

# NVIDIA DGX BasePOD: Deployment Guide Featuring NVIDIA DGX H200/H100 Systems

**Release latest** 

**NVIDIA Corporation** 

Jun 16, 2025

### Overview

1	Introd	luction	2
2	Hardw	vare Overview	3
3	Netwo 3.1 3.2 3.3 3.4 3.5 3.6	Drking         DGX H200/H100 System Network Ports         DGX BasePOD Network Overview         Managementnet and External networks         oobmanagementnet (ipminet)         externalnet uplink         computenet (ibnet)	<b>4</b> 4 5 5 7 7 7
4	<b>Softw</b> 4.1	<b>are Overview</b> Kubernetes (K8s)	<b>8</b> 8
5	NFS S <sup>1</sup>	torage	9
6	BCM I	ntroduction 1	0
7	Netwo 7.1 7.1.1 7.2 7.2 7.3 7.4 7.4.1 7.4.2 7.4.3	ork Deployment       1         SN4600C - managementnet ethernet switches       1         SN4600C-1 Reference Configuration       1         SN4600C-2 Reference Configuration       1         SN2201 - IPMI Switch for Out-of-Band Management       1         Computenet Configuration       1         QM9700 IB Switches       1         QM-9700-1       1         QM-9700-2       1         InfiniBand/Ethernet Storage Fabric Specific Configurations       1	<b>1</b> 1 2 4 5 5 6 6
8	BCM H 8.1 8.2 8.3 8.3.1 8.3.2 8.3.3 8.3.4	Headnodes Pre-install Preparation1NFS server1Verify DNS & NTP servers1Cluster Nodes Configuration1DGX BIOS Config and Network Interface Boot Order1Control Plane and Workload Management Nodes1Other Branded Appliances2RAID/Storage Configuration2	7 7 7 8 8 3 4
9	<b>BCM H</b> 9.1 9.2 9.3 9.4	Headnodes Installation2Download the Base Command Manager ISO2Primary Headnode preparation2Dell appliances with iDRAC92Other Vendor Appliances2	<b>25</b> 25 25 25 25 25 29

9.5 9.6 9.7 9.8 9.9	Booting the Base Command Manager Graphical InstallerLogin to the HeadnodeUpdate BCMActivate the BCM Cluster LicenseEnable Bonding on the Headnode	29 45 45 45 47
<b>10 Clusto</b> 10.1 10.2 10.2. 10.2. 10.2. 10.2. 10.3 10.4 10.4. 10.4. 10.4. 10.4. 10.4. 10.4. 10.4. 10.4. 10.4. 10.4.	er Bring UpEnable DeviceResolveAnyMACDefine Cluster Networks1 Change network names2 Computenet Config3 Storagenet Config4 IB StorageEnable Out-of-band Management of Cluster NodesDGX Node Bringup1 Software Image Setup for DGX's2 DGX Node category setup3 DGX Interface Definitions4 Defining the storage fabric:5 Ethernet (tcp)6 Infiniband (o2ib)7 Define DGX-01's MAC address8 Test Provisioning of DGX-01	<b>49</b> 49 49 50 50 51 51 51 51 52 52 52 53 53 53
11 BCM		57
13 Testir	ng BCM HA	73
14 Deplo 14.1 14.2 14.3 15 Comp 15.1	y Kubernetes Kubernetes Node Setup Kubernetes Deployment Add a kubernetes user wute network/ IB Interfaces Configuration Validate IB/Compute interfaces	<b>75</b> 75 77 94 <b>95</b> 95
15.2	Configure SR-IOV NetworkNodePolicy CR	98
16 Valida	ate the GPU status/health	108
17 Valida	ate the system topology/NVlink	110
18 Valida	ate the GPU/RDMA access within the container Validate GPU access from container	<b>113</b> 113
18.1 18.2	Validate the RDMA Network from the container	115
18.1 18.2 <b>19 Valida</b>	Validate the RDMA Network from the container	115 117
18.1 18.2 19 Valida 20 Valida	Validate the RDMA Network from the container	115 117 120
18.1 18.2 19 Valida 20 Valida 21 Site S 21.1 21.2	Validate the RDMA Network from the container	115 117 120 125 125 128

22.1 SI	N2201 (oobmanagementnet) Switch Configuration	130
22.2 SI	N4600C-1 (managementnet) Switches Configuration	131
22.2.1	SN4600C-1 Configuration	131
22.2.2	SN4600C-2 Configuration	135
22.3 Q	M9700 (computenet) Switches Configuration	138
22.3.1	QM9700-1	139
22.3.2	QM9700-2	140

Document NumberNAPublication Date2025-2-22

#### Тір

The NVIDIA DGX BasePOD: Deployment Guide Featuring NVIDIA DGX H200 and H100 Systems is also available as a PDF.

# Chapter 1. Introduction

Artificial Intelligence (AI) infrastructure requires significant compute resources to operate the latest state-of-the-art models efficiently, often requiring multiple nodes running in a distributed cluster.

While cloud computing provides an easy on-ramp to train AI models, many enterprises require an onpremises data center for a variety of technical or business reasons.

Building AI infrastructure on-premises can be a complex and confusing process. Careful planning and coordination will make the cluster deployment and the job of the cluster administrators tasked with the day-to-day operations easier.

NVIDIA DGX BasePOD<sup>™</sup> provides the reference design to accelerate deployment and execution of these new AI workloads. By building upon the success of NVIDIA DGX<sup>™</sup> systems, DGX BasePOD is a prescriptive AI infrastructure for enterprises, eliminating the design challenges, lengthy deployment cycle, and management complexity traditionally associated with scaling AI infrastructure.

The DGX BasePOD is built upon NVIDIA DGX H200/H100 systems, which offer unprecedented compute performance with eight NVIDIA H200/H100 Tensor Core GPUs connected with NVIDIA NVLink<sup>®</sup> and NVIDIA NVSwitch<sup>™</sup> technologies for fast inter-GPU communication.

Powered by NVIDIA Base Command<sup>™</sup>, DGX BasePOD provides the essential foundation for AI development optimized for the enterprise.

### Chapter 2. Hardware Overview

DGX BasePOD deployment in this example consists of compute nodes, five control plane servers (two for cluster management and three Kubernetes (K8s) control plane nodes), as well as associated storage and networking infrastructure.

An overview of the hardware is in Table 2.1. Details about the hardware that can be used and how it should be cabled are given in the NVIDIA DGX BasePOD Reference Architecture.

This deployment guide describes the steps necessary for configuring and testing a four-node DGX BasePOD after the physical installation has taken place. Minor adjustments to specific configurations will be needed for DGX BasePOD deployments of different sizes, and to tailor for different customer environments, but the overall procedure described in this document should be largely applicable to any DGX deployments.

Component	Technology
Compute nodes	DGX H200/H100 system
Compute fabric	NVIDIA Quantum QM9700 InfiniBand switches
Management fabric	NVIDIA SN4600C switches
Storage fabric	Option 1: NVIDIA SN4600C switches for Ethernet attached stor- age Option 2: NVIDIA Quantum QM9700 switches for InfiniBand at- tached storage
Out-of-band management fab- ric	NVIDIA SN2201 switches
Control plane and workload management nodes	Minimum Requirements (each server): > 64-bit x86 processor, AMD EPYC 7272 or equivalent > 256 GB memory > 1 TB SSD > Two 100 Gbps network ports

#### Table 2.1: DGX BasePOD components

# Chapter 3. Networking

This section covers the DGX system network ports and an overview of the networks used by DGX BasePOD.

### 3.1. DGX H200/H100 System Network Ports



Figure 3.1 shows the physical layout of the back of the DGX H200/H100 system.

Figure 3.1: Physical layout of the back of the DGX H200/H100 system

Figure 3.2 shows how the DGX H200/H100 network ports are used in this deployment guide.

The following ports are selected for DGX BasePOD networking:

- Eight ports in four OSFP connections are used for the InfiniBandcompute fabric (marked in green).
- Two ports of the dual-port ConnectX-7 cards are configured as abonded Ethernet interface for in-band management and storage ethernet networks. These are the left ports (port 2) from slot 1 and slot 2 (marked in blue).
- Optional: Two ports of the dual-port ConnectX-7 cards when InfinIBand storage is used instead of Ethernet. These are the right ports (port 1) from slot 1 and slot 2 (marked in red).
- > BMC network access is provided through the out-of-band network (marked in purple).



Figure 3.2: DGX H200/H100 network ports

The networking ports and their mapping are described in detail in the Network Ports section of the NVIDIA DGX H200/H100 System User Guide.

### 3.2. DGX BasePOD Network Overview

There are four networks in a DGX BasePOD configuration:

- managementnet (internalnet)—Network used exclusively within the cluster, for storage and inband management.
- externalnet—Network connecting the DGX BasePOD to an external network, such as a corporate or campus network.
- oobmanagementnet (ipminet)—Network for out of band management, connecting BMCs.
- computenet (ibnet)—InfiniBand network connecting all DGX systems' ConnectX-7 Compute Fabric HCAs.

These are shown in Figure 3.3.

### 3.3. Managementnet and External networks

managementnet (internalnet) and externalnet are configured on the SN4600C switches, which are the backbone of the DGX BasePOD Ethernet networking. Each DGX system connects to the SN4600C switches with a bonded interface that consists of two physical interfaces; slot 1 port 2 (storage 1-2) and slot 2 port 2 (storage 2-2) as described in the Network Ports section of the NVIDIA DGX H200/H100 System User Guide.

The K8s (Workload Manager nodes in the topology diagram) nodes and the NFS storage device have a similar bonded interface configuration connected to SN4600C switches. Two SN4600C switches with Multi-chassis Link Aggregation (MLAG) provides the redundancy for DGX systems, K8s nodes, and other devices with bonded interfaces. In this deployment guide, BGP (Border Gateway Protocol) is used for network connectivity between the managementnet (internalnet) and externalnet networks.



Figure 3.3: Network design and topology diagram.

Table 3.1: BGP Protocols

Proto- col	Description
eBGP	Used as required for routing between switches and customer network
iBGP	Configured between the two SN4600C switches using the MLAG peerlink.4094 interface

#### 3.4. oobmanagementnet (ipminet)

On the oobmanagementnet(ipminet) SN2201 switches, all the switch ports connected to the end hosts are configured as access ports. Each BCM headnode has its BMC interface connected to the IPMI switch. Uplinks are connected to SN4600C switches.

#### 3.5. externalnet uplink

All connected subnets are redistributed into BGP. IPMI switches can also be uplinked to a separate management network if required, rather than the SN4600C switches; still IPMI subnet route must be advertised to the in-band network so that BCM can control hosts using the IPMI network.

### 3.6. computenet (ibnet)

For the computenet (ibnet), 4 ports of the DGX OSFP ports are connected to QM9700-1 InfiniBand switch, and 4 ports are connected to QM9700-2 InfiniBand switch. To manage the InfiniBand fabric, at least one subnet manager is required to be enabled on the QM9700 switches.

The networking ports and their mapping are described in the Network Ports section of the NVIDIA DGX H200/H100 System User Guide.

### Chapter 4. Software Overview

Base Command Manager (BCM) is a key software component of DGX BasePOD. BCM is used to provision the OS on all hosts, deploy K8s, and provide monitoring and visibility of the cluster health.

An instance of BCM runs on a pair of head nodes in a High Availability (HA) configuration and is connected to all other nodes in the DGX BasePOD.

DGX systems within a DGX BasePOD have a DGX OS image installed by BCM. Similarly, the K8s control plane (workload manager) nodes are imaged by BCM with an Ubuntu LTS version equivalent to that of the DGX OS and the head nodes themselves.

#### 4.1. Kubernetes (K8s)

K8s is a platform for automating deployment, scaling, and operations of application containers across clusters of hosts. With K8s, it is possible to:

- ▶ Scale applications on the fly.
- Seamlessly update running services.
- > Optimize hardware availability by using only the needed resources.

The cluster manager provides the administrator with the required packages, allows K8s to be set up, and manages and monitors K8s.

# Chapter 5. NFS Storage

An NFS solution is required for a highly available (HA) BCM installation, and the required export path for that is described in this DGX BasePOD Deployment Guide. A DGX BasePOD typically also includes dedicated storage, but the configuration of that is outside the scope of this document. Contact the DGX BasePOD certified storage vendor being used for instructions on configuring the high performance storage portions of a DGX BasePOD.

# Chapter 6. BCM Introduction

This chapter details deploying the NVIDIA Base Command Manager (BCM) on NVIDIA DGX BasePOD<sup>™</sup> configurations.

Physical installation and network switch configuration must be completed before deploying BCM. In addition, information about the intended deployment should be recorded in the Site Survey. A sample Site Survey can be found in the Appendix.

It's essential to get familiar with the DGX BasePOD reference architecture, DGX H200/H100 motherboard connections and its network port designations before you proceed with this deployment guide:

- ► DGX BasePOD reference architecture
- DGX H200/H100 Motherboard connections
- ► DGX H200/H100 Network port designations

The following reference documents can be helpful during the deployment:

- BCM Installation Manual
- BCM Administrator Manual
- DGX OS User Guide
- ▶ Cumulus User Guide (for SN4600C and SN2201 switches)
- InfiniBand NVOS with QM9700 switches
- MLNX-OS with QM9700 switches

# Chapter 7. Network Deployment

It's essential to get familiar with the DGX BasePOD Reference Architecture before proceeding

Note

Before powering on any of the switches, ensure physical serial port connections have been established, then proceed to power on all the switches.

# 7.1. SN4600C – managementnet ethernet switches

The SN4600c managementnet fabric provides connectivity for inband management and provisioning of the nodes. The key configuration requirements are

- MLAG between the two SN4600C switches
- ▶ L3 SVI/VRRP for all the pod ethernet networks
- ▶ Each headnode / K8s node / DGX is dual homed to the SN4600C switches via bond interface
- External connectivity to customer network, using customer specified routing arrangements, like BGP (Border Gateway Protocol) or static or other dynamic routing protocols
- Link to IPMI Network for BCM to access node BMCs, either direct or indirect via customer network.

#### 7.1.1. SN4600C-1 Reference Configuration

```
# Basic management configuration
nv set system hostname 4600C-1
#
# Create SVIs for Internal/Management Network with VRRP as FHRP
nv set bridge domain br_default vlan 102
nv set interface vlan102 type svi
nv set interface vlan102 ip vrr mac-address 00:00:5E:00:01:01
nv set interface vlan102 ip vrr address 10.184.94.1/24
nv set interface vlan102 ip address 10.184.94.2/24
nv set interface vlan102 ip vrr state up
# Repeat the same for other SVI interfaces
```

(continues on next page)

(continued from previous page) # Configure MLAG # Define inter-chassis peerlink etherchannel/bond nv set interface peerlink bond member swp63, swp64 nv set interface peerlink type peerlink # Loopback for BGP/MLAG backup routing nv set interface lo ip address 10.160.254.22 # Configure Peerlink L3 parameters nv set interface peerlink.4094 base-interface peerlink nv set interface peerlink.4094 type sub nv set interface peerlink.4094 vlan 4094 nv set mlag backup 10.160.254.23 nv set mlag enable on nv set mlag mac-address 44:38:39:ff:00:02 nv set mlag peer-ip linklocal # MAG Primary nv set mlag priority 2048 # Example port configuration for head nodes (BCM, Kube) # BCM Head Nodes nv set interface bond1 bond member swp1 nv set interface bond1 description "BCM Headnode 1" nv set interface bond1 bond mlag id 1 nv set interface bond1 bridge domain br\_default access 102 nv set interface bond1 bond mlag enable on nv set interface bond1 bond lacp-bypass on # Repeat for other management/workloads/compute nodes # Uplink to the customer network. # Example configuration with BGP unnumbered nv set router bgp autonomous-system 4200004001 nv set router bgp enable on nv set router bgp router-id 10.160.254.22 nv set vrf default router bgp address-family ipv4-unicast enable on nv set vrf default router bgp address-family ipv4-unicast redistribute  $\rightarrow$  connected enable on nv set vrf default router bgp enable on # Uplinks via swp50 nv set vrf default router bgp neighbor swp50 type unnumbered # Peering to MLAG peer switch nv set vrf default router bgp neighbor peerlink.4094 remote-as internal nv set vrf default router bqp neighbor peerlink.4094 type unnumbered

Refer to the appendix for complete switch configuration.

#### 7.1.2. SN4600C-2 Reference Configuration

Same as SN4600C-1, with the following changes

```
# Basic management configuration
nv set system hostname 4600C-2
```

(continues on next page)

#

(continued from previous page)

```
#
# Create SVIs - Internal/Management Network with VRRP as FHRP
nv set bridge domain br_default vlan 102
nv set interface vlan102 type svi
nv set interface vlan102 ip vrr mac-address 00:00:5E:00:01:01
nv set interface vlan102 ip vrr address 10.184.94.1/24
nv set interface vlan102 ip address 10.184.94.3/24
nv set interface vlan102 ip vrr state up
# Configure MLAG
# Define inter-chassis peerlink etherchannel/bond
# BGP/MLAG backup routing loopback
nv set interface lo ip address 10.160.254.23
#
# Configure Peerlink L3 parameters
nv set mlag backup 10.160.254.22
nv set mlag mac-address 44:38:39:ff:00:02
# MLAG Secondary
nv set mlag priority 4096
# Example port configuration - head nodes (BCM, Kube)
# same as 4600-1
# Uplink to the customer network.
# Same as 4600-1
```

Refer to the appendix for complete switch configuration.

You can verify the MLAG status using the following command

```
root@mgmt-net-leaf-1:mgmt:/home/cumulus# clagctl
The peer is alive
    Our Priority, ID, and Role: 2048 9c:05:91:dd:cc:28 primary
   Peer Priority, ID, and Role: 2048 9c:05:91:f1:73:28 secondary
         Peer Interface and IP: peerlink.4094 fe80::9e05:91ff:fef1:7328
\rightarrow(linklocal)
                    Backup IP: 10.160.254.23 vrf mgmt (inactive)
                    System MAC: 44:38:39:ff:0a:00
CLAG Interfaces
                                  CLAG Id
Our Interface
                 Peer Interface
                                             Conflicts
                                                                   Proto-
→Down Reason
                                             _____
                  _____
.....
         bond1
                                    1
         bond10
                                    10
         bond11 -
                                    11
                                             _
         bond12 -
                                    12
         bond13
                  _
                                    13
                                             _
                                                                   _
         bond14
                                    14
```

For troubleshooting, you can use the consistency check command. Here is an example output from a

working MLAG pair.

<pre>cumulus@mgmt-net-leaf-2 Parameter</pre>	<b>:mgmt:~\$</b> nv show mlag consis LocalValue	stency-checker global PeerValue
anycast-ip	-	-
bridge-priority	32768	32768
→ bridge-stp-mode	rstp	rstp
bridge-stp-state ⊶-	on	on
bridge-type	vlan-aware	vlan-aware
clag-pkg-version	1.6.0-cl5.11.0u2	1.6.0-cl5.11.0u2
clag-protocol-version	1.7.0	1.7.0
peer-ip	fe80::9e05:91ff:fedd:cc28	fe80::9e05:91ff:fedd:cc28
peerlink-bridge-member	Yes	Yes
peerlink-mtu	9216	9216
peerlink-native-vlan	1	1
peerlink-vlans	1, 100->102	1, 100->102
redirect2-enable	yes	yes
⇒ system-mac ⇒-	44:38:39:ff:0a:00	44:38:39:ff:0a:00

### 7.2. SN2201 – IPMI Switch for Out-of-Band Management

All the BMCs are in the same oobmanagementnet subnet, configure all switch ports connected to the BMCs to be under the same VLAN. The oobmanagementnet should be accessible from the managementnet to allow the BCM headnodes to control the BMCs. In this example, the oobmanagementnet is routed via the managementnet SN4600C switches. It is recommended to add an additional uplink to the customer's OOB network.

Example Configuration for the SN2201 switch.

```
nv set system hostname IPMI-SW
#<Basic management configuration>
#
#
# VLAN - BMC ports. Adjust according to the customer
```

(continues on next page)

(continued from previous page)

```
specification
nv set bridge domain br_default vlan 101
# Enable the BMC Ports to the Access VLAN
#
nv set interface swp1-48 bridge domain br_default
nv set bridge domain br_default untagged 1
nv set interface swp1-48
nv set interface swp1-48 link state up
nv set interface swp1-48 description "BMC Ports"
nv set interface swp1-48 bridge domain br_default access 101
# Uplink to customer OOB/PIMI Network
# In this example the uplink is a layer 2 trunk with etherchannel/bond.
# Adjust according to the customer specification
nv set interface swp49-50 link state up
nv set interface bond1 bond member swp49, swp50
nv set interface bond1 bridge domain br_default untagged 1
nv set interface bond1 bridge domain br_default vlan all
```

Refer to the appendix for complete switch configuration.

Reference: Cumulus Network configuration Guide.

You can also use NVIDIA Air to simulate and model the network configuration.

Once the SN2201 switches have been successfully configured, verify that all devices out of band management interfaces are reachable from the network. (i.e. make sure you can access the BMC/iLO/iDRAC).

#### 7.3. Computenet Configuration

Before powering on any of the QM9700 switches in the Compute or Storage switch stacks ensure that serial port connectivity can be established (either via remote serial concentrator or physically interfacing with the serial port of the switch), then proceed to power on all Compute & Storage switches.

#### 7.4. QM9700 IB Switches

We recommend configuring the InfiniBand switches with subnet manager HA enabled.

Example configuration

#### 7.4.1. QM-9700-1

```
ib sm
ib sm virt enable
ib smnode 9700-1 create
ib smnode 9700-1 enable
```

(continues on next page)

(continued from previous page)

ib smnode 9700-1 sm-priority 15
ib ha infiniband-default ip <HA VIP> <mask>

#### 7.4.2. QM-9700-2

ib sm virt enable ib smnode 9700-1 create ib smnode 9700-1 enable ib smnode 9700-1 sm-priority 15

Verify IB SM HA status using the following command

```
QM9700-1[infiniband-default: master] # show ib smnodes
HA state of switch infiniband-default:
IB Subnet HA name: infiniband-default
HA IP address : 10.185.230.247/22
Active HA nodes : 2
HA node local information:
        : 9700-2 (active)
 Name
 SM-HA state: standby
 SM Running : stopped
 SM Enabled : disabled
 SM Priority: 0
 IP
          : 10.185.230.243
HA node local information:
 Name
         : 9700-1 (active) <--- (local node)
 SM-HA state: master
 SM Running : running
 SM Enabled : enabled - master
 SM Priority: 15
 TΡ
         : 10.185.231.43
```

Refer to the Appendix for complete switch configuration.

Reference: Nvidia QM9700 InfiniBand Switch user manual

# 7.4.3. InfiniBand/Ethernet Storage Fabric Specific Configurations

A DGX BasePOD typically also includes dedicated storage, but the configuration is outside the scope of this document. Contact the vendor of the storage solution being used for instructions on configuring the high-performance storage portions of a DGX BasePOD.

# Chapter 8. BCM Headnodes Pre-install Preparation

### 8.1. NFS server

NFS is used for BCM headnode HA. User home directories (/home) and shared data directories (/ cm\_shared, which includes files such as the DGX OS image) must be shared between head nodes and are stored on an NFS filesystem that both headnodes mount.

Because DGX BasePOD does not mandate the nature of the NFS storage, the configuration is outside the scope of this document. This DGX BasePOD deployment uses the NFS export path provided in the BasePOD site survey.

The parameters below are recommended for the NFS server export (for example, as specified in the file /etc/exports). In particular, the exported NFS directory must be mountable read-write, and files must be allowed to be owned by UID 0 (root); these are indicated by the rw and no\_root\_squash directives in the example below.

/var/nfs/general \*(rw,sync,no\_root\_squash,no\_subtree\_check)

### 8.2. Verify DNS & NTP servers

Make sure the DNS and NTP servers are reachable from within the cluster environment.

### 8.3. Cluster Nodes Configuration

As cluster nodes – control plane (BCM headnodes), workload management nodes, DGX's, compute & storage fabric switches are racked and cabled. It is recommended to configure each appliance's BIOS & out of band management interface (sometimes referred to as a BMC, IPMI, etc) ahead of time before the installation of BCM.

Once all cluster nodes have been successfully configured verify that all cluster nodes' out of band management interfaces are reachable from within the working cluster network space. (i.e. make sure you can load access the BMC/iLO/iDRAC).

#### 8.3.1. DGX BIOS Config and Network Interface Boot Order

Refer to the DGX System User Guide for specific steps on changing the boot order to use the 2 primary in-band interfaces to PXE boot first.

#### 8.3.2. Control Plane and Workload Management Nodes

The following is an example for Dell appliances.

Refer to the below references for configuring an appliance with iDRAC9:

•••	idrac-76VCW54, PowerEdge R760, User: root, FPS: 0.4		
Not Secure https://10.150.123.19/res	stgui/vconsole/index.html		A V
	Boot Power Chat Keyboo	ard Screen Capture Refresh Full Screen Virtual Media	Disconnect Viewer Console Controls
	Mult Technologies Boot Manager	Help   About   Exit	
	Boot Manager		
	Boot Manager Main Menu		
	Continue Normal Boot		
	One-shot UEFI Boot Menu		
	Launch System Setup		
	Launch Lifecycle Controller		
	System Utilities		
	This selection will direct the sustant to continue to beating process.		
	This selection will direct the system to continue to booting process.		
	PowerEdge R760 Service Tag : 76VCW54	Finish	
			_

Interrupt the boot cycle to enter the Boot Manager and select "Launch System Setup". Next select "Device Settings".

Vell lechnologies System Setup	Heip   About   Exit
System Setup	
System Setup Main Menu	
System BIOS	
iDRAC Settings	
Device Settings	

Select the Card that needs the mode flipped from Infiniband (IB Mode) to Ethernet (ETH Mode).

System Setun	
Device Settings	
Integrated NIC 1 Port 1: Broadcom NetXtreme Gigabit Ethernet (BCt 6C:92:CF:11:AC:C6 Integrated NIC 1 Port 2: Broadcom NetXtreme Gigabit Ethernet (BCt 6C:92:CF:11:AC:C7 Integrated NIC 1 Port 3: Broadcom NetXtreme Gigabit Ethernet (BCt 6C:92:CF:11:AC:C8 Integrated NIC 1 Port 4: Broadcom NetXtreme Gigabit Ethernet (BCt 6C:92:CF:11:AC:C9 BOSS in SL 12: BOSS-N1 Boot Controller InfiniBand Controller in Slot 2: NVIDIA ConnectX-7 Single Port NDR2t A0:88:C2:34:44:CC InfiniBand Controller in Slot 3: Nvidia ConnectX-7 Single Port VPI ND C4:70:BD:DD:34:6E InfiniBand Controller in Slot 6: Nvidia ConnectX-7 Single Port VPI ND C4:70:BD:DD:33:E2 InfiniBand Controller in Slot 6: Nvidia ConnectX-7 Single Port NDR2t A0:88:C2:34:44:E8 PCIe SSD in Slot 0 in Bay 1: Dell NVMe PCIe SSD Configuration Data	M5720) - (M5720) - (M5720) - (M5720) - (M5720) - (M5720) OSFP Adapter - (M200 OSFP

On this screen change the "Network Link Type" from "Infiniband" to "Ethernet" and select "Finish".

InfiniBand Controller in Slot 6: Nvidia C	onnectX-7 Single Port VPI NDR200 OSFP Adapter - C4:70:BD
Main Configuration Page	
Firmware Image Properties	
NIC Configuration	
iSCSI Configuration	
Device Level Configuration	
Blink LEDs	
Device Name	Nvidia ConnectX-7 Single Port VPI NDR200 OSFP Adapter
Chip Type	ConnectX-7
PCI Device ID	
PCI Address	
Link Status	Disconnected
Network Link Type	

On seeing the following confirmation message, the card is now in Ethernet mode. Click OK.

		•
Main Configuration Page		
Firmware Image Properties		
NIC Configuration		
iSCSI Configuration	Success	
Device Level Configuration	Saving Changes	
Blink LEDs	The settings were saved	
Device Name	successfully.	t VPI NDR200 OSFP Adapter
Chip Type		
PCI Device ID		
PCI Address	OK	
Link Status	Disconnected	

After confirming the CX card ports are in the correct mode we can proceed to configure the boot order. Return to the "System Setup" screen and select "System BIOS".

Vot Secure     Name     Call     Repair & Repair	•••	idrac-76VCW54, PowerEdge R760, User: root, FPS: 4.2		
Notice     Notice <th>S Not Secure https://10.150.123.19</th> <th>restgui/vconsole/index.html</th> <th></th> <th><b>A</b></th>	S Not Secure https://10.150.123.19	restgui/vconsole/index.html		<b>A</b>
D&LLTechnologies       System Setup         System Setup       Main Menu         System BIOS		Boot Power Chat Keyboa	ard Screen Capture Refresh Full Screen Virtual Media Disconnect Viewer Console C	entrols
System Setup         System Setup Main Menu         System BIOS         IDRAC Settings         Device Settings		DelLTechnologies System Setup	Help   About   Exit	
System Setup Main Menu System BIOS IDRAC Settings Device Settings		System Setup		
System BIOS IDRAC Settings Device Settings		System Setup Main Menu		
DEvice Settings		System BIOS		
		IDRAC Settings		
Select to configure system BIOS settings.		Select to configure system BIOS settings.		
		PowerEdna R760		
Service Tag: 76VCW54		Service Tag : 76VCW54	Finish	

Select "Network Settings".

drac-76VCW54, PowerEdge R760, User: root, FPS: 0.2     Not Secure https://10.150.123.19/restgu/yconsole/index.html		
Room Chai	Keyboard Screen Capture Refresh Full Screen Virtual Media	Disconnect Viewer Console Controls
D&LLTechnologies, System Setup	Help   About   Exit	
System BIOS		
System BIOS Settings		
System Information	i i	
Memory Settings Processor Settings		
SATA Settings		
NVMe Settings		
Network Settings	- I	
Integrated Devices	_	
Serial Communication		
	•	
This field controls the system network settings.		
PowerEdge R760	Default	
Service Tag: 76VCW54	Social Inight	

Enable a minimum of 2 PXE Devices, then define each PXE Device Setting such that the In-Band network port is selected as one of the 2 PXE boot interfaces.

idrac-76VCW54 123.19/restgui/vconsole/index.html	, PowerEdge R760,	User: root, FPS: 0.2		I 🔮 🖌
		Boot Power Chat Keybox	ard Screen Capture Refresh Full Screen Virtual	Media Disconnect Viewer Console Contr
DelLTechnologies System Setup			Help   About   Exit	
Network Settings				
System BIOS Settings • Network Settings				
UEFI PXE Settings Number of PXE Devices PXE Device1 PXE Device2 PXE Device2 PXE Device4 PXE Device4 PXE Device4 Settings PXE Device3 Settings PXE Device4 Setting PXE Device	4 © Enabled O Enabled O Enabled O Enabled	O Disabled © Disabled © Disabled © Disabled		
This field controls the configuration for this PXE dev	rice.			•
PowerEdge R760 Service Tag : 76VCW54			Back	

	idrac-76VCW5	1, PowerEdge R760, User: root, FPS: 1	
Not Secure https://10.150.123.19/restg	gui/vconsole/index.html		👽 🔺
		Boot Power Chat Keyboard Screen Capture Refresh Full S	creen Virtual Media Disconnect Viewer Console Controls
	LTechnologies System Setup	Help I Ab	out L Exit
Ν	Network Settings		
S	System BIOS Settings • Network Settings • PXE	E Device1 Settings	
	Interface	NIC in Slot 3 Port 1 Partition 1	
	Protocol	Pv4 OPv6	-
	VI AN		
	VLANID		_
	VENID		
	VLAN Priority	U	_
	NIC interface used for this PXE device.		
	•		
	PowerEdge R/60		Back
	Service Tag: 76VCVV54		

With both interfaces defined as PXE Devices click "Back" to return to the "System BIOS" screen with a "Warning - Save Changes" prompt. Select "Yes" to confirm saving the changes. Then click "Finish" to return to the System Setup Main Menu.



Select "Finish", and on the "Warning - Confirm Exit" prompt select "Yes" to confirm the appliance reboot.

•••	idrac-76VCW54, PowerEdge R760, User: root, FPS: 4
Not Secure https://10.150.123.19/restgui/vconsole/index.html	I 🔽
	Boot Power Chat Keyboard Screen Capture Refresh Full Screen Virtual Media Disconnect Viewer Console Co
Del Technologies System Satur	Help I About I Evit
System Setup	Hop Phone Pene
System Setup	
System Setup Main Menu	
· · ·	
System BIOS	
iDRAC Settings	
	Warping
Device Settings	vval i mg
	Confirm Exit
	Are you sure you want to exit
	and reboot?
	Tes
Select to configure system BIOS setti	ngs.
PowerEdge P760	
Service Tag : 76VCW54	Finish
2017/00 Hag. 10701104	

#### 8.3.3. Other Branded Appliances

On the ConnectX card that facilitates the In-Band network connections for the management nodes, ensure to set the port mode to Ethernet (not InfiniBand)

If the Connect-X card mode is not correctly set to Ethernet mode, the appliance will fail to communicate on the In-Band network.

The card's port mode can be modified by temporarily booting the appliance to a linux environment to install the NVIDIA Firmware Tools application which can flip the port mode using the below EXAMPLE command:

mlxconfig -d /dev/mst/mt4119\_pciconf0 set LINK\_TYPE\_P1=2 LINK\_TYPE\_P2=2

#### Note

The specified command needs to be used with the correct device id, do not run the below example *as is* on a production system. Refer to NVIDIA Firmware Tools for usage details.

#### 8.3.4. RAID/Storage Configuration

If available, configure the hardware RAID controller and disks to minimum RAID level 1 using the appliance's BMC or BIOS. The procedure varies depending on the appliance vendor and RAID controller. Refer to the specific vendor documentation for the configuration procedure.

# Chapter 9. BCM Headnodes Installation

### 9.1. Download the Base Command Manager ISO

Download the BCM ISO image from the BCM website.

Be sure to select the following options for the download:

Version: Base Command Manager 10 Architecture: x86\_64/amd64 Linux Base Distribution: Ubuntu 22.04 Hardware Vendor: NVIDIA DGX Additional Features: Include MOFED Packages & Include NVIDIA DGX OS software images for DGX H100 & DGX A100

Validate the downloaded file by verifying the MD5 checksum

\$md5sum bcm-10.0-ubuntu2204-dgx-os-6.3.iso

66ecc05da5b0ed89cf365a168af86ec9 bcm-10.0-ubuntu2204-dgx-os-6.3.iso

Burn the BCM ISO to a DVD or to a bootable USB device. The ISO can also be mounted as virtual media and installed using appliance BMC Virtual Console.

#### 9.2. Primary Headnode preparation

Before starting the installation process it's important to ensure the Headnode's storage media are all in a freshly wiped state, and perform a BIOS configuration default. This resets all boot devices/order priorities.

The following is an example of Dell appliances.

#### 9.3. Dell appliances with iDRAC9

Access the appliance's iDRAC9 web portal, click the boot button and select Lifecycle Controller. Power cycle the appliance and boot into the Lifecycle Controller. Select "OS Deployment" on the left side of the screen and then click "Deploy OS".



Select "Go Directly to OS Deployment" then click "Next".

DelLTechnologies Lifecycle Controlle	r Hei	p Abou	t   Exit
Select OS Deployment Path	OS Deployment: Deploy OS		
Select an Operating System	Step 1 of 5: Select Deployment path		
Select Installation Mode	Select OS Deployment Path		
Select OS Media	<ul> <li>Configure RAID First</li> <li>Go Directly to OS Deployment</li> </ul>		
Reboot the System	RAID can be configured prior to OS Deployment; the current wizard will be restarted after the RAID wizard is completed		
PowerEdge R7525 Service Tag : BMGY9R3	Cancel Ba	ack	Next

Ensure that Boot Mode is set to UEFI, Secure Boot is Disabled, Secure Boot Policy is set to Standard, and lastly that "Any Other Operating System" is set for the Available Operating System. Click "Next" to proceed.



Next select the option for "Manual Install" and click "Next".



Proceed to choose the appropriate Media/Virtual Media containing the BCM10 Installation ISO and then select "Next".

D&LLTechnologies Lifecycle Controller	,		Help   At	bout   Exit
Select OS Deployment Path 🗸	OS Deployment: Deploy OS			
Select an Operating System 🗸	Step 4 of 5: Select OS Media			
Select Installation Mode 🗸	Select OS media for the following OS, then click Any Other Operating System	Next.		
Select OS Media	Select Media	USB0 (Front USB 1)		
PowerEdge R760 Service Tag : G7VCW54		Cancel	Back	Next

Confirm the selected options, if any adjustments need to be made click the "Back" button to return to the appropriate screen to make a correction. If all options have been confirmed as correct select "Finish".

DeutTechnologies Lifecycle Controller	Help   About   Exit
Select OS Deployment Path 🗸	OS Deployment: Deploy OS
Select an Operating System 🗸	Step 5 of 5: Reboot the System
Select Installation Mode 🗸	1. Boot Mode: UEFI
Select OS Media	Secure boot disabled Secure Boot Policy : Standard
Reboot the System	Secure Boot Mode: Deployed Mode
	2. Keep the following media in the Device:USB0 (Front USB 1)
	Any Other Operating System
	3. Selected Installation Mode:
	Manual Install
	4. Click Finish to continue with the installation.
	The system reboots to start the operating system installation. After reboot, you may be prompted to "Press Any Key" to boot to the operating system media and start installation. If a key is not pressed, the system bypasses the installation.
PowerEdge R760 Service Tag : G7VCW54	Cancel Back Finish

The Dell appliance will proceed to boot as normal.

### 9.4. Other Vendor Appliances

Attach the BCM10 installation media to the designated Headnode1 appliance.

Power on Headnode1 and proceed to boot from the BCM installation media. The specific procedure may vary by vendor, follow the respective vendor's user manual for details.

### 9.5. Booting the Base Command Manager Graphical Installer

After booting from the BCM ISO, at the grub menu, highlight **Start Base Command Manager Graphical Installer** using the arrow keys then press enter/return to select the option.

This step has an automated countdown timer, to interrupt the timer simply use the up or down arrow key.



If you see the following after selecting "Start Base Command Manager Graphical Installer" this is expected and patience is needed while the installer loads up.



From here we can proceed to use the mouse to click Start installation on the Installer splash screen.



Accept the terms of the NVIDIA EULA by checking I agree and then select Next.

🔍 NVIDIA.	Base Command Manager installer		JNTU2204)
NVIDIA EULA	NVIDIA		
Kernel modules			A
Hardware info	NYJUR AL PRODUCT ANRENENT		
Installation source	Throwing while - Please kew and acked before USING WIDIA AI PRODUCTS.		
Cluster settings	Inis AI Product Agreement is entered into between the entity you represent or you individually if you do not designate an entity ("Customer") and NVIDIA Corporation ("NVIDIA"). This AI Product Agreement consists of the terms and		
Workload manager	conditions below and all documents attached to or referenced in this AI Product Agreement (together, the "Agreement"). The AI Product catalogs include products		
Network topology	that can be used without payment and paid products and services. Key terms are defined in Section 17.		
Head node	By using or registering to use AI Products, Customer is affirming that Customer		
Compute nodes	has read the Agreement and agrees to its terms. If Customer does not have the required authority to enter into the Agreement or if Customer does not accept		
BMC configuration	all the terms and conditions below, do not use (or register to use) AI Products.		
Networks	1. AI PRODUCTS OFFERINGS.		
Head node interfaces	1.1 Grant.		
Image: Compute nodes interfaces	Subject to the terms of the Agreement, Customer's Order Agreement and Subscription or Perpetual license parameters, and payment of applicable fees,		
Disk layout	NVIDIA grants Customer a non-exclusive, non-transferable, non-sublicensable (except as expressly provided in the Agreement) license to do the following for		
Disk layout settings	the duration of the license:		
Additional software	1.1.1 install and use copies of AI Products,		
Summary	<ol> <li>1.1.2 create Derivative Samples and Derivative Models to develop and test services and applications,</li> </ol>		
Deployment	<ol> <li>1.1.3 configure the AI Product using the configuration files provided (as applicable),</li> </ol>		
	📄 I agree		
	Continue remotely Load config Show	r config Bac	:k Next

#### Accept the terms of the **Ubuntu Server EULA** by checking **I agree** and then select **Next**.

	Base Command Manager installer v10.0 (UBUNTU2204)
NVIDIA EULA	Ubuntu Server 22.04
Kernel modules	
Hardware info	Ubuntu is a collection of thousands of computer programs and documents created by a range of individuals, teams and companies.
Installation source	Each of these programs may come under a different licence. This licence policy describes the process that
Cluster settings	we follow in determining which software will be included by default in the Ubuntu operating system.
Workload manager	Copyright licensing and trademarks are two different areas of law, and we consider them separately in Ubuntu. The following policy applies only to copyright licences. We evaluate trademarks on a case-by-case
Network topology	basis.
Head node	Categories of software in Ubuntu The thousands of software packages available for Ubuntu are organised into four key groups or components:
Compute nodes	main, restricted, universe and multiverse. Software is published in one of these components based on whether or not it meets our free software philosophy, and the level of support we can provide for it. In addition,
BMC configuration	software may be published for Ubuntu as a universal Linux snap package, in which case licenses are determined by the snap publisher and documented in the snap store.
Networks	This policy only addresses the software that you will find in main and restricted, which contain software
Head node interfaces	that is fully supported by the Ubuntu team and must comply with this policy.
Compute nodes interfaces	Ubuntu 'main' component licence policy All application software included in the Ubuntu main component:
Disk layout	Must include source code. The main component has a strict and non-negotiable requirement that application
Disk layout settings	Must allow modification and distribution of modified copies under the same licence. Just having the source
Additional software	software, the Ubuntu community cannot support software, fix bugs, translate it, or improve it.
Summary	Ubuntu 'main' and 'restricted' component licence policy
Deployment	Nust allow redistribution. Your right to sell or give away the software alone, or as part of an aggregate software distribution. is important because:
	I agree
	Continue remotely Load config Show config Back Next

Unless instructed otherwise, select **Next** without modifying the **Kernel modules** to be loaded at boot time.

	Base Command Manag	ger installer	v10.0 (UBU	NTU2204)	
NVIDIA EULA	Kernel modules				
Kernel modules	in order to be able to use all the hardware, it is important that the correct set of kernel modules are loaded at boot-				
Hardware info	time. The hardware in this machine has been pro- circumstances it is not necessary to modify the	obed and the kernel modules list kernel modules selection, but if y	ed below were loaded. Un ou wish to do so, you may	der most add or	
Installation source	remove kernel modules here.				
Cluster settings				€ ⊕	
Workload manager	Name	Parameters	Path		
Network topology	acpi_ipmi	-	-	$\oslash$	
needer to be	acpi_power_meter		-	$\oslash$	
Head node	aesni_intel		-	$\oslash$	
Compute nodes	ahci		-	$\otimes$	
BMC configuration	amd64_edac	-	-	$\oslash$	
Networks	async_memcpy	-	-	$\oslash$	
Head node interfaces	async_pq	-	-	$\oslash$	
Compute nodes interfaces	async_raid6_recov	-	-	$\oslash$	
Disk layout	async_tx	-	-	$\oslash$	
Disk layout settings	async_xor	-	-	$\oslash$	
Additional software	autofs4	-	-	$\oslash$	
Summary	сср		-	$\oslash$	
Deployment	cec	-	-	$\oslash$	
	crc32_pclmul	-	-	$\oslash$	
	crct10dif pclmul			$\bigcirc$	
		Continue remotely	Show config Bac	k Next	

Verify the Primary Headnode Hardware info is correct and then select Next.

The key components that need to be validated are as follows:

- Network interfaces Verify that a minimum of 2 Ethernet mode interfaces are detected, this is typically indicated via the device naming convention.
- Devices starting with e = ethernet, i = InfiniBand
- Storage devices It is advisable to install the operating system on a redundant storage device, such as a hardware or software RAID array.
|                          |   | Base Co             | mmand Manager instal | ler v10.0 (UBUNTU2204)                  |
|--------------------------|---|---------------------|----------------------|---|
| NVIDIA EULA              | Ð | Keyboard            |                      | Mouse                                   |
| Kernel modules           |   |                     | /dev/input/mice      | Raritan D2CIM-VUSB                      |
| Hardwara info            |   |                     | /dev/input/mice      | Raritan D2CIM-VUSB                      |
| naruware into            | Ð | Mouse               | /dev/input/mice      | Dell DRAC 5 Virtual Keyboard and        |
| Installation source      |   |                     |                      | House                                   |
| Cluster settings         |   |                     | enol                 | Network interface [tg3]                 |
| Workload manager         |   | Network Interfaces  | eno2                 | Network interface [tg3]                 |
| Network topology         |   |                     | eno34                | Network interface [tg3]                 |
| Head node                |   |                     | eno35                | Network interface [tg3]                 |
|                          | ÷ |                     | eno36                | Network interface [tg3]                 |
| Compute nodes            |   |                     | enp129s0np0          | Network interface [mlx5_core]           |
| BMC configuration        |   |                     | enp226s0np0          | Network interface [mlx5_core]           |
| Networks                 |   |                     | enp37s0np0           | Network interface [mlx5_core]           |
| Head node interfaces     |   |                     | enp65s0np0           | Network interface [mlx5_core]           |
| Compute nodes interfaces |   |                     | /dev/sda             | 959GB PERC H345 Front                   |
| Disk lavout              |   | Storage             | /dev/nvme0n1         | Dell Ent NVMe AGN MU U.2 3.2TB          |
|                          | Č |                     | /dev/nvmeln1         | Dell Ent NVMe AGN MU U.2 3.2TB          |
| Disk layout settings     |   |                     | /dev/nvme2n1         | 🄊 Dell Ent NVMe AGN MU U.2 3.2TB        |
| Additional software      |   |                     | -                    |   |
| Summary                  |   |                     |                      |   |
| Deployment               | Ð | Storage Controllers | -                    |   |
|                          |   |                     | -                    |   |
|                          |   |                     | •                    |   |
|                          |   |                     |                      | Continue remotely Show config Back Next |

On the **Installation source** screen, choose the appropriate source for the installation media and then select **Next**.



On the General cluster settings screen, enter the required information according to the Site Survey

and then select **Next**.

	Base Command Manager insta	ller	v10.0	(UBUN1	U220	4)
NVIDIA EULA	General cluster settings					Î
Kernel modules	Cluster name:					
Hardware info	ExampleCluster					
Installation source	Organization name:					
Cluster settings						
Workload manager	Example Org					
Network topology	Administrator email:					
Head node	┣dmin@example.org					
Compute nodes	Send email to the administrator on first boot					
BMC configuration	Time zone:					
Networks	(GMT-08:00) America/Los_Angeles				~	
Head node interfaces	Time servers:					
Compute nodes interfaces	× 0.pool.ntp.org × 1.pool.ntp.org × 2.pool.ntp.org				× •	
Disk layout						
Disk layout settings	Nameservers:					
Additional software	x 8.8.8.8 Leave this field empty if you intend to use DHCP for external network				× v	
Summary	Search domains:					
Deployment	x example.org				× Ŧ	
	Leave this field empty if you intend to use DHCP for external network					
		Continue remotely	Show config	Back	Next	

On the Workload manager screen, choose None and then select Next.

	Base Command		v10.0 (UBUNTU2204)
NVIDIA EULA	HPC workload manage	er	
Kernel modules	A workload management system is high	ly recommended to run compute jobs. Ple	ase choose the one that should be
Hardware info	configured or choose 'None' to prevent	configuration.	
Installation source	Please select workload manager	:	
Cluster settings			
Workload manager	PBS Works <sup></sup>	PBS Works <sup>**</sup>	slurm
Network topology	OpenPBS	PBS Pro	uarklaad manager Slurm
Head node			
Compute nodes	IBM Spectrum I SE		
BMC configuration		Gunnendine	None
Networks	IBM Spectrum LSF	Univa Grid Engine	
Head node interfaces	No wor	kload manager will be configured on first l	boot.
Compute nodes interfaces		, ,	
Disk layout			
Disk layout settings			
Additional software			
Summary			
Deployment			
		Continue remotely	Show config Back Next

On the **Network topology** screen, choose the network type for the data center environment and then select **Next**.

#### Note

In this deployment example we are using a type 2 network. More information on the different types of networks can be found in the BCM Installation Manual



#### Update head node settings

On the **Head node** screen enter the Hostname and Administrator password as defined on the **Site Survey**.

Choose Other for Hardware manufacturer, and then select Next.

	Base Command Manager installer v10.0 (UBUNT	U2204)
	Head node settings	
Kernel modules	Hostname:	
Hardware info	bcm10-headnode1	
Installation source	A deviation to a second s	
Cluster settings	Administrator password:	
Workload manager	•••••	<i>\$</i>
Network topology	Confirm administrator password:	
Head node		I)
Compute nodes	Hardware manufacturer:	
BMC configuration	Other	~
Networks		
Head node interfaces		
Compute nodes interfaces		
Disk layout		
Disk layout settings		
Additional software		
Summary		
Deployment		
	Continue remotely Show config Back	Next

Update compute node settings

On the **Compute nodes** screen, update node digits from 3 to 2 and select **Next**.

This will populate what will be referred to as a "template" node with the name of node01, which will be modified to create the appropriate DGX and workload management node identities.

	Base Command Manager installer	v10.0 (UBUNTU2204)
NVIDIA EULA	Compute nodes settings	
Kernel modules	Number of racks:	
Hardware info	1	
Installation source	Number of nodes	
Cluster settings	Number of nodes:	
Workload manager	1	
Network topology	Node start number:	
Head node	1	
Compute nodes	Node base name:	
BMC configuration	node	
Networks	Node digite:	
Head node interfaces		
Compute nodes interfaces	2	
Disk layout	Hardware manufacturer:	
Disk layout settings	NVIDIA DGX	-
Additional software		
Summary		
Deployment		
	Continue remotely Show	config Back Next

On the BMC Configuration screen, choose Yes for both Head Node and Compute Nodes and populate the following values for both the Head & Compute nodes

- 1. BMC network type select IPMI from the dropdown menu.
- 2. Choose No for "Use DHCP to obtain BMC IP addresses?"
- **3**. For the Head node, select No for "Automatically configure BMC when node boots?". Select Yes for Compute nodes.
- 4. Lastly select "New dedicated network" from the dropdown list for "To which Ethernet segment is BMC connected?".

	Base Command Manager install	ler v10.0 (UBUNTU2204)
NVIDIA EULA	BMC configuration	
Kernel modules		
Hardware info	Head Node	Compute Nodes
Installation source	Will head node have IPMI/iDRAC/iLO/CIMC compatible BMCs?	Will compute nodes have IPMI/iDRAC/iLO/CIMC compatible BMCs?
Cluster settings	Yes      No     No	Yes ONO
Workload manager	BMC network type:	BMC network type:
Network topology	IPMI -	IPMI -
Head node	Use DHCP to obtain BMC IP addresses?	Use DHCP to obtain BMC IP addresses?
Compute nodes	<ul> <li>See Sheri to obtain Side in dudressesh</li> <li>See ● No</li> </ul>	<ul> <li>Yes ● No</li> </ul>
BMC configuration	Automatically configure BMC when node boots?	Automatically configure BMC when node boots?
Networks	⊙ Yes ● No	Yes ONO
Head node interfaces	To which Ethernet segment is BMC connected?	To which Ethernet segment is BMC connected?
Compute nodes interfaces	New dedicated network	New dedicated network
Disk layout		
Disk layout settings		•
Additional software		2
Summary		
Deployment		
		Continue remotely Show config Back Next

Since a Type 2 network was specified and "New dedicated network" was selected in the prior step for IPMI, there will be a total of 2 networks defined: managementnet (internalnet) & oobmanagementnet (ipminet). Proceed to populate both network definitions with the defined values in the Site Survey.

	Base Command Manager installer	v10.0 (UBUNTU2204)
NVIDIA EULA	Networks	1
Kernel modules	The following IP networks have been pre-configured. Using the controls below, the ne	etwork settings may be altered.
Hardware info	internalnet inminet	Đ
Installation source	Name:	
Cluster settings	internalnet	
Workload manager	Base IP address:	
Network topology	10.184.94.0	
Head node	Netmask:	
Compute nodes	255.255.0(/24)	× •
BMC configuration		
Networks	Dynamic range start:	
Head node interfaces	10.184.94.160	
Compute nodes interfaces	Dynamic range end:	
Disk layout	10.184.94.223	
Disk layout settings	Domain name:	
Additional software	eth.cluster	
Summary	Gateway:	
Deployment	10.184.94.1	
	By default the head node will be used as the default gateway.	
	MTU: Continue remotely	Show config Back Next

	Base Command Manager installer	10.0 (UBUNTU2204)
NVIDIA EULA	Networks	
Kernel modules	The following IP networks have been pre-configured. Using the controls below, the network se	ttings may be altered.
Hardware info	internalnet ipminet	$\oplus$
Installation source	Name:	
Cluster settings	ipminet	
ेWorkload manager	Base IP address:	
Network topology	10.160.6.0	
Head node	Network	
Compute nodes		
BMC configuration	255.255.255.0(/24)	× *
Networks	Domain name:	
Head node interfaces	ipmi.cluster	
Compute nodes interfaces	Gateway:	
Disk layout	10.160.6.1	
Disk layout settings	By default the head node will be used as the default gateway.	
Additional software	MTU:	
Summary	1500	
Deployment	Management network	
Deployment	Bootable network	
	Continue remotely Show co	nfig Back Next

On the Head node interfaces screen, ensure that the correct interface is configured (refer to site survey) with the head node's target managementnet (internalnet) IP.

We will also need to remove the IPMI alias interface that was defined by default, in our example this interface is ens18:ipmi.

In other scenarios you will be looking for an interface name that ends with ":ipmi".

		Base Comi	mand Manager ir	nstaller	١	10.0 (UBUNT	U2204)
NVIDIA EULA	Head node	netwo	rk interface	s			
Kernel modules	incut incut						÷
Hardware info	Interface		Network		IP address		
Installation source	enol	× -	internalnet	× -	10.184.94.254	× -	Ċ
Cluster settings	ipmi0	× •	ipminet	× -	10.160.6.254	× •	Ċ
Workload manager			. Iprimiter				÷
Network topology	eno1:ipmi	× •	ipminet	× •	10.160.6.253	× •	•
Head node							
Compute nodes							
BMC configuration							
Networks							
Head node interfaces							
Compute nodes interfaces							
Disk layout				•			
Disk layout settings				<i>N</i>			
Additional software							
Summary							
Deployment							
				Continu	ie remotely Show co	nfig Back	Next

After deleting the alias interface ens18:ipmi.

			Command Mar				(UBUN	FU2204)
NVIDIA EULA	Head no	de net	work inter	faces				
Kernel modules								÷
Hardware info	Interface		Network		IP address			
Installation source	enol	× -	internalnet	× -	10.184.94.254		× •	亡
Cluster settings	ipmi0	× -	ipminet	× -	10.160.6.254		× -	Ċ
Workload manager								
Network topology								
Head node								
Compute nodes								
BMC configuration								
Networks				•				
Head node interfaces				~				
Compute nodes interfaces								
Disk layout								
Disk layout settings								
Additional software								
Summary								
Deployment								
				Cor	ntinue remotely	Show config	Back	Next

On the Compute node interfaces screen, update the IP offset for both listed items, and then select Next. Here we are setting the IP offset for any nodes that get provisioned into the cluster later in the deployment process. The offset effectively blocks off the first n number of IP's from the specified

network.

In our example we have it set to 0.0.0.4 for managementnet (internalnet) (10.141.225.0/16) which means the first IP available out of our managementnet (internalnet) ip range will be 10.141.225.4 instead of the expected 10.141.225.1 address. The offset allows the reserved IP addresses to be used for gateways, VRRP etc within the network subnet.

		Base Comr	mand Manager ins	taller		v10.0 (UBUN	TU2204)
NVIDIA EULA	Compute n	odes ne	etwork inter	faces			
Kernel modules							÷
Hardware info	Interface		Network		IP offset		
Installation source	BOOTIF	× -	internalnet	× -	0.0.0.4	× <del>-</del>	Ċ
Cluster settings	ipmi0	× -	ipminet	× -	0.0.0.4	× -	Ċ
Workload manager							
Network topology	IP addresses for eac addresses. The star	h of the inter ing address i	faces on all nodes will n the range is determi	be assigned autom ned by adding the s	atically from a co specified offset to	secutive range the base addres	of IP s of the
Head node	network. For examp to be assigned 10.1	le, a selected 41.0.1. and th	network base address e second node 10.141	s 10.141.0.0 with ar	IP offset of 0.0.0	0.0 will cause the	first node
Compute nodes							
BMC configuration							
Networks							
Head node interfaces							
Compute nodes interfaces							
Disk layout							
Disk layout settings							
Additional software							
Summary							
Deployment							
				Continue rem	otely Show o	config Back	Next

On the Disk layout screen, select the target install location (in this case /dev/sda) and then select Next.

When selecting the target installation location be sure to use a storage device with a minimum of RAID1 redundancy.

	Base Command Manager installer	v10.0 (UBUNTU220	)4)
NVIDIA EULA	Disk lavout		
Kernel modules	Installation drives		
Hardware info	Please select a drive to use for this head node installation. If a software RAID layout is inter	nded, then multiple drives	i
Installation source	should be selected.		
Cluster settings	Select install drive(s):		
Workload manager	/dev/sua (9590B FERC H345 Front) /dev/nvme0n1 (Dell Ent NVMe AGN MU U.2 3.2TB)		
Network topology	/dev/nvme1n1 (Dell Ent NVMe AGN MU U.2 3.2TB) /dev/nvme2n1 (Dell Ent NVMe AGN MU U.2 3.2TB)		
Head node			
Compute nodes			
BMC configuration			
Networks			
Head node interfaces			
Compute nodes interfaces			
Disk layout settings	λ.		
Additional software			
Summary			
Deployment			
	Continue remotely Show	config Back Nex	¢t

On the Disk layout settings screen, if hardware RAID storage was selected in the prior step accept defaults and then select Next.

	Base Command Manager installer	v10.0 (UBUNTU2204
NVIDIA EULA	Disk lavout settings	
Kernel modules	Disk layouts	
Hardware info	Please select a layout from the predefined list of node layouts. To view and edit the disk lay	out, click the edit button
Installation source	below.	
Cluster settings	Head node disk layout:	
Workload manager	One big partition	* 🗹 ⊕
Network topology	Compute nodes disk layout:	
Head node	One big partition	- Ľ ⊕
Compute nodes		
BMC configuration		
Networks	k	
Head node interfaces		
Compute nodes interfaces		
Disk layout		
Disk layout settings		
Additional software		
Summary		
Deployment		
	Continue remotely Show	config Back Next

If software RAID is the storage option, ensure that "One big partition RAID1" or "One big partition RAID5" is selected for Head node disk layout.

🚳 NVIDIA.	Base Command Manager installer	v10.0 (UBUNTU2204)
NVIDIA EULA	Disk lavout settings	
Kernel modules	Disk layouts	
Hardware info	Please select a layout from the predefined list of node layouts. To view and edit the disk lay	yout, click the edit button
Installation source	below.	
Cluster settings	Head node disk layout:	
Workload manager	Standard Layout RAID 5	- ピ ⊙
Network topology	Compute nodes disk layout:	
Head node	bne big partition	- ピ ⊕
Compute nodes		
BMC configuration		
Networks		
Head node interfaces		
Compute nodes interfaces		
Disk layout		
Disk layout settings		
Additional software		
Summary		
Deployment		
	Continue remotely Show	config Back Next

In the Additional software screen, select the newest version of OFED that is compatible with the DGX and select Next.



Confirm the information on the Summary screen and then select Next.

The Summary screen provides an opportunity to confirm the Head node/basic cluster configuration

#### before installation begins.

If values do not match site survey, use the back button to navigate to the appropriate screen to correct any errors.



Monitor the progress of the installation, once the deployment is complete, select Reboot.

#### Note

You can tick the "Automatically reboot after installation is complete" box to have the headnode automatically reboot after the installation completes.

NVDIA EULA     Installation progress       Kernel modules     Overview of installation       Hardware info     Parsing build config       Installation source     Mounting CD/DVD-ROM       Cluster settings     Parsing build config       Network topology     Installing ubuntu Server 22.04       Network topology     Installing head node distribution packages       Compute nodes     BMC configuration
Kernel modules     Overview of installation       Hardware info     Parsing build config       Installation source     Mounting CD/DVD-ROM       Cluster settings     Partitioning harddrives       Installing Ubuntu Server 22.04     Installing Ubuntu Server 22.04       Workload manager     Installing Ubuntu Server 22.04       Network topology     Head node       Compute nodes     BMC configuration
Hardware info <ul> <li>Parsing build config</li> <li>Installation source</li> <li>Mounting CD/DVD-ROM</li> <li>Partitioning harddrives</li> <li>Partitioning harddrives</li> <li>Installing build config</li> <li>Installing build config</li> <li>Installing build config</li> <li>Installing head node distribution packages</li> <li>Metwork topology</li> <li>Head node</li> <li>Compute nodes</li> <li>BMC configuration</li> <li>Partitioning harddrives</li> <l< th=""></l<></ul>
Installation source     ✓ Mounting C/J/VC-ROM       Cluster settings     ✓ Partitioning harddrives       Workload manager     ● Installing bard node distribution packages       Network topology     Head node       Compute nodes     Engeling       BMC configuration     BMC configuration
Cluster settings         Workload manager        installing Ubuntu Server 22.04        Network topology        installing head node distribution packages        More compute nodes        compute nodes        BMC configuration        BMC configuration
Workload manager     Description       Network topology     Head node       Groupute nodes     Compute nodes       BMC configuration     BMC configuration
Network topology       Head node       Compute nodes       BMC configuration
Head node Compute nodes BMC configuration
Compute nodes BMC configuration
BMC configuration
Networks
U     Head node interfaces
Compute nodes interfaces
Disk layout
Disk layout settings
Additional software
Summary
Deployment 5/14
Automatically reboot after installation is complete
Show config Install log Back Reboot

## 9.6. Login to the Headnode

Once the headnode has finished rebooting, ssh to it using the root credentials.

## 9.7. Update BCM

Use apt update followed by apt upgrade to get the latest version of tools/utilities. Reboot the system if prompted.

## 9.8. Activate the BCM Cluster License

License the cluster by running the request-license command and providing the product key and other pieces of information as per the site survey.

(continued from previous page) If setting up a second headnode for HA, enter the mac address for it's →primary in-band interface. Will this cluster use a high-availability setup with 2 head nodes? [y/N] y MAC Address of secondary head node for eth0 [XX:XX:XX:XX:XX:XX:XX]: -5c:6f:69:24:dd:54Certificate request data saved to /cm/local/apps/cmd/etc/cluster.csr.new Submit certificate request to http://licensing.brightcomputing.com/licensing/ →index.cgi ? [Y/n] Y Contacting http://licensing.brightcomputing.com/licensing/index.cgi... License granted. License data was saved to /cm/local/apps/cmd/etc/cluster.pem.new Install license? [Y/n] Y ======= Certificate Information ======== Version: 10 Edition: Advanced OEM: NVIDIA Common name: Demo Cluster Organization: NVIDIA Organizational unit: Demo Locality: Santa Clara State: California Country: US Serial: 2369865 Starting date: 04/0ct/2023 Expiration date: 01/Sep/2024 MAC address / Cloud ID: 08:C0:EB:F5:72:0F|5C:6F:69:24:DD:54 Licensed tokens: 8192 Pay-per-use nodes: Yes Accounting & Reporting: Yes Allow edge sites: Yes License type: Free \_\_\_\_\_ Is the license information correct ? [Y/n] Y Backup directory of old license: /var/spool/cmd/backup/certificates/2024-05- $\rightarrow 31 08.25.05$ Installed new license Revoke all existing cmd certificates Waiting for CMDaemon to stop: OK Installing admin certificates Waiting for CMDaemon to start: OK mysql: [Warning] Using a password on the command line interface can be ⇒insecure. Copy cluster certificate to 3 images / node-installers Copy cluster certificate to /cmimages/default-image//cm/local/apps/cmd/etc/ →cluster.pem

(continued from previous page)

Copy cluster certificate to /cm/node-installer//cm/local/apps/cmd/etc/cluster. →pem Copy cluster certificate to /cmimages/dgx-os-6.3-h100-image//cm/local/apps/ →cmd/etc/cluster.pem Copy cluster certificate to /cmimages/dgx-os-6.3-a100-image//cm/local/apps/ →cmd/etc/cluster.pem mysql: [Warning] Using a password on the command line interface can be →insecure.

Regenerating certificates for users

New license was installed. In order to allow compute nodes to obtain a new node certificate, all compute nodes must be rebooted.

Please issue the following command to reboot all compute nodes: pdsh -g computenode reboot

### 9.9. Enable Bonding on the Headnode

#### Note

We recommended that you perform this from a remote/physical KVM, not via SSH. Before attempting the following steps on the headnode, verify the headnode's out of band management or BMC interface/remote/physical KVM is reachable and in service.

In the event a mistake is made here, remote access will temporarily be lost to the host OS, and the out of band management or BMC interface or remote console/crash cart would be the only way to rectify the problem.

In this step, we'll clear the managementnet (internalnet) interface IP which was assigned to the primary interface during the installation and assign it to the newly created bond interface with the network interfaces. Refer to site survey for the network interface names/MAC addresses.

Login to headnode and run Cluster Manager Shell (cmsh).

```
root@HEAD-01:~# cmsh
[bcm10-headnode]% device
[bcm10-headnode->device]% use bcm10-headnode1
[bcm10-headnode->device[bcm10-headnode]]% interfaces
[bcm10-headnode1->device[bcm10-headnode1]->interfaces]% list
            Network device name IP
Type
                                                  Network
                                                                   Start if
               _____
bmc
            ipmi0
                                 <Change IP>
                                                 ipminet
                                                                  always
                                               internalnet
physical
            ens3f1np1 [prov] <Change IP>
[bcm10-headnode->device[bcm10-headnode]->interfaces]% clear ens3f1np1 ip
[bcm10-headnode->device*[bcm10-headnode*]->interfaces*]% clear ens3f1np1
→network
[bcm10-headnode->device*[bcm10-headnode*]->interfaces*]% add physical ens2np0
[bcm10-headnode->device*[bcm10-headnode*]->interfaces*[ens2np0*]]% set mac
```

(continued from previous page)

Verify the IP connectivity to the BCM headnode using ping/ssh before proceeding.

# Chapter 10. Cluster Bring Up

This section addresses configuration steps to be performed on Base Command Manager headnode1.

## 10.1. Enable DeviceResolveAnyMAC

The following section enables provisioning of the bonded interfaces on downstream appliances/nodes.

This process enables failover PXE booting for bonded interfaces.

Edit /cm/local/apps/cmd/etc/cmd.conf and add the following line

AdvancedConfig = { "DeviceResolveAnyMAC=1" } # modified value

Example:

```
nano /cm/local/apps/cmd/etc/cmd.conf
GNU nano 6.2
# Set one or more advanced config parameters, only do this when needed
# AdvancedConfig = { "param=value", "param=value" }
AdvancedConfig = { "DeviceResolveAnyMAC=1" } # modified value
```

Once the above parameter has been saved restart the CMDaemon

root@bcm10-headnode:~# systemctl restart cmd

## 10.2. Define Cluster Networks

Next we'll add and configure the additional networks needed for BasePOD.

Refer to Site Survey for the details.

### 10.2.1. Change network names

First we will change the default network names to align with the names defined in Networking section under Overview.

```
root@bcm10-headnode1:~# cmsh
[bcm10-headnode1]% network
[bcm10-headnode1->network]% list
```

(continued from previous page)

```
Name (key) Type Netmask bits Base address Domain name IPv6
globalnet Global 0 0.0.0.0 cm.cluster
internalnet Internal 24 10.184.94.0 eth.cluster
ipminet Internal 24 10.160.6.0 ipmi.cluster
[bcm10-headnode1->network]% use internalnet
[bcm10-headnode1->network[internalnet]]% set name managementnet
[bcm10-headnode1->network*[managementnet*]]% ...
[bcm10-headnode1->network*]% use ipminet
[bcm10-headnode1->network*[ipminet]]% set name oobmanagementnet
[bcm10-headnode1->network*[oobmanagementnet*]]% ...
[bcm10-headnode1->network*]% commit
Successfully committed 2 Networks
[bcm10-headnode1->network]% list
Name (key) Type Netmask bits Base address Domain name IPv6
globalnet Global 0 0.0.0.0 cm.cluster
managementnet Internal 24 10.184.94.0 eth.cluster
oobmanagementnet Internal 24 10.160.6.0 ipmi.cluster
```

### 10.2.2. Computenet Config

Starting with computenet, to facilitate gpu to gpu RDMA communication.

```
root@bcm10-headnode1:~# cmsh
[bcm10-headnode1]% network
[bcm10-headnode1]% add computenet
[bcm10-headnode1->network*[computenet*]]% set domainname ib.compute
[bcm10-headnode1->network*[computenet*]]% set baseaddress 100.126.0.0
[bcm10-headnode1->network*[computenet*]]% set netmaskbits 16
[bcm10-headnode1->network*[computenet*]]% commit
```

### 10.2.3. Storagenet Config

Ethernet (tcp) Storage

BasePOD typically has Ethernet attached Block Storage solutions to the managementnet (internalnet).

In such scenarios it is not necessary to define any additional networks.

### 10.2.4. IB Storage

In the event IB Storage is attached to the cluster an additional Infiniband network will need to be defined using the following commands.

```
[bcm10-headnode1->network[computenet]]% clone computenet storagenet
[bcm10-headnode1->network*[storagenet*]]% set domainname ib.storage
[bcm10-headnode1->network*[storagenet*]]% set baseaddress 100.127.0.0
[bcm10-headnode1->network*[storagenet*]]% commit
```

Verify using cmsh CLI

```
[bcm10-headnode1]% home;network;list -f
name:20,type:10,netmaskbits:10,baseaddress:15,domainname:20
name (key) type netmaskbit baseaddress domainname
computenet Internal 16 100.64.0.0 ib.compute
managementnet Internal 24 10.184.94.0 eth.cluster
oobmanagementnet Internal 24 10.160.6.0 ipmi.cluster
```

## 10.3. Enable Out-of-band Management of Cluster Nodes

Set the BMC Username and Password for all BCM managed nodes

## 10.4. DGX Node Bringup

### 10.4.1. Software Image Setup for DGX's

Next we'll create a backup image of the DGX software image on the headnode.

This is a safety step which lets us make changes to the in-use image, and revert back to a factory DGX OS image in the event that something goes wrong.

```
cmsh
[bcm10-headnode]% softwareimage
[bcm10-headnode->softwareimage]% clone dgx-os-6.2-h100-image dgx-os-6.2-h100-
image-orig
[bcm10-headnode->softwareimage*[dgx-os-6.2-h100-image-orig*]]% commit
```

### 10.4.2. DGX Node category setup

Next, we're going to define the DGX node identities in BCM.

All of the DGX nodes in the DGX BasePOD will be named using the following naming convention "dgxxx", this helps differentiate them from the other nodes.

We'll first start by defining dgx-01's node identity and DGX node category

```
cmsh

[bcm10-headnode]% device

[bcm10-headnode->device]% clone node01 dgx-01

[bcm10-headnode->device*[dgx-01*]]% set category dgx-h100

[bcm10-headnode->device*[dgx-01*]]% commit
```

### 10.4.3. DGX Interface Definitions

Consult the site survey for the specific interface/IP addresses to assign to DGX nodes.

First we'll define the BMC and managementnet bond interfaces

```
[bcm10-headnode1->device*[dgx-01*]]% interfaces
[bcm10-headnode1->device*[dgx-01*]->interfaces]% set ipmi0 ip 10.160.6.31
[bcm10-headnode1->device*[dgx-01*]->interfaces*]% set ipmi0 network
→oobmanagementnet
[bcm10-headnode1->device*[dgx-01*]->interfaces*]%
[bcm10-headnode1->device*[dgx-01*]->interfaces*[ipmi0*]]% add physical
→enp170s0f1np1; add physical enp41s0f1np1
[bcm10-headnode1->device*[dgx-01*]->interfaces*[enp41s0f1np1*]]% add bond
→bond0 10.133.15.31 managementnet
[bcm10-headnode1->device*[dgx-01*]->interfaces*[bond0*]]% append interfaces
→enp170s0f1np1 enp41s0f1np1
[bcm10-headnode1->device*[dgx-01*]->interfaces*[bond0*]]%
[bcm10-headnode1->device*[dgx-01*]->interfaces*]% remove bootif
[bcm10-headnode1->device*[dax-01*]->interfaces*]%
[bcm10-headnode1->device*[dgx-01*]]% set provisioninginterface bond0
[bcm10-headnode1->device*[dgx-01*]]% commit
```

Now add the ib interface definitions for the compute fabric

```
[bcm10-headnode->device*[dgx-01*]->interfaces[bond0]]% add physica ibp220s0

→100.126.0.31 computenet
[bcm10-headnode->device*[dgx-01*]->interfaces*[ibp154s0*]]% foreach -o

→ibp220s0 ibp154s0 ibp206s0 ibp192s0 ibp24s0 ibp64s0 ibp79s0 ibp94s0 ()
[bcm10-headnode->device*[dgx-01*]->interfaces*]% set ibp154s0 ip 100.126.1.31
[bcm10-headnode->device*[dgx-01*]->interfaces*]% set ibp206s0 ip 100.126.2.31
[bcm10-headnode->device*[dgx-01*]->interfaces*]% set ibp192s0 ip 100.126.3.31
[bcm10-headnode->device*[dgx-01*]->interfaces*]% set ibp79s0 ip 100.126.4.31
[bcm10-headnode->device*[dgx-01*]->interfaces*]% set ibp64s0 ip 100.126.5.31
[bcm10-headnode->device*[dgx-01*]->interfaces*]% set ibp64s0 ip 100.126.5.31
[bcm10-headnode->device*[dgx-01*]->interfaces*]% set ibp94s0 ip 100.126.5.31
[bcm10-headnode->device*[dgx-01*]->interfaces*]% set ibp24s0 ip 100.126.7.31
[bcm10-headnode->device*[dgx-01*]->interfaces*]% set ibp24s0 ip 100.126.7.31
```

### 10.4.4. Defining the storage fabric:

### 10.4.5. Ethernet (tcp)

For Ethernet attached block storage solutions no additional network interface definitions are necessary as the interface used will be the bond0 bonded interface.

### 10.4.6. Infiniband (o2ib)

For Infiniband attached block storage solutions run the following commands to define the 2 additional storagenet interfaces for the DGX appliance.

```
[bcm10-headnode->device[dgx-01]->interfaces]% add physical ibp170s0f0 100.127.

...0.31 storagenet
[bcm10-headnode->device*[dgx-01*]->interfaces*[ibp170s0f0*]]% add physical

...ibp41s0f0 100.127.1.31 storagenet
[bcm10-headnode->device*[dgx-01*]->interfaces*[ibp41s0f0*]]% commit
[bcm10-headnode->device[dgx-01]->interfaces[ibp41s0f0]]% exit
```

### 10.4.7. Define DGX-01's MAC address

Here we are going to assign MAC addresses to the two managementnet (internalnet) attached interfaces that belong to bond0.

When assigning MAC address there is no specific order/requirement of which MAC goes to which enumerated interface name, so long as both MAC addresses are recorded for the appropriate DGX node identity

Refer to site survey for interface MAC details.

```
[bcm10-headnode->device[dgx-01]->interfaces]% set enp170s0f1np1 mac

→94:6D:00:00:00:FB
[bcm10-headnode->device*[dgx-01*]->interfaces*]% set enp41s0f1np1 mac

→94:00:00:00:74:0B
[bcm10-headnode->device*[dgx-01*]->interfaces*]% exit
[bcm10-headnode->device*[dgx-01*]]% set mac 94:6D:00:00:00:FB
[bcm10-headnode->device*[dgx-01*]]% commit
```

### 10.4.8. Test Provisioning of DGX-01

DGX-01 is now ready to be provisioned. It can be powered on using the physical power button, by using the BMC, or via IPMI tool command from the Headnode.

```
[bcm10-headnode->device[dgx-01]]% power on
ipmi0 ..... [ ON ] dgx-01
```

OR

root@HEAD-01:~# module load ipmitool
ipmitool -I lanplus -U <BMC User> -P <pass> -H 10.160.6.31 power on

This DGX bootup process will take several minutes for it to go through POST. You can monitor the progress from a KVM or via BMC Virtual Console.

If DGX-01's boot options were properly configured in the BIOS (i.e PXE boot as the first boot option with the proper interface) the node should proceed to attempt PXE booting.

The DGX will load an installer environment to help facilitate the provisioning process and finally load into the Cluster Manager Node Installer environment.

In the event that the DGX successfully identifies itself, you will see the following automated "Confirm node" prompt. The timer will expire and then proceed to provision the DGX with the displayed identity.

If required, validate the DGX hostname/category/MAC/network IPs with the site survey.

Hostname:	dgx-01			
Category:	dgx-h100			
Mac:	46:63:BA:D1:CB:14	, Tag:		
Network: Power:	[no network switc	h port configure	d]	
Roles:	[no roles assigne	d]		
Interfaces:	ipmi0	10.160.6.31	ipminet	
	enp170s0f1n+ (bon	d0)		
	enp41s0f1np1 (bon	d0)		
	bond0 [pro	v] 10.184.94.11	internalnet	
	1bp220s0	100.64.0.1	computenet	
z x	10p154SU	100.64.1.1	computenet	
()				
		Accept		
	Manually s	elect another no	de	
		4		
		7		

The next screen shows when a DGX appliance has successfully PXE booted. This state is fully automated and no user intervention is required here.

Please AUTO	[ Install mode ] select the install mode, the option are: – Check all disk partitions and file systems. If no errors , app detected, attempt quick provisioning. In case of
FULL MAIN	<ul> <li>Re-create all partitions and request full provisioning.</li> <li>Re-create all partitions and request full provisioning.</li> <li>Do not check disk and drop to maintenance shell.</li> <li>Do not superprize disks upless file sustems are broken.</li> </ul>
SKIP	- Do not check and synchronize disks.
	AUTO
	FULL
	MAIN
	NOSYNC
	SKIP
	a de la companya de l

Cluster Manager Node Installer

From here the DGX will proceed to provision itself via the served identity from the BCM headnode.

As the DGX progresses through the PXE boot provisioning process, you can monitor the progress from the headnode via cmsh.

Once DGX-01 successfully shows a status of UP we can proceed to clone this node identity for the remaining needed DGX's, in this example we are adding 3 additional nodes for a total of 4.

#### Note

In the event the provisioning attempt fails or encounters problems refer to /var/log/messages and /var/log/node-installer log files to further diagnose the provisioning issue.

Set the MAC addresses for each of the new nodes. Repeat the steps below for each new DGX node, refer to the site survey for the details.

```
[bcm10-headnode->device]% use dgx-02; interfaces
[bcm10-headnode->device[dgx-02]->interfaces]% set enp170s0f1np1 mac

...94:6D:00:00:00:FD
[bcm10-headnode->device*[dgx-02*]->interfaces*]% set enp41s0f1np1 mac

...94:6D:00:00:00:FE
[bcm10-headnode->device*[dgx-02*]->interfaces*]% exit
[bcm10-headnode->device*[dgx-02*]]% set mac 94:6D:00:00:00:FD
[bcm10-headnode->device*[dgx-02*]]% commit
```

Proceed to power on and provision the remaining DGX nodes into the BCM Cluster.

You can verify the provisioning progress/status using cmsh

# Chapter 11. BCM HA

We will be configuring BCM head node high availability next by provisioning the second BCM headnode

Verify that the head node has power control over the cluster nodes.

```
% device
% power -c dgx-h100 status
[-head1->device]% power -c dgx-h100 status
ipmi0 ..... [ ON ] bcm-dgx-h100-01
ipmi0 ..... [ ON ] bcm-dgx-h100-02
ipmi0 ..... [ ON ] bcm-dgx-h100-03
ipmi0 ..... [ ON ] bcm-dgx-h100-04
[bcm-head-01->device]%
```

Power off the cluster nodes.

The cluster nodes must be powered off before configuring HA.

```
% power -c dgx-h100 off
ipmi0 ..... [ OFF ] bcm-dgx-h100-01
ipmi0 ..... [ OFF ] bcm-dgx-h100-02
ipmi0 ..... [ OFF ] bcm-dgx-h100-03
ipmi0 ..... [ OFF ] bcm-dgx-h100-04
```

Start the cmha-setup CLI wizard as the root user on the primary head node.

#cmha-setup

Choose Setup and then select SELECT.

Welcome t Please th Storage' if high a detailed	the Bright Cluster Manager High Availability Setup Utility. ose 'Setup' to enter the failover settings menu, 'Shared so setup shared storage, 'Status' to view the failover status, ailability has already been setup. Choose 'Help' to see a lescription of the options available.	
	SeturConfigure failover seturShared StorageConfigure shared storageStatusView failover statusHelpCmha-setup help	
	STIFCTS < QUIT >	

Choose Configure and then select NEXT.

Select 'Conf the failover 'Clone Insta head node, i 'Install Pro 'Undo Failov	igure' to configure 'setup if the second ll' to see the insta f the failover confi gress', if the secon er' to remove existi	failover setup, 'Finalize' to finalize lary head node has been installed. Choose llation instructions for the secondary guration has been completed. Choose dary head node is being installed. Choose ng failover configuration.
	Clone Install Clone Install Install Progress Finalize Undo Failover Main	Configure failover setup View install instructions View install progress Finalize failover setup Remove failover setup Main menu
	S NEXT S	< BACK >

Verify that the cluster license information/MAC found in cmha-setup screen is correct and then select CONTINUE.



Configure an external Virtual IP address (obtained from Site Survey) that will be used by the active head node in the HA configuration and then select NEXT.

#### Note

This will be the IP that should always be used for accessing the active head nodes once HA configuration is complete.

Cluster Manager Hig	h Availability Setup	
	Please enter the values for the shared internal interface parameters. The IP address must be in range of the management network. The interface must be one that is not in use already.           Name:         pond01ha           IP [10.184.94.0/24]:         10.184.94.251	

Provide the name of the secondary head node and then select NEXT.

Nama	barapad baad2	 
[name.		

Populate the failover network details as prescribed in the Site Survey

There are two options for the failover network:

- 1. Using dedicated interfaces on the BCM head nodes or
- 2. Utilizing the existing managementnet (internalnet).

If you are using a dedicated network interface, select the chosen failover interface and provide the IP information for the dedicated network.

In this example, we are using the internal network as the failover network by skipping dedicated failover interface configuration.

Refer to the BCM admin guide for BCM HA failover network configuration options.

Cluster Manager High	h Availability Setup
_	
	This screen takes parameters of a dedicated failover network that will be created. Please keep in mind, to select a network name different from
	those that have been defined already.
	Name: failovernet Base address: 10.151.0.0
	Netmask: 255.255.0.0 Domain name: failover.cluster

Configure the IP addresses for the secondary head node that the wizard is about to create and then select NEXT.

Please update the IP addresses for the secondary head node interfaces.           Name NetworkPrimary IP Secondary IP           ipmi0         [10.160.6.0/24]           10.160.6.254         10.160.6.253           bond0         [10.184.94.0/24]           10.184.94.253         10.184.94.253
< NEXT > < BACK >

The wizard shows a summary of the information that it has collected. The VIP will be assigned to the internal and external interfaces, respectively.

Cluster Manager Hig	h Availability Setup	
	SUMMARY	
	Failover Setup Sum	mary
	Shared Internal Interface:	
	Name: IP:	bond0:ha 10.184.94.251
	Shared External Interface:	SKIPPING
	Failover Network:	SKIPPING
	Enilover Network Interfaces(Primary):	
	Fallover Network Interfaces(Frimary).	SVIPPING
	Failover Network Interfaces(Secondary):	SKIPPING 715
	< EXIT >	

Select Yes to proceed with the failover configuration.



Enter the BCM root password and then select OK.

Please enter the mysql root password:	
< <u>0%</u> ≼Cancel≻	

The wizard implements the first steps in the HA configuration. If all the steps show OK, press ENTER to continue. The progress is shown here.

Initializing failover setup on master [ OK ]	
Updating shared internal interface [ OK ]	
Updating shared external interface [ OK ]	
Updating extra shared internal interfaces [ OK ]	
Cloning head node [ OK ]	
Updating secondary master interfaces [ OK ]	
Updating Failover Object [ OK ]	
Restarting cmdaemon [ OK ]	
Press any key to continue	

When the failover setup installation on the primary master is complete, select OK to exit the wizard to the main HA setup screen

We will come back to the HA setup screen once the secondary headnode is added to the cluster.



Power up and PXE boot the secondary head node and then select RESCUE from the grub menu.



After the secondary head node has booted into the rescue environment, run the

/cm/cm-clone-install --failover

command, then enter YES when prompted.

The secondary head node will be cloned from the primary.



Specify the inband interface on the secondary node

Press c to save the disk layout when prompted

When cloning is completed, enter y to reboot the secondary head node. The secondary must be set to boot from its hard drive. PXE boot should not be enabled. Use the appliance BMC/BIOS to change the boot order. Wait for the secondary head node to reboot.

Creating failover/clone modes:   Install the secondary head mode   \$ /cw/cm-clone-installfailover   Create a clone of the primary head mode   \$ /cw/cm-clone-installfailover   Clonehostname=mew-hostname   \$ /cw/cm-clone-installfailoverreboot   \$ /cw/cm-clone-installfailover ClusterManager 10gin: root (estomatic login) Linux ClusterManager 5.13.0-39-generic M4'20.04.1-thunts SPP Thm Har 24 16:43: root#ClusterManager 18 /cov/cm-clone-installfailover Network interManager 18 /cov/cm-clone Inter the password of the headmode mode to continue. /cov/cmeedia /cov/cmeedia /cov
<pre>i = install the secondary head mode i \$ /cm/cm-clone-installfailover i = Create a clone of the primary head mode i \$ /cm/cm-clone-installclonehostname-new-hostname i = install the secondary (failover) head mode and reboot antomatically i = install the secondary (failover) head mode and reboot antomatically i = install the secondary (failover) head mode and reboot antomatically i = install the secondary (failover) head mode and reboot antomatically i = install the secondary (failover) head mode and reboot antomatically i = install the secondary (failover) head mode and reboot antomatically i = install the secondary (failover) head mode and reboot antomatically i = install the secondary (failover) head mode and reboot antomatically i = install the secondary (failover) head mode and reboot antomatically i = install the secondary (failover) head mode and reboot antomatically i = install the secondary (failover) head mode and reboot antomatically i = install the secondary (failover) head mode and reboot antomatically i = install the secondary (failover) head mode and reboot antomatically i = install the installfailoverreboot Linex ClosterHamager: is //av/cm-clone-installfailover Network interface to use (default: eng01: ensifing) Please wit while bringing up network Please wit while installation begins Please wit while installation begins ProtHamager is passand: Please wit while installation begins I = GK i De head mode disk lagoet is saved in //cm/meadodedisksetup.xel info: Detecting device //dev/meadod1: //cm/meadodedisksetup.xel info: Usid device sumeNal. All checks have succeeded. The contents of the following disks will be erased. /dev/meadol /dev/meadol</pre>
<pre>\$ /cm/cm-clone-installfailover • Create a clone of the primary head mode \$ /cm/cm-clone-installclonehostname=new-hostname • install the secondary (failover) head mode and reboot antonatically \$ /cm/cm-clone-installfailoverreboot • help \$ /cm/cm-clone-installfailoverreboot • help \$ /cm/cm-clone-installhelp ClusterManager logim: root (automatic logim) Limux ClusterManager 5.13.0-39-generic B44'20.04.1-Ubuntm SNP Thm Mar 24 16:43:3 rootUC-clone-installfailover Metaork interface to use (default: exp0: cms1finpt Please wait while mitmeline in the ing art up Fater the password of the headmode mode to continue. rootUmater's password: Please wait while installation begins Merifying license Getting disk lagout is moved in /cms_headmodelSkuetup.xml for ours, e = edit, c = continue 1: c = formed info: becomed in /dev/mmedil': found info: becomed of the following disk will be erased. /dev/mmedil for contexts of the following disks will be erased. /dev/mmedil /dev/medil /dev/med</pre>
<pre>- Create a clone of the primary head mode \$ /cn/cm-clone-installclonehostname-new-hostname = install the secondary (failower) head mode and reboot antonatically \$ /cn/cm-clone-installfailowerreboot = Help \$ /cn/cm-clone-installfailowerreboot = Help \$ /cn/cm-clone-installhelp ClusterManager: 5.13.0-73-generic H44^20.04.1-Ubunta SMP Thm Mar 24 16:43:25 root(ClusterManager: 7.25 /cn/cm-clone-installfailower Network interface to use (default: exp0): ens(fing) Please wit while bringing up setwork Please wit while installation begins Please wit while installation begins Prost wit while installation begins Portfung liceme Current of the headmode mode to continue; rootManater's paraword; Please wit while installation begins Derifying liceme Current of the installation begins Current of the installation begins Current of the installation begins for the head mode disk layout is asced in /com/licemidediskuetsp.xxl Gutting disk layout is asced in</pre>
<pre>\$ /cn/cm-clone-installclonehostname=new-hostname = install the secondary (failover) head node and reboot antonatically \$ /cn/cm-clone-installfailoverreboot = Help = \$ /cn/cm-clone-installhelp = thelp = \$ /cn/cm-clone-installhelp = clusterHamager: 5.13.0-79-generic #44'20.04.1-Ubunts SMT The Mar 24 16:43:25 rootHClusterHamager: 5.13.0-79-generic #44'20.04.1-Ubunts SMT The Mar 24 16:43:25 Returnsk interface to use Idefault: exp01: ens1fing1 Please wait while bringing up network Please wait while hestinging up network Please wait while installation hegins Derifying 1/cense Please wait while installation hegins 10K 1 The head node disk lagout is sourd in /cwrheadmodedisketsp.xxl 10- view, e = edit, c = continue 1: c 10fo: Betcring device '.dew.rwmedwal': found 10fo: Wild device newedwil. All checks have succeeded. The contents of the following disks will be erased. /dev.rwmedwil /dev.rwmedwil</pre>
<pre>i = install the secondary (failover) head node and reboot automatically i \$ /cn/cm-clone-installfailoverreboot i = Help i \$ /cn/cm-clone-installfailoverreboot i = Help i \$ /cn/cm-clone-installhelp i \$ /cn/cm-clone-installfailover ClusterManager 5.13.0-39-generic #44~20.04.1-Ubunts SPP Thm Mar 24 16:43:35 rout64clusterManager 7: # /cn/cm-clone-installfailover Retwork interface to use (default: exp0): ens(fay) Flease usit while brighting up network Flease usit while hestication is being set up Flease usit while installation begins Verifying license</pre>
\$ rowcs-close-installfailoverreboot * Help S rowcs-close-installhelp ClusterManager logis: root (automatic logis) Lines ClusterManager 5.13.0-39-generic 844'20.04.1-Ubunts SMP The Mar 24 16:43:35 rootsClusterManager: & rowcs-close-installfailover Network interface to use (default: exp0): ext[figs] Please wit while bringing up metwork Please wit while authentication is being set up Enter the password of the headende mode to continue. rootmaster's password: Please wit while installation begins Use fuging licence
<pre>i = Help i \$ /Ow/CA-Clone-installhelp i \$ /Ow/CA-Clone-installhelp i ClusterManager logis: root (automatic logis) Linex ClusterManager: % /Ow/CA-Clone-installfailover Retwork interface to use idefault: explicient[flup1 Floase wit while helping up network Floase wit while inthemication is being set up Enter the password of the headnode mode to continue. rootMmaster's password: Flease wit while installation begins Verifying license Flease wit while installation begins f OK 1 for the password of the headnode index in to the installation begins f OK 1 for the password of /doc/waveb04': found info: Detecting double index[and in /Ow/headnodedisksetsp.ow1 fu = uiew, e = edit, c = continue 1: c finfo: betecting double index[and in found info: Wild double modeDate in found info: Wild double momeDat. All checks have succeeded. The contexts of the following disks will be erased. rdevrawe0a1 found to the following disks will be erased. rdevrawe0a1 found found</pre>
\$ /On/Ca-clone-installhelp     [     S /On/Ca-clone-installhelp     [     ClasterManager logis: root (automatic logis)     Linex ClasterManager: 5.13.0-29-generic 844"20.04.1-Ubunts SNP The Mar 24 16:43:25     rootRClasterManager: Tm /On/Ca-clone-installfailower     Metwork interface to use (default: eng0): ens[fng]     Flease wit while helnging up network     Flease wit while helnging up network     Flease wit while installation begins     Yenifying license     for the headmode sofe to continue.     Youring license     for the headmode sofe to continue.     Youring license     for the headmode sofe to continue.     To the node of the lagort is saved in /Cabeadmodeliskuetup.xel     lo - view, e = elit, c = continue 1: c     Info: Detecting device /dev/nweebal': found     Info: life to installation hegins     Headmode Jisk lagort is lowed in /Cabeadmodeliskuetup.xel     Info: betecting device /dev/nweebal': found     Info: Detecting device /dev/nweebal': found
ClusterManager logis: root (automatic logis) Linex ClusterManager 5.13.0-39-generic #44~20.04.1-Ubunts SMP Thm Mar 24 16:43:35 1 root#ClusterManager: "# .cov.cn-close-installfailower Network interface to use (default: exp0): enst[Inpl Please wait while bringing up metwork Please wait while authentication is being set up Enter the password of the bednode node to continue. rootMmaster's password: Please wait while installation begins Werifying license
DO GOB WEET, LO CONTINUE LUCS/NOTE DES

Then continue the HA setup procedure on the primary head node.

On the primary headnode, in the HA setup screen, select Finalize from the cmha-setup menu and then select NEXT.

This will clone the MySQL database from the primary to the secondary head node.



Select CONTINUE on the confirmation screen.

CONFIRM	
Confirmation is required to proceed. The following tasks will be performed:	
1. Update secondary head node MAC addr 2. Clone MySQL database(s) to the seco	ess in cmdaemon database. ndary head node.
Press 'CONTINUE' to proceed with the f or press 'BACK' to return to the failo	inalize operation, ver Setup menu.
CONTINUES	< BACK >

Enter the root password and then select OK.

Please enter the mysql	root password:	
[		
< 0K >	<cancel></cancel>	

The cmha-setup wizard continues. Press ENTER to continue when prompted.



The progress is shown in the console during the process.

Updating secondary master mac address [ OK ]	
<pre>Initializing failover setup on bcm-head-02 [ OK ]</pre>	
Stopping cmdaemon [ OK ]	
Cloning cmdaemon database [ OK ]	
Checking database consistency [ OK ]	
Starting cmdaemon, chkconfig services [ OK ]	
Cloning workload manager databases [ OK ]	
Cloning additional databases [ OK ]	
Update DB permissions [ OK ]	
Checking for dedicated failover network [ OK ]	
Press any key to continue	

The Finalize step is now completed. Select REBOOT and wait for the secondary head node to reboot.



You can verify the secondary node's status from the primary head node using cmsh

<pre>[bcm10-headnode1]% device list -f hostname:20,category:12,ip:20,status:15 hostname (key) category ip status</pre>	
bcm-head-01 10.130.122.254 [ UP ] bcm-head-02 10.130.122.253 [ UP ]	

						(continued from previous page)
bcm-dgx-h100-01	dgx-h100	10.130.122.5	[	DOWN	]	
bcm-dgx-h100-02	dgx-h100	10.130.122.6	[	DOWN	]	
bcm-dgx-h100-03	dgx-h100	10.130.122.7	[	DOWN	]	
bcm-dgx-h100-04	dgx-h100	10.130.122.8	[	DOWN	]	
# Chapter 12. Setup shared NFS storage.

On the primary headnode, in the HA setup screen, select Shared Storage from the cmha-setup menu and then select SELECT.

In this final HA configuration step, cmha-setup will copy the /cm/shared and /home directories to the shared storage and configure both head nodes and all cluster nodes to mount it.

Welcome to Please cho Storage' t if high av detailed o	the Bright Cluster P ose 'Setup' to enter o setup shared storag allability has alread escription of the opt Setup Setup Status Help	Anager High Availability Setup Utility. the failover settings menu, 'Shared e, 'Status' to view the failover status, by been setup. Choose 'Help' to see a tions available. Configure failover setup Configure failover setup Configure status Cmha-setup help	
	SILICIE	< QUIT >	

Choose NAS and then select SELECT.

type of shared storag	storage solutions are supported. Please select a e from the menu below.
DAS	Network Attached Storage Direct Attached Storage Distributed Redundant Block Device
	SILLOT < BACK >

Choose both /cm/shared and /home and then select NEXT.



Provide the IP address of the NFS host and the NFS Shared paths for /cmshared and /home folders and then select NEXT.

Refer to the site survey for the details.

Please fill in NAS p	arameters		]
NAS host: Path to /cm/shared: Path to /home:		10.160.0.4 /data/nas/cmshared /data/nas/home	
	< NEXT >	< BACK >	_

The wizard shows a summary of the information that it has collected. Select EXIT to continue.

Cluster Manager High Av	velied-lifty Setup	
	NeSs shared storage config	
	NET HOLT 10, 100, 116, 4	
	Mis Paransi	
	Path to /cm/sharedi /bcmvikingshered	
	THE COLOR	

When asked to proceed with the NAS setup, select Yes to continue.

This will initiate a copy and update the NFS mount on the head nodes.



The cmha-setup wizard proceeds with its work.

Preparing nas setup 100%

When setup completes, select ENTER to finish HA setup.

The progress is shown here:

Copying NAS data	[	0K	[]	
Mount NAS storage	[	0K	[]	
Remove old fsmounts	[	0K	[]	
Add new fsmounts	[	0K	[]	
Remove old fsexports	[	0K	[]	
Write NAS mount/unmount scripts	[	0K	[]	
Copy mount/unmount scripts	[	0K	[]	
Press any key to continue				

cmha-setup is now complete. EXIT the wizard to return to the shell prompt on the primary headnode



Run the cmha status command to verify that the failover configuration is correct and working as expected.

The active head node is indicated by an asterisk. Here is an example of a working HA configuration.

```
# cmha status
Node Status: running in active mode
bcm-head-01* -> bcm-head-02
failoverping [ OK ]
mysql [ OK ]
ping [ OK ]
status [ OK ]
bcm-head-02 -> bcm-head-01*
failoverping [ OK ]
mysql [ OK ]
ping [ OK ]
status [ OK ]
```

Verify that the /cm/shared and /home directories are mounted from the NAS server. On the BCM head-node, run the following command.

# Chapter 13. Testing BCM HA

Login to the secondary head node to be made active and run cmha makeactive.

Run the cmsh status command again to verify that the secondary head node has become the active head node. The active head node is indicated by an asterisk.

```
# cmha status
Node Status: running in active mode
bcm-head-02\* -> bcm-head-01
failoverping [ OK ]
mysql [ OK ]
ping [ OK ]
status [ OK ]
bcm-head-01 -> bcm-head-02\*
failoverping [ OK ]
mysql [ OK ]
ping [ OK ]
status [ OK ]
```

Manually failover back to the primary head node by running cmha makeactive

```
# ssh bcm10-headnode1
# cmha makeactive
This is the passive head node. Please confirm that this node should
become the active head node. After this operation is complete, the HA status
of the head nodes will be as follows:
bcm-head-01 will become active head node (current state: passive)
bcm-head-02 will become passive head node (current state: active)
```

(continued from previous page)

Continue(c)/Exit(e)? c Initiating failover...... [ OK ] bcm-head-01 is now active head node, makeactive successful

Run cmha status command again to verify that the primary head node has become the active head node. Then proceed to power on the cluster nodes

```
# cmha status
Node Status: running in active mode
bcm-head-01\* -> bcm-head-02
failoverping [ OK ]
mysql [ OK ]
ping [ OK ]
status [ OK ]
bcm-head-02 -> bcm-head-01\*
failoverping [ OK ]
mysql [ OK ]
ping [ OK ]
status [ OK ]
```

Power on the cluster nodes.

#cmsh -c "device ; power -c dgx-h100 on"
ipmi0 ..... [ ON ] bcm-dgx-h100-01
ipmi0 ..... [ ON ] bcm-dgx-h100-02
ipmi0 ..... [ ON ] bcm-dgx-h100-03
ipmi0 ..... [ ON ] bcm-dgx-h100-04

This concludes the setup and verification of HA.

# Chapter 14. Deploy Kubernetes

# 14.1. Kubernetes Node Setup

This section outlines the configuration steps required on the BCM head node to provision the 3 Kubernetes control nodes

First, we will create the software image and define the category for the kubernetes nodes and assign the disklayout.

Add necessary kernel modules for bonding and mellanox drivers.

```
cmsh
[bcm10-headnode1]% softwareimage
[bcm10-headnode1->softwareimage]% clone default-image k8s-control-plane-image
[bcm10-headnode1->softwareimage*[k8s-control-plane-image*]]% commit
[bcm10-headnode1->softwareimage[k8s-control-plane-image]]%kernelmodules
[bcm10-headnode1->softwareimage[k8s-control-plane-image]->kernelmodules]% add
→mlx5 core
[bcm10-headnode1->softwareimage[k8s-control-plane-image]->kernelmodules]% add
→bondina
[bcm10-headnode1->softwareimage*[k8s-control-plane-image*]->
→kernelmodules*[mlx5_core*]]%commit
[bcm10-headnode1->softwareimage[k8s-control-plane-image]]% category
[bcm10-headnode1->category]% clone default k8s-control-plane
[bcm10-headnode1->category*[k8s-control-plane*]]% set softwareimage k8s-
→control-plane-image
[bcm10-headnode1->category*[k8s-control-plane*]]% set disksetup /cm/local/
→apps/cmd/etc/htdocs/disk-setup/x86_64-slave-one-big-partition-ext4.xml
[bcm10-headnode1->category*[k8s-control-plane*]]% commit
```

Create the first kubernetes node - knode-01- by cloning the default node01 and move it to the k8scontrol-plane category.

cmsh
[bcm10-headnode1]% device
[bcm10-headnode1->device]% clone node01 knode-01
[bcm10-headnode1->device\*[knode-01\*]]% set category k8s-control-plane
[bcm10-headnode1->device\*[knode-01\*]]% commit

Add and configure the IPMI (BMC) and managementnet (internalnet) bond interfaces.

#### Note

The name of the interfaces will change depending on the hardware vendor of the node appliance.

```
#cmsh
[bcm10-headnode1]% device
[bcm10-headnode1->device]% use knode-01
[bcm10-headnode1->device[knode-01]]% interfaces
[bcm10-headnode1->device[knode-01]->interfaces]% add bmc ipmi0 10.160.6.4
→ipminet
[bcm10-headnode1->device*[knode-01*]->interfaces*[ipmi0*]]% commit
[bcm10-headnode1->device[knode-01]->interfaces]% add physica ens2f1np1; add
→physical ens1f1np1
[bcm10-headnode1->device*[knode-01*]->interfaces*[ens1f1np1*]]% commit
[bcm10-headnode1->device[knode-01]->interfaces]% add bond bond0 10.184.94.4
→managementnet
[bcm10-headnode1->device*[knode-01*]->interfaces*[bond0*]]% append interfaces
→ens2f1np1 ens1f1np1
[bcm10-headnode1->device*[knode-01*]->interfaces*[bond0*]]% remove bootif
[bcm10-headnode1->device*[knode-01*]->interfaces*[bond0*]]% ...
[bcm10-headnode1->device*[knode-01*]->interfaces*]% ...
[bcm10-headnode1->device*[knode-01*]]% set provisioninginterface bond0
[bcm10-headnode1->device*[knode-01*]]% commit
```

Clone knode-01 to create the two additional knodes.

```
[bcm10-headnode1]% device
[bcm10-headnode1->device]% foreach --clone knode-01 -n knode-02..knode-03 --
→next-ip ()
[bcm10-headnode1->device*]% commit
```

Set the MAC addresses for each of the knodes so that they can PXE boot from BCM. Refer to the site survey for details.

Repeat these steps for all 3 knodes, to get their interface/MAC mapping in BCM for PXE boot/provisioning.

Power on and BCM will provision the kubernetes nodes with PXE boot.

cmsh

(continued from previous page)

```
[bcm10-headnode1]% device
[bcm10-headnode1->device]% power on -c k8s-control-plane
```

Ensure all the nodes are up

### 14.2. Kubernetes Deployment

This section addresses configuration steps to be performed on the BCM head node.

In the root shell, run the Kubernetes setup script.

cm-kubernetes-setup

Select Deploy to continue



Select the newest version certified for NVIDIA AI Enterprise (i.e. version with \*).

		- (	hoose	a K	uberne	tes	versi	on. —	 	
"*" Ve	ersion	cert	ified	for	NVIDI	A AI	Ente	rprise		
( <b>X</b> ) Ki	bernet	es \	/1.28	٠						
( ) Ki	bernet	es v	1.27							
( ) KI	bernet	es 🗤	1.26							
<ul> <li>Kı</li> </ul>	bernet	es 🗤	1.25							
( ) KI	bernet	es 🗤	1.24							
			<	0k	>	< B	ack >		 	-

(Optional) Enter a private registry server here if required.

Please provide a registry mirror	for pulling	container	images fr	om DockerHub	(optional)
DockerHub registry mirror server	(optional)				
	< 0k >	< Back >			

Keep the default settings for the cluster.

Insert basic values of the ne	ew Kubernetes cluster
Kubernetes cluster name Kubernetes domain name Kubernetes external FQDN Service network base address Service network netmask bits Pod network base address Pod network netmask bits	default cluster.local bcm10-headnode.eth.cluster 10.150.0.0 16 172.29.0.0
< 0k >	< Back >

Select yes to allow the cluster to be used from the headnode.

Do	you	want	to	expose	the	Kubernetes	API	server	to	the	external	network?
n	es D											
						< 0k >	<	Back >				
							_					

Select managementnet (internalnet) for kubernetes networking.

This One	is the network Kubernetes nodes use to communicate with other Kubernetes nodes. network must be selected. Use <space> to select a single value.</space>
Inc	ase nodes do not share an IP on this network, their internal networks
NILL	be configured as fallback networks.
D	storagenet
(X)	internalnet
()	dgxnet1
()	loopback
()	computenet
()	ipminet
	e Ok s e Back s

Select the 3 knodes we just provisioned for the control plane.

] bcm10-headnoo	de la
] bcm10-headnoo	de2
] dgx-01	category:dgx-h100
[ ] dgx-02	category:dgx-h100
[ ] dgx-03	category:dgx-h100
[ ] dgx-04	category:dgx-h100
] dgx-05	category:dgx-h100
] dgx-06	category:dgx-h100
[ ] dgx-07	category:dgx-h100
] dgx-08	category:dgx-h100
[ ] dgx-09	category:dgx-h100
] dgx-10	category:dgx-h100
] dgx-11	category:dgx-h100
] dgx-12	category:dgx-h100
] dgx-13	category:dgx-h100
] dgx-14	category:dgx-h100
] dgx-15	category:dgx-h100
] dgx-16	category:dgx-h100
] dgx-17	category:dgx-h100
] dgx-18	category:dgx-h100
] dgx-19	category:dgx-h100
] dgx-20	category:dgx-h100
[X] knode-01	category:k8s-control-plane
[X] knode-02	category:k8s-control-plane
X] knode-03	category:k8s-control-plane

Select the dgx-h100 category for Kubernetes workers.

	_
Select node categories to use as Kubernetes workers	
[] default	
[X] dgx-h100 [] k8s-control-plane	
	'
	-
< Ok > < Back >	

Skip selecting individual worker nodes.

	la.	a hom10 hoods
	1e 1o2	bcm10-headn
	category:k8s-control-plane	[] knode-01
	category:k8s-control-plane	[] knode-01
	category:k8s-control-plane	[] knode-02
e	category:k8s-control-plane category:k8s-control-plane	[ ] knode-02 [ ] knode-03

Select the 3 kubernetes nodes to be etcd nodes.

] bcm10-headn	ode	
] bcm10-headn	ode2	
] dgx-01	category:dgx-h100	
] dgx-02	category:dgx-h100	
] dgx-03	category:dgx-h100	
] dgx-04	category:dgx-h100	
] dgx-05	category:dgx-h100	
] dgx-06	category:dgx-h100	
] dgx-07	category:dgx-h100	
] dgx-08	category:dgx-h100	
] dgx-09	category:dgx-h100	
] dgx-10	category:dgx-h100	
] dgx-11	category:dgx-h100	
] dgx-12	category:dgx-h100	
] dgx-13	category:dgx-h100	
] dgx-14	category:dgx-h100	
] dgx-15	category:dgx-h100	
] dgx-16	category:dgx-h100	
] dgx-17	category:dgx-h100	
] dgx-18	category:dgx-h100	
] dgx-19	category:dgx-h100	
] dgx-20	category:dgx-h100	
X] knode-01	category:k8s-control-plane	
X] knode-02	category:k8s-control-plane	
X] knode-03	category:k8s-control-plane	

Click "OK" to the symlink dialogue

The passive head node has /etc/kubernetes as a symlink pointing to: .
(This should not be problematic as long as the target is guaranteed to exist.)

Accept the default values for the main Kubernetes components.

Configure the values	for the main Kubernetes components:
API server proxy port API server port	10443 6443
Etcd spool directory	/var/lib/etcd
<	0k > < Back >

Select the Calico CNI plugin.

Select the K	ubernetes ne	twork p	olugin	
(X) Calico () Flannel	(recommended)	)		
	< 0k	>	< Back >	

Select no for installing the Kyverno Policy Engine.

o you mane co	instatt kyverno roti	icy Engine:		
(yverno is a p and generate c	olicy engine designed onfigurations using d	d for Kubernetes admission control	It can validate, s and background s	mutate, scans.
VAS				
no				
	< 0k x	< Back >		

(Optional) If an NVAIE license has been provided, select yes and enter the details on the following page. Otherwise, select no.

Do you have NVAIE lie	censes?		
yes no			
K	0k >	<mark>&lt; Back &gt;</mark>	

Select the following operators to install:

- ▶ NVIDIA GPU Operator
- cm-jupyter-kernel-operator
- cm-kubernetes-mpi-operator
- Network Operator
- ► Kubeflow Training Operator
- Prometheus Adapter
- Prometheus Operator Stack

Optional - MetalLB if you are planning to expose services directly from the DGX nodes, like NIM/inference workloads

Please choose the operators to install
<pre>[X] NVIDIA GPU Operator [X] cm-jupyter-kernel-operator [X] cm-kubernetes-mpi-operator</pre>
[ ] KubeFlow Training operator [X] Kubernetes Dashboard
[X] Kubernetes Metrics Server [X] Kubernetes State Metrics [X] MetalLB [ ] Net0
<pre>[X] Network Operator [ ] NIM Operator [ ] postgresgl-operator</pre>
<pre>[ ] Prometheus Adapter [ ] Prometheus Operator Stack [ ] Run:ai</pre>
[] spark-operator
< Ok > < Back >

Select the latest version certified for NVIDIA AI Enterprise.

"*" Ve	rsion (	certif	ied for		τα ατ ε	nternr	ise	
ve.	51011		icu ioi	11110		incer pr	130	
( ) la	test							
() v2	3.9.2							
( <mark>X</mark> ) v2	3.9.1							
() v2	3.9.0							
() v2	3.6.2							
() v2	3.6.1							
() v2	3.6.0							
() vZ	3.3.2							
() v2	3.3.1							
() v2	3.3.0							
() vZ	2.9.2							
() v2	2.9.1							
() v2	2.9.0							
			- Ok		e Bac	k s		 
			< UK	~	< bac	K >		

Select the latest version of network operator certified for NVIDIA AI Enterprise.

	Vansian	- Choo	icd for	NIVITOT	erator ve	ersion ·	
+	version	certif	lea for	NVIDI	A AL ENCO	erprise	
dD	latest						
$\bigcirc$	23.10.0						
$\alpha$	23.7.0						
$\dot{\odot}$	23.5.0						
õ	23.1.0						
			< 0k	>	< Back >		

Leave the custom YAML file blank for the GPU Operator.

uration paramete	rs will be availe	able in the next st
)		
e Ok >	e Back	
	)	)

Enable CDI (container device interface) and NFD (node feature discovery)

Configure NVIDIA	GPU Operator		
[X] cdi.enabled	Make GPUs access	ible to containers	
[X] nfd.enabled	Deploy Node Featu	ure Discovery plugin as a daemo	nset
	< 0k >	< Back >	

Leave the Network Operator custom YAML configuration file name as blank.

If empty, ba	sic configur	ation paramet	ers will be av	ailable in the n	ext step
Path to file	(optional)				
	., ,				
		< 0k >	< Back >		

Select the following configuration options for the Network Operator:

- NFD Node Feature discovery
- SRIOV SRIOV Network Operator
- CR Deploy NIC Cluster Policy CR
- ▶ IPoIB Enable IP over Infiniband CNI

All secondaryNetwork Components - To deploy RDMA interface to the containers.

[X]	nfd.enabled	Deploy Node Feature Discovery
[X]	<pre>sriovNetworkOperator.enabled</pre>	Deploy SR-IOV Network Operator
[X]	deployCR	Deploy NicClusterPolicy custom resource
[]	ofedDriver.deploy	Deploy the NVIDIA MLNX_OFED driver contain
[]	rdmaSharedDevicePlugin.deploy	Deploy RDMA shared device plugin
[]	sriovDevicePlugin.deploy	Deploy SR-IOV Network device plugin
[X]	ipoib.deploy	Deploy IPoIB CNI
[X]	secondaryNetwork.deploy	Deploy CNI Plugins Secondary Network
[X]	<pre>secondaryNetwork.cniPlugins.deploy</pre>	Deploy CNI Plugins Secondary Network
[X]	secondaryNetwork.multus.deploy	Deploy Multus Secondary Network
[X]	<pre>secondaryNetwork.ipamPlugin.deploy</pre>	Deploy IPAM CNI Plugin Secondary Network

If MetalLB is enabled, define the MetalLB IP pool. Allocate a free IP range for MetalLB from the internal network range or as appropriate. Refer to the MetalLB configuration guide for further details.

Configure MetalLB IP Address Pools
Address pools:
<alt><a> to add new line, <alt><d> to remove current line</d></alt></a></alt>
< Add > < Remove >
< Ok > < Back > < Help >

Deploy all the addons.

Which addons do you want to deploy?
[X] Ingress Controller (Nginx)
< Ok > < Back >

#### Select Yes to the Ingress default port

ſ	Do	you	want	to	expose	the	Kubernetes	Ingress	to	the	default	HTTPS	port	443?
	ye	S												
	no													
l							< 0k >	< Ba	ack	>				

Leave the ports as default.

Insert values	of the new Kubernetes cluster	
Ingress HTTP Ingress HTTPS	port 30080 port 30443	
	< Ok > < Back >	

Choose **yes** to install the Permission Manager.

This is only n	eeded if you	want to	have non-roo	t users on the	cluster
yes					
no					
	<	0k >	< Back >		

Select both enabled and default for the local storage path.

nfigure Kubernetes St	torageClass			
torageClass	enabled	default		
EPH	[]	()		
EPH is not available.	rx1	(1)		
o not set default		õ		
ocal path To not set default				

Leave the storage path as default.

Path to store the data /cm/shared/apps/kubernetes/default/var/volum Custom address of the registry (optional) Custom provisioner's image (optional)		Configure local path storage pool for Kubernetes
	es	Path to store the data /cm/shared/apps/kubernetes/o Custom address of the registry (optional) Custom provisioner's image (optional)
< 0k > < Back >		< Ok > < Back >

Choose to save config and deploy.

Summary		
Save config & deploy		
Show config		
Save config		
Save config & exit		
Exit		
	< Back >	

Keep the default file path for the config file and continue.

#### Note

This file contains the configuration for the kubernetes cluster.

					be installed in	ster tonight.
Please specify the filepath.						
/root/cm-kubernetes-setup.co	Inf					
Related the second second						
hin/						
<b>av</b>	77					
\$700/	8					
bon-18.8-sbuntu2284-dgx-os-	6.1.ise 17.36					
cn-docker-setup.conf	1.40					
C 1 Shaw bidden C	1 Resolve swelight	[ ] Show details				
F. J. 2000 1100000 F	7 NERVINE SPECERS	F 1 mon perature				
			< 0k >	e lack a		

BCM will deploy kubernetes to the nodes, once the Kubernetes setup has completed, verify that all the nodes are online.

```
root@bcm10-headnode1:~# kubectl get node -o wide
NAME STATUS ROLES AGE VERSION INTERNAL-IP EXTERNAL-IP OS-IMAGE KERNEL-VERSION
→CONTAINER-RUNTIME
dgx-01 Ready worker 14d v1.30.9 10.184.94.11 <none> Ubuntu 22.04.4 LTS 5.15.0-
→1063-nvidia containerd://1.7.21
dqx-02 Ready worker 14d v1.30.9 10.184.94.12 <none> Ubuntu 22.04.4 LTS 5.15.0-
→1063-nvidia containerd://1.7.21
dqx-03 Ready worker 14d v1.30.9 10.184.94.13 <none> Ubuntu 22.04.4 LTS 5.15.0-
→1063-nvidia containerd://1.7.21
dgx-04 Ready worker 14d v1.30.9 10.184.94.14 <none> Ubuntu 22.04.4 LTS 5.15.0-
→1063-nvidia containerd://1.7.21
knode1 Ready control-plane,master 14d v1.30.9 10.184.94.4 <none> Ubuntu 22.04.
→4 LTS 5.15.0-113-generic containerd://1.7.21
knode2 Ready control-plane,master 14d v1.30.9 10.184.94.5 <none> Ubuntu 22.04.
→4 LTS 5.15.0-113-generic containerd://1.7.21
knode3 Ready control-plane,master 14d v1.30.9 10.184.94.6 <none> Ubuntu 22.04.
→4 LTS 5.15.0-113-generic containerd://1.7.21
```

Validate the Kubernetes cluster by checking that the pods are in the "Running" state and ensuring that both the GPU operator and network operator pods are active.

```
root@bcm10-headnode1:~$ kubectl get pods -A | grep "network\|gpu"
gpu-operator gpu-feature-discovery-714bz 1/1 Running 0 13h
gpu-operator gpu-feature-discovery-bpzzq 1/1 Running 0 13h
- Output removed for brevity -
network-operator cni-plugins-ds-gx97k 1/1 Running 0 13h
```

(continued from previous page)

```
network-operator cni-plugins-ds-hw6sr 1/1 Running 0 13h
network-operator kube-multus-ds-kkbwz 1/1 Running 0 13h
```

### 14.3. Add a kubernetes user

Add a new user named 'k8suser' in BCM

```
root@bcm10-headnode1:~# cmsh
[bcm10-headnode1]% user;list
Name (key) ID (key) Primary group Secondary groups
```

-----

```
cmsupport 1000 cmsupport
[bcm10-headnode1]% user
[bcm10-headnode1->user]% add k8suser
[bcm10-headnode1->user*[k8suser*]]% set password bcm123
[bcm10-headnode1->user*[k8suser*]]% commit
```

Add a new user to Kubernetes:

cm-kubernetes-setup --add-user k8suser

Switch to the new user:

su - k8suser

# Chapter 15. Compute network/ IB Interfaces Configuration

# 15.1. Validate IB/Compute interfaces

Reference: Configuring SR-IOV (InfiniBand) section in OFED user guide

Check Physical link type (LINK\_TYPE\_P1), SRIOV(SRIOV\_EN) state and VF count (NUM\_OF\_VF) configuration for the DGX Compute interfaces.

Physical link should be type 1 for InfiniBand, SRIOV in enabled state with 8 VFs

On the BCM headnode, run cmsh

```
root@bcm10-headnode1:~# cmsh
[bcm10-headnode1]% device
[bcm10-headnode1->device]%
[bcm10-headnode1->device]%pexec -c dgx-h100 -j "for i in dc 9a ce c0 4f 40 5e
→18 ; do mst start; mlxconfig -d $i:00.0 q; done | grep -e \"SRIOV_EN\\|LINK_
→TYPE\\|NUM_OF_VFS\";"
[dgx-01..dgx-04]
NUM_OF_VFS 8
SRIOV_EN True(1)
LINK_TYPE_P1 IB(1)
NUM_OF_VFS 8
SRIOV_EN True(1)
```

(continued from previous page)

```
LINK_TYPE_P1 IB(1)
NUM_OF_VFS 8
SRIOV_EN True(1)
LINK_TYPE_P1 IB(1)
```

Refer to DGX H100 user guide for interface name/PCI address mapping

If SR-IOV is enabled, the interface type is InfiniBand, and eight or more VFs are already configured, proceed to section Configure SR-IOV NetworkNodePolicy CR

If SRIOV/IB/VFs are not configured, enable them using the following command

```
[bcm10-headnode1->device]% pexec -c dgx-h100 -j "for i in dc 9a ce c0 4f 40
→5e 18 ; do mst start; mlxconfig -d $i:00.0 -y set SRIOV_EN=1 NUM_OF_VFS=8
→LINK_TYPE_P1=1 ; done"
Device #1:
Device type: ConnectX7
Name: MCX750500B-0D00_Ax
Description: Nvidia adapter card with four ConnectX-7; each 400Gb/s NDR
IB; PCIe 5.0 x32; PCIe switch; secured boot; No Crypto
Device: 18:00.0
Configurations: Next Boot New
SRIOV_EN True(1) True(1)
NUM_OF_VFS 8 8
LINK_TYPE_P1 IB(1) IB(1)
Apply new Configuration? (y/n) [n] : y
Applying... Done!
-I- Please reboot the machine to load new configurations.
```

Reboot the DGX nodes from BCM to apply the changes.

[bcm10-headnode1->device]% reboot -c dgx-h100

Wait for the nodes to come back up

Configure 8 VFs per IB interface

```
[bcm10-headnode1->device] pexec -c dgx-h100 -j "for i in 0 3 4 5 6 9 10 11;

→do echo 8 > /sys/class/infiniband/mlx5_${i}/device/sriov_numvfs; done"

[dgx-01..dgx-04]
```

Check IB Interface status on the DGX nodes.

```
[bcm10-headnode1->device]% pexec -c dgx-h100 -j "for i in 0 3 4 5 6 9 10 11;
→do ibstat -d mlx5_${i} \| grep -i \\"mlx5_\\|state\\|infiniband\"; done"
[dgx-01..dgx-04]
CA 'mlx5 0'
State: Active
Physical state: LinkUp
Link layer: InfiniBand
CA 'mlx5_3'
State: Active
Physical state: LinkUp
Link layer: InfiniBand
CA 'mlx5_4'
State: Active
Physical state: LinkUp
Link layer: InfiniBand
CA 'mlx5_5'
State: Active
Physical state: LinkUp
Link layer: InfiniBand
CA 'mlx5_6'
State: Active
Physical state: LinkUp
Link layer: InfiniBand
CA 'mlx5_9'
State: Active
Physical state: LinkUp
Link layer: InfiniBand
CA 'mlx5 10'
State: Active
Physical state: LinkUp
Link layer: InfiniBand
CA 'mlx5_11'
State: Active
Physical state: LinkUp
Link layer: InfiniBand
```

All interfaces should be in Active and physical layers in LinkUp state.

Verify VFs under each IB interface PCI device, for e.g. 18:00.0 to 18:00.7 for OSFP4, Port 2, mlx5\_0

```
[bcm10-headnode1->device]% pexec -c dgx-h100 -j "lspci \| grep ConnectX"
[dgx-01..dgx-04]
16:00.0 PCI bridge: Mellanox Technologies MT2910 Family [ConnectX-7 PCIe
Bridge]
17:00.0 PCI bridge: Mellanox Technologies MT2910 Family [ConnectX-7 PCIe
Bridge]
17:02.0 PCI bridge: Mellanox Technologies MT2910 Family [ConnectX-7 PCIe
Bridge]
18:00.0 Infiniband controller: Mellanox Technologies MT2910 Family
[ConnectX-7]
18:00.1 Infiniband controller: Mellanox Technologies ConnectX Family
mlx5Gen Virtual Function
```

18:00.2 Infiniband controller: Mellanox Technologies ConnectX Family
mlx5Gen Virtual Function
18:00.3 Infiniband controller: Mellanox Technologies ConnectX Family
mlx5Gen Virtual Function
18:00.4 Infiniband controller: Mellanox Technologies ConnectX Family
mlx5Gen Virtual Function
18:00.5 Infiniband controller: Mellanox Technologies ConnectX Family
mlx5Gen Virtual Function
18:00.6 Infiniband controller: Mellanox Technologies ConnectX Family
mlx5Gen Virtual Function
18:00.7 Infiniband controller: Mellanox Technologies ConnectX Family
mlx5Gen Virtual Function
18:00.7 Infiniband controller: Mellanox Technologies ConnectX Family
mlx5Gen Virtual Function
18:00.7 Infiniband controller: Mellanox Technologies ConnectX Family
mlx5Gen Virtual Function
18:01.0 Infiniband controller: Mellanox Technologies ConnectX Family
mlx5Gen Virtual Function

# 15.2. Configure SR-IOV NetworkNodePolicy CR

Create a file named 'sriov-ib-network-node-policy.yaml' with the following information:

```
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodePolicy
metadata:
  name: ibp24s0
  namespace: network-operator
spec:
  deviceType: netdevice
  nodeSelector:
    feature.node.kubernetes.io/network-sriov.capable: "true"
  nicSelector:
    vendor: "15b3"
    pfNames: ["ibp24s0"]
  linkType: ib
  isRdma: true
  numVfs: 8
  priority: 90
  resourceName: resibp24s0
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodePolicy
metadata:
  name: ibp64s0
  namespace: network-operator
spec:
  deviceType: netdevice
  nodeSelector:
    feature.node.kubernetes.io/network-sriov.capable: "true"
  nicSelector:
    vendor: "15b3"
    pfNames: ["ibp64s0"]
```

(continues on next page)

(continued from previous page)

(continued from previous page)

```
linkType: ib
  isRdma: true
  numVfs: 8
  priority: 90
  resourceName: resibp64s0
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodePolicy
metadata:
  name: ibp79s0
  namespace: network-operator
spec:
  deviceType: netdevice
  nodeSelector:
    feature.node.kubernetes.io/network-sriov.capable: "true"
  nicSelector:
    vendor: "15b3"
    pfNames: ["ibp79s0"]
  linkType: ib
  isRdma: true
  numVfs: 8
  priority: 90
  resourceName: resibp79s0
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodePolicy
metadata:
  name: ibp94s0
  namespace: network-operator
spec:
  deviceType: netdevice
  nodeSelector:
    feature.node.kubernetes.io/network-sriov.capable: "true"
  nicSelector:
    vendor: "15b3"
    pfNames: ["ibp94s0"]
  linkType: ib
  isRdma: true
  numVfs: 8
  priority: 90
  resourceName: resibp94s0
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodePolicy
metadata:
  name: ibp154s0
  namespace: network-operator
spec:
  deviceType: netdevice
```

(continued from previous page)

```
nodeSelector:
    feature.node.kubernetes.io/network-sriov.capable: "true"
  nicSelector:
    vendor: "15b3"
    pfNames: ["ibp154s0"]
  linkType: ib
  isRdma: true
  numVfs: 8
  priority: 90
  resourceName: resibp154s0
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodePolicy
metadata:
  name: ibp192s0
  namespace: network-operator
spec:
  deviceType: netdevice
  nodeSelector:
    feature.node.kubernetes.io/network-sriov.capable: "true"
  nicSelector:
    vendor: "15b3"
    pfNames: ["ibp192s0"]
  linkType: ib
  isRdma: true
  numVfs: 8
  priority: 90
  resourceName: resibp192s0
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodePolicy
metadata:
  name: ibp206s0
  namespace: network-operator
spec:
  deviceType: netdevice
  nodeSelector:
    feature.node.kubernetes.io/network-sriov.capable: "true"
  nicSelector:
    vendor: "15b3"
    pfNames: ["ibp206s0"]
  linkType: ib
  isRdma: true
  numVfs: 8
  priority: 90
  resourceName: resibp206s0
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodePolicy
```

(continued from previous page)

```
metadata:
  name: ibp220s0
  namespace: network-operator
spec:
  deviceType: netdevice
  nodeSelector:
    feature.node.kubernetes.io/network-sriov.capable: "true"
  nicSelector:
    vendor: "15b3"
    pfNames: ["ibp220s0"]
  linkType: ib
  isRdma: true
  numVfs: 8
  priority: 90
  resourceName: resibp220s0
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodePolicy
metadata:
  name: ibp24s0
  namespace: network-operator
spec:
  deviceType: netdevice
  nodeSelector:
    feature.node.kubernetes.io/network-sriov.capable: "true"
  nicSelector:
    vendor: "15b3"
    pfNames: ["ibp24s0"]
  linkType: ib
  isRdma: true
  numVfs: 8
  priority: 90
  resourceName: resibp24s0
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodePolicy
metadata:
  name: ibp64s0
  namespace: network-operator
spec:
  deviceType: netdevice
  nodeSelector:
    feature.node.kubernetes.io/network-sriov.capable: "true"
  nicSelector:
    vendor: "15b3"
    pfNames: ["ibp64s0"]
  linkType: ib
  isRdma: true
  numVfs: 8
  priority: 90
```

(continued from previous page)

```
resourceName: resibp64s0
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodePolicy
metadata:
  name: ibp79s0
  namespace: network-operator
spec:
  deviceType: netdevice
  nodeSelector:
    feature.node.kubernetes.io/network-sriov.capable: "true"
  nicSelector:
    vendor: "15b3"
    pfNames: ["ibp79s0"]
  linkType: ib
  isRdma: true
  numVfs: 8
  priority: 90
  resourceName: resibp79s0
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodePolicy
metadata:
  name: ibp94s0
  namespace: network-operator
spec:
  deviceType: netdevice
  nodeSelector:
    feature.node.kubernetes.io/network-sriov.capable: "true"
  nicSelector:
    vendor: "15b3"
    pfNames: ["ibp94s0"]
  linkType: ib
  isRdma: true
  numVfs: 8
  priority: 90
  resourceName: resibp94s0
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodePolicy
metadata:
  name: ibp154s0
  namespace: network-operator
spec:
  deviceType: netdevice
  nodeSelector:
    feature.node.kubernetes.io/network-sriov.capable: "true"
  nicSelector:
    vendor: "15b3"
```

(continued from previous page)

```
pfNames: ["ibp154s0"]
  linkType: ib
  isRdma: true
  numVfs: 8
  priority: 90
  resourceName: resibp154s0
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodePolicy
metadata:
  name: ibp192s0
  namespace: network-operator
spec:
  deviceType: netdevice
  nodeSelector:
    feature.node.kubernetes.io/network-sriov.capable: "true"
  nicSelector:
    vendor: "15b3"
    pfNames: ["ibp192s0"]
  linkType: ib
  isRdma: true
  numVfs: 8
  priority: 90
  resourceName: resibp192s0
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodePolicy
metadata:
  name: ibp206s0
  namespace: network-operator
spec:
  deviceType: netdevice
  nodeSelector:
    feature.node.kubernetes.io/network-sriov.capable: "true"
  nicSelector:
    vendor: "15b3"
    pfNames: ["ibp206s0"]
  linkType: ib
  isRdma: true
  numVfs: 8
  priority: 90
  resourceName: resibp206s0
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodePolicy
metadata:
  name: ibp220s0
  namespace: network-operator
spec:
```

(continued from previous page)

```
deviceType: netdevice
nodeSelector:
    feature.node.kubernetes.io/network-sriov.capable: "true"
nicSelector:
    vendor: "15b3"
    pfNames: ["ibp220s0"]
linkType: ib
isRdma: true
numVfs: 8
priority: 90
resourceName: resibp220s0
```

Create the CRD using

kubectl apply -f sriov-ib-network-node-policy.yaml

Create SR-IOV IB Network CR

Create another file named 'sriov-ib-network.yaml' with the following information:

```
____
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovIBNetwork
metadata:
  name: ibp24s0
  namespace: network-operator
spec:
  ipam: |
      "type": "whereabouts",
      "datastore": "kubernetes",
      "kubernetes": {
        "kubeconfig": "/etc/cni/net.d/whereabouts.d/whereabouts.kubeconfig"
      },
      "range": "192.168.1.0/24",
      "log_file": "/var/log/whereabouts.log",
      "log_level": "info"
  resourceName: resibp24s0
  linkState: enable
  networkNamespace: default
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovIBNetwork
metadata:
  name: ibp64s0
  namespace: network-operator
spec:
  ipam: |
      "type": "whereabouts",
      "datastore": "kubernetes",
```
```
"kubernetes": {
        "kubeconfig": "/etc/cni/net.d/whereabouts.d/whereabouts.kubeconfig"
      },
      "range": "192.168.2.0/24",
      "log_file": "/var/log/whereabouts.log",
      "log_level": "info"
  resourceName: resibp64s0
  linkState: enable
  networkNamespace: default
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovIBNetwork
metadata:
  name: ibp79s0
  namespace: network-operator
spec:
  ipam: |
      "type": "whereabouts",
      "datastore": "kubernetes",
      "kubernetes": {
        "kubeconfig": "/etc/cni/net.d/whereabouts.d/whereabouts.kubeconfig"
      },
"range": "192.168.3.0/24",
      "log_file": "/var/log/whereabouts.log",
      "log_level": "info"
  resourceName: resibp79s0
  linkState: enable
  networkNamespace: default
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovIBNetwork
metadata:
  name: ibp94s0
  namespace: network-operator
spec:
  ipam: |
      "type": "whereabouts",
      "datastore": "kubernetes",
      "kubernetes": {
        "kubeconfig": "/etc/cni/net.d/whereabouts.d/whereabouts.kubeconfig"
      },
      "range": "192.168.4.0/24",
      "log_file": "/var/log/whereabouts.log",
      "log_level": "info"
  resourceName: resibp94s0
```

(continued from previous page)

```
linkState: enable
  networkNamespace: default
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovIBNetwork
metadata:
  name: ibp154s0
  namespace: network-operator
spec:
  ipam: |
      "type": "whereabouts",
      "datastore": "kubernetes",
      "kubernetes": {
        "kubeconfig": "/etc/cni/net.d/whereabouts.d/whereabouts.kubeconfig"
      },
"range": "192.168.5.0/24",
      "log_file": "/var/log/whereabouts.log",
      "log_level": "info"
  resourceName: resibp154s0
  linkState: enable
  networkNamespace: default
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovIBNetwork
metadata:
  name: ibp192s0
  namespace: network-operator
spec:
  ipam: |
      "type": "whereabouts",
      "datastore": "kubernetes",
      "kubernetes": {
        "kubeconfig": "/etc/cni/net.d/whereabouts.d/whereabouts.kubeconfig"
      },
      "range": "192.168.6.0/24",
      "log_file": "/var/log/whereabouts.log",
      "log_level": "info"
    }
  resourceName: resibp192s0
  linkState: enable
  networkNamespace: default
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovIBNetwork
metadata:
  name: ibp206s0
```

```
namespace: network-operator
spec:
  ipam: |
      "type": "whereabouts",
      "datastore": "kubernetes",
      "kubernetes": {
        "kubeconfig": "/etc/cni/net.d/whereabouts.d/whereabouts.kubeconfig"
      },
      "range": "192.168.7.0/24",
      "log_file": "/var/log/whereabouts.log",
      "log_level": "info"
  resourceName: resibp206s0
  linkState: enable
  networkNamespace: default
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovIBNetwork
metadata:
  name: ibp220s0
  namespace: network-operator
spec:
  ipam: |
      "type": "whereabouts",
      "datastore": "kubernetes",
      "kubernetes": {
        "kubeconfig": "/etc/cni/net.d/whereabouts.d/whereabouts.kubeconfig"
      },
      "range": "192.168.8.0/24",
      "log_file": "/var/log/whereabouts.log",
      "log_level": "info"
  resourceName: resibp220s0
  linkState: enable
  networkNamespace: default
```

Apply the CRD using

kubectl apply -f sriov-ib-network.yaml

Restart the services to apply the changes:

pdsh -g category=k8s-control-plane service containerd restart pdsh -g category=k8s-control-plane service kubelet restart

# Chapter 16. Validate the GPU status/health

Using cmsh run the following command

```
[bcm10-headnode1->device]% pexec -c dgx-h100 "nvsm show gpus \| grep -e \\
→"GPU.\" -e \\"Health\"'
[dgx-01] :
/systems/localhost/gpus/GPU0
Inventory_UUID = GPU-3a713db1-ff94-3f28-de11-2d6449b8d35f
Stats_UtilGPU = 0%
Status_Health = OK
/systems/localhost/gpus/GPU0/health
Health = OK
/systems/localhost/gpus/GPU1
Inventory_UUID = GPU-bc00e4ef-fb74-10c8-fa6a-a8c243304e14
Stats_UtilGPU = 0%
Status_Health = OK
/systems/localhost/gpus/GPU1/health
Health = OK
/systems/localhost/gpus/GPU2
Inventory_UUID = GPU-4f2e5158-c8be-b50e-2d99-e5380b4a8236
Stats_UtilGPU = 0%
Status_Health = OK
/systems/localhost/gpus/GPU2/health
Health = OK
/systems/localhost/gpus/GPU3
Inventory_UUID = GPU-0a8a7416-4c04-5038-19ca-345db5a1a0ad
Stats_UtilGPU = 0%
Status_Health = OK
/systems/localhost/gpus/GPU3/health
Health = OK
/systems/localhost/gpus/GPU4
Inventory_UUID = GPU-bbd31c7c-7845-d48a-df3b-1523adf86ee6
Stats_UtilGPU = 0%
Status_Health = OK
/systems/localhost/gpus/GPU4/health
Health = OK
/systems/localhost/gpus/GPU5
 output omitted for brevity
```

Ensure all GPUs are healthy.

# Chapter 17. Validate the system topology/NVlink

Using cmsh run the following command

root@bcm10-headnode1:~# cmsh [bcm10-headnode1]%device [bcm10-headnode1]% pexec -c dgx-h100 -j "nvidia-smi topo -m" bcm10-headnode1->device]% pexec -c dgx-h100 -j "nvidia-smi topo -m" [dgx-01..dgx-04] 76 NICs found in the topology, only displaying 56 in the matrix. **GPU0** GPU1 GPU2 GPU3 GPU4 GPU5 GPU6 GPU7 →NIC0 NIC1 NIC2 NIC3 NIC4 NIC5 NIC6 NIC7 NIC8 →NIC9 NIC10 NIC11 NIC12 NIC13 NIC14 NIC15 NIC16 NIC17 →NIC18 NIC19 NIC20 NIC21 NIC22 NIC23 NIC24 NIC25 NIC26 NIC28 NIC29 →NIC27 NIC30 NIC31 NIC32 NIC33 NIC34 NIC35 →NIC36 NIC37 NIC38 NIC39 NIC40 NIC41 NIC42 NIC43 NIC44 NIC48 NIC49 NIC50 NIC51 NIC52 →NIC45 NIC46 NIC47 NIC53 →NIC54 NIC55 CPU Affinitv GPU NUMA ID NUMA Affinity **GPU0** NV18 NV18 NV18 NV18 Х NV18 NV18 NODE NODE →NV18 PXB NODE NODE NODE SYS SYS SYS SYS **PXB** PXB PXB PXB **PXB** PXB PXB **PXB** 4 →NODE NODE NODE NODE NODE NODE NODE NODE NODE →NODE NODE NODE NODE NODE NODE NODE NODE NODE → NODE NODE NODE NODE NODE NODE SYS SYS SYS SYS SYS SYS  $\hookrightarrow$ SYS SYS SYS SYS SYS SYS SYS ⇔SYS 0-55,112-167 0 N/A GPU1 NV18 Х NV18 NV18 NV18 NV18 NV18 →NV18 NODE NODE NODE PXB NODE NODE SYS SYS SYS NODE NODE SYS NODE NODE NODE NODE NODE NODE  $\hookrightarrow$ →PXB PXB PXB PXB PXB PXB PXB PXB NODE →NODE NODE NODE NODE NODE NODE NODE NODE NODE →NODE NODE NODE NODE NODE NODE SYS  $\hookrightarrow$ →SYS 0-55,112-167 0 N/A GPU2 NV18 NV18 NV18 NV18 NV18 NV18 Х NODE NODE NODE NODE **PXB** NODE SYS →NV18 SYS SYS NODE NODE SYS NODE NODE NODE NODE NODE NODE →NODE NODE NODE NODE NODE NODE PXB PXB NODE NODE PXB **PXB** PXB PXB PXB **PXB** NODE NODE NODE \_ →NODE NODE NODE NODE NODE SYS SYS SYS SYS SYS (continues on next page)

(continued from previous page) SYS SYS SYS SYS SYS SYS SYS SYS SYS →0-55,112-167 0 N/A GPU3 NV18 NV18 NV18 Х NV18 NV18 NV18 NODE NODE NODE NODE NODE PXB →NV18 SYS SYS SYS → SYS NODE NODE NODE NODE NODE NODE NODE NODE NODE →NODE NODE NODE NODE NODE NODE NODE NODE →NODE NODE NODE NODE NODE NODE NODE РХВ PXB PXB → PXB PXB PXB PXB PXB SYS SYS SYS SYS SYS →SYS SYS SYS SYS SYS SYS SYS SYS SYS 0-55,112-167 0  $\hookrightarrow$ N/A NV18 NV18 NV18 GPU4 NV18 NV18 NV18 Х SYS →NV18 SYS SYS SYS SYS SYS PXB NODE →NODE NODE SYS SYS SYS SYS SYS SYS SYS SYS → SYS SYS SYS SYS SYS SYS SYS SYS SYS ⇔SYS SYS PXB PXB  $\rightarrow$  SYS SYS SYS SYS PXB PXB PXB PXB PXB NODE NODE NODE ⊶PXB NODE NODE →NODE 56-111,168-223 1 N/A GPU5 NV18 NV18 NV18 NV18 NV18 Х NV18 →NV18 SYS SYS SYS SYS SYS SYS NODE NODE SYS SYS SYS SYS →NODE PXB SYS SYS SYS SYS  $\hookrightarrow$ SYS ⇔SYS SYS SYS SYS SYS → SYS SYS SYS SYS SYS NODE NODE NODE →NODE NODE NODE NODE PXB PXB PXB PXB PXB **PXB** 56-111,168-223 1 N/A  $\hookrightarrow$ GPU6 NV18 NV18 NV18 NV18 NV18 NV18 Х NV18 SYS SYS SYS SYS SYS SYS NODE NODE  $\hookrightarrow$ NODE → NODE SYS SYS SYS SYS SYS SYS SYS SYS SYS ⇔SYS SYS SYS SYS SYS SYS SYS SYS SYS → SYS ⇔SYS NODE NODE NODE NODE →NODE NODE NODE NODE NODE NODE NODE NODE NODE →NODE 56-111,168-223 1 N/A NV18 NV18 NV18 NV18 NV18 NV18 NV18 GPU7 Х → SYS SYS SYS SYS SYS NODE NODE NODE NODE SYS SYS SYS SYS SYS SYS → SYS ⇔SYS SYS NODE NODE NODE NODE NODE ⇔SYS NODE →NODE NODE NODE NODE NODE NODE NODE NODE 56-→111,168-223 1 N/A PXB NODE NODE NODE SYS SYS SYS SYS NICO чХ NODE NODE NODE NODE NODE SYS SYS SYS SYS → PIX PIX PIX PIX PIX PIX PIX PIX NODE NODE →NODE NODE NODE NODE NODE NODE NODE NODE →NODE NODE SYS SYS SYS →NODE NODE NODE SYS SYS → SYS NIC1 NODE NODE NODE NODE SYS SYS SYS NODE →NODE PIX NODE NODE SYS SYS SYS Х SYS NODE NODE NODE NODE NODE NODE NODE NODE NODE (continues on next page)

							(continu	ed from prev	rious page)
→NODE	NODE	NODE							
→NODE	NODE	NODE							
→NODE	NODE	NODE	NODE	NODE	SYS	SYS	SYS	SYS	SYS
لا →	(S								

Verify the NVLink (NV18) status for GPU to GPU interconnect

Reference: Nvidia System Management Interface

# Chapter 18. Validate the GPU/RDMA access within the container

#### 18.1. Validate GPU access from container

Create a file named 'gpu-test.yaml' with the following information

With the Job file created we can now get ready to execute the job by loading the Kubernetes Module using the following command.

module load kubernetes

Now that the environment module is loaded we're finally ready to run the example job:

kubectl apply -f gpu-test.yaml

Next we'll need to monitor the progress of the job using the following command:

```
k8suser@bcm10-headnode1:~$kubectl get podsNAMEREADY STATUSRESTARTS AGEnvidia-smi-test 1/1Running 08m20s
```

Once the job has finished, verify the results using kubectl logs:

The log file should list all the 8 GPUs in the node where the job was executed.

k8suser@bcm10-headnode1:~\$ kubectl logs nvidia-smi-test Wed Feb 12 23:49:34 2025 \_\_\_\_\_ +------\_\_\_\_\_+ Driver Version: 550.90.07 CUDA | NVIDIA-SMI 550.90.07 →Version: 12.4 | <u>\_\_\_\_\_</u>+ Persistence-M | Bus-Id | GPU Name Disp.A | Volatile →Uncorr. ECC | | Fan Temp Perf Pwr:Usage/Cap | Memory-Usage | GPU-Util → Compute M. | → MIG M. →========| 0 NVIDIA H100 80GB HBM3 On | 00000000:1B:00.0 Off | ↔ 0 | | N/A 27C P0 69W / 700W | 1MiB / 81559MiB | 0% → Default | → Disabled | <u>\_\_\_\_+</u> | 1 NVIDIA H100 80GB HBM3 On | 00000000:43:00.0 Off | ↔ 0 | N/A 28C P0 70W / 700W | 1MiB / 81559MiB | 0% → Default | → Disabled | +-----+ ----+ 2 NVIDIA H100 80GB HBM3 On | 00000000:52:00.0 Off | ↔ 0 | N/A 31C P0 69W / 700W | 1MiB / 81559MiB | 0% → Default | → Disabled | ....+ 3 NVIDIA H100 80GB HBM3 On | 00000000:61:00.0 Off | ↔ Ø | | N/A 29C P0 72W / 700W | 1MiB / 81559MiB | 0% → Default | → Disabled | ----+ 4 NVIDIA H100 80GB HBM3 On | 00000000:9D:00.0 Off | → 0 | | N/A 28C P0 69W / 700W | 1MiB / 81559MiB | 0% → Default | (continues on next page)

(continued from previous page)

⊶ Disabled					· · · · ·		1 3 4
++   5 NVIDIA H100 80GB HBM3 → 0     N/A 26C P0 → Default   	70W	/	0n 700W		00000000:C3:00.0 Off 1MiB / 81559MiB	   	0%
→ Disabled   +				-+-		.+	
→+   6 NVIDIA H100 80GB HBM3 → 0			0n		00000000:D1:00.0 Off		
N/A 29C P0 → Default	69W	/	700W		1MiB / 81559MiB		0%
→ Disabled				I		I	
++				-+-		+	
7 NVIDIA H100 80GB HBM3 → 0			0n	Ι	00000000:DF:00.0 Off		
N/A 29C P0 → Default	68W	/	700W	Ι	1MiB / 81559MiB		0%
 ⊶ Disabled							
++				-+-		+	

Reference: Nvidia System Management Interface

# 18.2. Validate the RDMA Network from the container

Create a test file named 'ib-network-validation.yaml' with the following information:

```
- -c
  - sleep inf
securityContext:
 capabilities:
    add: ["IPC_LOCK"]
resources:
  requests:
      nvidia.com/resibp192s0: "1"
      nvidia.com/resibp206s0: "1"
      nvidia.com/resibp154s0: "1"
      nvidia.com/resibp220s0: "1"
      nvidia.com/resibp24s0: "1"
      nvidia.com/resibp64s0: "1"
      nvidia.com/resibp79s0: "1"
      nvidia.com/resibp94s0: "1"
  limits:
    nvidia.com/resibp192s0: "1"
    nvidia.com/resibp206s0: "1"
    nvidia.com/resibp154s0: "1"
    nvidia.com/resibp220s0: "1"
    nvidia.com/resibp24s0: "1"
    nvidia.com/resibp64s0: "1"
    nvidia.com/resibp79s0: "1"
    nvidia.com/resibp94s0: "1"
```

Apply the 'ib-network-validation.yaml' file:

```
kubectl apply -f ib-network-validation.yaml
```

Confirm the pod is in running state

```
k8suser@bcm10-headnode1:~$ k get pods
NAME READY STATUS RESTARTS AGE
network-validation-pod 1/1 Running 0 5s
```

Verify the InfiniBand/RDMA interfaces are available in the container

```
root@bcm10-headnode1:~# kubectl exec -it network-validation-pod --
/usr/sbin/ibdev2netdev
mlx5_14 port 1 ==> net5 (Up)
mlx5_25 port 1 ==> net6 (Up)
mlx5_33 port 1 ==> net7 (Up)
mlx5_43 port 1 ==> net8 (Up)
mlx5_49 port 1 ==> net3 (Up)
mlx5_58 port 1 ==> net1 (Up)
mlx5_61 port 1 ==> net2 (Up)
mlx5_70 port 1 ==> net4 (Up)
```

Delete the test pod

root@bcm10-headnode1:~# kubectl delete pod network-validation-pod

```
pod "network-validation-pod" deleted
```

# Chapter 19. Validate the node level NCCL test with 8 GPUs

Create a yaml named nccl-local.yaml with the following content

```
apiVersion: v1
kind: Pod
metadata:
    name: nccl-local-test
spec:
    restartPolicy: Never
    containers:
        - name: nvidia-smi
        image: docker.io/deepops/nccl-tests:2312
        command: ["/bin/bash", "-c", "nvidia-smi && tail -f /dev/null"]
        resources:
        limits:
            nvidia.com/gpu: 8 # Request 8 GPUs
nodeSelector:
        nvidia.com/gpu.present: "true" # Ensure it runs on a node with a GPU
```

Start the container with

kubectl apply -f nccl-local.yaml

Verify the container is running

```
k8suser@bcm10-headnode1:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
nccl-local-test 1/1 Running 0 17s
```

Login to the container and run NCCL broadcast perf with 8 GPUs

**Reference: NCCL primitives** 

Reference: NCCL Tests

					(cont	inued from p	revious page)
→H100 80GB HBM3							
# Rank 1 Group →H100 80GB HBM3	0 Pid	323 on ncc	l-local-test	device	1 [0	x43] NVI	DIA
# Rank 2 Group →H100 80GB HBM3	0 Pid	323 on ncc	l-local-test	device	2 [0	x52] NVI	DIA
# Rank 3 Group	0 Pid	323 on ncc	l-local-test	device	3 [0	x61] NVI	DIA
# Rank 4 Group	0 Pid	323 on ncc	l-local-test	device	4 [0	x9d] NVI	DIA
# Rank 5 Group →H100 80GB HBM3	0 Pid	323 on ncc	l-local-test	device	5 [0	xc3] NVI	DIA
# Rank 6 Group	0 Pid	323 on ncc	l-local-test	device	6 [0	xd1] NVI	DIA
# Rank 7 Group	0 Pid	323 on ncc	l-local-test	device	7 [0	xdf] NVI	DIA
#							
#						out-of-	place
$\hookrightarrow$	<b>in</b> -p	Lace					
# size	count	type	redop r	oot	time	algbw	busbw
→#wrong time	algbw	busbw #wro	ong				
# (B) (e	elements)				(us)	(GB/s)	(GB/s)
↔ (us)	(GB/s)	(GB/s)		-			
65536	16384	float	none	0	28.77	2.28	2.28
→ 0 27.55	2.38	2.38	0				
131072	32768	float	none	0	30.42	4.31	4.31
→ 0 30.28	4.33	4.33	0		0 = 04	- 10	- 10
262144	65536	float	none	0	35.31	7.42	7.42
→ 0 33.20	7.90	7.90	Ø	0	40.05	10 17	10 47
524288	131072	TLOat	none	0	42.05	12.4/	12.47
↔ U 43.83	11.90	11.90 floot	0	0	E2 00	10 10	10 40
10400/0	202144	20 02	none	0	53.98	19.42	19.42
→ Ø JZ.30	20.02	20.02	0	0	E4 20	20 62	20 62
2097132	20 70	20 70	none	0	34.29	30.03	30.03
→ 0 54.07 /10/30/	1048576	float	none	A	56 67	74 01	74 01
Ω 9 55 61	75 43	75 43	A	0	50.07	74.01	74.01
8388608	2097152	float	none	A	60 71	138 17	138 17
→ Ø 62.26	134.74	134.74	0	Ũ	00171	100117	100117
16777216	4194304	float	none	0	82.29	203.88	203.88
→ 0 82.32	203.80	203.80	0	-			
33554432	8388608	float	none	0	130.4	257.30	257.30
→ 0 130.1	257.98	257.98	0				
67108864	16777216	float	none	0	225.7	297.33	297.33
→ 0 226.5	296.35	296.35	0				
134217728	33554432	float	none	0	411.5	326.17	326.17
→ 0 412.5	325.41	325.41	0				
268435456	67108864	float	none	0	780.8	343.80	343.80
→ 0 783.4	342.64	342.64	0				
536870912	134217728	float	none	0 1	508.3	355.95	355.95
→ 0 1513.4	354.74	354.74	0				
1073741824	268435456	float	none	0 2	975.8	360.82	360.82
↔ Ø 2977.4	360.64	360.64	0				

```
(continued from previous page)
  2147483648
                                                       0
                                                            5896.4 364.20 364.20
                  536870912
                                 float
                                           none
       0
            5911.1 363.29 363.29
                                          0
\hookrightarrow
# Out of bounds values : 0 OK
# Avg bus bandwidth
                      : 175.205
#
root@nccl-local-test:/workspace# exit
exit
```

## Chapter 20. Validate the cluster level NCCL test with 4 nodes and 32 GPUs

Create a test file named 'nccl-test.yaml' with the contents shown below. In this example we are running NCCLtest across 4 DGX nodes, over a total of 32 GPUs, so "-np" is set to "4" in the mpirun command.

```
apiVersion: kubeflow.org/v2beta1
kind: MPIJob
metadata:
  name: nccltest
spec:
  slotsPerWorker: 1
  runPolicv:
    cleanPodPolicy: Running
  mpiReplicaSpecs:
    Launcher:
      replicas: 1
      template:
         spec:
           imagePullSecrets:
           - name: ngc-registry-default
           containers:
           - image: docker.io/deepops/nccl-tests:2312
             name: nccltest
             imagePullPolicy: IfNotPresent
             command:
             - sh
             - "-c"
             - |
               /bin/bash << 'EOF'</pre>
               mpirun --allow-run-as-root \
                 -np 4 \
                 -bind-to none -map-by slot \
                 -mca pml ob1 \
                 -mca btl ^openib \
                 -mca btl_tcp_if_include 192.168.0.0/16 \
                 -mca oob_tcp_if_include 172.29.0.0/16 \
                 all_reduce_perf_mpi -b 8 -e 16G -f2 -g 8 \
                 && sleep infinity
```

	(*************************************
EOF	
Worker:	
replicas: 4	
template:	
metadata:	
annotations:	
k8s.v1.cni.cncf.io/networks: ibp192s0,ibp206s0,	ibp154s0,ibp220s0,
→ibp24s0,ibp64s0,ibp79s0,ibp94s0	
spec:	
imagePullSecrets:	
- <b>name:</b> ngc-registry-default	
containers:	
- image: docker.io/deepops/nccl-tests:2312	
name: nccltest	
<pre>imagePullPolicy: IfNotPresent</pre>	
securitvContext:	
capabilities:	
add: [ "IPC_LOCK" ]	
resources:	
requests:	
nvidia.com/resibp192s0: "1"	
nvidia.com/resibp206s0: "1"	
nvidia.com/resibp154s0: "1"	
nvidia.com/resibp220s0: "1"	
nvidia.com/resibp24s0: "1"	
nvidia.com/resibp64s0: "1"	
nvidia.com/resibp79s0: "1"	
nvidia.com/resibp94s0: "1"	
nvidia.com/gpu: 8	
limits:	
nvidia.com/resibp192s0: "1"	
nvidia.com/resibp206s0: "1"	
nvidia.com/resibp154s0: "1"	
nvidia.com/resibp220s0: "1"	
nvidia.com/resibp24s0: "1"	
nvidia.com/resibp64s0: "1"	
nvidia.com/resibp79s0: "1"	
nvidia.com/resibp94s0: "1"	
nvidia.com/gpu: 8	

Run the 'nccl-test.yaml' file:

kubectl apply -f nccl-test.yaml

Monitor the progress of the job: Wait till the pods are "Running"

```
kubectl get pods
root@bcm10-headnode1:~# k get pods
NAME READY STATUS RESTARTS AGE
nccltest-launcher-8znll 1/1 Running 3 (54s ago) 87s
nccltest-worker-0 1/1 Running 0 87s
nccltest-worker-1 1/1 Running 0 87s
```

```
nccltest-worker-2 1/1 Running 0 87s
nccltest-worker-3 1/1 Running 0 87s
```

Once all are in Running state, (takes ~90 seconds), verify the results:

kubectl logs nccltest-launcher-NNNNN

For example

<mark>root@bcm10-headnode1:</mark> ~#kubectl #			tl	logs -f nccltest-launcher-8znll							
# # Uning doutoon											
# USING devices # Rank Ø Group	Q	Pid	43	on	ncclte	st-work	or-0	device	Q	[Øv1h]	Νντστα
→H100 80GB HBM3	0	I IU	40	UII	HCCIC	SL WOIN		device	0	[OVID]	NVIDIA
# Rank 1 Group	0	Pid	43	on	ncclte	est-worke	er-0	device	1	[0x43]	NVIDIA
⊸H100 80GB HBM3											
# Rank 2 Group	0	Pid	43	on	ncclte	est-worke	er-0	device	2	[0x52]	NVIDIA
→H100 80GB HBM3	_				_		_		_		
# Rank 3 Group	0	Pid	43	on	ncclte	est-worke	er-0	device	3	[0x61]	NVIDIA
→HI00 80GB HBM3	0	Did	10	<u>_</u>	noo]+/	ot work	or 0	davitaa	4		
	0	PIU	43	011	nccite	est-worke	er-0	device	4	[0x90]	NVIDIA
# Rank 5 Group	0	Pid	43	on	ncclte	est-worke	er-0	device	5	[0xc3]	NVIDIA
→H100 80GB HBM3				•							
# Rank 6 Group	0	Pid	43	on	ncclte	est-worke	er-0	device	6	[0xd1]	NVIDIA
→H100 80GB HBM3											
# Rank 7 Group	0	Pid	43	on	ncclte	est-worke	er-0	device	7	[0xdf]	NVIDIA
→H100 80GB HBM3	~	D: 1	40						~		
# Rank 8 Group	Ø	Pid	43	on	nccite	est-worke	er-1	device	0	[0x1b]	NVIDIA
$\Rightarrow \Pi 100 0000 \Pi DM3$	Q	Did	13	on	ncclta	set-work	or_1	dovico	1	[0×13]	ΝΥΤΟΤΛ
→H100 80GB HBM3	0	110	-0	011	neeree	JOC WOLK		ucvicc		[0740]	NULDIA
# Rank 10 Group	0	Pid	43	on	ncclte	est-worke	er-1	device	2	[0x52]	NVIDIA
⊸H100 80GB HBM3											
<pre># Rank 11 Group</pre>	0	Pid	43	on	ncclte	est-worke	er-1	device	3	[0x61]	NVIDIA
→H100 80GB HBM3	_				_						
# Rank 12 Group	0	Pid	43	on	nccite	est-worke	er-1	device	4	[0x9d]	NVIDIA
→HIUU 80GB HBM3	0	Did	12	on	noolta	ot-work	or_1	dovico	5	[0x02]	
- H100 80GB HBM3	0	FIU	43	011	HUUTU	SL-WUIK	ei - i	device	5	[ 0XC2 ]	NVIDIA
# Rank 14 Group	0	Pid	43	on	ncclte	est-worke	er-1	device	6	[0xd1]	NVIDIA
⊸H100 80GB HBM3											
<pre># Rank 15 Group</pre>	0	Pid	43	on	ncclte	est-worke	er-1	device	7	[0xdf]	NVIDIA
→H100 80GB HBM3					_	_					
# Rank 16 Group	0	Pid	43	on	ncclte	est-worke	er-2	device	0	[0x1b]	NVIDIA
→H100 80GB HBM3	0	Did	10	<b>~ ~</b>	noo]+/	ot work	or 0	davitaa	1	[0,42]	
	0	PIU	43	011	nccite	est-worke	er-z	device	1	[0X43]	NVIDIA
# Rank 18 Group	Ø	Pid	43	on	ncclte	est-work	er-2	device	2	[0x52]	NVTDTA
→H100 80GB HBM3	0	. 14	.0	011				20,100	-	[ 0//02 ]	
# Rank 19 Group	0	Pid	43	on	ncclte	est-work	er-2	device	3	[0x61]	NVIDIA
→H100 80GB HBM3											

•
<u>(</u>
ţ

										(CON	unueu non	i previous page
# Rank 20 Group	0	Pid	43	on	ncclt	es	t-wor	ker-2	device	. 4	[0x9d]	NVIDIA
# Rank 21 Group	0	Pid	43	on	ncclt	es	t-wor	ker-2	device	e 5	[0xc3]	NVIDIA
$\Rightarrow H100 80GB HBM3$ # Rank 22 Group	0	Pid	43	on	ncclt	es	t-wor	ker-2	device	6	[0xd1]	NVIDIA
→H100 80GB HBM3 # Rank 23 Group	0	Pid	43	on	ncclt	es	t-wor	ker-2	device	. 7	[0xdf]	NVIDIA
→H100 80GB HBM3											_	
# Rank 24 Group →H100 80GB HBM3	0	Pid	43	on	ncclt	es	t-wor	ker-3	device	9	[0x1b]	NVIDIA
# Rank 25 Group →H100 80GB HBM3	0	Pid	43	on	ncclt	es	t-wor	ker-3	device	. 1	[0x43]	NVIDIA
# Rank 26 Group	0	Pid	43	on	ncclt	es	t-wor	ker-3	device	2	[0x52]	NVIDIA
# Rank 27 Group	0	Pid	43	on	ncclt	es	t-wor	ker-3	device	9 3	[0x61]	NVIDIA
$\Rightarrow$ HI00 800B HBM3 # Papk 28 Group	Q	Did	13	on	ncclt	- 0 0	t-wor	kor-3	dovice	1	[0v0d]	
→H100 80GB HBM3	0	r Iu	43	UII	HUULI	.03		Kel-3	device	4	[0x9u]	NVIDIA
# Rank 29 Group	0	Pid	43	on	ncclt	es	t-wor	ker-3	device	. 5	[0xc3]	NVIDIA
→H100 80GB HBM3												
# Rank 30 Group →H100 80GB HBM3	0	Pid	43	on	ncclt	es	t-wor	ker-3	device	9 6	[0xd1]	NVIDIA
# Rank 31 Group →H100 80GB HBM3	0	Pid	43	on	ncclt	es	t-wor	ker-3	device	e 7	[0xdf]	NVIDIA
#												
#		•									out-of	f-place
⇔ # oizo		<b>1n</b> -p]	lace	+.			dan	root		ima	alahu	v buobw
# Size		alabw	hus	ty thw t	pe #wron	re 7	aop	FOOL	. l	лше	атдри	v busbw
# (B) (	ele	ements)	543		rwi ong	9			(	us)	(GB/s)	(GB/s)
		(GB/s)	(GB)	/s)					(	,	(0270)	(02,0)
8		2		flo	at		sum	-1	22	2.7	0.00	0.00
→ 0 52.78		0.00	0	.00		0						
16 0 45 37		4 0 00	Q	flo 00	at	Q	sum	-1	45	5.53	0.00	0.00
32		8	0	flo	at	0	sum	-1	43	.96	0.00	0.00
→ 0 46.49 64		0.00	0	.00 flo	at	0	sum	-1	44	. 13	0.00	9 9.99
→ 0 45.86		0.00	0	.00	ac	0	oum				0.00	0.00
128		32	Q	flo 01	at	0	sum	-1	44	.73	0.00	0.01
→ 0 44.28 256		64	0	flo	at	0	sum	-1	66	.13	0.00	0.01
→ 0 43.62 512		0.01 128	0	.01 flo	at	0	sum	-1	51	. 52	0.0	1 0.02
→ 0 46.60		0.01	0	.02		0			5.0	10	0.00	0.04
0 48.78		256	Ø	тто .04	ατ	Ø	sum	-	56	1.13	0.02	2 0.04
2048		512	Ĵ	flo	at		sum	-1	52	.50	0.04	4 0.08
→ 0 48.21		0.04	0	.08		0					_	
4096		1024	0	†10	at	0	sum	-1	50	.20	0.08	8 0.16
→ 0 50.25 8192		2048	0	flo	at	0	sum	-1	56	. 57	0.14	4 0.28

							(cont	inued from p	revious page)
$\hookrightarrow$	0 50.76	0.16	0.31	0					
	16384	4096	float		sum	-1	56.42	0.29	0.56
$\hookrightarrow$	0 53.13	0.31	0.60	0					
	32768	8192	float	-	sum	-1	55.20	0.59	1.15
$\hookrightarrow$	0 53.74	0.61	1.18	0		_			1 00
	65536	16384	float	~	sum	-1	69.90	0.94	1.82
$\hookrightarrow$	0 68.49	0.96	1.85	Ø		1	60 50	0.06	2 00
	131072	32/08	2 04	Q	sum	- 1	63.59	2.00	3.99
$\hookrightarrow$	0 03.04 2621 <i>44</i>	65536	5.04 float	0	siim	-1	74 29	3 53	6 84
<i></i>	A 77 A9	3 40	6 59	ß	Sum		74.27	0.00	0.04
	524288	131072	float	Ŭ	sum	-1	78.86	6.65	12.88
$\hookrightarrow$	0 82.36	6.37	12.33	0					
	1048576	262144	float		sum	-1	91.09	11.51	22.30
$\hookrightarrow$	0 88.45	11.85	22.97	0					
	2097152	524288	float		sum	-1	159.7	13.13	25.44
$\hookrightarrow$	0 139.5	15.03	29.13	0					
	4194304	1048576	float		sum	-1	151.1	27.75	53.77
$\hookrightarrow$	0 159.6	26.28	50.91	0					
	8388608	2097152	float	0	sum	-1	197.8	42.41	82.17
$\hookrightarrow$	0 203.7	41.19	/9.80	0		1	054 0	(F 01	107 50
	10///210	4194304	T10at	0	sum	- 1	254.9	05.81	127.50
$\hookrightarrow$	22551122	00.09	127.00 float	0	cum	_1	364 0	02 10	179 61
	0 462 5	72 56	140 58	ß	Sum		304.0	92.19	170.01
$\rightarrow$	67108864	16777216	float	0	sum	-1	562.0	119.41	231.35
$\hookrightarrow$	0 550.7	121.87	236.12	0			00200		
	134217728	33554432	float		sum	-1	982.6	136.59	264.64
$\hookrightarrow$	0 969.6	138.42	268.19	0					
	268435456	67108864	float		sum	-1	1842.7	145.67	282.24
$\hookrightarrow$	0 1874.8	143.18	277.41	0					
	536870912 1	34217728	float		sum	-1	3578.6	150.02	290.67
$\hookrightarrow$	0 3580.5	149.94	290.52	0		_	7404 0		
1	0/3/41824 2	150 00	float	~	sum	-1	/421.0	144.69	280.34
⇔ າ	0 /059.9	152.09	294.67	0		1	14051	150 04	206 12
2	.147403040 3 0 17061	152 72	205 00	Q	Suili	- 1	14051	152.04	290.12
∽ ⊿	294967296 14001	132.72	float	0	ciim	-1	28187	152 37	295 22
-	0 28145	152 60	295 67	ß	Sum		20107	102.07	290.22
8	589934592 21	47483648	float	Ŭ	sum	-1	56366	152.39	295.26
$\hookrightarrow$	0 56564	151.86	294.23	0					
17	179869184 42	94967296	float		sum	-1	112687	152.46	295.38
$\hookrightarrow$	0 112808	152.29	295.07	0					
<b>#</b> 0	out of bounds va	lues : 0	OK						
<b>#</b> A	vg bus bandwidt	:h :9₄	4.9049						

## Chapter 21. Site Survey

## 21.1. Sample Site Survey

General Information	
Country Name	US
State/Province	California
Locality	Santa Clara
Organization Name	Example Org
Administrator Email	admin@example.org
Organizational Unit	Demo
Cluster Name	ExampleCluster
Head Node Shared IP (HA Virtual IP)	10.184.94.251
Add Failover Network?	
NFS Server IP	10.160.0.4
NAS Path to /cm/shared	/nfs/data/nas/cmshared
NAS Path to /home	/nfs/data/nas/home
Timezone	US/Los_Angeles
Network Topology	Туре 2
IP Offset (Compute Nodes)	0.0.0.3
Partition Type	One Big Partition
OFED Stack Version	Mellanox OFED 23.10
OOB Management BMC Username	root
BCM Head Node Admin Username	root
BCM Head Node Admin Password	

Network Information											
DGX BasePOD RA Name	BCM Network Name	Network Address (Base IP Address)	Netmask (/Net- maskbits)	Gate- way							
Compute Fabric	computenet	100.64.0.0	255.255.0.0 (/16)	-							
Management & Stor- age Fabric	managementnet (in- ternalnet)	10.184.94.0	255.255.255.0 (/24)	10.184.94							
OOB Management Fabric	oobmanagementnet (ipminet)	10.160.6.0	255.255.255.0 (/24)	10.160.6.1							
IB Storage Fabric	storagenet	-	-								
Name Servers	Search Domains	Time Servers									
8.8.8.8	example.org	time.nist.gov									

BCM Head Node Information Managementnet											
Name/Uni ID	Hostname	BMC IP manageme net)	(oob- ent-	BMC Creden- tia	Node IP (man- agementnet)	MAC 1 (enp37s0np0	MAC 2 (enp226s0np0				
Head 1	bcm10- headnode1	10.160.6.2	54		10.184.94.254	E8:EB:D3:09:2	E8:EB:D3:09:2				
Head2	bcm10- headnode2	10.160.6.2	53		10.184.94.253	E8:EB:D3:09:2	E8:EB:D3:09:2				

DGX Node	Informa	tion (1)		Managementnet				
Name/Unic ID	Host- name	BMC IP (oobman- agementnet)	BMC Cre- dentials	Node IP (man- agementnet)	MAC 1 (enp37s0np0)	MAC 2 (enp226s0np0	)	
DGX-01	dgx- 01	10.160.6.31		10.184.94.11	94:6D:AE:AA:1	94:6D:AE:AA:1	4:89	
DGX-02	dgx- 02	10.160.6.32		10.184.94.12	A0:88:C2:A3:4	A0:88:C2:A3:4	B:05	
DGX-03	dgx- 03	10.160.6.33		10.184.94.13	94:6D:AE:1C:8	94:6D:AE:1C:8	0:7D	
DGX-04	dgx- 04	10.160.6.34		10.184.94.14	A0:88:C2:04:7	A0:88:C2:04:6	0:E1	

DGX Node Information (2)						computenet			
Name/Unique ID	Host- name	ibp220s(	ibp154s(	ibp206s(	ibp192s(	ibp79s0	ibp64s0	ibp94s0	ibp24s0
DGX-01	dgx-01	100.64.0	100.64.1	100.64.2	100.64.3	100.64.4	100.64.5	100.64.6	100.64.7.1
DGX-02	dgx-02	100.64.0	100.64.1	100.64.2	100.64.3	100.64.4	100.64.5	100.64.6	100.64.7.2
DGX-03	dgx-03	100.64.0	100.64.1	100.64.2	100.64.3	100.64.4	100.64.5	100.64.6	100.64.7.3
DGX-04	dgx-04	100.64.0	100.64.1	100.64.2	100.64.3	100.64.4	100.64.5	100.64.6	100.64.7.4

Kubernete	s Node Inf	ormation	ManagementNet			
Name/Unic ID	Host- name	BMC IP (oobman- agementnet)	BMC Cre- dentials	Node IP (man- agementnet)	MAC 1 (enp37s0np0	MAC 2 (enp226s0npC
Knode1	k8s- control- 01	10.160.6.4		10.184.94.4	10:70:FD:73:7	10:70:FD:73:71
Knode2	k8s- control- 02	10.160.6.5		10.184.94.5	10:70:FD:73:7	10:70:FD:73:70
Knode3	k8s- control- 03	10.160.6.6		10.184.94.6	E8:EB:D3:09:2	B8:CE:F6:63:EI

## 21.2. Blank Site Survey

General Information						
Country Name						
State/Province						
Locality						
Organization Name						
Administrator Email						
Organizational Unit						
Cluster Name						
Head Node Shared IP (HA Virtual IP)						
Add Failover Network?						
NFS Server IP						
NAS Path to /cm/shared						
NAS Path to /home						
Timezone						
Network Topology						
IP Offset (Compute Nodes)						
Partition Type						
OFED Stack Version						
OOB Management BMC Username						
OOB Management BMC Password						

Network Information					
DGX BasePOD RA Name	BCM Network Name	Network Ad- dress	Netmask maskbits)	(/Net-	Gate- way
Compute Fabric	computenet				
Management & Storage Fabric	managementnet (inter- nalnet)				
OOB Management Fab- ric	oobmanagementnet (ip- minet)				
IB Storage Fabric	storagenet				
Name Servers	Search Domains	Time Servers			

BCM Head Node Information							
Name/Uniq ID	Host- name	BMC IP (oobman- agementnet)	BMC Cre- dentials	Node IP (man- agementnet)	MAC 1 (enp37s0np0)	MAC 2 (enp226s0np0)	
Head 1							
Head2							

DGX Node Information (1)   Managementnet						
Name/Uniq ID	Host- name	BMC IP (oobman- agementnet)	BMC Cre- dentials	Node IP (man- agementnet)	MAC 1 (enp37s0np0)	MAC 2 (enp226s0np0
DGX-01						
DGX-02						
DGX-03						
DGX-04						

DGX Node Information (2)									
Name/Unique ID	Host- name	ibp220s(	ibp154s(	ibp206s(	ibp192s(	ibp79sC	ibp64sC	ibp94sC	ibp24s0
DGX-01									
DGX-02									
DGX-03									
DGX-04									

Kubernetes Node Information					
Name/Unique ID	Host- name	BMC IP (oobmanage- mentnet)	BMC Cre- dentials	Node IP (manage- mentnet)	MAC 1 (enp37s0np0)
Knode1					
Knode2					
Knode3					

## Chapter 22. Switch Configurations

### 22.1. SN2201 (oobmanagementnet) Switch Configuration

```
nv set bridge domain br_default vlan 101
nv set interface bond1 bond member swp49
nv set interface bond1 bond member swp50
nv set interface bond1 bridge domain br_default untagged 1
nv set interface bond1 bridge domain br_default vlan all
nv set interface bond1 type bond
nv set interface eth0 ip address dhcp
nv set interface eth0 ip vrf mgmt
nv set interface eth0 type eth
nv set interface swp1-48 bridge domain br_default access 101
nv set interface swp1-48 description 'BMC Ports'
nv set interface swp1-50 link state up
nv set interface swp1-50 type swp
nv set service ntp mgmt server 0.cumulusnetworks.pool.ntp.org
nv set service ntp mgmt server 1.cumulusnetworks.pool.ntp.org
nv set service ntp mgmt server 2.cumulusnetworks.pool.ntp.org
nv set service ntp mgmt server 3.cumulusnetworks.pool.ntp.org
nv set system aaa class nvapply action allow
nv set system aaa class nvapply command-path / permission all
nv set system aaa class nvshow action allow
nv set system aaa class nvshow command-path / permission ro
nv set system aaa class sudo action allow
nv set system aaa class sudo command-path / permission all
nv set system aaa role nvue-admin class nvapply
nv set system aaa role nvue-monitor class nvshow
nv set system aaa role system-admin class nvapply
nv set system aaa role system-admin class sudo
nv set system aaa user cumulus full-name cumulus,,,
nv set system aaa user cumulus hashed-password '*'
nv set system aaa user cumulus role system-admin
nv set system api state enabled
nv set system config auto-save state enabled
nv set system control-plane acl acl-default-dos inbound
nv set system control-plane acl acl-default-whitelist inbound
nv set system hostname IPMI-Basepod-01
```

nv	set	system	reboot mode cold
nv	set	system	<pre>ssh-server permit-root-login enabled</pre>
nv	set	system	ssh-server state enabled
nv	set	system	ssh-server vrf mgmt
nv	set	system	timezone America/Los_Angeles
nv	set	system	wjh channel forwarding trigger 12
nv	set	system	wjh channel forwarding trigger 13
nv	set	system	wjh channel forwarding trigger tunnel
nv	set	system	wjh enable on

### 22.2. SN4600C-1 (managementnet) Switches Configuration

#### 22.2.1. SN4600C-1 Configuration

nv	set	bridge dom	nain br_default vlan 100-102
nv	set	interface	bond1 bond member swp1
nv	set	interface	bond1 bond mlag id 1
nv	set	interface	bond1-11,13-48,51 bond lacp-bypass on
nv	set	interface	<pre>bond1-48 bridge domain br_default access 102</pre>
nv	set	interface	bond1-48,51 bond mlag enable on
nv	set	interface	bond1-48,51 type bond
nv	set	interface	bond2 bond member swp2
nv	set	interface	bond2 bond mlag id 2
nv	set	interface	bond3 bond member swp3
nv	set	interface	bond3 bond mlag id 3
nv	set	interface	bond4 bond member swp4
nv	set	interface	bond4 bond mlag id 4
nv	set	interface	bond5 bond member swp5
nv	set	interface	bond5 bond mlag id 5
nv	set	interface	bond6 bond member swp6
nv	set	interface	bond6 bond mlag id 6
nv	set	interface	bond7 bond member swp7
nv	set	interface	bond7 bond mlag id 7
nv	set	interface	bond8 bond member swp8
nv	set	interface	bond8 bond mlag id 8
nv	set	interface	bond9 bond member swp9
nv	set	interface	bond9 bond mlag id 9
nv	set	interface	bond10 bond member swp10
nv	set	interface	bond10 bond mlag id 10
nv	set	interface	bond11 bond member swp11
nv	set	interface	bond11 bond mlag id 11
nv	set	interface	bond12 bond member swp12
nv	set	interface	bond12 bond mlag id 12
nv	set	interface	bond13 bond member swp13
nv	set	interface	bond13 bond mlag id 13
nv	set	interface	bond14 bond member swp14
nv	set	interface	bond14 bond mlag id 14

nv set interface bond15 bond member swp15 nv set interface bond15 bond mlag id 15 nv set interface bond16 bond member swp16 nv set interface bond16 bond mlag id 16 nv set interface bond17 bond member swp17 nv set interface bond17 bond mlag id 17 nv set interface bond18 bond member swp18 nv set interface bond18 bond mlag id 18 nv set interface bond19 bond member swp19 nv set interface bond19 bond mlag id 19 nv set interface bond20 bond member swp20 nv set interface bond20 bond mlag id 20 nv set interface bond21 bond member swp21 nv set interface bond21 bond mlag id 21 nv set interface bond22 bond member swp22 nv set interface bond22 bond mlag id 22 nv set interface bond23 bond member swp23 nv set interface bond23 bond mlag id 23 nv set interface bond24 bond member swp24 nv set interface bond24 bond mlag id 24 nv set interface bond25 bond member swp25 nv set interface bond25 bond mlag id 25 nv set interface bond26 bond member swp26 nv set interface bond26 bond mlag id 26 nv set interface bond27 bond member swp27 nv set interface bond27 bond mlag id 27 nv set interface bond28 bond member swp28 nv set interface bond28 bond mlag id 28 nv set interface bond29 bond member swp29 nv set interface bond29 bond mlag id 29 nv set interface bond30 bond member swp30 nv set interface bond30 bond mlag id 30 nv set interface bond31 bond member swp31 nv set interface bond31 bond mlag id 31 nv set interface bond32 bond member swp32 nv set interface bond32 bond mlag id 32 nv set interface bond33 bond member swp33 nv set interface bond33 bond mlag id 33 nv set interface bond34 bond member swp34 nv set interface bond34 bond mlag id 34 nv set interface bond35 bond member swp35 nv set interface bond35 bond mlag id 35 nv set interface bond36 bond member swp36 nv set interface bond36 bond mlag id 36 nv set interface bond37 bond member swp37 nv set interface bond37 bond mlag id 37 nv set interface bond38 bond member swp38 nv set interface bond38 bond mlag id 38 nv set interface bond39 bond member swp39 nv set interface bond39 bond mlag id 39 nv set interface bond40 bond member swp40 nv set interface bond40 bond mlag id 40

(continued from previous page)

nv set interface bond41 bond member swp41 nv set interface bond41 bond mlag id 41 nv set interface bond42 bond member swp42 nv set interface bond42 bond mlag id 42 nv set interface bond43 bond member swp43 nv set interface bond43 bond mlag id 43 nv set interface bond44 bond member swp44 nv set interface bond44 bond mlag id 44 nv set interface bond45 bond member swp45 nv set interface bond45 bond mlag id 45 nv set interface bond46 bond member swp46 nv set interface bond46 bond mlag id 46 nv set interface bond47 bond member swp47 nv set interface bond47 bond mlag id 47 nv set interface bond48 bond member swp48 nv set interface bond48 bond mlag id 48 nv set interface bond51 bond member swp51 nv set interface bond51 bond mlag id 51 nv set interface bond51 bridge domain br\_default untagged 1 nv set interface bond51 bridge domain br\_default vlan all nv set interface eth0 ip address dhcp nv set interface eth0 ip vrf mgmt nv set interface eth0 type eth nv set interface lo ip address 10.160.254.22/32 nv set interface lo type loopback nv set interface peerlink bond member swp63 nv set interface peerlink bond member swp64 nv set interface peerlink type peerlink nv set interface peerlink.4094 base-interface peerlink nv set interface peerlink.4094 type sub nv set interface peerlink.4094 vlan 4094 nv set interface swp49-50 type swp nv set interface vlan101-102 ip vrr enable on nv set interface vlan101-102 ip vrr mac-address 00:1c:73:aa:bb:04 nv set interface vlan101-102 ip vrr state up nv set interface vlan101-102 type svi nv set interface vlan101 ip address 10.160.6.2/24 nv set interface vlan101 ip vrr address 10.160.6.1/24 nv set interface vlan101 vlan 101 nv set interface vlan102 ip address 10.184.94.2/24 nv set interface vlan102 ip vrr address 10.184.94.1/24 nv set interface vlan102 vlan 102 nv set mlag backup 10.160.254.23 nv set mlag enable on nv set mlag mac-address 44:38:39:FF:0A:00 nv set mlag peer-ip linklocal nv set mlag priority 2048 nv set router bgp autonomous-system 4200120327 nv set router bgp enable on nv set router bgp router-id 10.160.254.22 nv set router vrr enable on nv set service ntp mgmt server 0.cumulusnetworks.pool.ntp.org

(continued from previous page) nv set service ntp mgmt server 1.cumulusnetworks.pool.ntp.org nv set service ntp mgmt server 2.cumulusnetworks.pool.ntp.org nv set service ntp mgmt server 3.cumulusnetworks.pool.ntp.org nv set system aaa class nvapply action allow nv set system aaa class nvapply command-path / permission all nv set system aaa class nvshow action allow nv set system aaa class nvshow command-path / permission ro nv set system aaa class sudo action allow nv set system aaa class sudo command-path / permission all nv set system aaa role nvue-admin class nvapply nv set system aaa role nvue-monitor class nvshow nv set system aaa role system-admin class nvapply nv set system aaa role system-admin class sudo nv set system aaa user cumulus full-name cumulus,,, nv set system aaa user cumulus hashed-password '\*' nv set system aaa user cumulus role system-admin nv set system api state enabled nv set system config auto-save state enabled nv set system control-plane acl acl-default-dos inbound nv set system control-plane acl acl-default-whitelist inbound nv set system hostname SN4600C-1 nv set system reboot mode cold nv set system ssh-server permit-root-login enabled nv set system ssh-server state enabled nv set system ssh-server vrf mgmt nv set system timezone America/Los\_Angeles nv set system wih channel forwarding trigger 12 nv set system wjh channel forwarding trigger 13 nv set system wjh channel forwarding trigger tunnel nv set system wjh enable on nv set vrf default router bgp address-family ipv4-unicast enable on nv set vrf default router bgp address-family ipv4-unicast redistribute  $\rightarrow$  connected enable on nv set vrf default router bgp enable on nv set vrf default router bgp neighbor peerlink.4094 remote-as internal nv set vrf default router bgp neighbor peerlink.4094 timers connection-retry →10 nv set vrf default router bgp neighbor peerlink.4094 timers hold 10 nv set vrf default router bgp neighbor peerlink.4094 timers keepalive 3 nv set vrf default router bqp neighbor peerlink.4094 timers route-→advertisement auto nv set vrf default router bqp neighbor peerlink.4094 type unnumbered nv set vrf default router bgp neighbor swp49 remote-as external nv set vrf default router bgp neighbor swp49 timers connection-retry 10 nv set vrf default router bgp neighbor swp49 timers hold 10 nv set vrf default router bgp neighbor swp49 timers keepalive 3 nv set vrf default router bgp neighbor swp49 timers route-advertisement auto nv set vrf default router bgp neighbor swp49 type unnumbered nv set vrf default router bgp neighbor swp50 remote-as external nv set vrf default router bgp neighbor swp50 timers connection-retry 10 nv set vrf default router bgp neighbor swp50 timers hold 10 nv set vrf default router bgp neighbor swp50 timers keepalive 3

nv set vrf default router bgp neighbor swp50 timers route-advertisement auto nv set vrf default router bgp neighbor swp50 type unnumbered

#### 22.2.2. SN4600C-2 Configuration

nv set bridge domain br\_default vlan 100-102 nv set interface bond1 bond member swp1 nv set interface bond1 bond mlag id 1 nv set interface bond1-11,13-48,51 bond lacp-bypass on nv set interface bond1-48 bridge domain br\_default access 102 nv set interface bond1-48,51 bond mlag enable on nv set interface bond1-48,51 type bond nv set interface bond2 bond member swp2 nv set interface bond2 bond mlag id 2 nv set interface bond3 bond member swp3 nv set interface bond3 bond mlag id 3 nv set interface bond4 bond member swp4 nv set interface bond4 bond mlag id 4 nv set interface bond5 bond member swp5 nv set interface bond5 bond mlag id 5 nv set interface bond6 bond member swp6 nv set interface bond6 bond mlag id 6 nv set interface bond7 bond member swp7 nv set interface bond7 bond mlag id 7 nv set interface bond8 bond member swp8 nv set interface bond8 bond mlag id 8 nv set interface bond9 bond member swp9 nv set interface bond9 bond mlag id 9 nv set interface bond10 bond member swp10 nv set interface bond10 bond mlag id 10 nv set interface bond11 bond member swp11 nv set interface bond11 bond mlag id 11 nv set interface bond12 bond member swp12 nv set interface bond12 bond mlag id 12 nv set interface bond13 bond member swp13 nv set interface bond13 bond mlag id 13 nv set interface bond14 bond member swp14 nv set interface bond14 bond mlag id 14 nv set interface bond15 bond member swp15 nv set interface bond15 bond mlag id 15 nv set interface bond16 bond member swp16 nv set interface bond16 bond mlag id 16 nv set interface bond17 bond member swp17 nv set interface bond17 bond mlag id 17 nv set interface bond18 bond member swp18 nv set interface bond18 bond mlag id 18 nv set interface bond19 bond member swp19 nv set interface bond19 bond mlag id 19 nv set interface bond20 bond member swp20 nv set interface bond20 bond mlag id 20

nv set interface bond21 bond member swp21 nv set interface bond21 bond mlag id 21 nv set interface bond22 bond member swp22 nv set interface bond22 bond mlag id 22 nv set interface bond23 bond member swp23 nv set interface bond23 bond mlag id 23 nv set interface bond24 bond member swp24 nv set interface bond24 bond mlag id 24 nv set interface bond25 bond member swp25 nv set interface bond25 bond mlag id 25 nv set interface bond26 bond member swp26 nv set interface bond26 bond mlag id 26 nv set interface bond27 bond member swp27 nv set interface bond27 bond mlag id 27 nv set interface bond28 bond member swp28 nv set interface bond28 bond mlag id 28 nv set interface bond29 bond member swp29 nv set interface bond29 bond mlag id 29 nv set interface bond30 bond member swp30 nv set interface bond30 bond mlag id 30 nv set interface bond31 bond member swp31 nv set interface bond31 bond mlag id 31 nv set interface bond32 bond member swp32 nv set interface bond32 bond mlag id 32 nv set interface bond33 bond member swp33 nv set interface bond33 bond mlag id 33 nv set interface bond34 bond member swp34 nv set interface bond34 bond mlag id 34 nv set interface bond35 bond member swp35 nv set interface bond35 bond mlag id 35 nv set interface bond36 bond member swp36 nv set interface bond36 bond mlag id 36 nv set interface bond37 bond member swp37 nv set interface bond37 bond mlag id 37 nv set interface bond38 bond member swp38 nv set interface bond38 bond mlag id 38 nv set interface bond39 bond member swp39 nv set interface bond39 bond mlag id 39 nv set interface bond40 bond member swp40 nv set interface bond40 bond mlag id 40 nv set interface bond41 bond member swp41 nv set interface bond41 bond mlag id 41 nv set interface bond42 bond member swp42 nv set interface bond42 bond mlag id 42 nv set interface bond43 bond member swp43 nv set interface bond43 bond mlag id 43 nv set interface bond44 bond member swp44 nv set interface bond44 bond mlag id 44 nv set interface bond45 bond member swp45 nv set interface bond45 bond mlag id 45 nv set interface bond46 bond member swp46 nv set interface bond46 bond mlag id 46

(continued from previous page)

(continued from previous page)

nv set interface bond47 bond member swp47 nv set interface bond47 bond mlag id 47 nv set interface bond48 bond member swp48 nv set interface bond48 bond mlag id 48 nv set interface bond51 bond member swp51 nv set interface bond51 bond mlag id 51 nv set interface bond51 bridge domain br\_default untagged 1 nv set interface bond51 bridge domain br\_default vlan all nv set interface eth0 ip address dhcp nv set interface eth0 ip vrf mgmt nv set interface eth0 type eth nv set interface lo ip address 10.160.254.22/32 nv set interface lo type loopback nv set interface peerlink bond member swp63 nv set interface peerlink bond member swp64 nv set interface peerlink type peerlink nv set interface peerlink.4094 base-interface peerlink nv set interface peerlink.4094 type sub nv set interface peerlink.4094 vlan 4094 nv set interface swp49-50 type swp nv set interface vlan101-102 ip vrr enable on nv set interface vlan101-102 ip vrr mac-address 00:1c:73:aa:bb:04 nv set interface vlan101-102 ip vrr state up nv set interface vlan101-102 type svi nv set interface vlan101 ip address 10.160.6.2/24 nv set interface vlan101 ip vrr address 10.160.6.1/24 nv set interface vlan101 vlan 101 nv set interface vlan102 ip address 10.184.94.2/24 nv set interface vlan102 ip vrr address 10.184.94.1/24 nv set interface vlan102 vlan 102 nv set mlag backup 10.160.254.23 nv set mlag enable on nv set mlag mac-address 44:38:39:FF:0A:00 nv set mlag peer-ip linklocal nv set mlag priority 2048 nv set router bgp autonomous-system 4200120327 nv set router bgp enable on nv set router bgp router-id 10.160.254.22 nv set router vrr enable on nv set service ntp mgmt server 0.cumulusnetworks.pool.ntp.org nv set service ntp mgmt server 1.cumulusnetworks.pool.ntp.org nv set service ntp mgmt server 2.cumulusnetworks.pool.ntp.org nv set service ntp mgmt server 3.cumulusnetworks.pool.ntp.org nv set system aaa class nvapply action allow nv set system aaa class nvapply command-path / permission all nv set system aaa class nvshow action allow nv set system aaa class nvshow command-path / permission ro nv set system aaa class sudo action allow nv set system aaa class sudo command-path / permission all nv set system aaa role nvue-admin class nvapply nv set system aaa role nvue-monitor class nvshow nv set system aaa role system-admin class nvapply

```
nv set system aaa role system-admin class sudo
nv set system aaa user cumulus full-name cumulus,,,
nv set system aaa user cumulus hashed-password '*'
nv set system aaa user cumulus role system-admin
nv set system api state enabled
nv set system config auto-save state enabled
nv set system control-plane acl acl-default-dos inbound
nv set system control-plane acl acl-default-whitelist inbound
nv set system hostname SN4600C-2
nv set system reboot mode cold
nv set system ssh-server permit-root-login enabled
nv set system ssh-server state enabled
nv set system ssh-server vrf mgmt
nv set system timezone America/Los_Angeles
nv set system wjh channel forwarding trigger 12
nv set system wih channel forwarding trigger 13
nv set system wjh channel forwarding trigger tunnel
nv set system wjh enable on
nv set vrf default router bgp address-family ipv4-unicast enable on
nv set vrf default router bgp address-family ipv4-unicast redistribute
\rightarrow connected enable on
nv set vrf default router bgp enable on
nv set vrf default router bgp neighbor peerlink.4094 remote-as internal
nv set vrf default router bgp neighbor peerlink.4094 timers connection-retry
→10
nv set vrf default router bgp neighbor peerlink.4094 timers hold 10
nv set vrf default router bgp neighbor peerlink.4094 timers keepalive 3
nv set vrf default router bgp neighbor peerlink.4094 timers route-
→advertisement auto
nv set vrf default router bgp neighbor peerlink.4094 type unnumbered
nv set vrf default router bgp neighbor swp49 remote-as external
nv set vrf default router bgp neighbor swp49 timers connection-retry 10
nv set vrf default router bgp neighbor swp49 timers hold 10
nv set vrf default router bgp neighbor swp49 timers keepalive 3
nv set vrf default router bgp neighbor swp49 timers route-advertisement auto
nv set vrf default router bgp neighbor swp49 type unnumbered
nv set vrf default router bgp neighbor swp50 remote-as external
nv set vrf default router bgp neighbor swp50 timers connection-retry 10
nv set vrf default router bgp neighbor swp50 timers hold 10
nv set vrf default router bgp neighbor swp50 timers keepalive 3
nv set vrf default router bgp neighbor swp50 timers route-advertisement auto
nv set vrf default router bgp neighbor swp50 type unnumbered
```

### 22.3. QM9700 (computenet) Switches Configuration

#### 22.3.1. QM9700-1

```
##
## Running database "initial"
## Generated at 2025/02/15 06:42:32 +0000
## Hostname: QM9700-1
## Product release: 3.12.1002
##
##
## Running-config temporary prefix mode setting
##
no cli default prefix-modes enable
##
## IB Partition configuration
##
   ib partition Default defmember full force
##
## Subnet Manager configuration
##
   ib sm virt enable
##
## IB ports configuration
##
   interface ib 1/1/1-1/1/2 mtu 4K
   interface ib 1/1/1-1/1/2 op-vls 8
   interface ib 1/1/1-1/1/2 speed sdr qdr fdr edr hdr ndr
   interface ib 1/1/1-1/1/2 width 7
##
## Network interface configuration
##
no interface mgmt0 dhcp
   interface mgmt0 ip address 10.185.231.43 /22
   management-lldp enable
##
## Other IP configuration
##
   hostname QM9700-1
   ip domain-list nvidia.com
   ip name-server 10.126.136.6
   ip route 0.0.0.0/0 10.185.228.1
##
## Other IPv6 configuration
##
no ipv6 enable
##
```

```
(continued from previous page)
## Local user account configuration
##
   username admin password 7 <>
   username monitor password 7 <>
##
## AAA remote server configuration
##
# ldap bind-password *******
# radius-server key *******
# tacacs-server key *******
##
## Password restriction configuration
##
no password hardening enable
##
## Network management configuration
##
# web proxy auth basic password *******
##
## X.509 certificates configuration
##
#
# Certificate name system-self-signed, ID
→9f639fcad62931e3996712b59066cdda047fb176
# (public-cert config omitted since private-key config is hidden)
##
## IB nodename to GUID mapping
##
   ib ha infiniband-default ip 10.185.230.247 /22 force
   ib smnode CL-01 create
   ib smnode CL-01 enable
   ib smnode CL-01 sm-priority 15
##
## Persistent prefix mode setting
##
cli default prefix-modes enable
```

#### 22.3.2. QM9700-2

```
##
## Running database "initial"
## Generated at 2025/02/15 06:41:54 +0000
## Hostname: QM9700-2
## Product release: 3.12.1002
```
##

(continued from previous page)

```
##
## Running-config temporary prefix mode setting
##
no cli default prefix-modes enable
##
## IB Partition configuration
##
   ib partition Default defmember full force
##
## Subnet Manager configuration
##
   ib sm virt enable
##
## Other IP configuration
##
   hostname QM9700-2
##
## Local user account configuration
##
   username admin password 7 <>
   username monitor password 7 <>
##
## AAA remote server configuration
##
# ldap bind-password *******
# radius-server key *******
# tacacs-server key *******
##
## Password restriction configuration
##
no password hardening enable
##
## Network management configuration
##
# web proxy auth basic password *******
##
## X.509 certificates configuration
##
#
# Certificate name system-self-signed, ID
\rightarrow 146da5394146409cf2e60c4b7debbedd1e2e6ac4
# (public-cert config omitted since private-key config is hidden)
```

(continues on next page)

(continued from previous page)

## Copyright

©2024-2025, NVIDIA Corporation