



# NVIDIA Enterprise Partners

Deployment Guide for NVIDIA Enterprise RAs – Base Command Manager

## **NVIDIA Enterprise Reference Architecture Notice & Disclaimers**

NVIDIA Enterprise Reference Architecture documentation is NVIDIA confidential information and embodies substantial intellectual property and engineering know-how. NVIDIA grants you limited permissions to use NVIDIA Enterprise Reference Architecture to create NVIDIA platform-enabled data center clusters with NVIDIA technologies. You may not use the NVIDIA Enterprise Reference Architecture to develop competing products or technologies or assisting a third party in such activities. Except as expressly stated in this Notice & Disclaimer statement, no other license or right is granted to you by implication, estoppel or otherwise. NVIDIA objects to and rejects any additional or different terms you may propose.

NVIDIA Enterprise Reference Architecture is subject to change without notice. NVIDIA technologies may require enabled hardware, software, or service activation. Your costs and results may vary. This document is not a commitment to develop, release, or deliver any technology, code, or functionality. NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice. It is your sole responsibility to evaluate and determine the applicability of any information contained in this document and ensure that the product is suitable and fit for the application planned by you. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party or a license from NVIDIA.

NVIDIA products or services are subject to the applicable NVIDIA product or sales terms and conditions that are provided in conjunction with the relevant NVIDIA product or service. NVIDIA's provision of these resources does not expand or otherwise alter NVIDIA's applicable warranties or warranty disclaimers for the NVIDIA products or services.

You will not use the NVIDIA Enterprise Reference Architecture or NVIDIA's confidential information to: (i) identify or support an assertion or potential assertion of any intellectual property rights (including patent, copyright, or trade secret); or (ii) prepare, file, amend, or prosecute any patent applications. You agree to grant NVIDIA a limited, worldwide, nonexclusive, perpetual, irrevocable, non-sublicensable, nontransferable, royalty-free, fully-paid license to any patent claim that you may file after accessing NVIDIA Enterprise Reference Architecture that covers the subject matter disclosed herein.

**Warranty Disclaimer. NVIDIA Enterprise Reference Architecture materials disclosed are provided "AS IS". To the maximum extent permitted by applicable law, NVIDIA disclaims all warranties and representations of any kind, whether express, implied, or statutory, relating to or arising under this agreement, including, without limitation, the warranties of title, noninfringement, merchantability, fitness for a particular purpose, usage of trade, and course of dealing.**

**Limitation of Liability.** To the maximum extent permitted by applicable law, in no event will NVIDIA be liable for any (i) indirect, punitive, special, incidental, or consequential damages, or (ii) damages for the (a) cost of procuring substitute goods or (b) loss of profits, revenues, use, data, or goodwill arising out of or related to this document, whether based on breach of contract, tort (including negligence), strict liability, or otherwise, and even if NVIDIA has been advised of the possibility of such damages and even if a your remedies fail their essential purpose. Additionally, to the maximum extent permitted by applicable law, NVIDIA's total cumulative aggregate liability for any and all liabilities, obligations, or claims arising out of or related to this document will not exceed one-hundred U.S. dollars (US\$100).

NVIDIA, NVIDIA HGX, NVIDIA DGX SuperPOD, NVIDIA DGX Cloud, NVIDIA ConnectX, NVIDIA BlueField, NVIDIA Spectrum, NVIDIA NVLink, NVIDIA Base Command, NVIDIA CUDA, NVIDIA Magnum IO, NVIDIA NetQ, and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

© 2025 NVIDIA Corporation & Affiliates. All rights reserved.

Sunny Sehmi

# Contents

Overview .....	1
Prerequisites .....	2
Installation .....	4
Configure the Head Node Network Interfaces .....	15
Configure GRUB .....	15
Configure the Head Node Interfaces in cmsh .....	17
Licensing.....	21
Configure HA (High Availability) .....	24
Test the Connection to the NFS Share.....	25
Run cmha-setup on the Primary Head Node.....	29
Follow the Instructions on the Secondary Head Node.....	33
Finalize cmha-setup on the Primary Head Node .....	35
Configure Shared Storage.....	36
Test Failover and Reboot the Primary Head Node .....	39
Backup Defaults: Softwareimage and Category .....	44
Configure Defaults: Softwareimage and Category .....	47
Configure Device: Template Default Node .....	51
Create Custom Softwareimage and Category .....	54
Create Devices (Nodes).....	57
Appendix A. Related Links.....	68

---

# Overview

This Deployment Guide for NVIDIA Enterprise RAs – Base Command Manager provides a step-by-step procedure for installing **NVIDIA Base Command Manager (BCM)** on bare-metal cluster hardware. The instructions focus on concise, practical steps with minimal explanations so that a moderately experienced cluster administrator can get a cluster up and running in a standard configuration as quickly as possible. Base Command Manager is one of several orchestration options available for data center cluster management.

Base Command Manager is included in NVIDIA AI Enterprise, which brings together optimized microservices, frameworks and libraries for AI development with advanced GPU orchestration and infrastructure management into a production-grade software suite to power your business

It is strongly recommended that the cluster administrator be familiar with the NVIDIA Enterprise Reference Architecture document sets and the components that form their design and implementation. The steps are intended to be **hardware-agnostic**, meaning they can be adapted for hardware configurations different from those illustrated with minimal changes. Be sure to replace all example values with actual information relevant to the target environment, especially:

- > Cluster specific information such as names, locations and passwords
- > IP addresses, including name servers and gateways
- > Interface names

Information provided in code blocks reflects the context in which each command is executed. In most cases, these commands are run from within **cmsh** (Cluster Management Shell):

```
[nvidia-era-bcm-01]% device
```

However, some commands may be run as the **root** user directly from within the operating system:

```
root@nvidia-era-bcm-01:~# cmsh
```

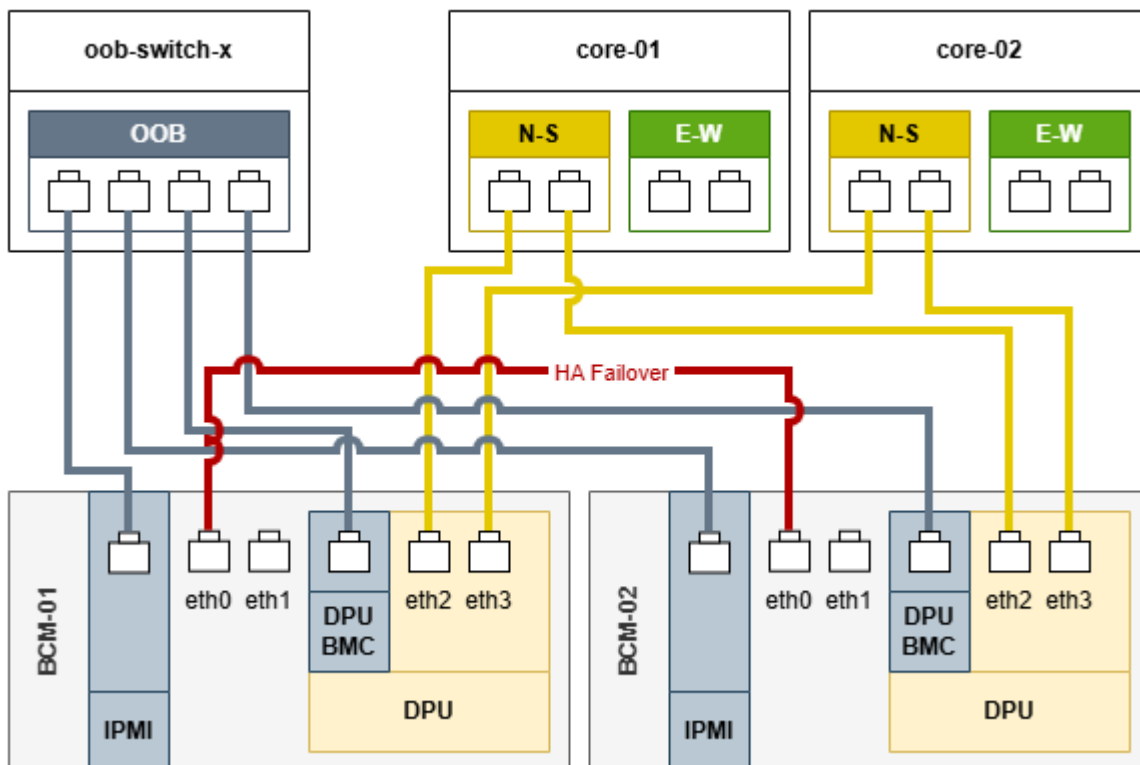
For more detailed information, please refer to [Appendix A: Related Links](#).

# Prerequisites

The following prerequisites must be met before deploying BCM:

- > The cluster administrator has access to the BCM installation media, a valid license, and a product key.
- > All hardware is installed and functional.
- > All firmware is updated to recommended versions.
- > All BIOS/UEFI firmware settings are properly configured.
- > For High Availability:
  - > A secondary head node, identical in hardware to the primary head node, is available and ready for configuration.
  - > An NFS share is created and fully accessible from the cluster.
  - > The network is configured in accordance with the Networking Configuration & Logical Architecture Guide for Enterprise RAs.

Diagram showing an example of the BCM head nodes network connections.



## Installation

Table describing the interfaces as shown in the diagram.

Port	Description	Network
IPMI	Server Management Interface	Out Of Band
eth0	Integrated NIC 1	High Availability Failover Network. A direct connection between the primary and secondary head nodes.
eth1	Integrated NIC 2	N/A
DPU BMC	DPU Management Interface	Out Of Band
eth2	DPU Network Interface 1	North-South
eth3	DPU Network Interface 2	North-South



**Note:** For High Availability, it is highly recommended that the NICs of both head nodes be directly connected.

If a direct connection is not feasible, a dedicated VLAN should be used.

---

# Installation

This section describes the minimal settings required to install NVIDIA Base Command Manager on the primary head node. While the options selected here should be customized to suit the target environment during installation, many settings can also be reconfigured post install.

For more detailed information, please refer to the [installation manual](#).

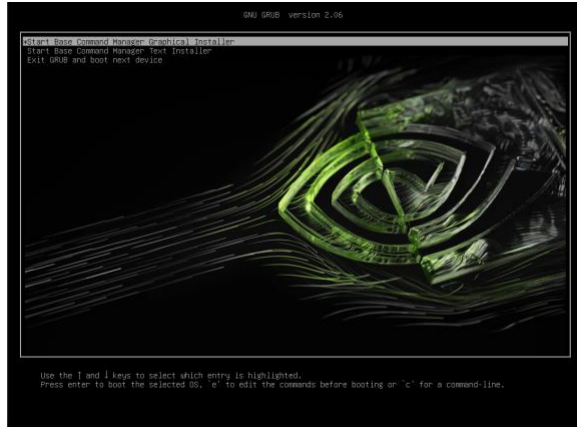
Details required to complete this section, with example values used in this guide:

Cluster Name	nvidia-era-bcm-cluster
Organization Name	NVIDIA
Time zone	(GMT-07:00) America/Los_Angeles
Name servers	192.168.255.25, 192.168.255.26
Search domains	nvidia.com
Hostname	nvidia-era-bcm-01
Base IP address	192.168.100.0
Netmask	255.255.255.0(/24)
Dynamic range start	192.168.100.101
Dynamic range end	192.168.100.250
Gateway	192.168.100.1
Head node interface	ens224
IP address	192.168.100.253
IP offset	0.0.0.20
Installation drives	/dev/nvme0n1



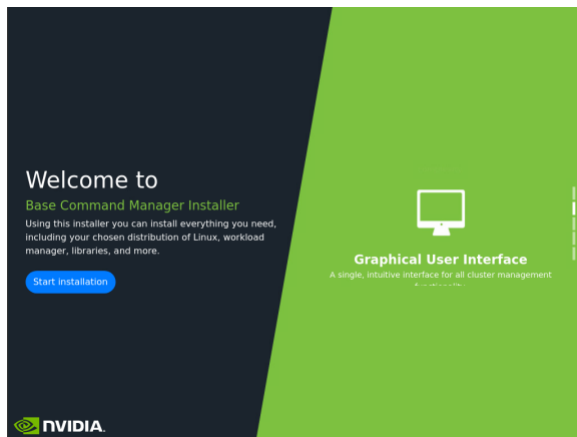
**Note:** Where applicable these values will be displayed in **red**. Be sure to replace all example values with actual information relevant to the target environment.

1. Power on and boot the primary head node server from the installation ISO.
2. Select **Start Base Command Manager Graphical Installer**.



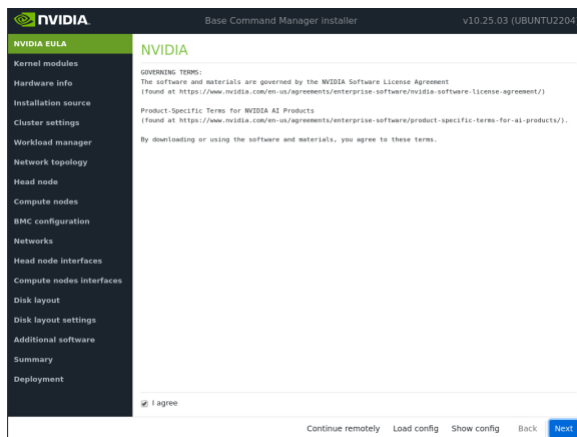
### 3. Welcome Screen.

- a. Select **Start Installation**.



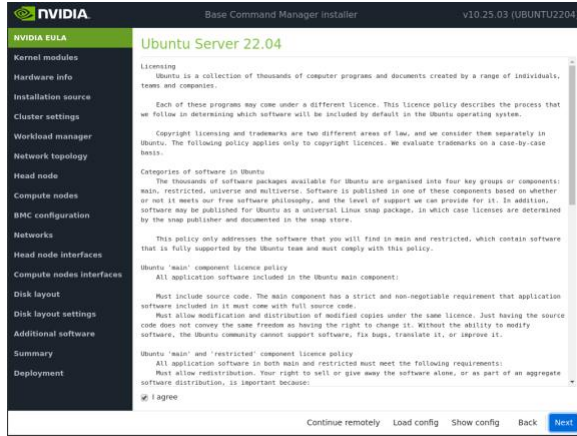
### 4. Software Licenses. NVIDIA EULA.

- a. Check the **I agree** checkbox.
- b. Click **Next** to continue.



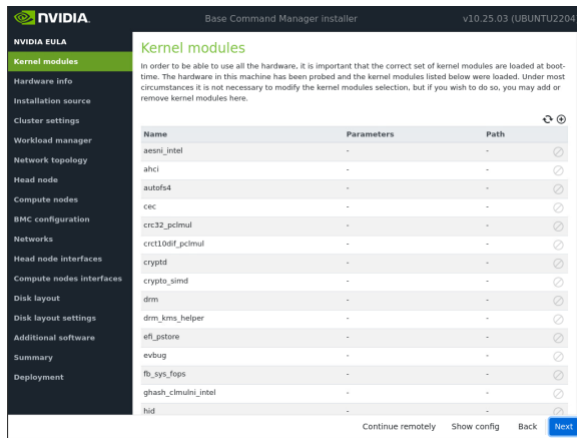
5. Software License. Ubuntu Server EULA.

- a. Check the **I agree** checkbox.
- b. Click **Next** to continue.



6. Kernel Modules Configuration.

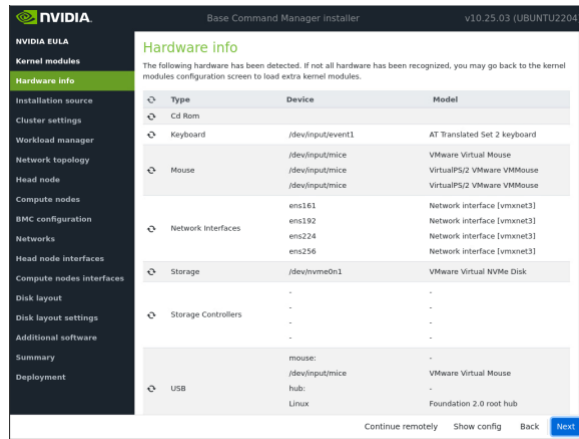
- a. Click **Next** to continue.



**Note:** The Kernel Modules screen displays the modules recommended for loading based on a hardware probe. In most cases, the default selections can be accepted without adding additional modules.

## 7. Hardware Info.

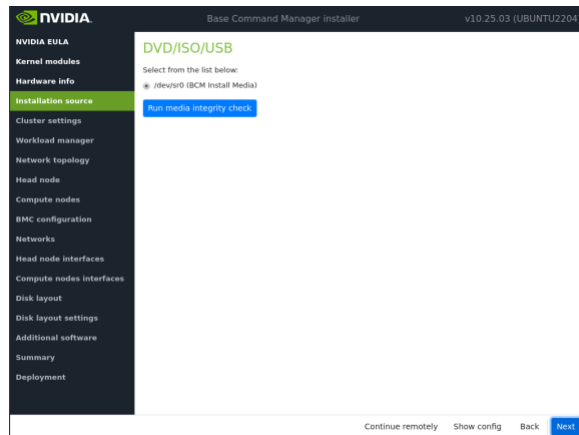
- a. Click **Next** to continue.



**Note:** The Hardware Info screen provides an overview of detected hardware based on the hardware probe used to load kernel modules. If any hardware is not detected at this stage, use the **Back** button to return to the Kernel Modules screen, add the appropriate modules, and then revisit the Hardware Info screen to verify detection.

## 8. Installation Source.

- a. Select the installation media from the list.
- b. Click **Next** to continue.



## 9. Cluster Settings.

- a. Enter information for the target environment:
  - i. **Cluster name:** `nvidia-era-bcm-cluster`
  - ii. **Organization name:** `NVIDIA`
  - iii. **Time zone:** `(GMT-07:00) America/Los_Angeles`
  - iv. **Name servers:** `192.168.255.25 192.168.255.26`
  - v. **Search domains:** `nvidia.com`
- b. Click **Next** to continue.

The screenshot shows the 'General cluster settings' screen in the NVIDIA Base Command Manager installer. The left sidebar lists various configuration steps, with 'Cluster settings' highlighted. The main area contains the following fields and options:

- Cluster name:** `nvidia-era-bcm-cluster`
- Organization name:** `NVIDIA`
- Administrator email:** (empty field)
- Send email to the administrator on first boot
- Time zone:** `(GMT-08:00) America/Los_Angeles`
- Time servers:** `0.pool.ntp.org 1.pool.ntp.org 2.pool.ntp.org`
- Name servers:** `192.168.255.25 192.168.255.26`
- Search domains:** `nvidia.com`

At the bottom, there are buttons for 'Continue remotely', 'Show config', 'Back', and 'Next'.

## 10. Workload Management Configuration.

- a. Select **None**.
- b. Click **Next** to continue.

The screenshot shows the 'HPC workload manager' screen in the NVIDIA Base Command Manager installer. The left sidebar lists various configuration steps, with 'Workload manager' highlighted. The main area contains the following information:

- HPC workload manager**
- A workload management system is highly recommended to run compute jobs. Please choose the one that should be configured or choose 'None' to prevent configuration.
- Please select workload manager:
- Options: **OpenPBS**, **PBS Pro**, **Slurm**, **IBM Spectrum LSF**, **Univa Grid Engine**, and **None** (selected).
- No workload manager will be configured on first boot.

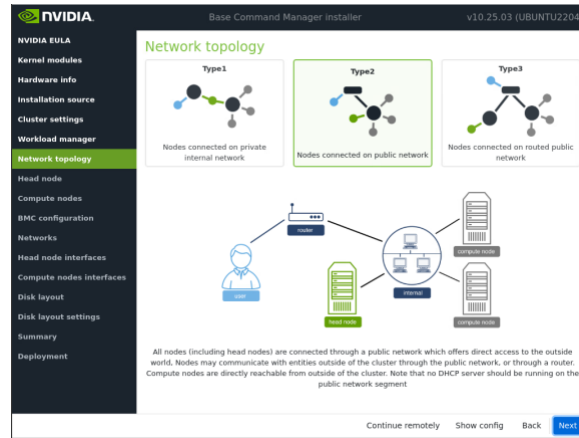
At the bottom, there are buttons for 'Continue remotely', 'Show config', 'Back', and 'Next'.



**Note:** If no workload manager is selected here, one can be installed later after completing the cluster installation without it. Instructions for installing a workload manager post-installation are provided in the BCM Administrator Manual.

## 11. Network Topology.

- a. Select **Type 2**.
- b. Click **Next** to continue.

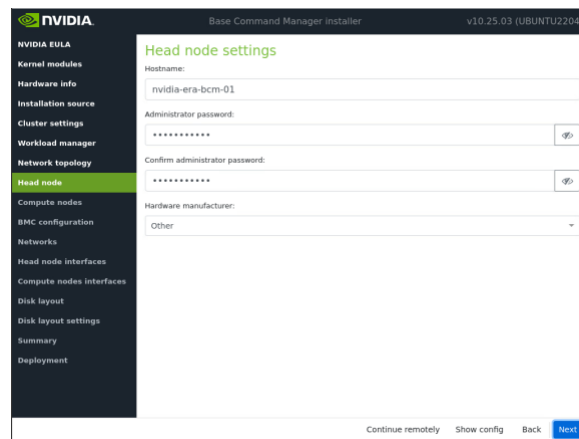


**Note:** The design provided is based on the deployment of a type 2 network - a type 2 network has the compute nodes connected via a router to a public network.

For more information on the different network topologies supported by BCM, please refer to the BCM Administrator Manual.

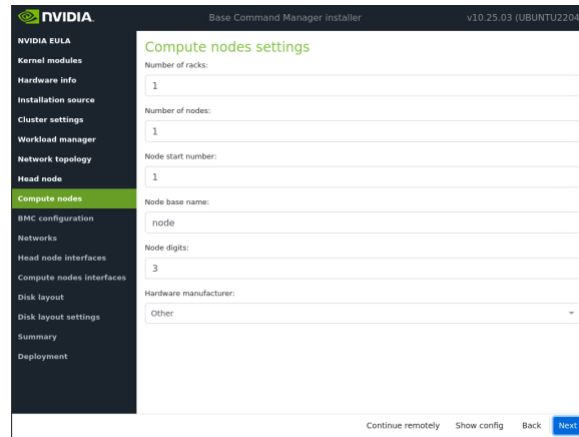
## 12. Head node Settings.

- a. Enter the following information:
  - i. **Hostname:** `nvidia-era-bcm-01`
  - ii. **Password:** <Password>
- b. Click **Next** to continue.



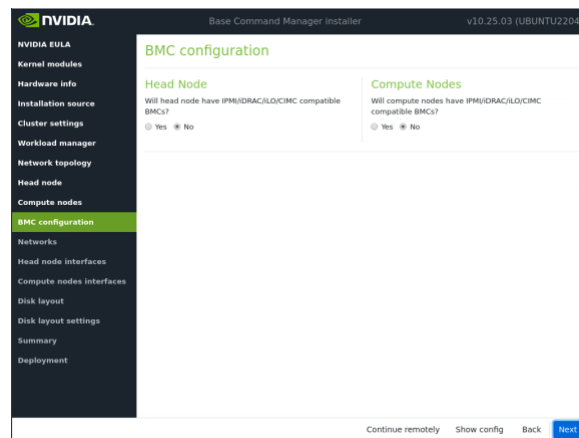
## 13. Compute Nodes Settings.

- a. Accept the default settings.
- b. Click **Next** to continue.



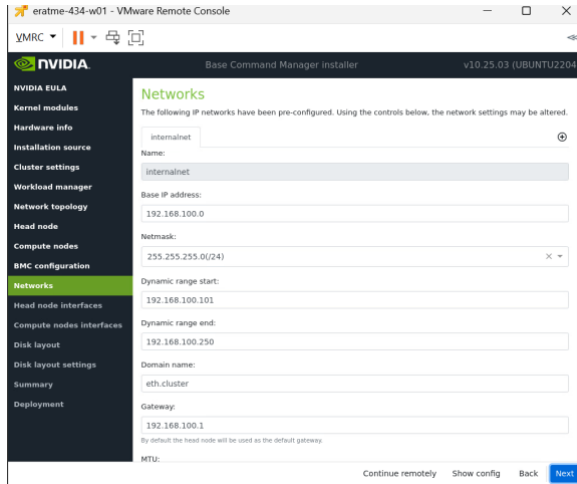
#### 14. BMC Configuration.

- a. Accept the default settings.
- b. Click **Next** to continue.



#### 15. Networks.

- a. Ensure **Internalnet** is selected.
- b. Enter the following information:
  - i. **Base IP address:** 192.168.100.0
  - ii. **Netmask:** 255.255.255.0 (/24)
  - iii. **Dynamic range start:** 192.168.100.101
  - iv. **Dynamic range end:** 192.168.100.250
  - v. **Gateway:** 192.168.100.1
- c. Click **Next** to proceed.



**Note:** As per the NVIDIA Enterprise Reference Architecture, the first 20 IP addresses are reserved for network services.

Setting the dynamic range start and end values suggested allows for other “static” or “reserved” IP addresses to be used in the range **192.168.100.21-192.168.100.100**.

The compute nodes utilize the DHCP assigned range **192.168.100.101-192.168.100.250**.

The BCM head nodes and High Availability shared IP address use the range **192.168.100.251-192.168.100.254**.

## 16. Head node Interfaces.

- a. Select the appropriate interface.



**Note:** This should be set to the first interface of the DPU.

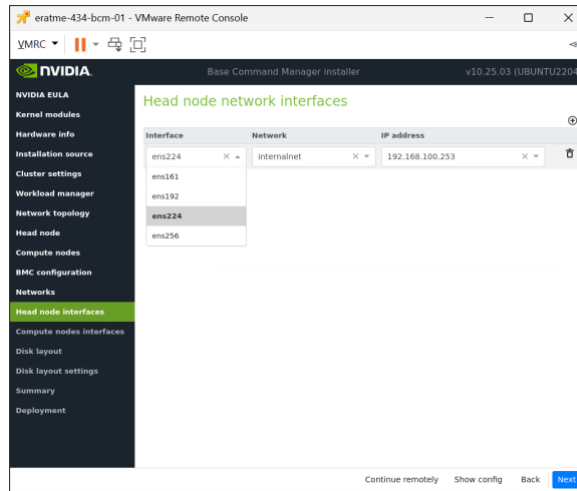
In this example, the interface is listed as **ens224**, but in the target environment it should be enumerated in a pattern similar to **enpXXs0f0np0**.

- b. Set the **IP address:** **192.168.100.253**



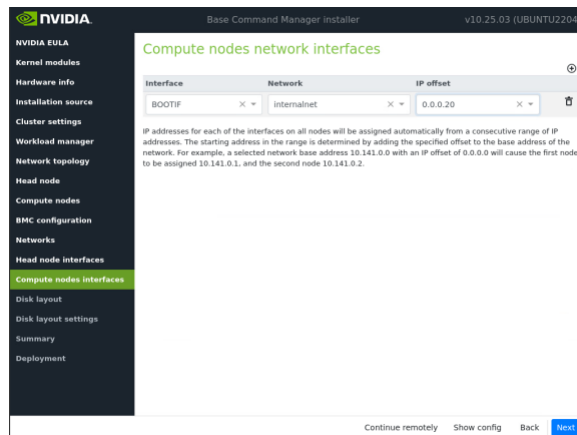
**Note:** It is suggested: use at least -1 from the last address in the usable range to allow for High Availability. For example: **192.168.100.253**.

c. Click **Next** to continue.



17. Compute node interfaces.

- a. Set the **IP offset**: `0.0.0.20`
- b. Click **Next** to continue.



**Note:** Setting the IP offset to 0.0.0.20 configures the IP address of the first compute node to 192.168.100.21. Later in this guide, this will be changed to use a DHCP assigned IP address.

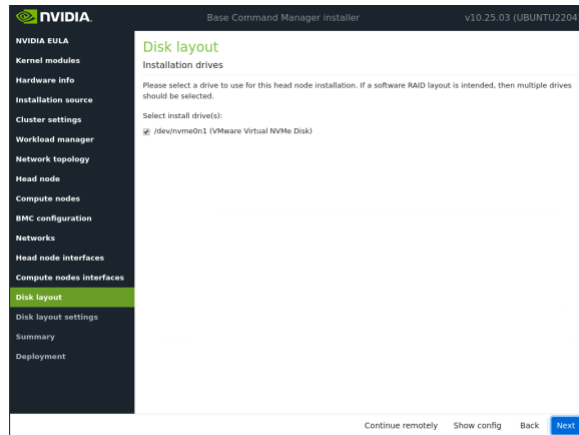
18. Disk Layout.

- a. Select the appropriate **installation drives**: `/dev/nvme0n1`



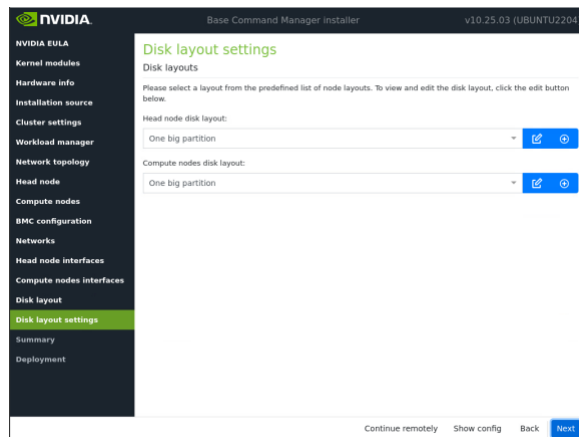
**Note:** In this example, there is only one drive listed, but in the target environment there may be multiple. It is recommended to select the hardware RAID configured disk.

- b. Click **Next** to proceed.



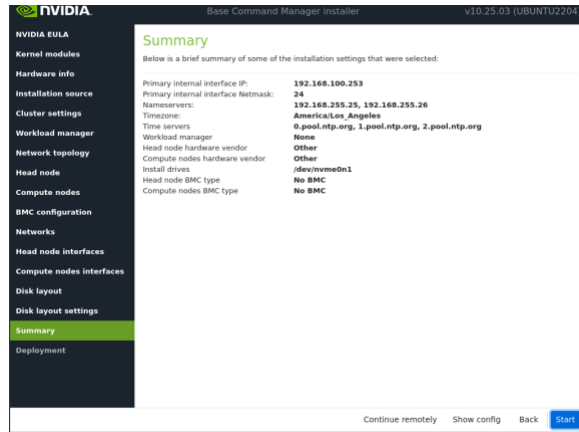
## 19. Disk Layout Settings.

- a. Head node disk layout:
  - i. Select **One big partition**. This setting cannot be not easily changed after installation.
- b. Compute node disk layout:
  - i. Select **One big partition**. This will be further configured after installation.
- c. Click **Next** to continue.



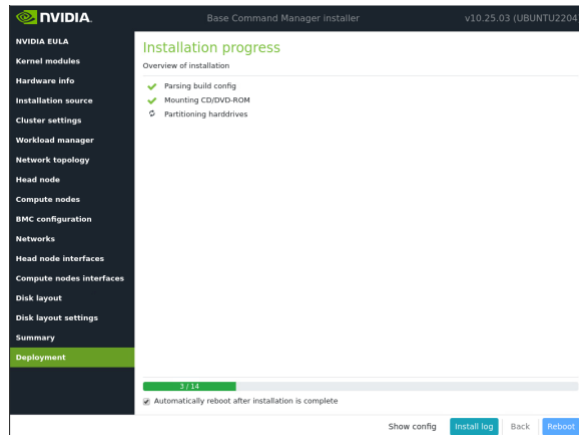
## 20. Summary.

- a. Review all configuration settings to ensure they are correct.
- b. Click **Start** to begin the installation.



## 21. Deployment.

- a. Check the **Automatically reboot after installation is complete** checkbox.



---

# Configure the Head Node Network Interfaces

In this section, the primary head node is configured with a bonded LACP interface on the N-S internal network. This interface is used for node image deployment as well as BCM management.

The head node is also configured to use the Pre-RHEL7 naming convention. This is a convenience option, as referring to traditional interface names (eth0, eth1) instead of the Post-RHEL7 scheme (ens192, enp0s3) can be easier when managing a large number of nodes with varying interfaces.

In this example, eth0 and eth1 are the integrated device interfaces on the out-of-band network, and eth1 and eth2 are the DPU interfaces on the N-S internal network.

Details required to complete this section, with example values used in this guide:

Existing interface name	ens224
IP address	192.168.100.253
Bond interfaces	eth2 eth3



**Note:** Where applicable these values will be displayed in **red**. Be sure to replace all example values with actual information relevant to the target environment.

## Configure GRUB

The first stage is to update the interfaces in the base operating system. This is typically handled by GRUB (Grand Unified Bootloader) and can be edited and updated using the commands below.

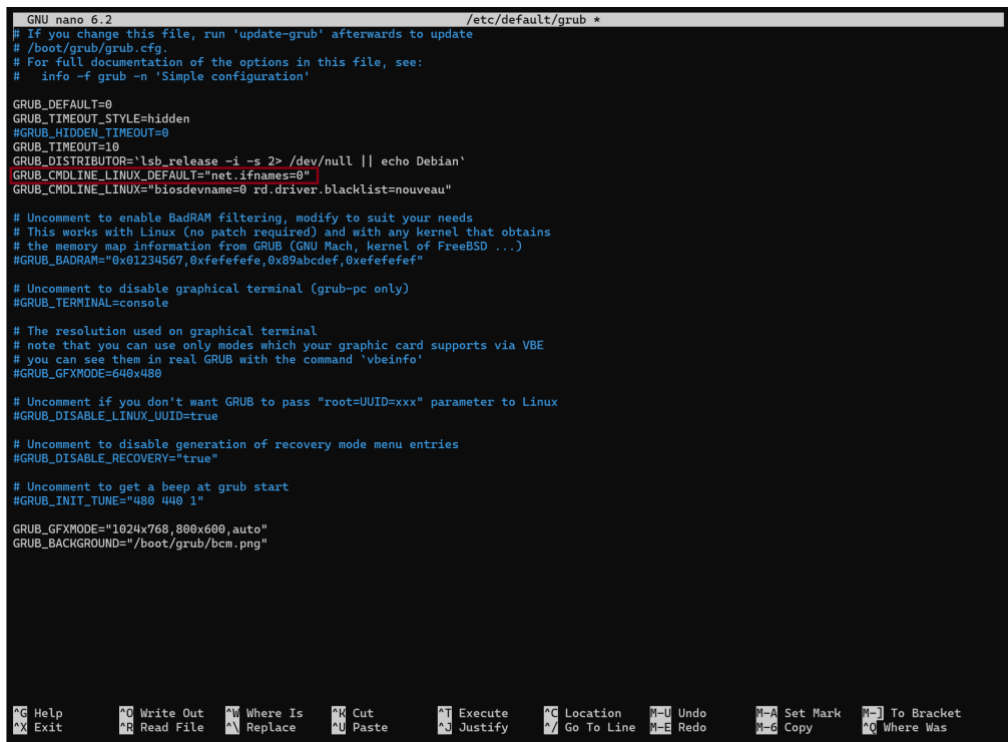
1. As a precaution, back up the existing grub configuration.

```
root@nvidia-era-bcm-01:# cp /etc/default/grub /etc/default/grub.BAK
```

2. Edit the grub configuration by amending the line `GRUB_CMDLINE_LINUX_DEFAULT` to include `net.ifnames=0`.

In this example nano is used as the editor, but the administrator should use the editor they are most comfortable with.

```
root@nvidia-era-bcm-01:~# nano /etc/default/grub
```



```
GNU nano 6.2 /etc/default/grub *
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
# info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
#GRUB_HIDDEN_TIMEOUT=0
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR='lsb_release -i -s 2> /dev/null || echo Debian'
GRUB_CMDLINE_LINUX_DEFAULT='net.ifnames=0'
GRUB_CMDLINE_LINUX='biosdevname=0 rd.driver.blacklist=nouveau'

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command 'vbeinfo'
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_RECOVERY="true"

# Uncomment to get a beep at grub start
#GRUB_INIT_TUNE="480 440 1"

GRUB_GFXMODE="1024x768,800x600,auto"
GRUB_BACKGROUND="/boot/grub/bcm.png"

^G Help      ^O Write Out  ^W Where Is  ^K Cut        ^T Execute   ^C Location  ^U Undo      ^M Set Mark  ^] To Bracket
^X Exit      ^R Read File  ^\ Replace   ^P Paste      ^J Justify   ^_ Go To Line ^E Redo      ^G Copy      ^O Where Was
```

Save (ctrl+s) and exit (ctrl+x).

3. Run the command `update-grub` for the changes to take effect on subsequent reboots. However, do not reboot the system yet.

```
root@nvidia-era-bcm-01:~# update-grub
```

The resulting output should be similar as below.

```
root@nvidia-era-bcm-01:~# update-grub
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found background: /boot/grub/bcm.png
Found background image: /boot/grub/bcm.png
Found linux image: /boot/vmlinuz-5.15.0-113-generic
Found initrd image: /boot/initrd.img-5.15.0-113-generic
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
Adding boot menu entry for UEFI Firmware Settings ...
done
```

## Configure the Head Node Interfaces in `cmsh`

Next, configure BCM to also use the traditional interface names.

1. Enter `cmsh`.

```
root@nvidia-era-bcm-01:~# cmsh
```

2. Enter the `device` mode.

```
[nvidia-era-bcm-01]% device
```

3. Select the active head node (in this example, using the “master” alias).

```
[nvidia-era-bcm-01->device]% use master
```

4. Enter the **interfaces** submenu.

```
[nvidia-era-bcm-01->device[nvidia-era-bcm-01]]% interfaces
```

5. List the current interfaces, noting the network device name and IP.

```
[nvidia-era-bcm-01->device[nvidia-era-bcm-01]->interfaces]% list
```

```
[nvidia-era-bcm-01->device[nvidia-era-bcm-01]->interfaces]% list
Type          Network device name  IP                Network          Start if
-----
physical     ens224 [prov]        192.168.100.253  internalnet      always
```

6. Remove the existing interface(s).

In this case, the interface is called **ens224**.

```
[nvidia-era-bcm-01->device[nvidia-era-bcm-01]->interfaces]% remove ens224
```

7. Add the physical interface **eth2**.

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]->interfaces*]% add physical eth2
```

8. After adding the interface, the context changes to the new interface. Exit to return to the **interfaces** submenu.

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]->interfaces*[eth2*]]% exit
```

9. Add the physical interface **eth3**.

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]->interfaces*]% add physical eth3
```

10. Exit to return to the **interfaces** submenu.

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]->interfaces*[eth3*]]% exit
```

11. Add and configure the bond interface bond0.

- a. Add the bond interface.

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]->interfaces*]% add bond bond0
```

- b. Disable the use of DHCP to acquire an IP address.

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]->interfaces*[bond0*]]% set dhcp no
```

- c. Set the IP address to the one used during installation.

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]->interfaces*[bond0*]]% set ip  
192.168.100.253
```

- d. Set the network to internalnet.

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]->interfaces*[bond0*]]% set network  
internalnet
```

- e. Set the interfaces (members) of the bond to use the newly created physical interfaces.

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]->interfaces*[bond0*]]% set interfaces  
eth2 eth3
```

- f. Set the bonding mode to 4 (802.3ad).

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]->interfaces*[bond0*]]% set mode 4
```

- g. Set the bonding options to `miimon=100` (media independent interface monitoring at 100 milliseconds.) and `xmit_hash_policy=layer3+4` (transmit hash policy).

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]->interfaces*[bond0*]]% set options  
"miimon=100 xmit_hash_policy=layer3+4"
```

12. Exit to return to the **interfaces** submenu.

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]->interfaces*[bond0*]]% exit
```

13. List and verify the interfaces.

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]->interfaces*]% list
```

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]->interfaces*]% list
Type           Network device name IP           Network      Start if
-----
bond           bond0             192.168.100.253 internalnet  always
physical      eth2 (bond0)      0.0.0.0
physical      eth3 (bond0)      0.0.0.0      always
```

14. Exit to return to the **device [head node]** submenu.

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]->interfaces*]% exit
```

15. Set the provisioning interface to bond0.

```
[nvidia-era-bcm-01->device[nvidia-era-bcm-01]]% set provisioninginterface bond0
```



**Caution:** The next two commands should be executed in quick succession. After committing the changes to the head node interface in cmsH, a reboot is required immediately. Otherwise, remote access will be lost and recovery will require the server's remote console/KVM or other intervention.

16. Commit the changes.

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]]% commit
```

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]]% commit
[nvidia-era-bcm-01->device[nvidia-era-bcm-01]]%
Thu Aug 7 05:36:03 2025 [notice] nvidia-era-bcm-01: nvidia-era-bcm-01 [ UP ], restart required (interface:bond0)
```

17. Reboot immediately.

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]]% reboot
```

---

# Licensing

A license will now be requested and installed to enable the full features of BCM.

Details required to complete this section, with example values used in this guide:

Product Key	000354-515786-112224-207440-186713
Country	US
State or Province Name	California
Locality Name	Santa Clara
Organization Name	NVIDIA
Organizational Unit Name	nvidia-era
Cluster Name	nvidia-era-bcm-cluster
MAC Address of primary head node	00:50:56:04:34:00
MAC Address of secondary head node	00:50:56:04:34:10



**Note:** Where applicable these values will be displayed in **red**. Be sure to replace all example values with actual information relevant to the target environment.



**Note:** It is recommended to use the MAC address of the first integrated network interface (eth0) for licensing, as opposed to a PCIe or other peripheral interface as it is less likely to be replaced. There are several ways to obtain the MAC address, which are beyond the scope of this guide. Possible methods include referencing the IPMI BMC, or from an operating system running on the host.

1. Enter **cmsh** (if not already within it).

```
root@nvidia-era-bcm-01:~# cmsh
```

2. Enter the **main** mode

```
[nvidia-era-bcm-01]% main
```

3. Get the current license information.

```
[nvidia-era-bcm-01->main]% licenseinfo
```

```
[nvidia-era-bcm-01->main]% licenseinfo
License Information
-----
Licensee                /C=US/ST=California/L=Santa Clara/O=NVIDIA Inc/OU=Bright Licensing/CN=nvidia-era-bcm-01
Serial number           99
Start time              Wed Aug 6 03:56:19 2025
End time                Fri Aug 7 03:56:19 2026
Version                 19 and above
Edition                 Advanced
Type                    Temporary
Licensed nodes          1 / 2
Licensed accelerators   1 / 2
Allow edge sites        No
MAC address / Cloud ID  08:58:56:04:34:02
```

4. Identify the MAC address of both primary and secondary head nodes.



**Note:** It is recommended to use a MAC address that will not change if peripheral components (such as PCIe NICs) are replaced. Therefore, obtain the MAC address of an integrated NIC (eth0) for licensing. There are several ways to obtain this information, which are beyond the scope of this guide. Possible methods include using IPMI BMC or checking directly from the operating system running on the host.

5. Request a license.

- a. Exit cmsh and return to the OS root.

```
[nvidia-era-bcm-01]% quit
```

```
[nvidia-era-bcm-01->main]% quit
root@nvidia-era-bcm-01:~#
```

- b. Run the request-license command.

```
root@nvidia-era-bcm-01:~# request-license
```

- c. Follow the prompts, entering information specific to the target environment.

- > Product Key
- > Country Name
- > State or Province Name
- > Locality Name
- > Organization Name
- > Organizational Unit Name
- > Cluster Name
- > MAC Address of primary head node
- > 'y' to the question **"Submit certificate request...?"**

- > 'y' to the question **“Will this cluster use a high-availability setup with 2 head nodes?”**
- > MAC Address of secondary head node
- > 'y' to the question **“Install license?”**
- > 'y' to the question **“Is the license information correct?”**

```

root@nvidia-era-bcm-01:~# request-license
Product Key (XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX): 000354-515786-112224-207440-186713

Country Name (2 letter code): US
State or Province Name (full name): California
Locality Name (e.g. city): Santa Clara
Organization Name (e.g. company): NVIDIA
Organizational Unit Name (e.g. department): nvidia-era
Cluster Name: nvidia-era-bcm-cluster
Private key data saved to /cm/local/apps/cmd/etc/cluster.key.new

MAC Address of primary head node (nvidia-era-bcm-01) for eth2 [00:50:56:04:34:02]: 00:50:56:04:34:00
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]: 00:50:56:00:34:10

Certificate 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
mysql: [Warning] Using a password on the command line interface can be insecure.
===== Certificate Information =====
Version:          10
Edition:          Advanced
OEM:              NVIDIA
Common name:      nvidia-era-bcm-cluster
Organization:     NVIDIA
Organizational unit: nvidia-era
Locality:         Santa Clara
State:            California
Country:          US
Serial:           3256787
Starting date:    03/Oct/2023
Expiration date:  07/Aug/2026
MAC address / Cloud ID: 00:50:56:04:34:00|00:50:56:04:34:10
Licensed tokens: 8192
Accounting & Reporting: Yes
Allow edge sites: Yes
License type:
=====

Is the license information correct ? [Y/n] y
Backup directory of old license: /var/spool/cmd/backup/certificates/2025-08-07_05.57.34
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 2 images / node-installers
Copy cluster certificate to /cm/images/default-image/cm/local/apps/cmd/etc/cluster.pem
Copy cluster certificate to /cm/node-installer/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

```

---

# Configure HA (High Availability)

In a cluster with a single head node, the head node is a single point of failure for the entire cluster. In many environments, it is unacceptable for the failure of a single machine to disrupt daily operations.

High Availability (HA) configuration for a head node involves adding a secondary head node to provide redundant head node services. If one head node fails, the other can take over, ensuring service continuity with minimal downtime.

HA is typically configured using shared storage, commonly provided by an NFS service, which serves the /home directory on the active head node, and on the regular nodes. In this case, it will also be configured to provide access to the /cm/shared directory. This ensures that no matter which head node is active, access to these folders can be maintained.

In brief, the steps to be taken are:

- > Test the connection to the NFS share.
- > On the primary head node, configure and initialize the HA setup.
- > On the secondary head node, boot from network into “Rescue” mode and execute the cloning process. Note that the BCM ISO is not required.
- > Back on the primary head node, finalize the HA setup.
- > Configure shared storage to use the NFS share.
- > Test the failover, allowing for rebooting the head nodes sequentially.

Details required to complete this section, with example values used in this guide:

NFS Share	192.168.255.31:/nvidia-era-bcm-cluster
HA shared interface name	bond0:ha
HA shared internal interface IP address	192.168.100.254
Hostname of secondary head node	nvidia-era-bcm-02
Failover network name	failovernet
Failover network base address	192.168.120.0
Failover network netmask	255.255.255.0
Failover network domain name	failover.cluster
Failover network interface (primary)	eth0
Failover network IP (primary)	192.168.120.11
Failover network interface (secondary)	eth0
Failover network IP (secondary)	192.168.120.12

IP address for secondary head node	192.168.100.252
------------------------------------	-----------------



**Note:** Where applicable these values will be displayed in **red**. Be sure to replace all example values with actual information relevant to the target environment.



**Note:** While the creation of the NFS share is beyond the scope of this guide, ensure that the exports are configured to allow access from the internalnet subnet (in this example, 192.168.100.0/24) and that the “no\_root\_squash” option is enabled.

## Test the Connection to the NFS Share

Before setting up HA, it is recommended to test the connection to the NFS share. If the share is inaccessible, the node health check will fail, as the /home and /cm/shared directories will not be unavailable.

This test will be performed in two stages. In the first stage, the NFS share will be mounted from the root operating system, where a “ha” directory will be created with a test txt file within it. In the second stage, the ha folder will be mounted from within cmsh, and the test file will be listed to confirm its creation. Two additional directories, home and cmshared, will then be created to map the folders after the main high-availability setup is completed.

1. Mount the NFS share from the root OS.
  - a. Exit cmsh and return to the OS root.

```
[nvidia-era-bcm-01]% quit
```

- b. Mount the NFS share to the /mnt directory.

```
root@nvidia-era-bcm-01:~# mount 192.168.255.31:/nvidia-era-bcm-cluster/ /mnt
```

- c. List the contents of the /mnt/ directory to ensure there are no existing files.

```
root@nvidia-era-bcm-01:~# ls -alh /mnt/
```

```
root@nvidia-era-bcm-01:~# ls -alh /mnt/
total 4.0K
drwxr-xr-x  2 root root 4.0K Aug  7 09:17 .
drwxr-xr-x 23 root root 330 Aug  7 03:47 ..
```

- d. Create the “ha” directory.

```
root@nvidia-era-bcm-01:~# mkdir /mnt/ha
```

- e. List the contents of the /mnt/ directory to validate the folder exists.

```
root@nvidia-era-bcm-01:~# ls -alh /mnt/
```

- f. Create the test.txt file.

```
root@nvidia-era-bcm-01:~# touch /mnt/ha/test.txt
```

- g. List the contents of the /mnt/ha/ directory to verify that the file was created.

```
root@nvidia-era-bcm-01:~# ls -alh /mnt/ha/
```

```
root@nvidia-era-bcm-01:~# ls -alh /mnt/ha/
total 8.0K
drwxr-xr-x 2 root root 4.0K Aug  7 11:18 .
drwxr-xr-x 3 root root 4.0K Aug  7 09:23 ..
-rw-r--r-- 1 root root   0 Aug  7 11:18 test.txt
```

- h. Unmount the share.

```
root@nvidia-era-bcm-01:~# umount /mnt
```

2. Mount the NFS share using cmsh and validate the directory and test file.

- a. Enter cmsh.

```
root@nvidia-era-bcm-01:~# cmsh
```

- b. Enter the **device** mode.

```
[nvidia-era-bcm-01]% device
```

- c. Use the active head node (in this case using the master alias).

```
[nvidia-era-bcm-01->device]% use master
```

- d. Enter the **fsmounts** submenu.

```
[nvidia-era-bcm-01->device[nvidia-era-bcm-01]]% fsmounts
```

- e. Add a mount point /nfs/general.

```
[nvidia-era-bcm-01->device[nvidia-era-bcm-01]->fsmounts]% add /nfs/general
```

- f. Set the device to the NFS share.

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]->fsmounts*[nvidia-era-bcm-01*]]% set device  
192.168.255.31:/nvidia-era-bcm-cluster/ha
```

- g. Set the filesystem to NFS.

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]->fsmounts*[nvidia-era-bcm-01*]]% set  
filesystem nfs
```

- h. Commit the changes.

```
[nvidia-era-bcm-01->device*[nvidia-era-bcm-01*]->fsmounts*[nvidia-era-bcm-01*]]% commit
```

- i. List the fsmounts to verify.

```
[nvidia-era-bcm-01->device[nvidia-era-bcm-01]->fsmounts[/nfs/general]]% list
```

```
[nvidia-era-bcm-01->device[nvidia-era-bcm-01]->fsmounts[/nfs/general]]% list
Device                               Mountpoint (key)                    Filesystem
-----
192.168.255.31:/nvidia-era-bcm-cluster/ha /nfs/general                        nfs
```

- a. Exit cmsh and return to the OS root.

```
[nvidia-era-bcm-01->device[nvidia-era-bcm-01]->fsmounts[/nfs/general]]% quit
```

- j. List the contents of /nfs/general. The test.txt file created previously should be present.

```
root@nvidia-era-bcm-01:~# ls -alh /nfs/general/
```

```
root@nvidia-era-bcm-01:~# ls -alh /nfs/general/
total 4.0K
drwxr-xr-x 2 root root 4.0K Aug  7 11:18 .
drwxr-xr-x 3 root root  21 Aug  7 12:58 ..
-rw-r--r-- 1 root root   0 Aug  7 11:18 test.txt
```

- k. Create the home folder.

```
root@nvidia-era-bcm-01:~# mkdir /nfs/general/home
```

- l. Create the cmshared folder.

```
root@nvidia-era-bcm-01:~# mkdir /nfs/general/cmshared
```

- m. List the contents of /nfs/general to confirm the folders exist.

```
root@nvidia-era-bcm-01:~# ls -alh /nfs/general/
```

```
root@nvidia-era-bcm-01:~# ls -alh /nfs/general/
total 12K
drwxr-xr-x 4 root root 4.0K Aug  7 13:18 .
drwxr-xr-x 3 root root  21 Aug  7 12:58 ..
drwxr-xr-x 2 root root 4.0K Aug  7 13:18 cmshared
drwxr-xr-x 2 root root 4.0K Aug  7 13:18 home
-rw-r--r-- 1 root root   0 Aug  7 11:18 test.txt
```

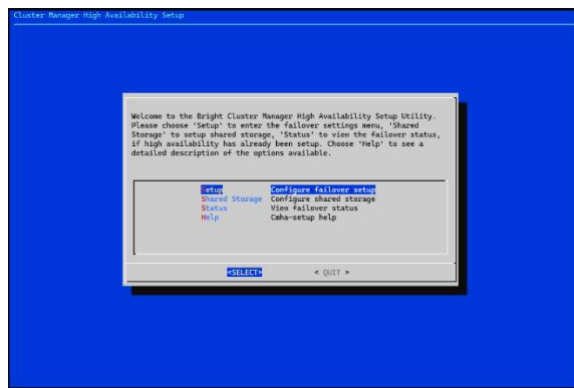
# Run cmha-setup on the Primary Head Node

HA can now be configured on the primary head node.

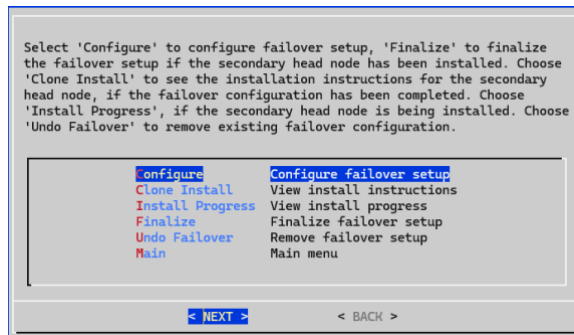
1. Run `cmha-setup`.

```
[nvidia-era-bcm-01]% quit # only required if in cmsh  
root@nvidia-era-bcm-01:~# cmha-setup
```

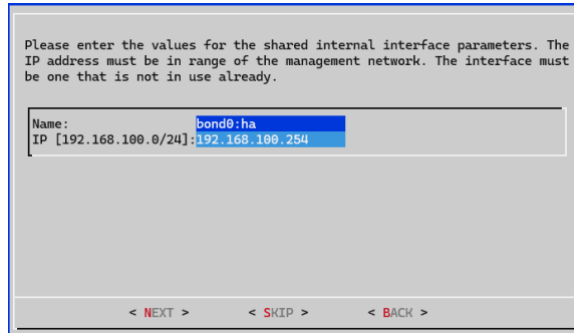
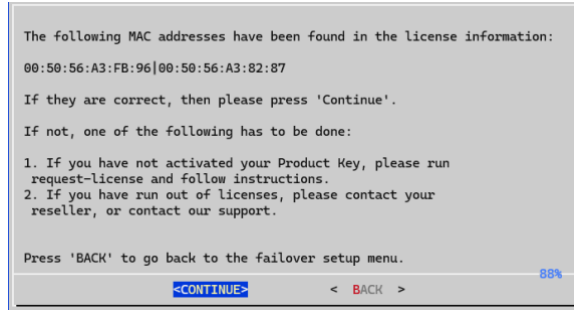
2. Select **Setup**.



3. Select **Configure**.

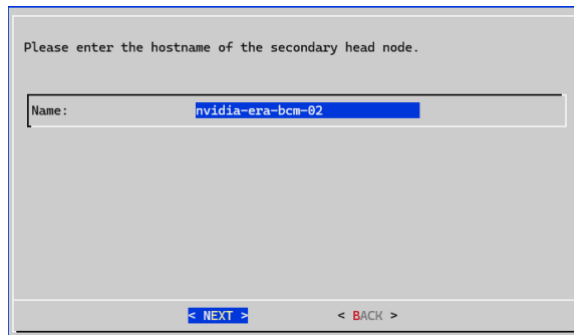


4. Select **Continue**.
5. Enter the HA shared interface name and IP address.
  - a. **Name:** `bond0:ha`
  - b. **IP:** `192.168.100.254`



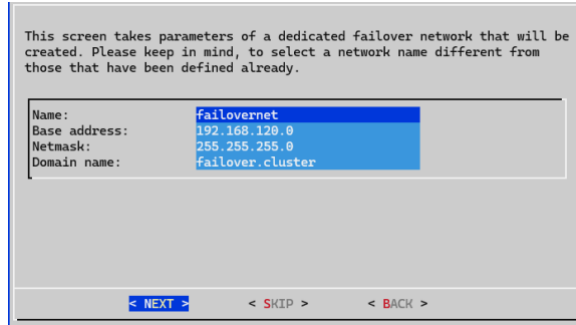
6. Enter the hostname of the secondary node.

- a. **Name:** nvidia-era-bcm-02



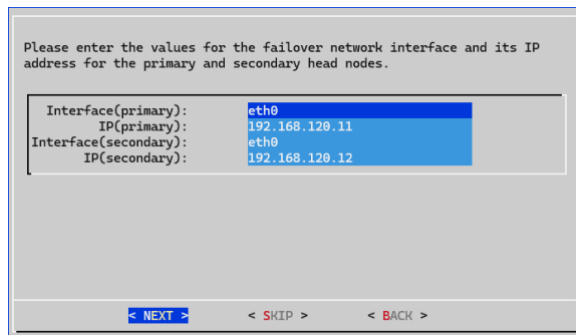
7. Set the failover network.

- a. **Name:** failovernet  
b. **Base address:** 192.168.120.0  
c. **Netmask:** 255.255.255.0  
d. **Domain name:** failover.cluster



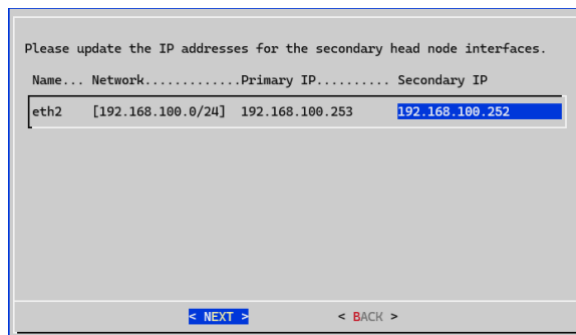
8. Set the failover network interfaces.

- a. **Interface (primary):** eth0
- b. **IP (primary):** 192.168.120.11
- c. **Interface (secondary):** eth0
- d. **IP (secondary):** 192.168.120.12

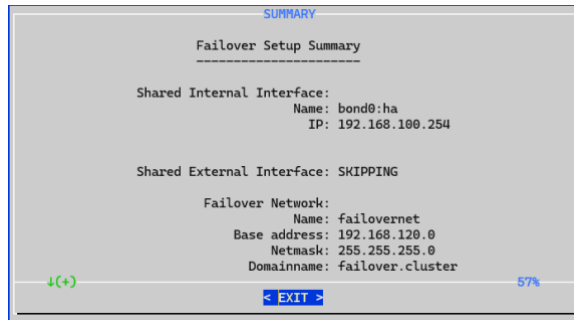


9. Set the IP address for the secondary head node.

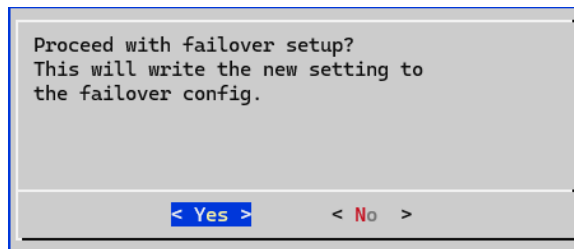
- a. **Secondary IP:** 192.168.100.252



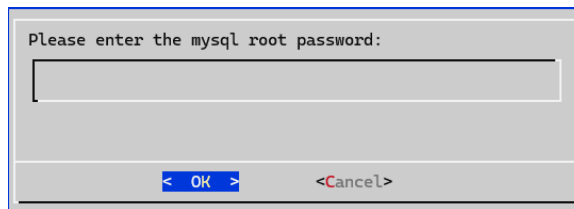
10. Review the summary and select **Exit**.



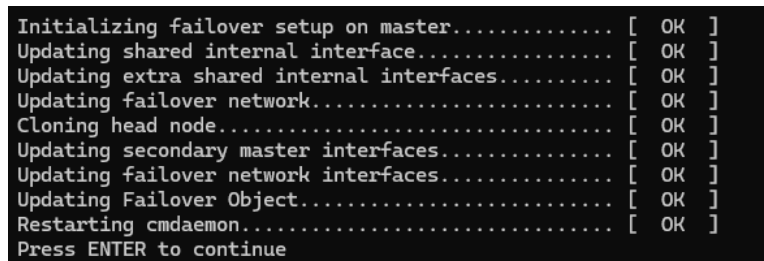
11. Select **Yes** to proceed.



12. Enter the **mysql root password**.



13. Wait until all tasks are completed, then press **Enter**.



```
The failover setup initialization on the primary master is done.
Now boot the secondary master into the rescue environment and run the
following command:

/cm/cm-clone-install --failover

and follow the instructions.

Once the installation has begun, select 'Install Progress' from the
Failover setup menu, to see
the installation progress of the clone machine. When the installation is
complete, and the
secondary master is up, select 'Finalize' from the Failover setup menu,
to complete the
failover setup process.

94%
< OK >
```

# Follow the Instructions on the Secondary Head Node

The secondary head node can now be powered on, booted from network into the rescue environment, and cloned from the primary head node using the failover parameter. Note that the BCM ISO is not required for this process.

- 1. Power on and boot the secondary head node server from the network (PXE).
- 2. Select **Start rescue environment**.

```
Cluster Manager PXE Environment
AUTO      - Normal node boot
MAIN      - Drop to maintenance shell
RESCUE    - Start rescue environment
MEMPLUS   - Start Mementest86+
MEMPLUS   - Start Mementest86+

Press [Tab] to edit options
```

```
«Welcome to the Cluster Manager rescue environment»
-----
|
| Creating failover/clone nodes:
| * Install the secondary head node
|   $ /cm/cm-clone-install --failover
|
| * Create a clone of the primary head node
|   $ /cm/cm-clone-install --clone --hostname=new-hostname
|
| * Install the secondary (failover) head node and reboot automatically
|   $ /cm/cm-clone-install --failover --reboot
|
| * Help
|   $ /cm/cm-clone-install --help
|
-----
ClusterManager login: root (automatic login)
Linux ClusterManager 5.15.0-113-generic #123-Ubuntu SMP Mon Jun 10 08:16:17 UTC 2024 x86_64
Last login: Tue May 13 15:10:48 UTC 2025 on tty2
root@ClusterManager:~#
```

- Determine the network interface to use for configuration.

```
root@ClusterManager:~# ip a
```

```
root@ClusterManager:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
   link/ether 00:50:56:a3:41:13 brd ff:ff:ff:ff:ff:ff
   altname enp4s0
   altname ens161
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
   link/ether 00:50:56:a3:f6:77 brd ff:ff:ff:ff:ff:ff
   altname enp11s0
   altname ens192
4: eth2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
   link/ether 00:50:56:a3:82:87 brd ff:ff:ff:ff:ff:ff
   altname enp12s0
   altname ens193
5: eth3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
   link/ether 00:50:56:a3:ca:38 brd ff:ff:ff:ff:ff:ff
   altname enp27s0
   altname ens256
root@ClusterManager:~# _
```

- Clone install failover (non-interactive).

```
root@ClusterManager:~# /cm/cm-clone-install \
--failover \
--noninteract \
--headnodepasswd=secret \
--interface=eth2 \
--addbiosboot
```

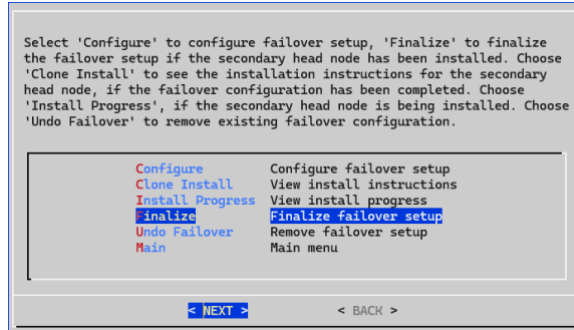
Wait until all tasks are completed. The secondary head node will automatically reboot.

```
Interface specified at the command line: eth2
Please wait while bringing up network...
Please wait while authentication is being set up...
Enter the password of the headnode node to continue.
Please wait while installation begins...
Verifying license ..... [ OK ]
Getting disk layout ..... [ OK ]
The head node disk layout is saved in /cm/_headnodedisksetup.xml
Detecting device '/dev/sda': found
Valid device sda. All checks have succeeded.
The contents of the following disks will be erased.
/dev/sda
Getting mount points ..... [ OK ]
Partitioning hard drive ..... [ OK ]
Mounting partitions ..... [ OK ]
Syncing hard drive ..... [ OK ]
Finalizing installation ..... [ OK ]
```

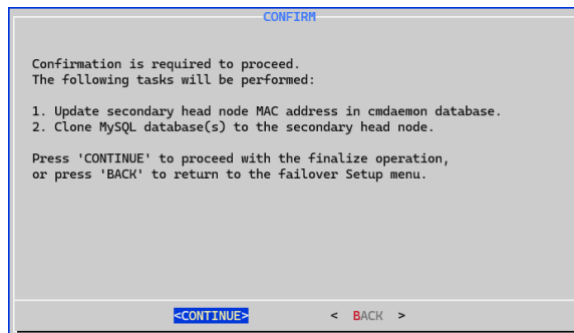
# Finalize cmha-setup on the Primary Head Node

Back on the primary head node, the HA process must now be finalized.

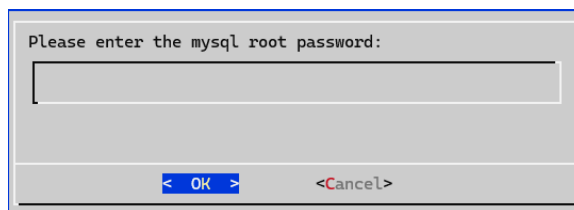
1. From the **cmha-setup** > **Setup TUI**, select **Finalize**.



2. Select **Continue**.



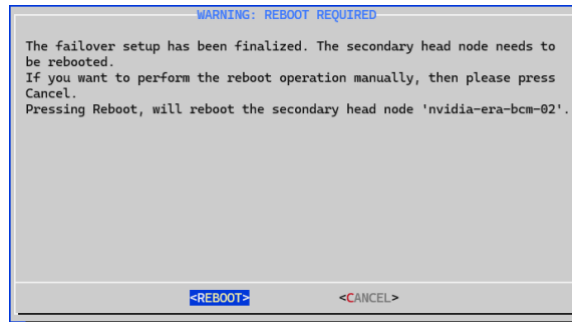
3. Enter the **mysql root password**.



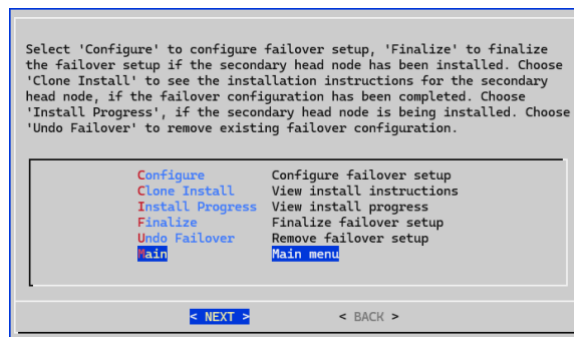
4. Wait until all tasks are complete, then press **Enter**.

```
Updating secondary master mac address..... [ OK ]
Initializing failover setup on nvidia-era-bcm-02... [ OK ]
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 ENTER to continue
```

5. Select **Reboot** to reboot the secondary head node.



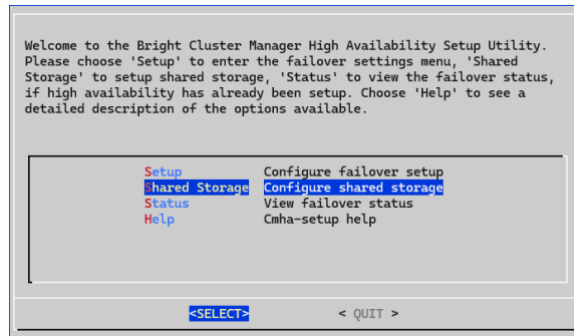
6. Select **Main**.



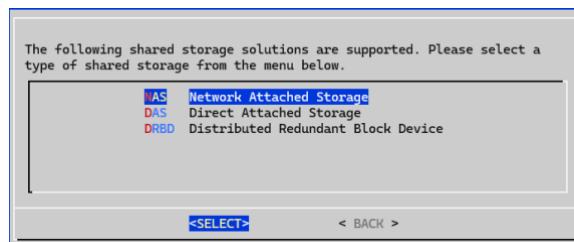
## Configure Shared Storage

Shared storage can now be configured to use the home and cmshared directories from earlier.

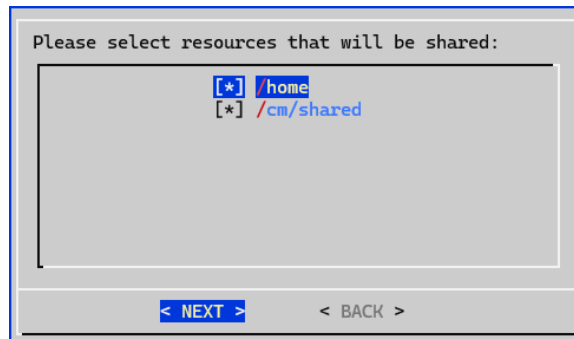
1. In **cmha-setup > TUI**, select **Shared Storage**.



2. Select **NAS**.

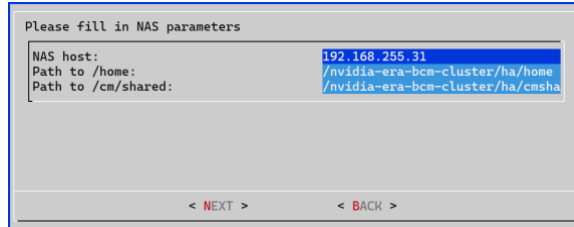


3. Select the resources that will be shared:
  - a. **/home**
  - b. **/cm/shared**

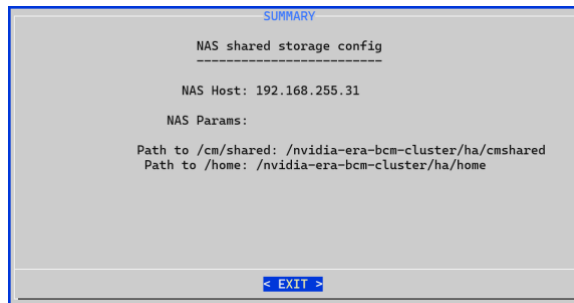


4. Enter the NAS parameters.

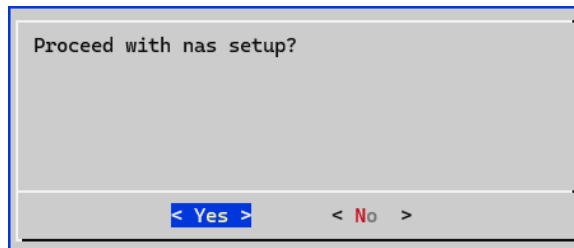
- a. **NAS host:** 192.168.255.31
- b. **Path to /home:** /nvidia-era-bcm-cluster/ha/home
- c. **Path to /cm/shared:** /nvidia-era-bcm-cluster/ha/cmshared



5. Review the summary and select **Exit**.



6. Proceed with nas setup by selecting **Yes**.



7. Wait for tasks to complete, then press **Enter**.



8. Exit **cmha-setup**. Select **Exit**, then **Quit**, then **Yes**.

## Test Failover and Reboot the Primary Head Node

From this point forward, it is recommended to log in using the HA shared internal interface IP address defined during the HA setup (in this example, **192.168.100.254**). This ensures that administrators are always directed to the active head node, which is currently still the primary head node.

In the event of a failover, this IP address will be assumed by the secondary head node, minimizing downtime and service disruption. To proceed, the primary head node will need a reboot. A failover will be simulated by making the secondary head node the active head node, allowing the primary head node to reboot gracefully.

1. Log off from the primary head node.

```
root@nvidia-era-bcm-01:~# exit
```

2. Log in using the HA shared internal interface IP address.

```
ssh root@192.168.100.254
```

```
Node Status: running in active master mode
-----
root@nvidia-era-bcm-01:~#
```



**Note:** The primary head node (nvidia-era-bcm-01) is still the active head node.

3. Enter **cmsh**.

```
root@nvidia-era-bcm-01:~# cmsh
```

4. Check the failover status and make the secondary head node the active head node.

- a. Enter the **partition** mode.

```
[nvidia-era-bcm-01]% partition
```

- b. Enter the **failover** submenu.

```
[nvidia-era-bcm-01->partition[base]]% failover
```

- c. Get the failover status.

```
[nvidia-era-bcm-01->partition[base]->failover]% status
```

```
[nvidia-era-bcm-01->partition[base]->failover]% status
nvidia-era-bcm-01* -> nvidia-era-bcm-02
failoverping [ OK ]
mysql [ OK ]
ping [ OK ]
status [ OK ]
nvidia-era-bcm-02 -> nvidia-era-bcm-01*
failoverping [ OK ]
mysql [ OK ]
ping [ OK ]
status [ OK ]
```

- d. Make the secondary head node active. (Pressing **Enter** after this command will disconnect the client.)

```
[nvidia-era-bcm-01->partition[base]->failover]% makeactive nvidia-era-bcm-02
```

```
[nvidia-era-bcm-01->partition[base]->failover]% makeactive nvidia-era-bcm-02
client_loop: send disconnect: Connection reset
```

5. Log in again using the HA shared internal interface IP address.

```
ssh root@192.168.100.254
```

```
Node Status: running in active master mode
```

```
-----
root@nvidia-era-bcm-02:~#
```



**Note:** The secondary head node (nvidia-era-bcm-02) is now the active head node.

6. Enter cmsh again.

```
root@nvidia-era-bcm-02:~# cmsh
```

7. Enter the **partition** mode.

```
[nvidia-era-bcm-02]% partition
```

8. Enter the **failover** submenu.

```
[nvidia-era-bcm-02->partition[base]]% failover
```

9. Get the failover status.

```
[nvidia-era-bcm-02->partition[base]->failover]% status
```

```
[nvidia-era-bcm-02->partition[base]->failover]% status
nvidia-era-bcm-02* -> nvidia-era-bcm-01
failoverping [ OK ]
mysql        [ OK ]
ping         [ OK ]
status       [ OK ]
nvidia-era-bcm-01 -> nvidia-era-bcm-02*
failoverping [ OK ]
mysql        [ OK ]
ping         [ OK ]
status       [ OK ]
```

10. Reboot the primary head node.

- a. Enter the **device** mode

```
[nvidia-era-bcm-02->partition[base]->failover]% device
```

- b. List the devices.

```
[nvidia-era-bcm-02->device]% list
```

```
[nvidia-era-bcm-02->device]% list
```

Type	Hostname (key)	MAC	Category	IP	Network	Status
HeadNode	nvidia-era-bcm-01	96:7C:AB:AE:9C:A0		192.168.100.253	internalnet	[ UP ], restart required (i+
HeadNode	nvidia-era-bcm-02	00:50:56:04:34:12		192.168.100.252	internalnet	[ UP ]
PhysicalNode	node001	00:00:00:00:00:00	default	192.168.100.21	internalnet	[ DOWN ], unassigned

- c. Reboot the primary head node.

```
[nvidia-era-bcm-02->device]% reboot nvidia-era-bcm-01
```

```
[nvidia-era-bcm-02->device]% reboot nvidia-era-bcm-01
Reboot in progress for: nvidia-era-bcm-01
```

- d. Wait for the primary head node to come back up, then list the devices to verify status.

```
[nvidia-era-bcm-02->device]% list
```

```
[nvidia-era-bcm-02->device]% list
```

Type	Hostname (key)	MAC	Category	IP	Network	Status
HeadNode	nvidia-era-bcm-01	96:7C:AB:AE:9C:A0		192.168.100.253	internalnet	[ UP ]
HeadNode	nvidia-era-bcm-02	00:50:56:04:34:12		192.168.100.252	internalnet	[ UP ]
PhysicalNode	node001	00:00:00:00:00:00	default	192.168.100.21	internalnet	[ DOWN ], unassigned

11. Make the primary head node is the active head node again.

- a. Enter the **partition** mode.

```
[nvidia-era-bcm-02->device]% partition
```

- b. Enter the **failover** submenu.

```
[nvidia-era-bcm-02->partition[base]]% failover
```

- c. Make the primary head node active. (Pressing **Enter** after this command will show that the client is disconnected.)

```
[nvidia-era-bcm-02->partition[base]->failover]% makeactive nvidia-era-bcm-01
```

```
[nvidia-era-bcm-02->partition[base]->failover]% makeactive nvidia-era-bcm-01  
client_loop: send disconnect: Connection reset
```

12. Log in again using the HA shared internal interface IP address.

```
ssh root@192.168.100.254
```

---

# Backup Defaults: Softwareimage and Category

At this stage, it is recommended to create a backup of the default software image and category included with the BCM installation. This serves as a precaution so that if changes are made that negatively affect future deployments, these backups can be used to restore the original “out-of-the-box” configuration.

1. Backup the default-image softwareimage.

- a. Enter cmsh.

```
root@nvidia-era-bcm-01:~# cmsh
```

- b. Enter the **softwareimage** mode.

```
[nvidia-era-bcm-01]% softwareimage
```

- c. List the current software images.

```
[nvidia-era-bcm-01->softwareimage]% list
```

```
[nvidia-era-bcm-01->softwareimage]% list
Name (key)      Path (key)      Kernel version  Nodes
-----
default-image  /cm/images/default-image  5.15.0-113-generic  1
```

- d. Clone the default image, naming the new image `backup-default-image`.

```
[nvidia-era-bcm-01->softwareimage]% clone default-image backup-default-image
```

- e. Commit the changes to start the cloning process, and wait for it to complete.

```
[nvidia-era-bcm-01->softwareimage*[backup-default-image*]]% commit
```

```
[nvidia-era-bcm-01->softwareimage*[backup-default-image*]]% commit
Fri Aug 8 03:21:22 2025 [notice] nvidia-era-bcm-01: Started to copy: /cm/images/default-image -> /cm/images/backup-default-image (1)
[nvidia-era-bcm-01->softwareimage*[backup-default-image*]]% commit[nvidia-era-bcm-01->softwareimage*[backup-default-image*]]%
Fri Aug 8 03:23:00 2025 [notice] nvidia-era-bcm-01: Copied: /cm/images/default-image -> /cm/images/backup-default-image (2)
[nvidia-era-bcm-01->softwareimage*[backup-default-image*]]%
Fri Aug 8 03:23:00 2025 [notice] nvidia-era-bcm-01: Initial ramdisk for image backup-default-image is being generated
[nvidia-era-bcm-01->softwareimage*[backup-default-image*]]%
Fri Aug 8 03:23:00 2025 [notice] nvidia-era-bcm-01: Provisioning started: sending nvidia-era-bcm-01:/cm/images/backup-default-image
to nvidia-era-bcm-02:/cm/images/backup-default-image, mode UPDATE, dry run = no
[nvidia-era-bcm-01->softwareimage*[backup-default-image*]]%
Fri Aug 8 03:24:03 2025 [notice] nvidia-era-bcm-01: Initial ramdisk for image backup-default-image was generated successfully
[nvidia-era-bcm-01->softwareimage*[backup-default-image*]]%
Fri Aug 8 03:24:31 2025 [notice] nvidia-era-bcm-01: Provisioning completed: sent nvidia-era-bcm-01:/cm/images/backup-default-image
to nvidia-era-bcm-02:/cm/images/backup-default-image, mode UPDATE, dry run = no
```

- f. Exit to return to the **softwareimage** mode and list the softwareimages to confirm creation.

```
[nvidia-era-bcm-01->softwareimage*[backup-default-image*]]% exit
[nvidia-era-bcm-01->softwareimage]% list
```

```
[nvidia-era-bcm-01->softwareimage]% list
```

Name (key)	Path (key)	Kernel version	Nodes
backup-default-image	/cm/images/backup-default-image	5.15.0-113-generic	0
default-image	/cm/images/default-image	5.15.0-113-generic	1

2. Backup the default category and set the softwareimage.

- a. Enter the **category** mode.

```
[nvidia-era-bcm-01->softwareimage]% category
```

- b. List the current categories.

```
[nvidia-era-bcm-01->category]% list
```

```
[nvidia-era-bcm-01->category]% list
```

Name (key)	Software image	Nodes
default	default-image	1

- c. Clone the default category, setting the new category name to **backup-default**.

```
[nvidia-era-bcm-01->category]% clone default backup-default
```

- d. Set the softwareimage to point to the **backup-default-image** created in the previous steps.

```
[nvidia-era-bcm-01->category*[backup-default*]]% set softwareimage backup-default-image
```

- e. Commit the changes.

```
[nvidia-era-bcm-01->category*[backup-default*]]% commit
```

- f. Exit to return to the **category** mode and list the categories to creation.

```
[nvidia-era-bcm-01->category*[backup-default]]% exit  
[nvidia-era-bcm-01->category]% list
```

```
[nvidia-era-bcm-01->category]% list  
Name (key)           Software image       Nodes  
-----  
backup-default      backup-default-image 0  
default             default-image        1
```

---

# Configure Defaults: Softwareimage and Category

In this section, the `default-image` softwareimage is configured to use the Pre-RHEL7 naming convention, matching the configuration applied to the head node in earlier steps.

The default category disk setup is updated to use a single large partition formatted with the XFS filesystem. For this, an existing xml file will be copied and modified to only use a single disk, `nvme0n1`.

## 2. Configure the `default-image` softwareimage.

- a. Enter `cmsh` (if not already within it).

```
root@nvidia-era-bcm-01:~# cmsh
```

- b. Enter the **softwareimage** mode.

```
[nvidia-era-bcm-01]% softwareimage
```

- c. Use the `default-image` softwareimage to edit its settings.

```
[nvidia-era-bcm-01->softwareimage]% use default-image
```

- d. Append the kernel parameters to include “ `net.ifnames=0 biosdevname=0`”.



**Note:** There is a space between the first quotation mark “ and `net.ifnames=0`.

The **append** command adds the new parameters to the end of the existing parameters, so a space is required to separate the old from the new.

```
[nvidia-era-bcm-01->softwareimage[default-image]]% append kernelparameters "
net.ifnames=0 biosdevname=0"
```

- e. Commit the changes.

```
[nvidia-era-bcm-01->softwareimage*[default-image*]]% commit
```

- f. Return to the cmsh home menu.

```
[nvidia-era-bcm-01->softwareimage*[default-image*]]% home
```

### 3. Create the disk setup xml file.

- a. Exit cmsh and return to the OS root.

```
[nvidia-era-bcm-01]% quit
```

```
[nvidia-era-bcm-01->main]% quit  
root@nvidia-era-bcm-01:~#
```

- b. Change directory to **/cm/local/apps/cmd/etc/htdocs/disk-setup/**

```
root@nvidia-era-bcm-01:~# cd /cm/local/apps/cmd/etc/htdocs/disk-setup/
```

- c. Copy the existing file **x86\_64-slave-one-big-partition-xfs.xml** to **nvidia-era-nvme0n1-one-big-partition-xfs.xml**.

```
root@nvidia-era-bcm-01:/cm/local/apps/cmd/etc/htdocs/disk-setup# cp x86_64-slave-one-  
big-partition-xfs.xml nvidia-era-nvme0n1-one-big-partition-xfs.xml
```

- d. Modify the copied file, removing references to any device other than **nvme0n1**.

In this example nano is used as the editor, but the administrator should use the editor they are most comfortable with.

```
root@nvidia-era-bcm-01:/cm/local/apps/cmd/etc/htdocs/disk-setup# nano nvidia-era-  
nvme0n1-one-big-partition-xfs.xml
```

Before:

```
GNU nano 6.2 nvidia-era-nvme0n1-one-big-partition-xfs.xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- One swap partition and the rest of the filesystem on 1 partition -->
<diskSetup xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <device>
    <blockdev>/dev/sda</blockdev>
    <blockdev>/dev/hda</blockdev>
    <blockdev>/dev/vda</blockdev>
    <blockdev>/dev/xvda</blockdev>
    <blockdev>/dev/nvme0n1</blockdev>
    <blockdev mode="cloud"/>/dev/sdb</blockdev>
    <blockdev mode="cloud"/>/dev/hdb</blockdev>
    <blockdev mode="cloud"/>/dev/vdb</blockdev>
    <!-- the following for paravirtual rhel6: -->
    <blockdev mode="cloud"/>/dev/xvdf</blockdev>
  </device>
  <partition id="a0" partitiontype="esp">
    <size>100M</size>
    <type>linux</type>
    <filesystem>fat</filesystem>
    <mountPoint>/boot/efi</mountPoint>
    <mountOptions>defaults,noatime,nodiratime</mountOptions>
  </partition>
  <partition id="a1">
    <size>16G</size>
    <type>linux swap</type>
  </partition>
  <partition id="a2">
    <size>max</size>
    <type>linux</type>
    <filesystem>xfs</filesystem>
    <mountPoint>/</mountPoint>
    <mountOptions>defaults,noatime,nodiratime</mountOptions>
  </partition>
</diskSetup>
```

After:

```
GNU nano 6.2 nvidia-era-nvme0n1-one-big-partition-xfs.xml *
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- One swap partition and the rest of the filesystem on 1 partition -->
<diskSetup xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <device>
    <blockdev>/dev/nvme0n1</blockdev>
  </device>
  <partition id="a0" partitiontype="esp">
    <size>100M</size>
    <type>linux</type>
    <filesystem>fat</filesystem>
    <mountPoint>/boot/efi</mountPoint>
    <mountOptions>defaults,noatime,nodiratime</mountOptions>
  </partition>
  <partition id="a1">
    <size>16G</size>
    <type>linux swap</type>
  </partition>
  <partition id="a2">
    <size>max</size>
    <type>linux</type>
    <filesystem>xfs</filesystem>
    <mountPoint>/</mountPoint>
    <mountOptions>defaults,noatime,nodiratime</mountOptions>
  </partition>
</diskSetup>
```

Save (ctrl+s) and exit (ctrl+x).

- e. Change directory back to home.

```
root@nvidia-era-bcm-01:/cm/local/apps/cmd/etc/htdocs/disk-setup# cd ~
```

- f. Enter cmsh.

```
root@nvidia-era-bcm-01:~# cmsh
```

#### 4. Configure the default category.

- a. Enter the **category** mode.

```
[nvidia-era-bcm-01]% category
```

- b. Use the **default** category to edit its settings.

```
[nvidia-era-bcm-01->category]% use default
```

- c. Set the disk layout to the newly create disk setup xml file. Use tab completion to avoid typing the entire filename manually.

```
[nvidia-era-bcm-01->category[default]]% set disksetup nvidia-era-nvme0n1-one-big-partition-xfs.xml
```

- d. Commit the changes.

```
[nvidia-era-bcm-01->category*[default*]]% commit
```

---

# Configure Device: Template Default Node

The device (node) that was created as part of the BCM installation process will now be used to create a template. Similar to backing up the default softwareimage and category, this node can be cloned later to create additional nodes with the out-of-the-box configuration... with one modification: the default static IP address assignment will be changed to use DHCP instead.

Some settings, such as **category** and **provisioning interface**, may already be correctly configured. This procedure can also serve as an example of how to apply similar settings to other devices in the future.



**Note:** BOOTIF is a special name used by BCM for one of the possible interfaces. The node-installer automatically translates BOOTIF into the name of the device (such as eth0 or eth1) used for network booting. This is useful for systems with multiple network interfaces, as it removes the need to manually identify the boot interface. Using the name BOOTIF instead means that the underlying device does not need to be specified in the first place.

For more information about the BOOTIF interface, please refer to the BCM Administrator Manual.

## 1. Reconfigure node001.

- a. Enter cmsh (if not already within it).

```
root@nvidia-era-bcm-01:~# cmsh
```

- b. Enter the **device** submenu.

```
[nvidia-era-bcm-01]% device
```

- c. List the devices.

```
[nvidia-era-bcm-01->device]% list
```

```
[nvidia-era-bcm-01->device]% list
-----
Type                Hostname (key)  MAC                Category           IP                Network           Status
-----
HeadNode            nvidia-era-bcm-01  96:7C:AB:AE:9C:A0  default            192.168.100.253  internalnet       [ UP ]
HeadNode            nvidia-era-bcm-02  00:50:56:04:34:12  default            192.168.100.252  internalnet       [ UP ]
PhysicalNode        node001          00:00:00:00:00:00  default            192.168.100.21   internalnet       [ DOWN ], unassigned
```

- d. Rename node001 to template-default-node

```
[nvidia-era-bcm-01->device]% rename node001 template-default-node
```

```
[nvidia-era-bcm-01->device]% rename node001 template-default-node  
Renamed: node001 to template-default-node
```

- e. List the devices again to confirm the change.

```
[nvidia-era-bcm-01->device]% list
```

```
[nvidia-era-bcm-01->device*]% list
```

Type	Hostname (key)	MAC	Category	IP	Network	Status
HeadNode	nvidia-era-bcm-01	96:7C:AB:AE:9C:A0		192.168.100.253	internalnet	[ UP ]
HeadNode	nvidia-era-bcm-02	00:50:56:04:34:12		192.168.100.252	internalnet	[ UP ]
PhysicalNode	template-default-node	00:00:00:00:00:00	default	192.168.100.21	internalnet	[ DOWN ], unassigned

- f. Use template-default-node to edit its settings.

```
[nvidia-era-bcm-01->device*]% use template-default-node
```

- g. Set the category to default.

```
[nvidia-era-bcm-01->device*[template-default-node*]]% set category default
```

- h. Enter the **interfaces** submenu.

```
[nvidia-era-bcm-01->device*[template-default-node*]]% interfaces
```

- i. List the interfaces.

```
[nvidia-era-bcm-01->device*[template-default-node*]->interfaces]% list
```

```
[nvidia-era-bcm-01->device*[template-default-node*]->interfaces]% list
```

Type	Network device name	IP	Network	Start if
physical	BOOTIF [prov]	192.168.100.21	internalnet	always

- j. Use bootif to edit its settings.

```
[nvidia-era-bcm-01->device*[template-default-node*]->interfaces]% use bootif
```

- k. Enable DHCP.

```
[nvidia-era-bcm-01->device*[template-default-node*]->interfaces[BOOTIF]]% set dhcp yes
```

- l. Clear the IP address.

```
[nvidia-era-bcm-01->device*[template-default-node*]->interfaces*[BOOTIF*]]% clear ip
```

- m. Set the network to internalnet.

```
[nvidia-era-bcm-01->device*[template-default-node*]->interfaces*[BOOTIF*]]% set network  
internalnet
```

- n. Exit to the **interfaces** submenu, then exit again to return to the device->template-default-node submenu.

```
[nvidia-era-bcm-01->device*[template-default-node*]->interfaces*[BOOTIF*]]% exit  
[nvidia-era-bcm-01->device*[template-default-node*]->interfaces*]% exit
```

- o. Set the provisioning interface to bootif.

```
[nvidia-era-bcm-01->device*[template-default-node*]]% set provisioninginterface bootif
```

- p. Commit the changes.

```
[nvidia-era-bcm-01->device*[template-default-node*]]% commit
```

---

# Create Custom Softwareimage and Category

New software images and categories can now be created.

Although they can be built from scratch using the add command, cloning the previously created defaults ensures that all previously applied settings (such as kernelparameters and disksetup) are copied across automatically. This reduces the amount of customization required after creation.

In this example, the software image is a simple clone of the default-image with no modifications, while the category is cloned from the default but configured to use the newly cloned software image.



**Note:** Managing packages in a software image is beyond the scope of this guide. For more information about managing packages in a software image, please refer to the BCM Administrator Manual.

## 1. Create the custom-image softwareimage.

### a. Enter cmsh.

```
root@nvidia-era-bcm-01:~# cmsh
```

### b. Enter the **softwareimage** mode.

```
[nvidia-era-bcm-01]% softwareimage
```

### c. List the current software images.

```
[nvidia-era-bcm-01->softwareimage]% list
```

```
[nvidia-era-bcm-01->softwareimage]% list
```

Name (key)	Path (key)	Kernel version	Nodes
backup-default-image	/cm/images/backup-default-image	5.15.0-113-generic	0
default-image	/cm/images/default-image	5.15.0-113-generic	1

- d. Clone the default image, naming the new image to custom-image.

```
[nvidia-era-bcm-01->softwareimage]% clone default-image custom-image
```

- e. Commit the changes to start the cloning process and wait for it to complete.

```
[nvidia-era-bcm-01->softwareimage*[custom-image*]]% commit
```

```
[nvidia-era-bcm-01->softwareimage*[custom-image*]]% commit
Fri Aug 8 04:43:54 2025 [notice] nvidia-era-bcm-01: Started to copy: /cm/images/default-image -> /cm/images/custom-image (8)
[nvidia-era-bcm-01->softwareimage*[custom-image*]]% commit[nvidia-era-bcm-01->softwareimage*[custom-image*]]%
Fri Aug 8 04:45:30 2025 [notice] nvidia-era-bcm-01: Copied: /cm/images/default-image -> /cm/images/custom-image (9)
[nvidia-era-bcm-01->softwareimage*[custom-image*]]%
Fri Aug 8 04:45:30 2025 [notice] nvidia-era-bcm-01: Initial ramdisk for image custom-image is being generated
[nvidia-era-bcm-01->softwareimage*[custom-image*]]%
Fri Aug 8 04:45:30 2025 [notice] nvidia-era-bcm-01: Provisioning started: sending nvidia-era-bcm-01:/cm/images/custom-image to nvid
ia-era-bcm-02:/cm/images/custom-image, mode UPDATE, dry run = no
[nvidia-era-bcm-01->softwareimage*[custom-image*]]%
Fri Aug 8 04:46:35 2025 [notice] nvidia-era-bcm-01: Initial ramdisk for image custom-image was generated successfully
```

- f. Exit to return to the **softwareimage** mode and list the softwareimages to confirm creation.

```
[nvidia-era-bcm-01->softwareimage*[custom-image]]% exit
[nvidia-era-bcm-01->softwareimage]% list
```

```
[nvidia-era-bcm-01->softwareimage]% list
Name (key)          Path (key)          Kernel version      Nodes
-----
backup-default-image /cm/images/backup-default-image 5.15.0-113-generic 0
custom-image        /cm/images/custom-image          5.15.0-113-generic 0
default-image       /cm/images/default-image        5.15.0-113-generic 1
```

## 2. Backup the default category and set the softwareimage.

- a. Enter the **category** mode.

```
[nvidia-era-bcm-01->softwareimage]% category
```

- b. List the current categories.

```
[nvidia-era-bcm-01->category]% list
```

```
[nvidia-era-bcm-01]% category list
Name (key)          Software image      Nodes
-----
backup-default      backup-default-image 0
default             default-image       1
```

- c. Clone the default category, naming the new category `custom`.

```
[nvidia-era-bcm-01->category]% clone default custom
```

- d. Set the `softwareimage` to point to the `custom-image` created in the previous steps.

```
[nvidia-era-bcm-01->category*[custom*]]% set softwareimage custom-image
```

- e. Commit the changes.

```
[nvidia-era-bcm-01->category*[custom*]]% commit
```

- f. Exit to return to the **category** mode and list the categories to confirm creation.

```
[nvidia-era-bcm-01->category*[custom]]% exit  
[nvidia-era-bcm-01->category]% list
```

```
[nvidia-era-bcm-01->category]% list  
Name (key)           Software image       Nodes  
-----  
backup-default      backup-default-image 0  
custom              custom-image         0  
default             default-image        1
```

---

# Create Devices (Nodes)

In this section, a device (node) will be created to represent a physical node in the cluster.

While devices can be created from scratch using the **add** command, cloning the previously created template-default-node will copy the settings already applied (such as interfaces).

At this stage, it is likely unclear which interfaces on the physical node (as enumerated by the operating system) correspond to the DPU interfaces on the N-S internal network that form the logical bond interface.

To determine this, the node will first be brought up using the default category (which uses the default-image software image) and the BCM special interface BOOTIF. From within the operating system, the network interfaces will be listed to identify which of the interfaces correspond to the DPU interfaces.

In BCM, the node object interfaces will then be updated by adding the two physical interfaces and the bond interface.

After this, the node category will be changed to the custom category (which uses the custom-image software image).

In a homogeneous environment, where all nodes are identical, it can be assumed that the OS will enumerate the interfaces in the same way. In this case, the newly created node can be used as a template to clone additional nodes.

In a heterogeneous environment, these preliminary and subsequent steps may need to be repeated for each group of similar node types.



**Note:** Example values displayed in **red** should be replaced with actual information relevant to the target environment.

## 1. Create the device.

- a. Enter **cmsh**.

```
root@nvidia-era-bcm-01:~# cmsh
```

- b. Enter the **device** mode.

```
[nvidia-era-bcm-01]% device
```

- c. List the current devices.

```
[nvidia-era-bcm-01->device]% list
```

```
[nvidia-era-bcm-01->device]% list
```

Type	Hostname (key)	MAC	Category	IP	Network	Status
HeadNode	nvidia-era-bcm-01	96:7C:AB:AE:9C:A0		192.168.100.253	internalnet	[ UP ]
HeadNode	nvidia-era-bcm-02	00:50:56:04:34:12		192.168.100.252	internalnet	[ UP ]
PhysicalNode	template-default-node	00:00:00:00:00:00	default	0.0.0.0	internalnet	[ DOWN ], unassigned

- d. Clone template-default-node to **oem-node001**.

```
[nvidia-era-bcm-01->device]% clone template-default-node oem-node001 --next-ip
```

- e. Set the category to default.

```
[nvidia-era-bcm-01->device*[oem-node001*]]% set category default
```

- f. Commit the changes.

```
[nvidia-era-bcm-01->device*[oem-node001*]]% commit
```

- g. Exit to return to **device** mode and list the devices to confirm creation.

```
[nvidia-era-bcm-01->device[oem-node001]]% exit
[nvidia-era-bcm-01->device]% list
```

```
[nvidia-era-bcm-01->device]% list
```

Type	Hostname (key)	MAC	Category	IP	Network	Status
HeadNode	nvidia-era-bcm-01	00:50:56:04:34:03		192.168.100.253	internalnet	[ UP ]
HeadNode	nvidia-era-bcm-02	00:50:56:04:34:12		192.168.100.252	internalnet	[ UP ]
PhysicalNode	oem-node001	00:00:00:00:00:00	default	0.0.0.0	internalnet	[ DOWN ], unassigned
PhysicalNode	template-default-node	00:00:00:00:00:00	default	0.0.0.0	internalnet	[ DOWN ], unassigned

2. Preliminary. Bring up the node.

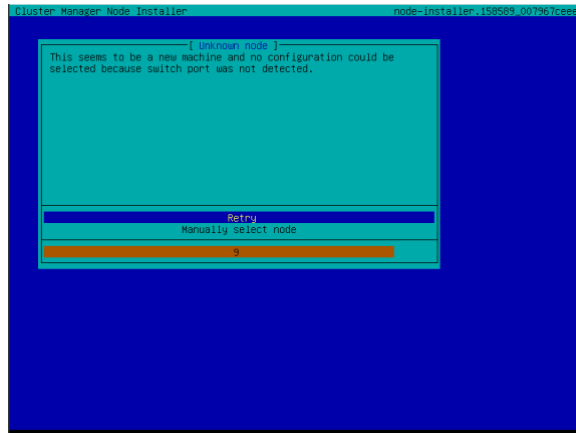


**Note:** Instructions will vary by OEM. Refer to the manufacturer's documentation for detailed instructions.

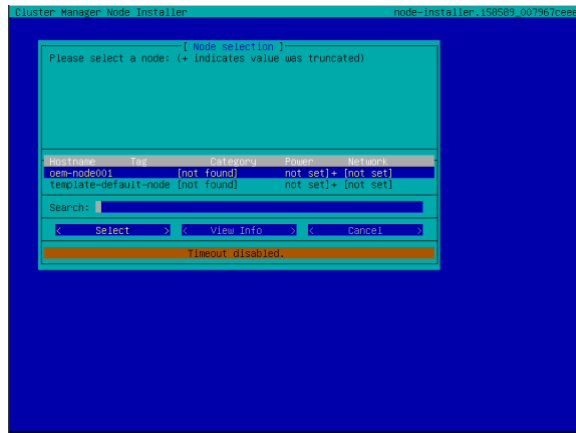
- a. Power on and boot the node from the network (PXE).
- b. Allow the node to boot to **AUTO**.



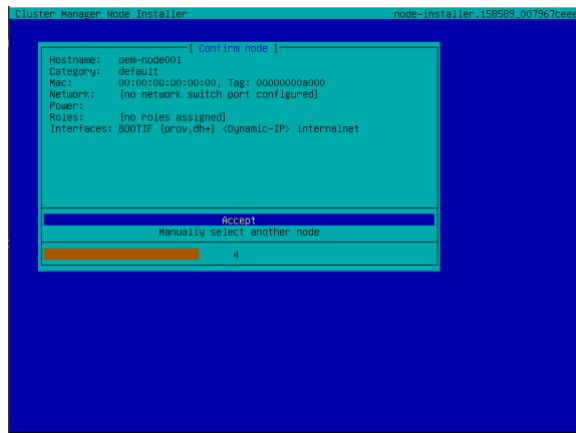
- c. When prompted as an unknown node, the Cluster Manager Node Installer will loop. Select **Manually select node**.



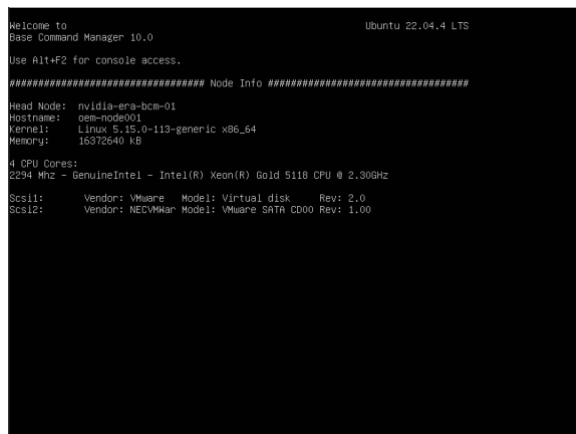
- d. Select the newly created node.



- e. **Accept** the selection.



- f. Wait until the node is fully booted.



3. From cmsh on the head node, log in to the new node using the rshell (remote shell) command.

- a. Enter cmsh.

```
root@nvidia-era-bcm-01:~# cmsh
```

- b. Enter the device mode.

```
[nvidia-era-bcm-01]% device
```

- c. Use the node previously created.

```
[nvidia-era-bcm-01->device]% use oem-node001
```

- d. Use the command rshell to connect to the node.

```
[nvidia-era-bcm-01->device[oem-node001]]% rshell
```

```
root@oem-node001:~#
```

4. On the new node, identify the interfaces and IP addresses.

- a. Run the command `ip a`.

```
root@oem-node001:~# ip a
```

```
root@oem-node001:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 00:50:56:04:34:5e brd ff:ff:ff:ff:ff:ff
    altname enp4s0
    altname ens161
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 00:50:56:04:34:6a brd ff:ff:ff:ff:ff:ff
    altname enp1s0
    altname ens192
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:04:34:c2 brd ff:ff:ff:ff:ff:ff
    altname enp19s0
    altname ens224
    inet 192.168.100.194/24 brd 192.168.100.255 scope global dynamic eth2
        valid_lft 86128sec preferred_lft 86128sec
    inet6 fe80::250:56ff:fe04:34c2/64 scope link
        valid_lft forever preferred_lft forever
5: eth3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 00:50:56:04:34:c3 brd ff:ff:ff:ff:ff:ff
    altname enp27s0
    altname ens256
```



**Note:** In this example, **eth2** has an IP address in the internal network range (**192.168.100.0/24**), and is therefore presumed to be the first DPU interface. It is logical that the next interface, **eth3** is the second DPU interface. This can be confirmed in the next steps.

If no interface has an IP address in this range, the server's boot device may not be configured to be the first interface of the DPU. Adjust the server's PXE boot order and repeat the procedure. Instructions will vary by OEM and are not covered in this guide.



**Note:** The interfaces shown in this example may differ from the target environment, depending on the chosen cluster architecture and OEM.

For example, with the additional network interfaces in compute nodes, the DPU interfaces could very well be enumerated as **eth7** and **eth8**, or **eth10** and **eth11**. It is important to identify the correct interfaces at this stage.

- b. Run the command `dhclient` to bring up any inactive interfaces that are down.

```
root@oem-node001:~# dhclient
```

```
root@oem-node001:~# dhclient
RTNETLINK answers: File exists
```

- c. Run the command `ip a` again.

```
root@oem-node001:~# ip a
```

```
root@oem-node001:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:04:34:5e brd ff:ff:ff:ff:ff:ff
    altname enp490
    inet6 fe80::250:56ff:fe04:345e/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:04:34:6a brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    altname ens192
    inet6 fe80::250:56ff:fe04:346a/64 scope link
        valid_lft forever preferred_lft forever
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:04:34:c2 brd ff:ff:ff:ff:ff:ff
    altname enp19s0
    altname ens224
    inet 192.168.100.194/24 brd 192.168.100.255 scope global dynamic eth2
        valid_lft 85691sec preferred_lft 85691sec
    inet6 fe80::250:56ff:fe04:34c2/64 scope link
        valid_lft forever preferred_lft forever
5: eth3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:04:34:c3 brd ff:ff:ff:ff:ff:ff
    altname enp27s0
    altname ens256
    inet 192.168.100.166/24 brd 192.168.100.255 scope global dynamic eth3
        valid_lft 100794sec preferred_lft 100794sec
    inet6 fe80::250:56ff:fe04:34c3/64 scope link
        valid_lft forever preferred_lft forever
```



**Note:** At this stage, both eth2 and eth3 should have IPs in **192.168.100.0/24**, confirming they are DPU interfaces. If not, further investigation is required.

- d. Exit rshell to return to the head node.

```
root@oem-node001:~# exit
```

```
root@oem-node001:~# exit
logout
[nvidia-era-bcm-01->device[oem-node001]]% |
```

5. While still in the context of device -> oem-node001, set the category to custom, which in turn will apply the custom-image software image when the node is rebooted.

```
[nvidia-era-bcm-01->device[oem-node001]]% set category custom
```

6. Modify the nodes interfaces to add the physical interfaces and the bond interface.

- a. Enter the **interfaces** submenu.

```
[nvidia-era-bcm-01->device[oem-node001]]% interfaces
```

- b. List the interfaces.

```
[nvidia-era-bcm-01->device[oem-node001]->interfaces]% list
```

```
[nvidia-era-bcm-01->device[oem-node001]->interfaces]% list
Type          Network device name IP          Network      Start if
-----
physical     BOOTIF [prov]      192.168.100.194 internalnet  always
```

- c. Remove the BOOTIF interface.

```
[nvidia-era-bcm-01->device[oem-node001]->interfaces]% remove bootif
```

- d. Add the physical interface eth2 and exit.

```
[nvidia-era-bcm-01->device*[oem-node001*]->interfaces*]% add physical eth2
[nvidia-era-bcm-01->device*[oem-node001*]->interfaces*[eth2*]]% exit
```

- e. Add the physical interface eth3 and exit.

```
[nvidia-era-bcm-01->device*[oem-node001*]->interfaces*]% add physical eth3
[nvidia-era-bcm-01->device*[oem-node001*]->interfaces*[eth3*]]% exit
```

- f. Add the bonded interface bond0.

```
[nvidia-era-bcm-01->device*[oem-node001*]->interfaces*]% add bond bond0
```

- g. Set DHCP on bond0.

```
[nvidia-era-bcm-01->device*[oem-node001*]->interfaces*[bond0*]]% set dhcp yes
```

- h. Clear the current IP address.

```
[nvidia-era-bcm-01->device*[oem-node001*]->interfaces*[bond0*]]% clear ip
```

- i. Set the network to internalnet.

```
[nvidia-era-bcm-01->device*[oem-node001*]->interfaces*[bond0*]]% set network
internalnet
```

- j. Set the bond interfaces to eth2 and eth3.

```
set
[nvidia-era-bcm-01->device*[oem-node001*]->interfaces*[bond0*]]% set interfaces eth2
eth3
```

- k. Set the bond mode to 4 (LACP).

```
[nvidia-era-bcm-01->device*[oem-node001*]->interfaces*[bond0*]]% set mode 4
```

- l. Set the options to `miimon=100` and `xmit_hash_policy=layer3+4`.

```
[nvidia-era-bcm-01->device*[oem-node001*]->interfaces*[bond0*]]% set options
"miimon=100 xmit_hash_policy=layer3+4"
```

- m. Exit to the interfaces submenu.

```
[nvidia-era-bcm-01->device*[oem-node001*]->interfaces*[bond0*]]% exit
```

- n. List the interfaces.

```
[nvidia-era-bcm-01->device*[oem-node001*]->interfaces*]% list
```

```
[nvidia-era-bcm-01->device*[oem-node001*]->interfaces*]% list
Type      Network device name IP      Network      Start if
-----
bond      bond0 [prov,dhcp]    0.0.0.0      internalnet   always
physical  eth2 (bond0)        0.0.0.0
physical  eth3 (bond0)        0.0.0.0      always
```

- o. Exit to the device submenu.

```
[nvidia-era-bcm-01->device*[oem-node001*]->interfaces*]% exit
```

- p. Set the provisioning interface to bond0.

```
[nvidia-era-bcm-01->device*[oem-node001*]]% set provisioninginterface bond0
```

- q. Commit the changes.

```
[nvidia-era-bcm-01->device*[oem-node001*]]% commit
```

- r. Reboot the node.

```
[nvidia-era-bcm-01->device[oem-node001]]% reboot
```

7. At this point, the node should come back up with the bond interface active.

- a. Connect to the node again.

```
[nvidia-era-bcm-01->device[oem-node001]]% rshell
```

- b. Run command `ip a`.

```
root@oem-node001:~# ip a
```

```
root@oem-node001:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
   link/ether 00:50:56:04:34:5e brd ff:ff:ff:ff:ff:ff
   altname enps4s0
   altname ens161
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
   link/ether 00:50:56:04:34:6a brd ff:ff:ff:ff:ff:ff
   altname enp11s0
   altname ens192
4: eth2: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc mq master bond0 state UP group default qlen 1000
   link/ether 42:09:c3:56:a8:7b brd ff:ff:ff:ff:ff:ff permaddr 00:50:56:04:34:c2
   altname enp19s0
   altname ens224
5: eth3: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc mq master bond0 state UP group default qlen 1000
   link/ether 42:09:c3:56:a8:7b brd ff:ff:ff:ff:ff:ff permaddr 00:50:56:04:34:c3
   altname enp27s0
   altname ens256
6: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
   link/ether 42:09:c3:56:a8:7b brd ff:ff:ff:ff:ff:ff
   inet 192.168.100.194/24 brd 192.168.100.255 scope global dynamic bond0
       valid_lft 68706sec preferred_lft 68706sec
   inet6 fe80::4009:c3ff:fe56:a87b/64 scope link
       valid_lft forever preferred_lft forever
```



**Note:** Notice that the bond0 interface is up.

- c. Exit the remote shell.

```
root@oem-node001:~# exit
```

Once verified, this node configuration can be cloned to create additional nodes with the same bonded interface setup. For example, 2 additional nodes can be created by using the following commands:

```
[nvidia-era-bcm-01->device]% clone oem-node001 oem-node002 --next-ip  
[nvidia-era-bcm-01->device*[oem-node002*]]% exit  
[nvidia-era-bcm-01->device*]% clone oem-node001 oem-node003 --next-ip  
[nvidia-era-bcm-01->device*[oem-node003*]]% exit  
[nvidia-era-bcm-01->device*]% commit  
[nvidia-era-bcm-01->device]% list
```

```
[nvidia-era-bcm-01->device]% list
```

Type	Hostname (key)	MAC	Category	IP	Network	Status
HeadNode	nvidia-era-bcm-01	00:50:56:04:34:03		192.168.100.253	internalnet	[ UP ]
HeadNode	nvidia-era-bcm-02	00:50:56:04:34:12		192.168.100.252	internalnet	[ UP ]
PhysicalNode	oem-node001	00:50:56:A3:57:BC	custom	192.168.100.194	internalnet	[ UP ]
PhysicalNode	oem-node002	00:00:00:00:00:00	custom	192.168.100.224	internalnet	[ DOWN ], unassigned
PhysicalNode	oem-node003	00:00:00:00:00:00	custom	192.168.100.225	internalnet	[ DOWN ], unassigned
PhysicalNode	template-default-node	00:00:00:00:00:00	default	0.0.0.0	internalnet	[ DOWN ], unassigned

For nodes with different configurations (for example, in a heterogenous cluster with nodes of differing models or from different manufacturers) the entire section should be repeated until all nodes are configured. (i.e. clone the template-default node, identify and configure the interfaces, set the applicable category, clone again.)

---

## Appendix A. Related Links

Document	Link
Base Command Manager Installation Manual	<a href="#">installation-manual.pdf</a>
Base Command Manager Administrator Manual	<a href="#">admin-manual.pdf</a>

## Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation (“NVIDIA”) makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality. NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer (“Terms of Sale”). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer (“Terms of Sale”). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer’s own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer’s product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, “MATERIALS”) ARE BEING PROVIDED “AS IS.” NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA’s aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

## Trademarks

NVIDIA, the NVIDIA logo, are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

## VESA DisplayPort

DisplayPort and DisplayPort Compliance Logo, DisplayPort Compliance Logo for Dual-mode Sources, and DisplayPort Compliance Logo for Active Cables are trademarks owned by the Video Electronics Standards Association in the United States and other countries.

## HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

## Arm

Arm, AMBA, and Arm Powered are registered trademarks of Arm Limited. Cortex, MPCore, and Mali are trademarks of Arm Limited. All other brands or product names are the property of their respective holders. “Arm” is used to represent Arm Holdings plc; its operating company Arm Limited; and the regional subsidiaries Arm Inc.; Arm KK; Arm Korea Limited.; Arm Taiwan Limited; Arm France SAS; Arm Consulting (Shanghai) Co. Ltd.; Arm Germany GmbH; Arm Embedded Technologies Pvt. Ltd.; Arm Norway, AS, and Arm Sweden AB.

## OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

## Copyright

© 2025 NVIDIA Corporation. All rights reserved.

