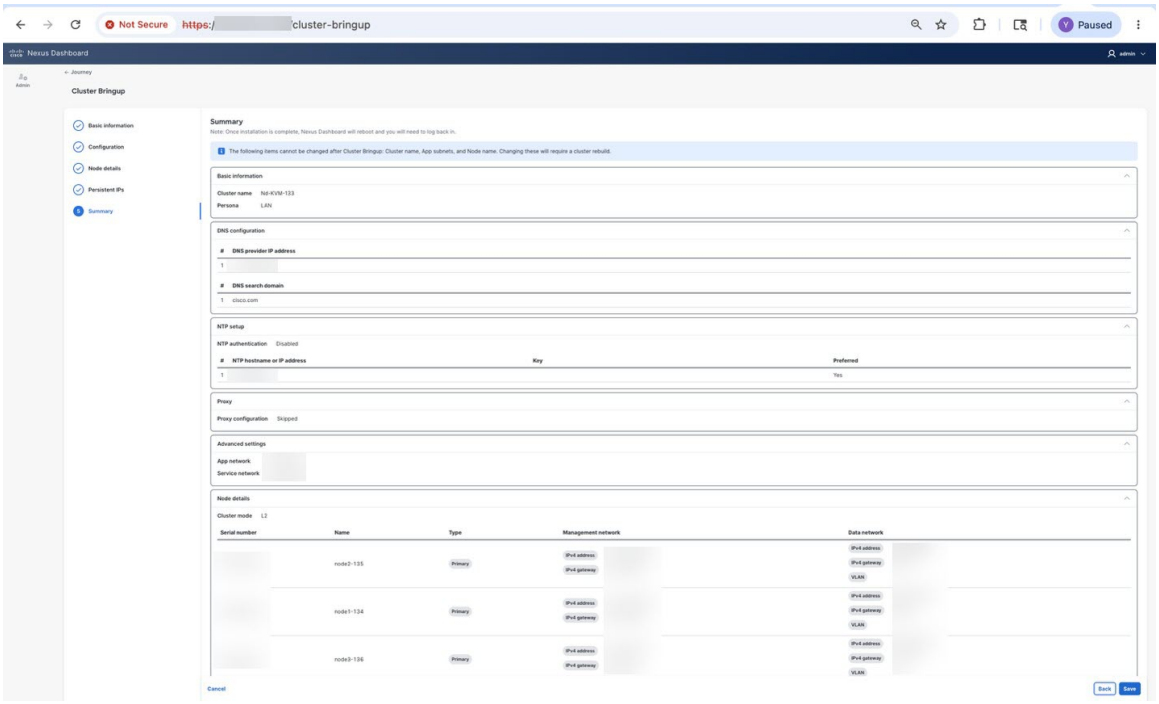
a) **RHEL ホストでのVMs の確認**：www:// を開きます<RHEL_host _ip>:9090 を実行し、3 つすべての VMs の電源がオンで、正常であることを確認します。



b) **クラスタの形成**：すべてのノードが実行されたら、設定の管理IPアドレスを使用して、Nexus Dashboard UI を介して Nexus Dashboard クラスタのセットアップを完了します。




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


Not Secure
https:/ cluster-bringup
Paused
Nexus Dashboard
Cluster Bringup
Basic information
Configuration
Node details
Persistent IPs
Summary
Summary
Basic information
DNS configuration
NTP setup
Proxy
Advanced settings
Node details
Cancel
Back
Save

オプションの展開後のタスク

トラブルシューティング

症状	検証
Authentication failure（認証失敗）	設定でユーザ名 および パスワードを確認します
VM が IP を取得できない	cloud-init management_networkが有効な CIDR 表記法および正しいゲートウェイであることを確認します。

デプロイメントを破棄します

デプロイメントを破棄したい場合このコマンドを入力します：

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

例：

```
$ 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
$
```

Linux KVM での Nexus ダッシュボードの展開

ここでは、Linux KVM で Cisco Nexus ダッシュボード クラスタを展開する方法について説明します。

始める前に

- Linux KVM で Nexus Dashboard クラスタを展開するための前提条件と注意事項 （1 ページ） に記載されている要件とガイドラインを満たしていることを確認します。

手順

ステップ 1 Cisco Nexus ダッシュボード イメージをダウンロードします。

a) [ソフトウェア ダウンロード（Software Download）] ページを参照します。

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

b) **[Nexus ダッシュボード ソフトウェア]** をクリックします。

c) 左側のサイドバーから、ダウンロードする Nexus ダッシュボードのバージョンを選択します。

d) Linux KVM の Cisco Nexus ダッシュボードイメージをダウンロードします（`nd-dk9.<version>.qcow2`）。

ステップ 2 ノードをホストする Linux KVM サーバーにイメージをコピーします。

`scp` を使用してイメージをコピーできます。次に例を示します。

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

次の手順は、イメージを `/home/nd-base` ディレクトリにコピーしたことを前提としています。

ステップ 3 各 KVM ホストで次の構成を行います。

a) `/etc/libvirt/qemu.conf` を編集、 Nexus Dashboard の展開に使用する予定のストレージの所有権に基づいて、ユーザーとグループが正しく設定されていることを確認します。

これは、デフォルトの `libvirtd`とは異なるディスク ストレージ パスを使用する場合にのみ必要です。

b) `/etc/libvirt/libvirt.conf` を編集、 `uri_default` のコメントを外します。

c) ルートから `systemctl restart libvirtd` コマンドを使用して設定を更新した後、 `libvirtd` サービスを再起動します。

ステップ 4 KVM ホストにルートユーザーとしてログインし、各ノードで必要なディスク イメージを作成するために次の手順を実行します。

システム リソースを理解する （3 ページ） で説明されているとおり、2 つのディスク イメージを作成するには、合計 550 GB または 3 TB の SSD ストレージが必要です。

- ダウンロードした QCOW2 イメージに基づいてディスクを起動します。
- データ ディスク：

a) VMディスクを保存するのに十分な空き容量があるディレクトリ（例：`/home/nd-node1`）があることを確認するか、ストレージ ディスク（raw ディスクまたは LVM）を `/opt/cisco/nd` ディレクトリにマウントしてください。

b) ルート ディレクトリの下に `/root/create_vm.sh` として次のスクリプトを作成します。



（注） この情報を手動で入力する場合は、これらの行の後に空白がないことを確認します。

システム リソースを理解する （3 ページ） に記載されている情報に基づき、次のとおり、スクリプトを作成します。

- 1ノードまたは3ノードKVM（**アプリ**）フォームファクタの場合：

```
#!/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
```

- 1ノードまたは3ノードKVM（**データ**）フォームファクタの場合：

```
#!/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
```

（注） 自動PCI割り当てによって、管理インターフェイスとデータインターフェイスがそれぞれ 00:03.0 と 00:04.0 に配置されていない場合、接続が失敗する可能性があります。その場合、これらの二つの --ネットワーク ラインを交換します：

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

これらのラインと交換します：

```
--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
 \
```

ステップ 5 create_vm.sh スクリプトを実行可能にし、これらのコマンドを使用して実行します。

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

ステップ 6 以前のステップを繰り返し、2 番目と 3 番目のノードを展開して、すべての VM を開始します。

（注） 単一のノードクラスタを展開している場合は、この手順をスキップできます。

ステップ 7 ノードのコンソールのいずれかを開き、ノードの基本情報を設定します。

a) いずれかのキーを押して、初期設定を開始します。

初回セットアップユーティリティの実行を要求するプロンプトが表示されます。

```
[ 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) admin パスワードを入力して確認します。

このパスワードは、rescue-user SSH ログインおよび初期 GUI パスワードに使用されます。

（注） すべてのノードに同じパスワードを指定する必要があります。指定しない場合、クラスタ作成に失敗します。

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

c) 管理ネットワーク情報を入力します。

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

d) 最初のノードのみ、「クラスタ リーダー」として指定します。

クラスタ リーダー ノードにログインして、設定を完了し、クラスタの作成を完了します。

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

e) 入力した譲歩をレビューし、確認します。

入力した情報を変更するかどうかを尋ねられます。すべてのフィールドが正しい場合は、nを選択して続行します。入力した情報を変更する場合は、yを入力して基本設定スクリプトを再起動します。

```
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
```

ステップ 8 前の手順を繰り返して、2 番目と 3 番目のノードの初期情報を構成します。

最初のノードの設定が完了するのを待つ必要はありません。他の 2 つのノードの設定を同時に開始できます。

（注） すべてのノードに同じパスワードを指定する必要があります。指定しない場合、クラスタ作成に失敗します。

2 番目と 3 番目のノードを展開する手順は同じですが、**クラスタ リーダー**ではないことを示す必要がある点が異なります。

ステップ 9 初期ブートストラップ プロセスを待機して、すべてのノードで完了します。

管理ネットワーク情報を入力して確認すると、最初のノード（クラスタ リーダー）初期設定でネットワーキングが設定され、UI が表示されます。この UI を使用して、他の 2 つのノードを追加し、クラスタの展開を完了します。

```
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.
```

ステップ 10 ブラウザを開き、https://<node-mgmt-ip> に移動して、GUI を開きます。

残りの設定ワークフローは、ノードの GUI の 1 つから実行します。展開したノードのいずれか 1 つを選択して、ブートストラッププロセスを開始できます。他の 2 つのノードにログインしたり、これらを直接構成したりする必要はありません。

前の手順で入力したパスワードを入力し、**[ログイン（Login）]** をクリックします。

ステップ 11 ログイン後、 **Meet Nexus Dashboard** の ウェルカム画面を確認し、 **ジャーニー** 画面に進みます。**クラスタの起動** 手順の下の **Go** をクリックして、ブートストラップ プロセスを開始します。

ステップ 12 **[クラスタのブリングアップ（Cluster Bringup）**] ウィザードの **[基本情報（Basic Information）]** ページに、必要な情報を入力します。

a) **[クラスタ名（Cluster Name）]** には、Nexus Dashboard クラスタの名前を入力します。

クラスタ名は、 RFC-1123 の要件に従う必要があります。

b) **[Nexus Dashboard の実装タイプの選択（Nexus Dashboard Implementation type）]**で、**LAN]** または **[SAN]** を選択して、 **[次へ（Next）]**をクリックします。

ステップ 13 **[クラスタのブリングアップ（Cluster Bringup）**] ウィザードの **[構成（Configuration）]** ページで、必要な情報を入力します。

a) （任意） クラスタの IPv6 機能を有効にする場合は、**[IPv6 を有効にする（Enable IPv6）]** チェックボックスをオンにします。

b) をクリックして、1 つ以上の DNS サーバーを追加し、DNS プロバイダーの IP アドレスを入力し、チェックマーク アイコンをクリックします。

c) （任意） **[+ DNS 検索ドメインの追加]**をクリックして、検索ドメインを追加し、DNS 検索ドメインの IP アドレスを入力し、チェックマーク アイコンをクリックします。

d) （任意） NTP サーバー認証を有効にする場合は、**[NTP 認証]** チェックボックスをオンにします。

e) NTP 認証を有効にした場合、**+ Add Key** をクリックし、必要な情報を入力し、チェックマーク アイコンをクリックして情報を保存します。

- **キー**：NTP 認証キーを入力します。Nexus Dashboard と NTP サーバ間の NTP トラフィックを認証するために使用される暗号キーです。次の手順で NTP サーバーを定義します。複数の NTP サーバーで同じ NTP 認証キーを使用できます。
- **ID**： NTP ホストのキー ID を入力します。各 NTP キーに一意のキー ID を割り当てる必要があります。この ID は、NTP パケットの検証時に使用する適切なキーを識別するために使用されます。
- **認証タイプ**：NTPキーの認証タイプを選択します。
- このキーを信頼したい場合には、**[信頼済み（Trusted）]** チェックボックスをオンにします。信頼できないキーは NTP 認証に使用できません。

NTP 認証の要件とガイドラインの完全なリストについては、全般的な前提条件とガイドライン を参照してください。

追加のNTPキーを入力する場合は、 **[+ キーの追加（+ Add Key）]** を再度クリックして、情報を入力します。

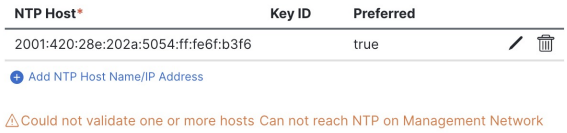
f) NTP認証を有効にした場合は、 **[+ NTP ホスト名/ IPアドレスの追加（+Add NTP Host Name/ IP Address）]**をクリックし、必要な情報を入力し、チェックマークアイコンをクリックして情報を保存します。

- **NTP ホスト**：IP アドレスを入力する必要があります。完全修飾ドメイン名（FQDN）はサポートされていません。
- **キー ID**：前のサブステップで定義した NTP キーのキー ID を入力します。

NTP 認証が無効になっている場合、このフィールドはグレー表示されます。

- このホストを優先したい場合は、**[優先（Preferred）]** チェックボックスをオンにします。

（注） ログインしているノードにIPv4アドレスのみが設定されているが、前の手順で**[IPv6 を有効にする（Enable IPv6）]** をオンにして NTP サーバーの IPv6 アドレスを指定した場合は、次の検証エラーが表示されます。



これは、ノードに IPv6 アドレスがまだなく、NTP サーバーの IPv6 アドレスに接続できないためです。次の手順で IPv6 アドレスを入力します。この場合、次の手順の説明に従って他の必要な情報の入力を完了し、**[次へ（Next）]** をクリックして次の画面に進み、ノードの IPv6 アドレスを入力します。

追加の NTP サーバーを入力する場合は、**[+ Add NTP Host Name/IP Address）]** を再度クリックし、情報を入力します。

g) **[プロキシ サーバー（Proxy Server）]** について、プロキシ サーバーの URL または IP アドレスを入力します。

Cisco Cloud に直接接続できないクラスタの場合は、接続を確立するためにプロキシ サーバを構成することをお勧めします。これにより、ファブリック内の非適合ハードウェアおよびソフトウェアにさらされるリスクを軽減できます。

+Add Ignore Host をクリックして、トラフィックがプロキシの使用をスキップする 1 つ以上の接続先 IP アドレスを入力します。

プロキシ サーバでは、永続 URL を許可する必要があります：

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

プロキシを構成しない場合は、**[プロキシをスキップ（Skip Proxy）]**をクリックして、**[確認（Confirm）]** をクリックします。

h) （任意）プロキシサーバーで認証が必要な場合は、**[プロキシに必要な認証（Authentication required for Proxy）]** をオンにして、ログイン資格情報を指定します。

i) （任意）**[詳細設定（Advanced Settings）]** カテゴリを展開し、必要に応じて設定を変更します。

詳細設定では、次の設定を行うことができます。

- **アプリ ネットワーク**：Nexus Dashboard でアプリケーションで使用されるアドレス空間です。ターゲット ネットワークの IP アドレスとネットマスクを入力します。

- **サービス ネットワーク**：Nexus Dashboard とそのプロセスで使用される内部ネットワークです。ターゲット ネットワークの IP アドレスとネットマスクを入力します。
- **[アプリ ネットワーク IPv6（App Network IPv6）]**：先ほど **[IPv6 の有効化（Enable IPv6）]** チェックボックスをオンにした場合は、アプリ ネットワークの IPv6 サブネットを入力します。
- **[サービス ネットワーク IPv6（Service Network IPv6）]**：先ほど **[IPv6 を有効にする（Enable IPv6）]** チェックボックスをオンにした場合は、サービス ネットワークの IPv6 サブネットを入力します。

アプリケーションおよびサービス ネットワークの詳細については、全般的な前提条件とガイドライン を参照してください。

j) [次へ（Next）] をクリックします。

ステップ 14 **[ノードの詳細（Node Details）]** ページで、最初のノードの情報を更新します。

前の手順の初期ノード構成時に現在ログインしているノードの管理ネットワークと IP アドレスを定義しましたが、他のプライマリ ノードを追加し、クラスタを作成する進む前に、ノードのデータ ネットワーク情報も指定する必要があります。

a) **クラスタ接続** について、クラスタが L3 モードで展開されている場合は、**BGP** を選択します。それ以外の場合は、 **L2**を選択します。

テレメトリで使用される永続的な IP アドレス機能には、BGP 構成が必要です。この機能の詳細については、 BGP 構成と永続的な IP アドレス および Nexus Dashboardの 永続 IP アドレス セクションで説明します。

（注） BGP をこの時点で、またはクラスタの展開後に Nexus ダッシュボード GUI で有効にすることができます。BGP が構成されている場合は、残りのすべてのノードで BGP を構成する必要があります。ノードのデータネットワークに異なるサブネットがある場合は、ここで BGP を有効にする必要があります。

b) 最初のノードの横にある **[編集（Edit）]**ボタンをクリックします。

ノードの**[シリアル番号（Serial Number）]**、**[管理ネットワーク（Management Network）]**情報、および**[タイプ（Type）]**が自動的に入力されます。ただし、他の情報は入力する必要があります。

c) **[名前（Name）]**に、サービス ノードのノード名を入力します。

ノードの **名前** はホスト名として設定されるため、 RFC-1123 の要件に従う必要があります。

（注） **[名前（Name）]**フィールドが編集できない場合には、CIMC の検証を再度実行して、この問題を修正してください。

d) **[タイプ（Type）]**で、 **[プライマリ（Primary）]**を選択します。

クラスタの最初のノードは **[プライマリ（Primary）]** に設定する必要があります。より大規模なスケールを有効にする必要がある場合は、後の手順でセカンダリ ノードを追加します。

e) **[データ ネットワーク（Data Network）]** エリアで、ノードのデータ ネットワークを入力します。

データ ネットワークの IP アドレス、ネットマスク、およびゲートウェイを入力します。オプションで、ネットワークの VLAN ID を指定することもできます。構成に VLAN が不要な場合は、 [VLAN ID] フィールドを空白のままにします。**データ接続** に **BGP** を選択した場合は、 ASNを入力します。

前のページで IPv6 機能を有効にした場合は、IPv6 アドレス、ネットマスク、およびゲートウェイも入力する必要があります。

（注） IPv6 情報を提供する場合は、クラスタ ブートストラップ プロセス中に行う必要があります。後で IP アドレス構成を変更するには、クラスタを再展開する必要があります。

クラスタ内のすべてのノードは、 IPv4 のみ、IPv6 のみ、またはデュアル スタック IPv4/IPv6 のいずれかで構成する必要があります。

f) **クラスタ接続** に **BGP** を選択した場合は、 **[BGP ピアの詳細（BGP peer details）]** 領域で、ピアの IPv4 アドレスとASNを入力します。

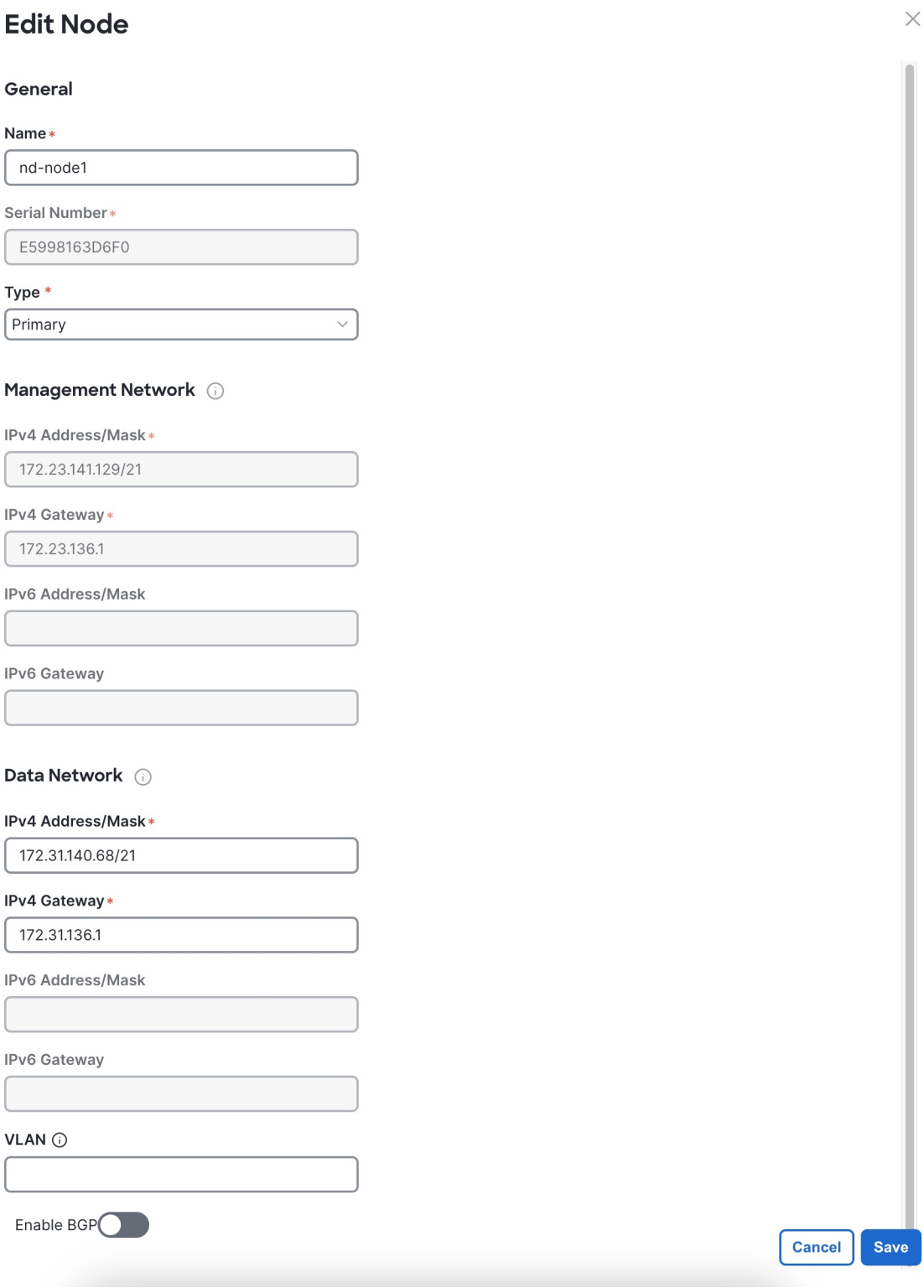
[+ IPv4 BGP ピアの追加（+ Add IPv4 BGP peer）] をクリックして、ピアを追加できます。

前のページで IPv6 機能を有効にした場合は、ピアの IPv6 アドレスと ASN も入力する必要があります。

g) **[Save]**をクリックして、変更内容を保存します。

ステップ 15 [**ノードの詳細（Node Details）**] 画面で、[**ノードの追加（Add Node**）] をクリックして、クラスタに 2 番目のノードを追加します。

単一ノード クラスタを展開する場合は、この手順をスキップします。



a) **[展開の詳細（Deployment Details）]** エリアで、2 番目のノードに **[管理 IP アドレス（Management IP Address）]** および **[パスワード（Password）]** を指定します。

ノードの初期構成手順で、管理ネットワーク情報とパスワードを定義しました。

b) **[検証（Validate）]** をクリックして、ノードへの接続を確認します。

接続が検証されると、ノードの**シリアル番号**と**管理ネットワーク**情報が自動的に入力されます。

c) ノードの **[名前（Name）]** を入力します。

d) **[タイプ（Type）]** ドロップダウンから `[プライマリ（Primary）]` を選択します。

クラスタの最初の 3 つのノードは `[プライマリ（ Primary）]` に設定する必要があります。より大規模なスケールを有効にする必要がある場合は、後の手順でセカンダリ ノードを追加します。

e) **[データ ネットワーク（Data Network）]** エリアで、ノードの**データ ネットワーク**を提供します。

データ ネットワークの IP アドレス、ネットマスク、およびゲートウェイを指定する必要があります。オプションで、ネットワークの VLAN ID を指定することもできます。ほとんどの導入では、[VLAN ID] フィールドを空白のままにできます。

前の画面で IPv6 機能を有効にした場合は、IPv6 アドレス、ネットマスク、およびゲートウェイも入力する必要があります。

（注） IPv6 情報を提供する場合は、クラスタ ブートストラップ プロセス中に行う必要があります。後で IP 構成を変更するには、クラスタを再展開する必要があります。

クラスタ内のすべてのノードは、 IPv4 のみ、IPv6 のみ、またはデュアル スタック IPv4/IPv6 のいずれかで構成する必要があります。

f) （オプション）クラスタが L3 モードで展開されている場合は、データ ネットワークの **[BGP を有効にする（Enable BGP）]** をオンにします。

永続 IP アドレス機能には BGP 設定が必要です。この機能については、BGP 構成と永続的な IP アドレス と『*Cisco Nexus Dashboard ユーザーガイド*』の「永続的な IP アドレス」セクションで詳しく説明されています。

（注） BGP をこの時点で、またはクラスタの展開後に Nexus ダッシュボード GUI で有効にすることができます。

BGP を有効にする際、次の情報も入力する必要があります。

- このノードの **ASN**（BGP 自律システム番号）。

  すべてのノードに同じ ASN を構成することも、ノードごとに異なる ASN を構成することもできます。

- IPv6 専用の場合、このノードの**ルータ ID**。

  ルータ ID は、`1.1.1.1` などの IPv4 アドレスである必要があります。

- ピアの IPv4 または IPv6 アドレスとピアの ASN を含む **BGP ピアの詳細**。

g) **[Save]**をクリックして、変更内容を保存します。

h) クラスタの最後の（3 番目の）プライマリ ノードでこの手順を繰り返します。

ステップ 16 **[ノードの詳細（Node Details）]** ページで、入力した情報を確認してから、**[次へ（Next）]**をクリックします。

ステップ 17 クラスタの**展開モード**を選択します。

a) **[永続的サービスIP/プールの追加]** をクリックして、必要な永続的IPアドレスを指定します。

永続 IP アドレスの詳細については、Nexus Dashboardの 永続 IP アドレスのセクションを参照してください：

b) [次へ（Next）] をクリックして続行します。

ステップ 18 **[サマリー（Summary）]** 画面で設定情報を見直して確認し、**[保存（Save）]** をクリックしてクラスタを構築します。

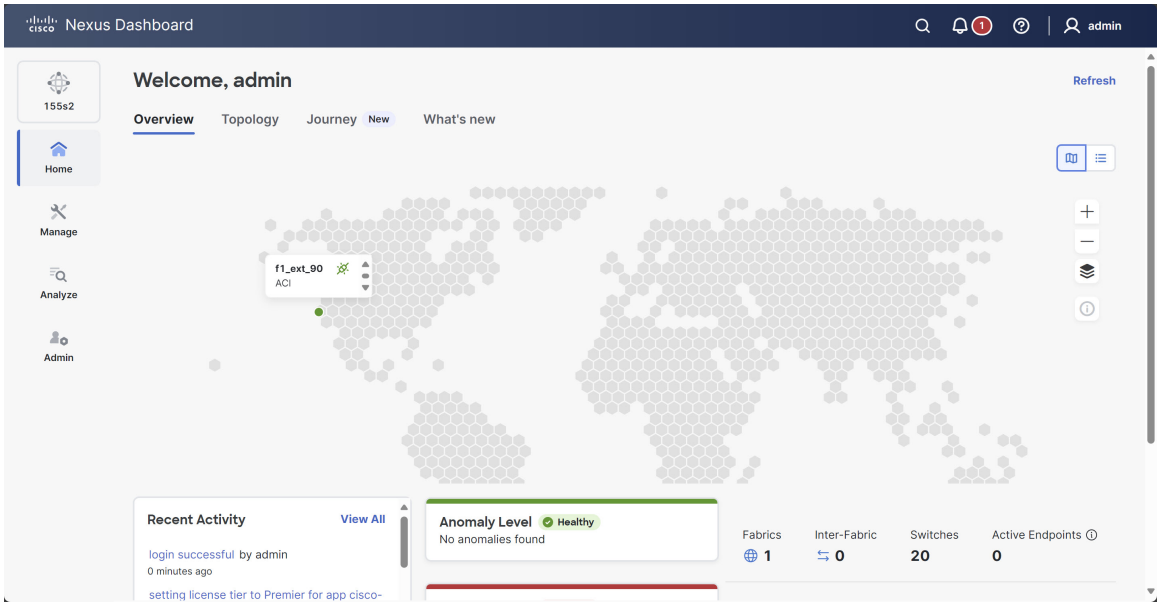
ノードのブート ストラップとクラスタの起動中に、全体的な進捗状況と各ノードの個々の進捗状況がUIに表示されます。ブートストラップの進行状況が表示されない場合は、ブラウザでページを手動で更新し、ステータスを更新してください。

クラスタが形成され、すべてのサービスが開始されるまでに最大30分かかる場合があります。クラスタの設定が完了すると、ページが Nexus ダッシュボード GUI にリロードされます。

ステップ 19 クラスタが健全であることを検証します。

クラスタが使用可能になったら、ノードの管理 IP アドレスのいずれかを参照してアクセスできます。`admin`ユーザーのデフォルトパスワードは、最初のノードに選択した`rescue-user`のパスワードと同じです。この間、UIは上部に「サービスのインストールが進行中です。Nexus Dashboard の設定タスクは現在無効になっています」という意味のバナーを表示します。

すべてのクラスタが展開され、すべてのサービスが開始されたら**[ホーム（Home）]**>**[概要（Overview）]** ページの **異常レベル（Anomaly Level）** でクラスタが正常であることを確認できます。



または、SSH を使用し、`rescue-user` として、ノード展開中に入力したパスワードを使っていずれかのノードにログインし、`acs health` コマンドを実行してクラスタの状態を確認できます。

- クラスタが収束している間、次の出力が表示されることがあります：

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

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

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

- クラスタが稼働している場合は、次の出力が表示されます。

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



（注） 場合によっては、ノードの電源を再投入（電源をオフにしてから再度オン）すると、この段階でスタックが停止することがある可能性があります。

`deploy base system services`

これは、pND（物理 Nexus Dashboard）クラスタの再起動後のノードの `etcd` の問題が原因です。

この問題を解決するには、影響を受けるノードで `acs reboot clean` コマンドを入力します。

ステップ 20 （オプション） Cisco Nexus DashboardクラスターをCisco Intersightに接続、可視性と利点を強化します。詳細な手順については、 「*Cisco Intersight*の操作」 を参照してください。

ステップ 21 Nexus Dashboardを展開した後、設定情報については、このリリースのコレクションページを参照してください。

次のタスク

次のタスクは、ファブリックとファブリック グループを作成することです。 Cisco Nexus Dashboardのコレクション ページ にある、このリリースの 「ファブリックとファブリック グループの作成」の記事を参照してください。

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。