NVIDIA DGX B200 BasePOD Deployment Guide with NVIDIA Mission Control

Release latest

NVIDIA Corporation

Jul 18, 2025

Contents

Chapter 1. Introduction

This document provides the steps for deploying NVIDIA DGX BasePOD with DGX B200 systems and NVIDIA Mission Control 1.1.

Chapter 2. NVIDIA Mission Control

NVIDIA Mission Control 1.1 for DGX B200 includes NVIDIA Base Command Manager and NVIDIA Run:ai functionality as part of integrated software delivery across configuration, validation, and operations for cluster management and workload orchestration. This release has built-in dashboards scoped for DGX B200 providing better visibility into cluster health, ensuring faster triaging.

NVIDIA Mission Control 1.1 release leverages Base Command Manager (BCM) 10.25.03 and NVIDIA Run:ai.

Chapter 3. Reference

NVIDIA Mission Control

NVIDIA DGX BasePOD Reference Architecture

NVIDIA DGX BasePOD Deployment Guide with DGX H100 or DGX H200 systems

Chapter 4. NVIDIA DGX BasePOD with DGX B200 Systems Deployment

The deployment and configuration is standardized across NVIDIA DGX B200 and DGX H100/H200 systems. Please refer to the DGX BasePOD Deployment guide featuring DGX H100/H200 for detailed instructions, with the following changes for DGX B200.

Note

We strongly recommend reading the BasePOD Deployment guide prior to initiating the DGX B200 deployment.

4.1. Hardware Overview

An overview of the hardware is in Table 4.1. 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.

Table 4.1: DGX BasePOD Components

Component	Technology
Compute nodes	DGX B200 system
Cluster Management	NVIDIA Mission Control
Compute fabric	NVIDIA Quantum QM9700 InfiniBand switches
Management fabric	NVIDIA SN4600C switches
Storage fabric	Option 1: NVIDIA SN4600C switches for Ethernet attached storage Option 2: NVIDIA Quantum QM9700 switches for InfiniBand attached storage
Out-of-band management fabric	NVIDIA SN2201 switches
Control plane and workload management nodes	Minimum Requirements (each server): 2 × Intel x86 Xeon Gold or better 512 GB memory 2 × 480 GB M.2 RAID for OS 4 × 200 Gbps network 2 × 100 GbE network

4.2. Networking

This section covers the DGX system network ports and an overview of the networks used by DGX B200 System Network Ports.

Figure 4.1 shows the physical layout of the back of the DGX B200 system.

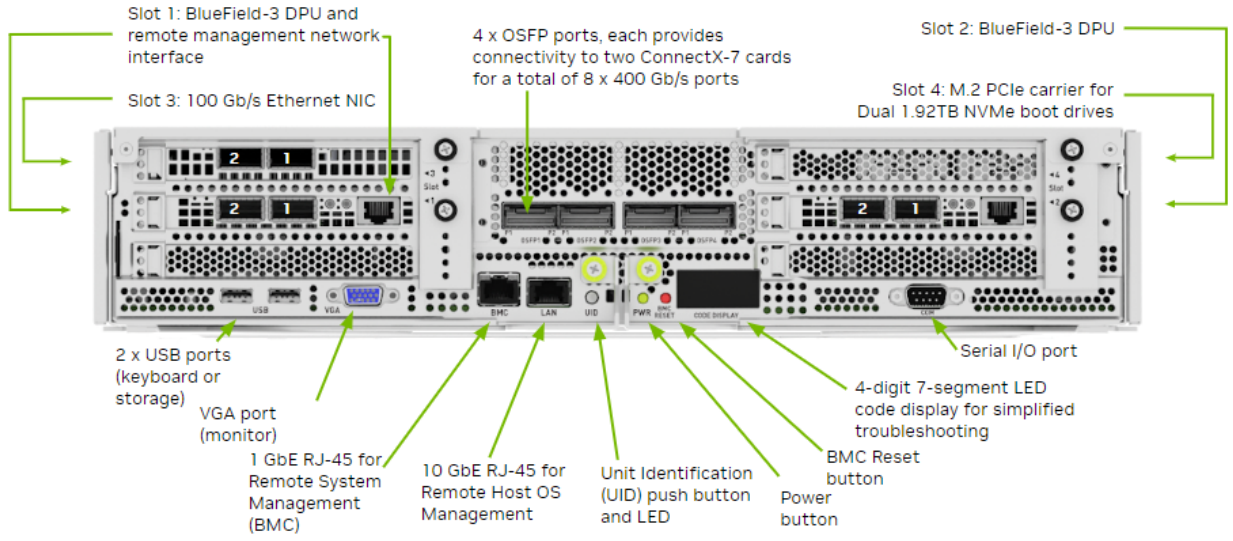


Figure 4.1: Physical layout of the back of the DGX B200 system

Figure 4.2 shows how the DGX B200 network ports are used in this deployment guide.

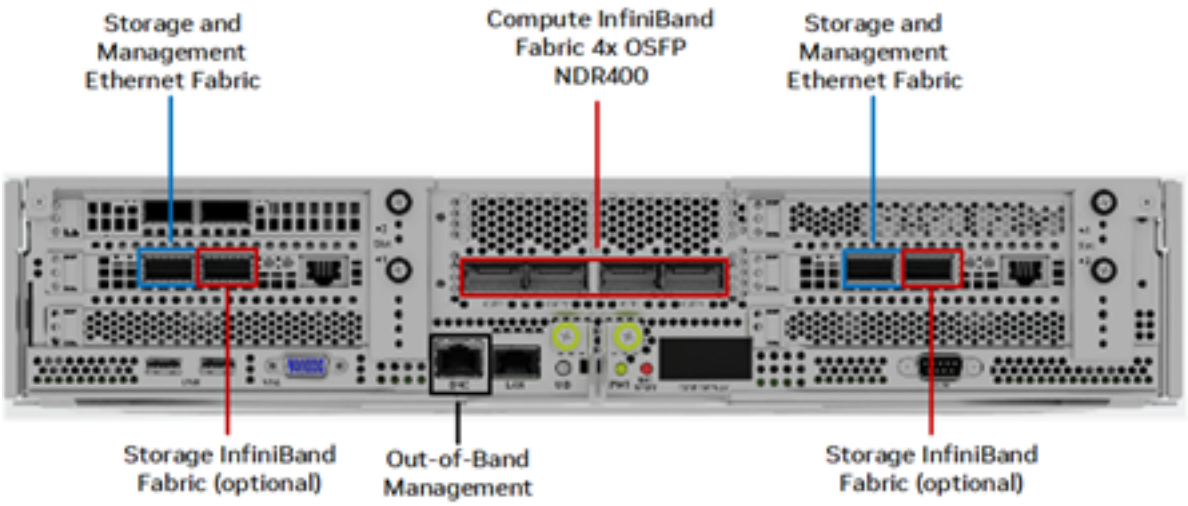


Figure 4.2: DGX B200 network ports used in this deployment guide

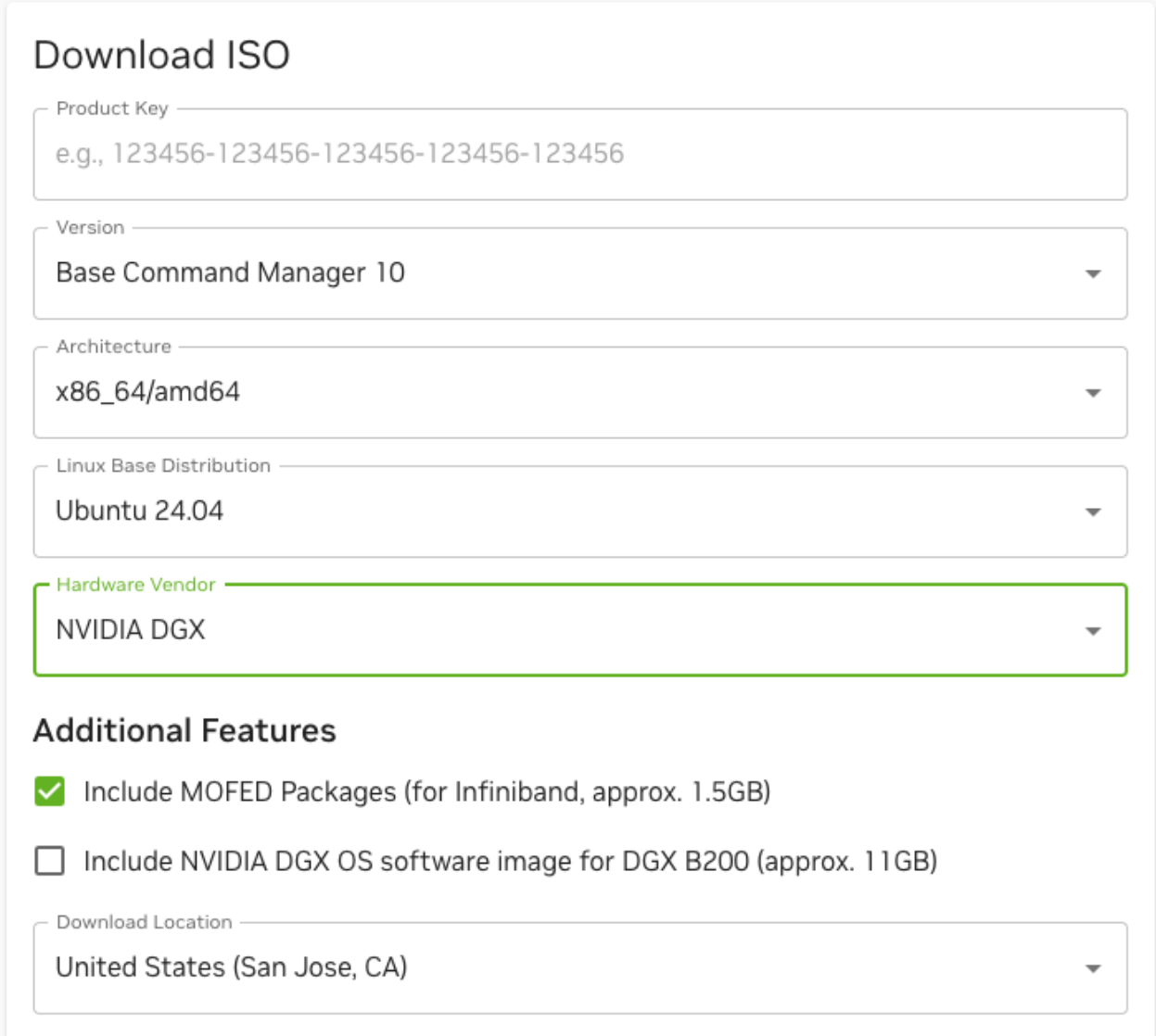
The following ports are selected for DGX BasePOD networking:

- Eight ports in four OSFP connections are used for the InfiniBand compute fabric
- Each pair of dual-port NVIDIA BlueField-3 HCAs (NIC mode) provide parallel pathways to the storage and management fabrics.
- Optional One port of dual-port BlueField-3 HCAs (IB mode) provides access to IB storage fabrics.
- BMC network access is provided through the out-of-band network
- The networking ports and their mapping are described in detail in the Network Ports section of the NVIDIA DGX B200 System User Guide.

Chapter 5. Base Command Manager Headnodes Installation

5.1. Download the Base Command Manager (BCM) ISO

Download the latest BCM 10.x ISO image from the BCM website with the following options.



5.2. DGX Node Bringup

Continue with the installation steps in the DGX BasePOD Deployment Guide for DGX H100 or DGX H200 systems till the Cluster bring up section and follow the instructions below to provision the DGX B200.

5.2.1. Importing Base OS7

We will be using the following pre-generated Base OS image for the DGX B200 Systems.

Note

This includes the latest available cuda-driver and the latest DCGM packages in addition to the latest Base OS packages as 03-15-2025.

Download the above tar.gz image to the head node under /cm/images.

```
root@bcm10-headnode1:~# wget https://support2.brightcomputing.com/
↪bcm10-b200-image/DGXOS-7.0.2-DGX-B200-03-20-2025-1.tar.gz
root@bcm10-headnode1:~# ls -al
-rw-r--r-- 1 root root 4.0G Mar 15 04:32  DGXOS-7.0.2-DGX-B200-03-20-
↪2025-1.tar.gz
```

Extract the tar.gz image on the headnode.

```
root@bcm10-headnode1:~# mkdir /cm/images/baseos7
root@bcm10-headnode1:~# tar -xzz DGXOS-7.0.2-DGX-B200-03-20-2025-1.
↪tar.gz -C /cm/images/baseos7
root@bcm10-headnode1:~# cd /cm/images/baseos7
```

Next create BCM install image by running:

```
root@bcm10-headnode1:/cm/images# cm-create-image -d /cm/images/
↪baseos7 -n baseos7 -s
Running validate image dir..................... [ OK ]
Running sanity check........................... [ OK ]
Finalize base distribution..................... [ OK ]
Copying cm repo files.......................... [ OK ]
Validating repo configuration.................. [ OK ]
Installing distribution packages............... [ OK ]
Finalizing image services...................... [ OK ]
Installing CM packages......................... [ OK ]
Finalizing cluster services.................... [ OK ]
Copying cluster certificate to image........... [ OK ]
Adding/Updating software image................. [ OK ]
```

5.2.2. Update Base OS7

The following steps utilize the cm-chroot-sw-img command to modify the imported Base OS image.

```
# enter into chroot mode to modify the image

root@bcm10-headnode1:~# cm-chroot-sw-img /cm/images/baseos7

# symlink enroot.conf to /cm/shared/apps/slurm/var/etc/enroot.conf

root@baseos7:/# ln -sf /cm/shared/apps/slurm/var/etc/enroot.conf /
↪etc/enroot/enroot.conf

# create /etc/enroot/environ.d/60-pmix.env with the following contents.

root@baseos7:/# cat /etc/enroot/environ.d/60-pmix.env

PMIX_MCA_ptl=^usock
PMIX_MCA_psec=native
PMIX_SYSTEM_TMPDIR=/var/empty
PMIX_MCA_gds=hash
```

(continues on next page)

(continued from previous page)

```
# create /etc/enroot/environ.d/30-cuda.env with the following contents.

root@baseos7:/# cat /etc/enroot/environ.d/30-cuda.env

CUDA_DEVICE_ORDER=PCI_BUS_ID

# create /etc/enroot/environ.d/20-mlnx.env with the following contents.

root@baseos7:/# cat /etc/enroot/environ.d/20-mlnx.env

MELLANOX_VISIBLE_DEVICES=4,7,8,9,10,13,14,15
OMPI_MCA_btl_tcp_if_include=bond0
OMPI_MCA_btl_openib_warn_default_gid_prefix=0
CUDA_CACHE_DISABLE=1

# Update /etc/sysctl.d/90-cm-sysctl.conf to the following:

net.ipv4.conf.all.arp_ignore = 0
root@baseos7:/# cat /etc/sysctl.d/90-cm-sysctl.conf | grep net.ipv4.
↪conf.all.arp_ignore

# exit out of chroot mode

$ exit
```

Enter cmsh verify that the DGX OS image has been created.

```
root@bcm10-headnode1:/cm/images# cmsh
[bcm10-headnode1]% softwareimage
[bcm10-headnode1->softwareimage]% ls
Name (key) Path (key) Kernel version Nodes

------------------------------------------------------------------------
↪----------------

**baseos7** /cm/images/baseos7 6.8.0-55-generic 0
default-image /cm/images/default-image 6.8.0-31-generic 1
k8s-ctrl-image /cm/images/k8s-ctrl-image 6.8.0-31-generic 0
k8s-ctrl-image-orig /cm/images/k8s-ctrl-image-orig 6.8.0-31-generic 0
slogin-image /cm/images/slogin-image 6.8.0-31-generic 0
slogin-image-orig /cm/images/slogin-image-orig 6.8.0-31-generic 0
```

After creating the DGXOS7 image we'll add the "bonding" kernel module into the image.

```
[bcm10-headnode1->softwareimage]% use baseos7
[bcm10-headnode1->softwareimage[baseos7]]% kernelmodules
[bcm10-headnode1->softwareimage[baseos7]->kernelmodules]% ls
Module (key) Parameters
------------------------------------------------------------------------
↪---

bridge
```

(continues on next page)

(continued from previous page)

```
i40e
dm-persistent-data
aacraid
nfsv4
usbhid
aic7xxx
hv_netvsc
ixgbe
forcedeth
bnx2
hpilo
nvme
igb
jfs
sata_sil
ipmi_si
sata_nv
hpsa
sata_svw
btrfs
ahci
mptscsih
igbvf
isofs
ipmi_devintf
megaraid
aic79xx
xfs
bnxt_en
udf
hv_utils
ixgbevf
arcmsr
e1000
hv_storvsc
nfsv3
dm-bufio
nfs
dm-bio-prison
megaraid_sas
mptspi
reiserfs
tg3
mptsas
bnx2x
e1000e
mpt3sas
dm-thin-pool
br_netfilter
hv_vmbus
[bcm10-headnode1->softwareimage[baseos7]->kernelmodules]% add bonding
[bcm10-headnode1->softwareimage*[baseos7\*]->kernelmodules*[bonding\
```

(continues on next page)

(continued from previous page)

```
↪*]]% commit
Thu Mar 16 12:53:38 2025 [notice] bcm10-headnode1: Initial ramdisk
↪foR image baseos7 is being generated

#

# wait for confirmation that ramdisk was generated...this may take a
↪few seconds.

#

[bcm10-headnode1->softwareimage[baseos7]->kernelmodules[bonding]]%

Thu Mar 16 12:55:04 2025 [notice] bcm10-headnode1: Initial ramdisk
↪for image baseos7 was generated successfully
```

Then clone baseos7 to baseos7-backup to have a backup in case a rollback is needed.

```
[bcm10-headnode1->softwareimage[baseos7]]% clone baseos7 baseos7-
↪backup
[bcm10-headnode1->softwareimage*[baseos7-backup\*]]% commit
```

5.2.3. Define DGX Node udev Rules

The following link shows how to manually define the udev rules for DGX A100 & DGX H100 systems. DGX B200 will utilize the same rules as DGX H100. KB Article

Create the following udev rule for the DGX B200 disksetup.

```
root@bcm10-headnode1:~# cat /cm/node-installer/usr/lib/udev/rules.d/
↪60-persistent-storage-b200.rules
########## persistent nvme rules by HW address ##########
KERNEL=="nvme[0-9]n[0-9]", ATTRS{address}=="0000:01:00.0", SYMLINK+=
↪"disk/by-id/osdisk-1"
KERNEL=="nvme[0-9]n[0-9]", ATTRS{address}=="0000:02:00.0", SYMLINK+=
↪"disk/by-id/osdisk-2"
KERNEL=="nvme[0-9]n[0-9]", ATTRS{address}=="0000:ab:00.0", SYMLINK+=
↪"disk/by-id/raiddisk-1"
KERNEL=="nvme[0-9]n[0-9]", ATTRS{address}=="0000:ac:00.0", SYMLINK+=
↪"disk/by-id/raiddisk-2"
KERNEL=="nvme[0-9]n[0-9]", ATTRS{address}=="0000:ad:00.0", SYMLINK+=
↪"disk/by-id/raiddisk-3"
KERNEL=="nvme[0-9]n[0-9]", ATTRS{address}=="0000:ae:00.0", SYMLINK+=
↪"disk/by-id/raiddisk-4"
KERNEL=="nvme[0-9]n[0-9]", ATTRS{address}=="0000:2a:00.0", SYMLINK+=
↪"disk/by-id/raiddisk-5"
KERNEL=="nvme[0-9]n[0-9]", ATTRS{address}=="0000:2b:00.0", SYMLINK+=
↪"disk/by-id/raiddisk-6"
KERNEL=="nvme[0-9]n[0-9]", ATTRS{address}=="0000:2c:00.0", SYMLINK+=
↪"disk/by-id/raiddisk-7"
KERNEL=="nvme[0-9]n[0-9]", ATTRS{address}=="0000:2d:00.0", SYMLINK+=
```

(continues on next page)

(continued from previous page)

```
↪"disk/by-id/raiddisk-8"
########## persistent nvme rules by HW address ##########
```

Copy the rule from /cm/node-installer to the /cm/images.

```
root@bcm10-headnode1:~# cp
/cm/node-installer/usr/lib/udev/rules.d/60-persistent-storage-b200.
↪rules /cm/images/baseos7/usr/lib/udev/rules.d/
```

Append the following kernel parameters to the baseos7 softwareimage.

> **Note**
>
> Please include the space between the " and nvme_core

```
root@bcm10-headnode1:~# cmsh
[bcm10-headnode1]% softwareimage
[bcm10-headnode1->softwareimage]% use baseos7
[bcm10-headnode1->softwareimage[baseos7]]% append kernelparameters "
↪nvme_core.multipath=n iommu=pt"
[bcm10-headnode1->softwareimage*[baseos7*]]% commit
```

Next create the disksetup.xml file at /cm/local/apps/cmd/etc/htdocs/disk-setup/dgx-disk-udev.xml for the DGX B200 with the following content.

```
<?xml version="1.0" encoding="UTF-8"?>
<diskSetup>
    <device>
        <blockdev>/dev/disk/by-id/osdisk-1</blockdev>
        <partition id="boot1" partitiontype="esp">
            <size>512M</size>
            <type>linux</type>
            <filesystem>fat</filesystem>
            <mountPoint>/boot/efi</mountPoint>
            <mountOptions>defaults,noatime,nodiratime</mountOptions>
        </partition>
        <partition id="slash1">
            <size>max</size>
            <type>linux raid</type>
        </partition>
    </device>
    <device>
        <blockdev>/dev/disk/by-id/osdisk-2</blockdev>
        <partition id="boot2" partitiontype="esp">
            <size>512M</size>
            <type>linux</type>
            <filesystem>fat</filesystem>
            <mountOptions>defaults,noatime,nodiratime</mountOptions>
        </partition>
        <partition id="slash2">
            <size>max</size>
```

(continues on next page)

(continued from previous page)

```
            <type>linux raid</type>
        </partition>
    </device>
    <device>
        <blockdev>/dev/disk/by-id/raiddisk-1</blockdev>
        <partition id="raid1" partitiontype="esp">
            <size>max</size>
            <type>linux raid</type>
        </partition>
    </device>
    <device>
        <blockdev>/dev/disk/by-id/raiddisk-2</blockdev>
        <partition id="raid2" partitiontype="esp">
            <size>max</size>
            <type>linux raid</type>
        </partition>
    </device>
    <device>
        <blockdev>/dev/disk/by-id/raiddisk-3</blockdev>
        <partition id="raid3" partitiontype="esp">
            <size>max</size>
            <type>linux raid</type>
        </partition>
    </device>
    <device>
        <blockdev>/dev/disk/by-id/raiddisk-4</blockdev>
        <partition id="raid4" partitiontype="esp">
            <size>max</size>
            <type>linux raid</type>
        </partition>
    </device>
    <device>
        <blockdev>/dev/disk/by-id/raiddisk-5</blockdev>
        <partition id="raid5" partitiontype="esp">
            <size>max</size>
            <type>linux raid</type>
        </partition>
    </device>
    <device>
        <blockdev>/dev/disk/by-id/raiddisk-6</blockdev>
        <partition id="raid6" partitiontype="esp">
            <size>max</size>
            <type>linux raid</type>
        </partition>
    </device>
    <device>
        <blockdev>/dev/disk/by-id/raiddisk-7</blockdev>
        <partition id="raid7" partitiontype="esp">
            <size>max</size>
            <type>linux raid</type>
        </partition>
    </device>
```

(continues on next page)

(continued from previous page)

```
    <device>
        <blockdev>/dev/disk/by-id/raiddisk-8</blockdev>
        <partition id="raid8" partitiontype="esp">
            <size>max</size>
            <type>linux raid</type>
        </partition>
    </device>
    <raid id="slash">
        <member>slash1</member>
        <member>slash2</member>
        <level>1</level>
        <filesystem>ext4</filesystem>
        <mountPoint>/</mountPoint>
        <mountOptions>defaults,noatime,nodiratime</mountOptions>
    </raid>
    <raid id="raid">
        <member>raid1</member>
        <member>raid2</member>
        <member>raid3</member>
        <member>raid4</member>
        <member>raid5</member>
        <member>raid6</member>
        <member>raid7</member>
        <member>raid8</member>
        <level>0</level>
        <filesystem>ext4</filesystem>
        <mountPoint>/raid</mountPoint>
        <mountOptions>defaults,noatime,nodiratime</mountOptions>
    </raid>
</diskSetup>
```

5.2.4. Defining DGX B200 Node Category

Clone the default category to dgx-b200 & link the previously defined disksetup.xml and define the softwareimage.

```
root@bcm10-headnode1:~# cmsh
[bcm10-headnode1]% category
[bcm10-headnode1->category]% clone default dgx-b200
[bcm10-headnode1->category[dgx-b200]]% set disksetup /cm/local/apps/
↪cmd/etc/htdocs/disk-setup/dgx-disk-udev.xml
[bcm10-headnode1->category*[dgx-b200*]]% set softwareimage baseos7
[bcm10-headnode1->category*[dgx-b200*]]% commit
```

Create the following `finalizescript`.

```
root@bcm10-headnode1:~# cat finalizescript.sh
#!/bin/bash
sed -i "s/.*\/boot\/efi.*/UUID=\"$(blkid -l -t PARTLABEL=/boot/efi -
↪s UUID -o value)\" \/boot\/efi    vfat       defaults,noatime,
↪nodiratime              0 2/" /localdisk/etc/fstab
```

Then link the finalizescript to the dgx-b200 category, save changes, and exit.

```
[bcm10-headnode1->category[dgx-b200]]% set finalizescript /root/
↪finalizescript.sh
[bcm10-headnode1->category*[dgx-b200\*]]% commit
```

5.3. Test provisioning of DGX Nodes

The rest of the procedure is the same as outlined in the DGX BasePOD Deployment Guide. with DGX H100 or DGX H200 systems.

Attaching the DGX B200 node definition here for reference.

5.3.1. DGX Nodes

Define the first DGX B200 node identity.

```
[bcm10-headnode1]% device
[bcm10-headnode1->device]% add physicalnode dgx-01 10.133.11.25 bond0
[bcm10-headnode1->device*[dgx-01*]]% set category dgx-b200
[bcm10-headnode1->device*[dgx-01*]]% set mac c4:70:bd:d2:0b:79
```

Set the interfaces and MAC addresses of the inband management interfaces for the specified DGX.

```
[bcm10-headnode1->device*[dgx-01*]]% interfaces
[bcm10-headnode1->device*[dgx-01*]->interfaces]% remove bootif
[bcm10-headnode1->device*[dgx-01*]->interfaces*]% add bmc ipmi0 10.
↪150.123.25 ipminet
Switched power control for this node to: ipmi0
[bcm10-headnode1->device*[dgx-01*]->interfaces*[ipmi0*]]% add
↪physical enp170s0f1np1
[bcm10-headnode1->device*[dgx-01*]->interfaces*[enp170s0f1np1*]]%
↪set mac c4:70:bd:d2:0b:79
[bcm10-headnode1->device*[dgx-01*]->interfaces*[enp170s0f1np1*]]%
↪add physical enp41s0f1np1
[bcm10-headnode1->device*[dgx-01*]->interfaces*[enp41s0f1np1*]]% set
↪mac c4:70:bd:d2:11:b5
[bcm10-headnode1->device*[dgx-01*]->interfaces*[enp41s0f1np1*]]% use
↪bond0
[bcm10-headnode1->device*[dgx-01*]->interfaces*[bond0]]% set network
↪dgxnet
[bcm10-headnode1->device*[dgx-01*]->interfaces*[bond0]]% set mode 4
[bcm10-headnode1->device*[dgx-01*]->interfaces*[bond0*]]% set
↪interfaces enp170s0f1np1 enp41s0f1np1
[bcm10-headnode1->device*[dgx-01*]->interfaces*[bond0*]]% ..
[bcm10-headnode1->device*[dgx-01*]->interfaces*]% commit
[bcm10-headnode1->device[dgx-01]->interfaces]% ..
[bcm10-headnode1->device[dgx-01]]% set managementnetwork dgxnet
[bcm10-headnode1->device*[dgx-01*]]% commit
```

Define the IB interfaces for the DGX B200.

```
[bcm10-headnode1->device*[dgx-01\*]->interfaces]% add physical
↪ibp154s0 100.126.0.25 computenet
[bcm10-headnode1->device*[dgx-01\*]->interfaces*[ibp154s0\*]]%
↪foreach -o ibp154s0 ibp192s0 ibp206s0 ibp220s0 ibp24s0 ibp64s0
↪ibp79s0 ibp94s0 ()
[bcm10-headnode1->device*[dgx-01\*]->interfaces\*]% set ibp192s0 ip
↪100.126.1.25
[bcm10-headnode1->device*[dgx-01\*]->interfaces\*]% set ibp206s0 ip
↪100.126.2.25
[bcm10-headnode1->device*[dgx-01\*]->interfaces\*]% set ibp220s0 ip
↪100.126.3.25
[bcm10-headnode1->device*[dgx-01\*]->interfaces\*]% set ibp24s0 ip
↪100.126.4.25
[bcm10-headnode1->device*[dgx-01\*]->interfaces\*]% set ibp64s0 ip
↪100.126.5.25
[bcm10-headnode1->device*[dgx-01\*]->interfaces\*]% set ibp79s0 ip
↪100.126.6.25
[bcm10-headnode1->device*[dgx-01\*]->interfaces\*]% set ibp94s0 ip
↪100.126.7.25
[bcm10-headnode1->device*[dgx-01\*]->interfaces\*]% commit
[bcm10-headnode1->device[dgx-01]->interfaces]% add physical
↪ibp170s0f0 100.127.0.25 storagenet
[bcm10-headnode1->device*[dgx-01\*]->interfaces*[ibp170s0f0\*]]% add
↪physical ibp41s0f0 100.127.1.25 storagenet
[bcm10-headnode1->device*[dgx-01\*]->interfaces*[ibp41s0f0\*]]%
↪commit
```

Ensure that all the interfaces are set up properly.

```
[bcm10-headnode1->device[dgx-01]->interfaces]% ls

Type Network device name IP Network Start if
------------ ---------------------- ---------------- ---------------
↪--------
bmc ipmi0 10.150.123.25 ipminet always
bond bond0 [prov] 10.150.125.25 dgxnet always
physical enp170s0f1np1 (bond0) 0.0.0.0 always
physical enp41s0f1np1 (bond0) 0.0.0.0 always
physical ibp154s0 100.126.0.25 computenet always
physical ibp170s0f0 100.127.0.25 storagenet always
physical ibp192s0 100.126.1.25 computenet always
physical ibp206s0 100.126.2.25 computenet always
physical ibp220s0 100.126.3.25 computenet always
physical ibp24s0 100.126.4.25 computenet always
physical ibp41s0f0 100.127.1.25 storagenet always
physical ibp64s0 100.126.5.25 computenet always
physical ibp79s0 100.126.6.25 computenet always
physical ibp94s0 100.126.7.25 computenet always
```

Clone dgx-01 to create the rest of the DGX nodes.

```
[bcm10-headnode1->device]%foreach -o dgx-01 -n DGX-02..DGX-04 () --
↪next-ip
```

Assign the respective management interface mac addresses for each node.

```
home;device
use dgx-02
set mac C4:70:BD:D2:0B:79
interfaces
use enp170s0f1np1
set mac C4:70:BD:D2:0B:79
..
use  enp41s0f1np1
set mac C4:70:BD:D2:11:B5
commit
```

5.3.2. Provision Nodes into the Cluster

Power on all the nodes. They should boot into their assigned roles automatically.

5.4. Deploy the Cluster

Continue deploying the cluster as prescribed in the DGX BasePOD Deployment Guide with DGX H100 or DGX H200 systems.

Chapter 6. NVIDIA Run:ai

The NVIDIA Mission Control 1.1 license purchase includes NVIDIA Run:ai. Please contact runai-order@nvidia.com for your included NVIDIA Run:ai licenses and installation instructions.

Copyright