



NVIDIA DGX OS 7 User Guide

NVIDIA Corporation

Dec 30, 2025

Contents

1	About DGX OS 7	1
1.1	DGX OS 7 Features	1
1.2	Supported NVIDIA DGX Systems	1
1.3	Installation and Upgrade	2
1.4	Related Documentation	3
1.5	NVIDIA Enterprise Support	4
2	Release Guidance	5
2.1	Release Mechanisms	5
2.1.1	DGX OS ISO Image	5
2.1.2	Linux Software Repositories	6
2.2	Release Numbering Convention	6
2.3	Release Cadence	7
3	Release Notes	9
3.1	Current Software Versions	9
3.2	Supported DGX Systems	12
3.3	Latest Release	13
3.3.1	DGX OS 7.3.1	14
3.4	Previous Releases	15
3.4.1	DGX OS 7.3.0	16
3.4.2	DGX OS 7.2.3	17
3.4.3	DGX OS 7.2.1	20
3.4.4	DGX OS 7.1.0	21
3.4.5	DGX OS 7.0.2	23
3.4.6	DGX OS 7.0.1	24
3.5	DGX OS ISO Releases	25
3.5.1	DGX OS ISO 7.3.1	26
3.5.2	DGX OS ISO 7.3.0	27
3.5.3	DGX OS ISO 7.2.3	28
3.5.4	DGX OS ISO 7.1.0	29
3.5.5	DGX OS ISO 7.0.2	30
4	Initial Setup	31
4.1	Connecting to the DGX System	31
4.2	First Boot Setup Wizard	31
4.2.1	First Boot Process for DGX Servers	32
4.2.2	First Boot Process for DGX Station	34
4.3	Post-Installation Tasks	35
4.3.1	Performing Package Updates for DGX Servers	35
4.3.1.1	Add the DOCA Driver to the DGX Server	35
4.3.1.2	Update Remaining Drivers and Packages	35
4.3.1.3	Verify the DGX OS Update	36

4.3.2	Performing Package Updates for DGX Stations	36
4.3.3	Adding Support for Additional Languages to the DGX Station	36
4.3.4	Configuring Your DGX Station	37
4.3.5	Enabling Multiple Users to Remotely Access the DGX System	38
5	Reimaging the System	39
5.1	Obtaining the DGX OS ISO Image	40
5.2	Installing the DGX OS Image	40
5.2.1	Installing the DGX OS Image Remotely Through the BMC	40
5.2.2	Installing the DGX OS Image from a USB Flash Drive or DVD-ROM	41
5.2.2.1	Creating a Bootable USB Flash Drive by Using the dd Command	41
5.2.2.2	Creating a Bootable USB Flash Drive by Using Akeo Rufus	42
5.2.3	Booting the DGX OS ISO image	43
5.3	DGX OS ISO Boot Options	43
5.3.1	Install DGX OS with Reformatting the Data RAID	44
5.3.2	Install DGX OS Without Reformatting the Data RAID	44
5.3.3	Advanced Installation Options (Encrypted Root)	45
5.3.4	Boot Into a Live Environment	45
5.3.5	Check Disc for Defects	46
6	Customizing Ubuntu Installation with DGX Software	47
6.1	Prerequisites	47
6.1.1	Ubuntu Software Requirements	47
6.1.2	Access to Software Repositories	48
6.2	Installation Considerations	48
6.2.1	System Drive Mirroring (RAID-1) [recommended]	48
6.2.2	Data Drive RAID-0 or RAID-5	48
6.2.3	System Drive Encryption [optional]	48
6.2.4	Data Drive Encryption [optional]	49
6.2.5	System Drive Partitioning	49
6.3	Installing Ubuntu	49
6.3.1	Booting from the Installation Media	49
6.3.2	Running the Ubuntu Installer	49
6.4	Installing the DGX Software Stack	55
6.4.1	Installing DGX System Configurations and Tools	55
6.4.2	Configuring Data Drives	57
6.4.3	Installing the GPU Driver	58
6.4.4	Installing the DOCA-OFED Package	61
6.4.5	Installing Docker and the NVIDIA Container Toolkit	62
6.4.6	Installing the NVIDIA System Management (NVSM) Tool [Recommended]	63
6.4.7	Additional Software Installed By DGX OS	63
6.5	Next Steps and Additional Information	63
7	Upgrading the OS	65
7.1	DGX OS 7 Release Upgrade Advisory	66
7.2	Getting Release Information for DGX Systems	66
7.3	Preparing to Upgrade the Software	67
7.3.1	Connect to the DGX System Console	67
7.3.2	Verifying the DGX System Connection to the Repositories	68
7.4	Performing a Release Upgrade from DGX OS 6	68
7.4.1	Upgrade DGX OS 6 to the Latest Version	68
7.4.2	Performing the Release Upgrade	69
7.4.3	Resolving Release Upgrade Conflicts	70
7.4.4	Verifying the Upgrade	72

7.4.5	Recovering from an Interrupted or Failed Update	72
7.5	Performing Package Upgrades	73
7.5.1	Enabling Extended Security Maintenance Upgrades	73
7.5.1.1	Renewing Extended Security Maintenance	74
7.5.2	Performing Package Upgrades Using the CLI	74
7.5.3	Managing Software Upgrades on DGX Station	75
7.5.4	Performing Package Upgrades Using the GUI	75
7.5.5	Checking for Updates to DGX Station Software	76
8	Managing OS and Software Updates	79
8.1	Upgrading the System	79
8.2	Changing Your GPU Driver Branch	79
8.2.1	Checking the Currently Installed Driver Branch	80
8.2.2	Determining the New Available Driver Branches	80
8.2.3	Upgrading Your GPU Driver Branch	80
8.3	Installing or Upgrading to a Newer CUDA Toolkit Release	85
8.3.1	CUDA Compatibility Matrix and Forward Compatibility	85
8.3.2	Checking the Currently Installed CUDA Toolkit Release	85
8.3.3	Installing or Upgrading the CUDA Toolkit	86
8.4	Installing the Latest DOCA-OFED Package	86
8.5	Installing GPUDirect Storage Support	87
8.5.1	Installing GDS Components	88
8.5.2	Enabling Relaxed Ordering for NVMe Drives	88
8.5.3	Configuring NVMe Interrupt Coalescing	88
8.5.4	Next Steps	89
9	Installing Firmware	91
9.1	Installing ConnectX-7 Firmware	91
9.2	Installing ConnectX-8 Firmware	92
9.3	Installing BlueField-3 NIC Firmware	93
9.3.1	Resetting the BlueField-3 BMC Password	95
10	System Configurations	97
10.1	Network Configuration	97
10.1.1	Configuring Network Proxies	97
10.1.2	For the OS and Most Applications	97
10.1.3	For the apt Package Manager	98
10.2	Configuring ConnectX from InfiniBand to Ethernet	98
10.2.1	Determining the Current Port Configuration	98
10.2.2	Configuring the Port	99
10.2.3	Configuring the ConnectX-8 Port for Splitting	100
10.3	Docker Configuration	101
10.3.1	Preparing the DGX System to be Used With Docker	101
10.3.2	Enabling Users To Run Docker Containers	101
10.3.3	Configuring Docker IP Addresses	102
10.3.4	Connectivity Requirements for NGC Containers	102
10.3.5	Configuring Static IP Addresses for the Network Ports	103
10.4	Managing CPU Mitigations	104
10.4.1	Determining the CPU Mitigation State of the DGX System	105
10.4.2	Disabling CPU Mitigations	105
10.4.3	Re-enable CPU Mitigations	106
10.5	Performance Impact of Enabling init_on_alloc	106
10.6	Managing the DGX Crash Dump Feature	107
10.6.1	Using the Script	107

10.7	Connecting to Serial Over LAN	107
10.8	Filesystem Quotas	107
10.9	Running Workloads on Systems with Mixed Types of GPUs	108
10.9.1	Running with Docker Containers	108
10.9.2	Running on Bare Metal	109
10.10	Using Multi-Instance GPUs	109
10.11	Updating the containerd Override File for MIG configurations	112
10.12	Data Storage Configuration	113
10.12.1	Using Data Storage for NFS Caching	113
10.12.2	Using cachefilesd	113
10.12.3	Disabling cachefilesd	114
10.12.4	Changing the RAID Configuration for Data Drives	114
10.13	Running NGC Containers	115
10.13.1	Obtaining an NGC Account	115
10.13.2	Running NGC Containers with GPU Support	115
11	Managing Self-Encrypting Drives	117
11.1	Overview	117
11.2	Installing the Software	118
11.3	Configuring Trusted Computing	118
11.3.1	Determining Whether Drives Support SID	119
11.3.2	Enabling the TPM and Preventing the BIOS from Sending Block SID Requests	119
11.4	Initializing the System for Drive Encryption	122
11.5	Enabling Drive Locking	123
11.6	Initialization Examples	123
11.6.1	Example 1: Passing in the JSON File	123
11.6.1.1	Determining Which Drives Can be Managed as Self-Encrypting	123
11.6.1.2	Creating the Drive/Password Mapping JSON Files and Using it to Initialize the System	124
11.6.2	Example 2: Generating Random Passwords	125
11.6.3	Example 3: Specifying Passwords One at a Time When Prompted	125
11.7	Disabling Drive Locking	126
11.8	Enabling Drive Locking	126
11.9	Exporting the Vault	126
11.10	Erasing Your Data	126
11.11	Clearing the TPM	127
11.12	Changing Disk Passwords, Adding Disks, or Replacing Disks	127
11.13	Recovering From Lost Keys	128
12	Resolved Issues	129
12.1	DGX Station A100 Fails to Boot After Applying MIG Configurations	129
12.1.1	Issue	129
12.1.2	Workaround	129
12.1.3	Status	129
12.2	Update the MLNX Firmware for the Connect-X and Bluefield-3 Adapters	129
12.2.1	Issue	129
12.2.2	Workaround	130
12.2.3	Status	130
12.3	Errors Occur When Loading Mirrored Repositories on Air-Gapped Systems	130
12.3.1	Issue	130
12.3.2	Explanation	130
12.3.3	Status	131
13	Known Issues	133

13.1	NVMe Opcode 0x2 (I/O Cmd) QID 256 Timeout	133
13.1.1	Issue	133
13.2	Upgrade Error During Package Configuration	133
13.2.1	Issue	133
13.2.2	Workaround	134
13.3	Error Messages Reported by GPU Driver	134
13.3.1	Issue	134
13.4	DGX GB200 System Failure During Upgrade	134
13.4.1	Issue	134
13.4.2	Workaround	134
13.5	RShim Devices Not Created After Starting rshim.service	135
13.5.1	Issue	135
13.5.2	Workaround	135
13.6	Firmware Crash on DGX GB200 During Boot	135
13.6.1	Issue	135
13.6.2	Workaround	136
13.7	The nvidia_peermem Module Does Not Load After an OTA Update	136
13.7.1	Issue	136
13.7.2	Workaround	137
13.8	The systemd-modules-load Service Failed to Insert the nvidia_peermem Module	137
13.8.1	Issue	137
13.8.2	Workaround	137
13.9	BMC Redfish Interface Not Active on the First Boot After Installation	137
13.9.1	Issue	137
13.9.2	Workaround	138
13.10	No Permissions to Access /var/run/nvidia-fabricmanager for Non-Root Users on DGX B200	138
13.10.1	Issue	138
13.10.2	Workaround	138
13.11	Kernel OOPS When Activating VFs from SR-IOV Network Operator	138
13.11.1	Issue	138
13.11.2	Workaround	139
13.12	ACCESS_REG Command Failure with Err(-22)	141
13.12.1	Issue	141
13.12.2	Workaround	142
13.13	DGX B200 Display Goes Blank Shortly After GRUB Menu	142
13.13.1	Issue	142
13.13.2	Workaround	142
13.14	nv-disk-encrypt Failed on Pre-Owned NVMe Drives	142
13.14.1	Issue	142
13.14.2	Workaround	143
13.15	GPUs Cannot Be Reset During MIG Configurations on DGX A100 and A800 Systems	143
13.15.1	Issue	143
13.15.2	Workaround	144
13.16	DGX System Device ID Not Found in /usr/share/misc/pci.ids	144
13.16.1	Issue	144
13.16.2	Workaround	144
13.17	Missing the nvidia-system-station Metapackage on DGX Station A100 and DGX Station A800	144
13.17.1	Issue	145
13.17.2	Workaround	145
13.18	Virtualization Not Supported	145
13.18.1	Issue	145

14 DGX OS Connectivity Requirements	147
14.1 In-Band Management, Storage, and Compute Networks	147
14.2 Out-of-Band Management	148
15 DGX Software Stack	149
15.1 NVIDIA DGX Software Packages	149
15.1.1 nvidia-system-core	149
15.1.2 nvidia-system-utils	150
15.1.3 nvidia-system-mlnx-drivers	150
15.2 DGX Kernel Parameters, System Configuration Settings, and Runtime Commands . . .	151
15.2.1 Kernel Parameters	151
15.2.2 System Configuration Settings	152
15.2.3 Runtime Commands	153
15.3 DGX Platform JSON Configuration	153
15.4 DGX Platform JSON Configuration Definitions	162
16 PXE Boot Setup	165
16.1 Requirements	165
16.2 Overview of the PXE Server	166
16.2.1 PXE Server Configuration	166
16.3 TFTP and HTTP Server Verification	170
16.4 Useful Parameters for Configuring Your System's Network Interfaces	170
16.5 Parameters Unique to the Base OS Installer	170
16.6 Configure Your DHCP Server	171
16.7 (Optional) Configure CX-4/5/6/7 Cards to PXE Boot	172
16.7.1 Enable UEFI for ConnectX Cards	172
16.8 (Optional) Configure the DGX-Server to PXE Boot Automatically	173
16.8.1 Add PXE to the Top of the UEFI Boot Order	173
16.9 Configure Network Boot Priorities	174
16.10 Make the DGX-Server PXE Boot	176
16.10.1 Automated PXE Boot Process	176
16.10.2 Manual PXE Boot Process	176
16.11 Other IPMI Boot Options	177
16.12 Autoinstall Customizations	177
16.13 NVIDIA-Specific Autoinstall Variables	178
16.14 Common Customizations	180
16.15 Network Configuration	180
16.16 Creating a User	180
17 Air-Gapped Installations	181
17.1 Creating a Local Mirror of the NVIDIA and Canonical Repositories	181
17.2 Creating the Mirror of the Repositories	182
17.3 Configuring the Target Air-Gapped System	184
18 Cloud-init Configuration File	187
18.1 Modifying the Configuration File	187
18.2 Drive Partitioning	190
19 Installing Docker Containers	193
20 Third-Party License Notices	195
20.1 Micron msecli	195
20.2 Mellanox (OFED)	196
21 Notices	197

21.1	Notice	197
21.2	Trademarks	198
22	About DGX OS 7	199
22.1	DGX OS 7 Features	199
22.2	Supported NVIDIA DGX Systems	199
22.3	Installation and Upgrade	200
22.4	Related Documentation	201
22.5	NVIDIA Enterprise Support	202

Chapter 1. About DGX OS 7

NVIDIA DGX OS provides a customized installation of Ubuntu Linux with system-specific optimizations and configurations, additional drivers, and diagnostic and monitoring tools. It provides a stable, fully tested, and supported OS to run AI, machine learning, and analytics applications on DGX Supercomputers.

NVIDIA DGX™ systems are shipped preinstalled with DGX OS to provide a turnkey solution for running AI and analytics workloads. Initial system configuration is deferred to a setup wizard that runs after the first boot. The setup wizard offers users a fast onboarding experience for using DGX systems.

The DGX OS installer is released as an ISO image to reimage a DGX system. The additional software, the NVIDIA DGX Software Stack included in DGX OS is provided as packages that are available from software repositories over the internet.

You also have the option to install the NVIDIA DGX Software Stack on a regular Ubuntu 24.04 while still benefiting from the advanced DGX features. This installation method supports more flexibility, such as custom partition schemes. Cluster deployments also benefit from this installation method by taking advantage of Ubuntu's standardized automated and non-interactive installation process.

1.1. DGX OS 7 Features

The following are the key features of the DGX OS 7 release:

- ▶ Based on Ubuntu 24.04 with the Linux kernel version 6.8 for the recent hardware and security updates and updates to software packages, such as Python, GCC, and OpenJDK.
- ▶ Includes the Ubuntu generic kernel (DGX servers based on x86_64) and the NVIDIA-optimized Linux kernel (DGX servers based on ARM64.)
- ▶ Provides access to all NVIDIA GPU driver branches and CUDA toolkit versions.
- ▶ Uses the NVIDIA DOCA™ OFED (OpenFabrics Enterprise Distribution) software, which is the successor to MLNX_OFED.
- ▶ Provides the Ubuntu Pro Client's Extended Security Maintenance (ESM) subscription from the Ubuntu Universe repository.
- ▶ Supports Emerald Rapids CPUs.

1.2. Supported NVIDIA DGX Systems

DGX OS 7 supports the following systems:

Table 1: Supported DGX Systems

Architecture x86_64	DGX Systems	Minimum DGX OS ISO Release
	DGX B300 2.3 TB	7.3.0
	DGX B200 1,440 GB	7.0.2
	DGX H200 1,128 GB	6.3.1
	DGX H100 640 GB	6.0.11
	DGX H800 640 GB	6.1.0
	DGX A100 640 GB	5.5.1
	DGX A100 320 GB	5.5.1
	DGX A800 640 GB	5.5.1
	DGX Station A100 320 GB	5.0.2
	DGX Station A100 160 GB	5.0.2
	DGX Station A800 320 GB	5.0.2
	DGX Systems	
Architecture ARM64		Minimum DGX OS ISO Release
	DGX GB300	7.2.3
	DGX GB200	7.1.0
	DGX Spark	7.2.3

DGX OS 7 does not support the following systems:

Table 2: Unsupported DGX Systems

Support	DGX Systems	End-of-Support Date
End of support	<ul style="list-style-type: none"> ▶ DGX-2 ▶ DGX-1 (V100) ▶ DGX Station (V100) 	August 2025

1.3. Installation and Upgrade

This document covers installation and upgrade options for DGX OS. It also provides instructions for setting up the system and installing additional software.

Initial Setup

If DGX OS 7 is already installed on your DGX system, refer to [Initial Setup](#) for information about setting up the system on first boot.

After initial setup, refer to [Upgrading the OS](#) to perform a *package upgrade* to the latest software package versions.

Upgrading the OS

To upgrade your DGX OS to the latest software package versions or for information about performing a *release upgrade* from DGX OS 5 or DGX OS 6 to DGX OS 7, refer to [Upgrading the OS](#).

Reimaging the System

To restore a DGX system to a default DGX OS installation and erase all data, you can use the ISO image that includes an autonomous installer. Refer to [Reimaging the System](#) for more information.

Customizing Ubuntu Installation with DGX Software

To install Ubuntu and the DGX Software Stack, refer to [Customizing Ubuntu Installation with DGX Software](#) for information about automating the installation process, such as a cluster deployment.

Managing OS and Software Updates

DGX OS and Ubuntu provide additional software packages, including additional NVIDIA software and driver options. Refer to [Managing OS and Software Updates](#) for more information and installation instructions.

Important

Before you upgrade or install any new software, always consult the [Release Notes](#) for the latest information about available upgrades. You can find out more about the release cadence and release methods for DGX OS in [Release Guidance](#).

1.4. Related Documentation

Refer to the following documents that are related to DGX OS 7:

- ▶ [DGX Documentation](#)

All documentation for DGX products, including product user guides, software release notes, and firmware update container information

- ▶ [MIG User Guide](#)

The Multi-Instance GPU (MIG) feature allows the NVIDIA A100 GPU to be securely partitioned into up to seven discrete GPU instances.

- ▶ [NGC Private Registry](#)

How to access the NGC container registry for using containerized deep learning GPU.

- ▶ [NVSM Software User Guide](#)

Contains instructions for using the NVIDIA System Manager software.

- ▶ [DCGM Software User Guide](#)

Contains instructions for using the Data Center GPU Manager software.

1.5. NVIDIA Enterprise Support

NVIDIA Enterprise Support is the support resource for DGX customers and can assist with hardware, software, or NGC application issues. For more information about how to obtain support, visit [NVIDIA Enterprise Support](#).

Chapter 2. Release Guidance

This information helps you understand the DGX OS release mechanism, release numbering convention, and options to install and upgrade your DGX OS software.

Important

Keeping your DGX OS software and firmware up to date is the most important task for protecting your DGX systems. Security-related updates are available from the Ubuntu and NVIDIA repositories. Refer to [Upgrading the OS](#) for information about performing a package upgrade to the latest software releases and the [Upgrades](#) page on Ubuntu's Wiki for additional information.

Ubuntu security updates are announced on [Ubuntu Security Notices \(USN\)](#), which lists known Common Vulnerabilities and Exposures (CVEs) for Ubuntu packages. To keep up to date on NVIDIA security announcements for software and firmware, refer to the [Product Security](#) page.

2.1. Release Mechanisms

This section provides information about the DGX OS release mechanisms that are available to install or upgrade DGX systems to the latest version of the DGX OS.

2.1.1. DGX OS ISO Image

DGX OS is released in the form of an ISO image that includes an autonomous installer and all the software required for running AI, machine learning, and analytics applications. Basic system configuration is deferred to a setup wizard on first boot. The ISO is intended for cases that require reimaging a single system.

Updated versions of the ISO image are available between releases with the following:

- ▶ Critical bug fixes and security mitigations.
- ▶ Improved installation experience.
- ▶ Support for new DGX systems and hardware components.

Always use the latest ISO image unless you need to restore the system to an earlier version.

2.1.2. Linux Software Repositories

The DGX OS software, upgrades to DGX OS, and a large pool of additional software are available from software repositories in the form of software packages. Software repositories are storage locations from which your system retrieves the software packages. The repositories used by DGX OS are hosted by Canonical for the Ubuntu OS and NVIDIA for DGX specific software and other NVIDIA software.

Software packages contain the files of the respective software component or closely related components. This includes executables, configuration files, documentation, and metadata, such as dependencies.

New versions of these packages contain bug fixes and security updates. These packages provide an upgrade to a DGX OS release. The repositories are also updated between releases for critical bug fixes, security mitigations, and additional hardware enablement. Updates for supporting a new system or a new hardware component, such as a network card or disk drive, do not affect existing hardware configurations.

System upgrades are cumulative. Your system always receives the latest package versions, which can be newer than the current DGX OS release. You cannot select which upgrades to make or limit upgrades to a specific DGX OS version by default.

Warning

NVIDIA recommends that you do not update packages individually. For example, do not run `apt install <package-name>`.

2.2. Release Numbering Convention

The NVIDIA DGX OS release numbering convention is <major>.<minor>. The meaning for major and minor are as follows:

Major Release

Major releases are typically based on Ubuntu releases. These releases include new kernel versions and new features that might not be backwards compatible. They also include the latest releases of the NVIDIA software.

For example:

- ▶ DGX OS 5 is based on Ubuntu 20.04.
- ▶ DGX OS 6 is based on Ubuntu 22.04.
- ▶ DGX OS 7 is based on Ubuntu 24.04.

Minor Release

Minor releases include new NVIDIA features, bug fixes, and security updates.

- ▶ Minor releases deliver upgrades to the NVIDIA software that are backward compatible.
- ▶ When you upgrade a system, the system remains on the currently installed NVIDIA GPU driver release unless the driver branch reaches the end of support.
- ▶ The ISO releases provide a snapshot of all accumulated changes. The release can include a newer NVIDIA GPU driver LTS branch.

Important bug fixes and security updates are provided between releases through the software repositories without having a release number assigned to the change.

2.3. Release Cadence

DGX OS is released twice a year, typically around February and August, for the first two years as minor releases. Updates are provided between releases and after the initial two years for the remainder of the supported duration.

Chapter 3. Release Notes

Note

Because software upgrades are cumulative, your systems will always receive the most recent versions of all installed components. The packages in the repositories might be newer than the current DGX OS release. Carefully review release information and advisories for all relevant upgrades.

3.1. Current Software Versions

The following table shows the current version information of the software packages provided in the NVIDIA and Ubuntu repositories for the NVIDIA DGX Software Stack.

Table 1: Current Software Versions (Last Updated on November 18, 2025)

Component		Version	Additional Information
GPU Driver		580.105.08	OS 7: DEB download OS 6: DEB download
GPU Driver		570.195.03	OS 7: DEB download OS 6: DEB download
GPU Driver		535.274.02	OS 6: DEB download OS 5: DEB download (535.261.03)
CUDA Toolkit		13.0 Update 2	R580: 13.0 Update 2 download
CUDA Toolkit		12.8 Update 1	R570: 12.8 Update 1 download
CUDA Toolkit		12.2 Update 2	R535: 12.2 Update 2 download
DOCA OFED		2.9.3	2.9.3 download
Inbox OFED		<ul style="list-style-type: none"> ▶ 50.0-2 for OS 7 ▶ 39.0-1 for OS 6 	
NCCL		2.28.7	
cuDNN		9.15.0	
DCGM		4.4.1	
GPUDirect (GDS)	Storage	<ul style="list-style-type: none"> ▶ 1.15 for CUDA Toolkit 13.0 ▶ 1.14 for CUDA Toolkit 12.9 ▶ 1.13 for CUDA Toolkit 12.8 ▶ 1.11 for CUDA Toolkit 12.6 ▶ 1.10 for CUDA Toolkit 12.5 ▶ 1.9 for CUDA Toolkit 12.4 ▶ 1.8 for CUDA Toolkit 12.3 ▶ 1.7 for CUDA Toolkit 12.2 	
NVIDIA Toolkit	Container	1.18.0	NVIDIA Container Toolkit includes the following packages: <ul style="list-style-type: none"> ▶ nvidia-container-toolkit: 1.18.0 ▶ libnvidia-container-tools: 1.18.0 ▶ libnvidia-container1: 1.18.0
nvidia-peer-memory		1.3	

Note

- For all DGX Stations installed with DGX OS, the CUDA Toolkit is installed by default. For all DGX servers installed with DGX OS and all systems installed with NVIDIA DGX Software for Red Hat Enterprise, the CUDA Toolkit is not installed by default; however, you can manually install a qualified CUDA Toolkit release. Refer to the [CUDA Release Notes](#) for driver compatibility information.
- Relevant Information:

For information on	Refer to
CUDA Toolkit minor version compatibility and the minimum required driver version	CUDA Compatibility
Driver and CUDA Toolkit lifecycles	Supported Drivers and CUDA Toolkit Versions
MLNX_OFED release transition	Adapter Software

The following table provides information about the matching firmware versions for the NVIDIA DOCA™ Host package with the doca-ofed installation profile ([v2.9.1](#) and [v3.1.0](#)).

Table 2: ConnectX Firmware Versions

OS	DGX A100	DGX Hx00/B200 (DOCA 2.9.1-3.0.2)	DGX B200/B300 (DOCA 3.1.0-091548)	DGX GB200/GB300 (DOCA 3.1.0-091513)
OS 7	CX6: 20.43.2026 CX7: 28.43.2026	CX7: 28.43.2026	CX7: 28.46.5500 CX8: 40.46.5500	CX7: 28.46.3048 CX8: 40.46.3048

Firmware for the NVIDIA® BlueField® DPU in NIC mode:

Table 3: BlueField-3 Firmware Versions

OS	DGX Hx00	DGX B200 (DOCA 2.9.1-3.0.2)	DGX B200/B300 (DOCA 3.1.0-091548)	DGX GB200/GB300 (DOCA 3.1.0-091513)
OS 7	BF3: 32.43.2026	BF3: 32.43.2026	BF3: 32.46.5500	BF3: 32.46.3048

For installation instructions, refer to

- ConnectX®-7, ConnectX®-8, and BlueField®-3 adapter cards: [Installing Firmware](#)

- ConnectX®-6 adapter cards: [Firmware Downloads](#)

In addition to upgrading to the versions described in this section, performing a *package upgrade* can update the software component versions, the Ubuntu 24.04 LTS version, and the Ubuntu kernel, depending on when you perform the upgrade.

Note

For information about LTS software versions for related networking components, refer to the [Networking Long-Term Support Releases](#) page.

Important

This release incorporates the following updates:

- Ubuntu ConnectX drivers and DOCA-OFED stack.
- Customers are advised to consider these updates and any effect they might have on their application. For example, some MOFED-dependent applications can be affected.
- Best practices support upgrading select systems and verifying that your applications are working as expected before you deploy additional systems.

3.2. Supported DGX Systems

DGX OS 7 supports the following systems:

Table 4: Supported DGX Systems

x86_64	DGX Systems	Minimum DGX OS ISO Release	Firmware Release
	DGX B300 2.3 TB	7.3.1	25.11.1
	DGX B200 1,440 GB	7.0.2	25.09.1
	DGX H200 1,128 GB	6.3.1	25.10.1
	DGX H100 640 GB	6.0.11	25.10.1
	DGX H800 640 GB	6.1.0	25.10.1
	DGX A100 640 GB	5.5.1	25.5.4
	DGX A100 320 GB	5.5.1	25.5.4
	DGX A800 640 GB	5.5.1	25.5.4
	DGX Station A100 320 GB	5.0.2	25.10.1
	DGX Station A100 160 GB	5.0.2	25.10.1
	DGX Station A800 320 GB	5.0.2	25.10.1
ARM64	DGX Systems	Minimum DGX OS ISO Release	Firmware Release
	DGX GB300	7.2.3	1.0
	DGX GB200	7.1.0	1.3
	DGX Spark	7.2.3	

DGX OS 7 does not support the following systems:

Table 5: Unsupported DGX Systems

Support	DGX Systems
End of support (August 2025)	<ul style="list-style-type: none"> ► DGX-2 ► DGX-1 (V100) ► DGX Station (V100)

3.3. Latest Release

3.3.1. DGX OS 7.3.1

Release Date: November 18, 2025

This release introduces the following new features and updates:

- ▶ The *DGX OS ISO 7.3.1* has been released.
- ▶ OS base: *Ubuntu 24.04.3 LTS*
- ▶ Integrates over-the-air (OTA) update functionality for the NVIDIA DGX™ Spark System.
- ▶ NVIDIA GPU drivers:

Architecture	DGX Systems	GPU Driver
x86_64	DGX B300	580.105.08
x86_64	DGX B200 DGX Hx00 DGX A100 DGX Station A100	570.195.03 and 580.105.08 (optional)
ARM64	DGX GB300 DGX GB200	580.105.08
ARM64	DGX Spark	580.95.05 (Canonical signed driver)

- ▶ Updated support for the NVIDIA DOCA™ Host package with the doca-ufed installation profile:

Architecture	DGX Systems	GPU Driver	DOCA Support
x86_64	DGX B300	580.105.08	<ul style="list-style-type: none"> ▶ DOCA Documentation v3.1.0 ▶ DOCA 3.1.0-091548 download
x86_64	DGX B200 DGX Hx00 DGX A100 DGX Station A100	570.195.03 and 580.105.08 (optional)	<ul style="list-style-type: none"> ▶ DOCA Documentation v2.9.1 LTS ▶ DOCA 2.9.1-3.0.2 download ▶ DOCA 3.1.0-091548 download (optional)
ARM64	DGX GB300 DGX GB200	580.105.08	<ul style="list-style-type: none"> ▶ DOCA Documentation v3.1.0 ▶ DOCA 3.1.0-091513 download
ARM64	DGX Spark	580.95.05 (Canonical signed)	<ul style="list-style-type: none"> ▶ Inbox OFED 50.0-2 ▶ No doca-ofed support

- ▶ The following changes were made to the repositories:

- ▶ Kernel:

Architecture	Kernel
x86_64	6.8.0-87-generic
ARM64 (DGX GB300, DGX GB200)	6.14.0-1013-nvidia-64k
ARM64 (DGX Spark)	6.14.0-1013-nvidia

- ▶ CUDA Toolkit:
 - ▶ [12.8 Update 1](#) for GPU driver 570.195.03
 - ▶ [13.0 Update 2](#) for GPU drivers 580.105.08 and 580.95.05
- ▶ NCCL: [2.28.7](#)
- ▶ NVIDIA Container Toolkit: [1.18.0](#)
- ▶ MIG Configuration Tool: [0.13.0](#)
- ▶ DLFW: [25.10](#)

3.4. Previous Releases

3.4.1. DGX OS 7.3.0

Release Date: October 30, 2025

This release introduces the following new features and updates:

- ▶ The [DGX OS ISO 7.3.0](#) has been released.
- ▶ OS base: [Ubuntu 24.04.3 LTS](#)
- ▶ Introduces support for the NVIDIA DGX™ B300 system.
- ▶ NVIDIA GPU drivers:

Architecture	GPU Driver
x86_64 (DGX B300)	580.95.05
x86_64 (excluding DGX B300)	570.195.03 and 580.95.05 (optional)

- ▶ Updated NVIDIA® BlueField®-3 DPU in NIC mode support for newer firmware and host software:

Architecture	DGX Systems	BlueField-3 Support
x86_64	DGX B300	BlueField-3 firmware 32.46.3052
x86_64	DGX B200 DGX Hx00 DGX A100 DGX Station A100	BlueField-3 firmware 32.43.2026 download

- ▶ Updated support for the NVIDIA DOCA™ Host package with the doca-ufed installation profile:

Architecture	GPU Driver	DOCA Support
x86_64 (DGX B300)	580.95.05	<ul style="list-style-type: none"> ▶ DOCA Documentation v3.1.0 ▶ NVIDIA DOCA 3.1.0-091544 download
x86_64 (excluding DGX B300)	580.95.05 (optional)	<ul style="list-style-type: none"> ▶ DOCA Documentation v3.1.0 ▶ NVIDIA DOCA 3.1.0-091544 download
x86_64 (excluding DGX B300)	570.195.03	<ul style="list-style-type: none"> ▶ DOCA Documentation v2.9.1 LTS ▶ NVIDIA DOCA 2.9.1-3.0.2 download

- ▶ The following changes were made to the repositories:

- ▶ Kernel:
 - ▶ x86_64: 6.8.0-86-generic
- ▶ CUDA Toolkit:
 - ▶ 12.8 Update 1 for GPU driver 570.195.03
 - ▶ 13.0 Update 2 for GPU driver 580.95.05
- ▶ cuDNN: 9.14.0
- ▶ NVSM: 25.09.03
- ▶ Docker Engine: 28.5.1
- ▶ DLFW: 25.09

The following table provides information about the matching firmware versions for the NVIDIA DOCA™ Host package with the doca-oped installation profile (v2.9.1 and v3.1.0).

Table 6: ConnectX Firmware Versions

OS	DGX A100 ConnectX-6 ConnectX-7	DGX Hx00/B200 ConnectX-7	DGX B300 ConnectX-7 ConnectX-8	DGX GB200/GB300 ConnectX-7 ConnectX-8
OS 7	CX6: 20.43.2026 CX7: 28.43.2026	28.43.2026	CX7: 28.46.3052 CX8: 40.46.3052	CX7: 28.46.3048 CX8: 40.46.3048

Firmware for the NVIDIA® BlueField® DPU in NIC mode:

Table 7: BlueField-3 Firmware Versions

OS	DGX H100/H200	DGX B200	DGX B300	DGX GB200/GB300
OS 7	32.43.2026	32.43.2026	32.46.3052	32.46.3048

3.4.2. DGX OS 7.2.3

Release Date: October 10, 2025

This release introduces the following new features and updates:

- ▶ The *DGX OS ISO 7.2.3* has been released.
- ▶ Introduces support for the NVIDIA DGX™ Spark and NVIDIA DGX™ GB300 systems.
- ▶ Updated NVIDIA® BlueField®-3 DPU in NIC mode support for newer firmware and host software:

Architecture	DGX Systems	BlueField-3 Support
x86_64	DGX B200 DGX H200 DGX H100	<ul style="list-style-type: none"> ▶ NVIDIA BlueField-3 DPU NIC Firmware Release Notes v32.43.2402 LTS ▶ NVIDIA BlueField-3 firmware download
ARM64	DGX GB300 DGX GB200	NVIDIA BlueField-3 firmware 32.46.3048

- ▶ Updated support for the NVIDIA DOCA™ Host package with the doca-ofed installation profile:

Architecture	GPU Driver	DOCA Support
x86_64	570.195.03	<ul style="list-style-type: none"> ▶ DOCA Documentation v2.9.1 LTS ▶ NVIDIA DOCA 2.9.1-3.0.2 download
x86_64	580.95.05 (optional)	<ul style="list-style-type: none"> ▶ DOCA Documentation v3.0.0 ▶ NVIDIA DOCA 3.0.0-058218 download
ARM64 (GB300, GB200)	580.95.05	<ul style="list-style-type: none"> ▶ DOCA Documentation v3.1.0 ▶ NVIDIA DOCA 3.1.0-091513 download
ARM64 (DGX Spark)	580.95.05	<ul style="list-style-type: none"> ▶ Inbox OFED 50.0-2 ▶ No doca-ofed support

- ▶ The following changes were made to the repositories:

- ▶ OS base: [Ubuntu 24.04.3 LTS](#)
- ▶ Kernel:

Architecture	Kernel
x86_64	6.8.0-85-generic
ARM64 (DGX GB300, DGX GB200)	6.14.0-1008-nvidia-64k
ARM64 (DGX Spark)	6.11.0-1016-nvidia

- ▶ NVIDIA GPU drivers:

Architecture	GPU Driver
x86_64	570.195.03 and 580.95.05 (optional)
ARM64 (DGX GB300, DGX GB200)	580.95.05
ARM64 (DGX Spark)	580.95.05 (Canonical signed driver)

- ▶ CUDA Toolkit:
 - ▶ 12.8 Update 1 for GPU driver 570.195.03
 - ▶ 13.0 Update 1 for GPU driver 580.95.05
- ▶ NCCL: 2.28.3
- ▶ cuDNN: 9.13.0
- ▶ DCGM: 4.4.1
- ▶ NVSM: 25.03.09
- ▶ Docker Engine: 28.3.3
- ▶ GDRCopy: 2.5.1
- ▶ DLFw: 25.08

The following table provides information about the matching firmware versions for the NVIDIA DOCA™ Host package with the doca-ified installation profile (v2.9.1 and v3.1.0).

Table 8: ConnectX Firmware Versions

OS	DGX A100 ConnectX-6 ConnectX-7	DGX Hx00/B200 ConnectX-7	DGX Spark ConnectX-7	DGX GB200 ConnectX-7 ConnectX-8	DGX GB300 ConnectX-7 ConnectX-8
OS 7	CX6: 20.43.2026 CX7: 28.43.2026	28.43.2026	28.45.4028	CX7: 28.46.3048 CX8: 40.46.3048	CX7: 28.46.3048 CX8: 40.46.3048

Firmware for the NVIDIA® BlueField® DPU in NIC mode:

Table 9: BlueField-3 Firmware Versions

OS	DGX H100/H200	DGX B200	DGX GB200	DGX GB300
OS 7	32.43.2402	32.43.2402	32.46.3048	32.46.3048

3.4.3. DGX OS 7.2.1

Release Date: August 19, 2025

This release introduces the following new features and updates:

- ▶ An ISO release was not issued for DGX OS 7.2.1.
- ▶ Removed support for the DGX-1™V, DGX-2™, and DGX Station™ systems.
- ▶ Updated NVIDIA® BlueField®-3 DPU in NIC mode support for newer firmware and host software:

Architecture	DGX Systems	BlueField-3 Support
x86_64	DGX B200 DGX H200 DGX H100	<ul style="list-style-type: none"> ▶ NVIDIA BlueField-3 DPU NIC Firmware Release Notes v32.43.2402 LTS ▶ NVIDIA BlueField-3 firmware download
ARM64	DGX GB200	NVIDIA BlueField-3 firmware 32.44.1600

- ▶ Updated support for the NVIDIA DOCA™ Host package with the doca-ofd installation profile:

Architecture	GPU Driver	DOCA Support
x86_64	570.172.08	<ul style="list-style-type: none"> ▶ DOCA Documentation v2.9.1 LTS ▶ NVIDIA DOCA 2.9.1-3.0.2 download
x86_64 (DGX B200)	580.65.06 (optional)	<ul style="list-style-type: none"> ▶ DOCA Documentation v3.0.0 ▶ NVIDIA DOCA 3.0.0-058218 download
ARM64 (DGX GB200)	570.172.08	<ul style="list-style-type: none"> ▶ DOCA Documentation v2.10.0 ▶ NVIDIA DOCA 2.10.0-093520 download

- ▶ The following changes were made to the repositories:
 - ▶ OS base: [Ubuntu 24.04.2 LTS](#)
 - ▶ Kernel:
 - ▶ x86_64: [6.8.0-71-generic](#)
 - ▶ ARM64 (DGX GB200): [6.11.0-1013-nvidia-64k](#)
 - ▶ NVIDIA GPU drivers:
 - ▶ [570.172.08](#) for CUDA ToolKit 12.8 Update 1
 - ▶ [580.65.06](#) (optional) for CUDA ToolKit 13.0

- ▶ CUDA Toolkit: [13.0](#)
- ▶ NCCL: [2.27.6](#)
- ▶ cuDNN: [9.11.0](#)
- ▶ DCGM: [4.4.0](#)
- ▶ NVSM: [25.03.08](#)
- ▶ Docker Engine: [28.3.2](#)
- ▶ NVIDIA Container Toolkit: [1.17.8](#)
- ▶ MIG Configuration Tool: [0.12.3](#)
- ▶ DLFW: [25.06](#)

The following table provides information about the matching firmware versions for the NVIDIA DOCA™ Host package with the doca-ufed installation profile ([v2.9.1](#) and [v2.10.0](#)).

Table 10: ConnectX-6 and ConnectX-7 Firmware Versions

OS	DGX A100 ConnectX-6	DGX A100 ConnectX-7	DGX H100/H200/B200 ConnectX-7	DGX GB200 ConnectX-7
OS 7	20.43.2026	28.43.2026	28.45.0350	28.44.2524

Firmware for the NVIDIA® BlueField® DPU in NIC mode:

Table 11: BlueField-3 Firmware Versions

OS	DGX H100/H200 BlueField-3	DGX B200 BlueField-3	DGX GB200 BlueField-3
OS 7	32.43.2402	32.43.2402	32.44.1600

3.4.4. DGX OS 7.1.0

Release Date: April 30, 2025

This release introduces the following new features and updates:

- ▶ The [DGX OS ISO 7.1.0](#) has been released.
- ▶ Introduces support for the NVIDIA DGX™ GB200 system.
- ▶ Ends support for DGX 1V, DGX2, and DGX Station systems.
- ▶ Includes support for the NVIDIA® BlueField®-3 DPU in NIC mode on DGX H100, DGX H200, and DGX B200 systems:

- ▶ [NVIDIA BlueField-3 DPU NIC Firmware Release Notes v32.43.2402 LTS](#)
- ▶ [NVIDIA BlueField-3 firmware download](#)

DGX GB200 system: NVIDIA BlueField-3 firmware 32.44.1600

- ▶ Includes support for the NVIDIA DOCA™ Host package with the doca-ofed installation profile [v2.9.1](#) (containing the MLNX_OFED functionality).
 - ▶ x86_64: [NVIDIA DOCA 2.9.1-3 download](#)
 - ▶ ARM64: [NVIDIA DOCA 2.10.0-093509 download](#)
- ▶ The following changes were made to the repositories:
 - ▶ OS base: [Ubuntu 24.04.1](#)
 - ▶ Kernel:
 - ▶ x86_64: [6.8.0-58-generic](#)
 - ▶ ARM64: [6.8.0-1025-nvidia-64k](#)
 - ▶ NVIDIA GPU Driver: [570.124.06](#)
 - ▶ NCCL: [2.26.2](#)
 - ▶ DCGM: [4.2.2](#)
 - ▶ NVSM: [25.03.03](#)
 - ▶ Docker Engine: [28.0.4](#)
 - ▶ GDRCopy: [2.5](#)
 - ▶ MIG Configuration Tool: [0.12.1](#)
 - ▶ DLFW: [25.03](#)

The following table provides information about the matching firmware versions for the NVIDIA DOCA™ Host package with the doca-ofed installation profile [v2.9.1](#).

Table 12: Matching Firmware Versions

OS	DGX A100 ConnectX-6	DGX A100 ConnectX-7	DGX H100/H200/B200 ConnectX-7	DGX GB200 ConnectX-7
OS 7	20.43.2026	28.43.2026	28.43.2026	28.44.2506

Firmware for the NVIDIA® BlueField® DPU in NIC mode:

Table 13: BlueField-3 Firmware Versions

OS	DGX H100/H200 BlueField-3	DGX B200 BlueField-3	DGX GB200 BlueField-3
OS 7	32.43.2402	32.43.2402	32.44.1600

3.4.5. DGX OS 7.0.2

Release Date: March 20, 2025

This release introduces the following new features and updates:

- ▶ The [DGX OS ISO 7.0.2](#) has been released.
- ▶ Support for the NVIDIA DGX™ B200 system.
- ▶ Discontinues support for DGX 1V, DGX2, and DGX Station systems.
- ▶ Enables upgrades to the latest DGX OS 7 release from the latest DGX OS 6 release.
- ▶ Includes support for the NVIDIA® BlueField®-3 DPU in NIC mode v32.43.2402 LTS on DGX H100, DGX H200, and DGX B200 systems.
 - ▶ [NVIDIA BlueField-3 DPU NIC Firmware Release Notes v32.43.2402 LTS](#)
 - ▶ [NVIDIA BlueField-3 firmware download](#)
- ▶ Includes support for the NVIDIA DOCA™ Host package with the doca-ufed installation profile v2.9.1 (containing the MLNX_OFED functionality).
 - ▶ [NVIDIA DOCA 2.9.1-3.0.0 download](#)
 - ▶ [MLNX_OFED release transition](#)
- ▶ The following changes were made to the repositories:
 - ▶ OS base: [Ubuntu 24.04.1](#)
 - ▶ Kernel: [6.8.0-55-generic](#)
 - ▶ NVIDIA GPU Driver:
 - ▶ Release 570 family: [570.124.06](#)
 - ▶ CUDA Toolkit: [12.8.1](#)
 - ▶ cuDNN: [9.8.0](#)
 - ▶ DCGM: [4.1.1](#)
 - ▶ Docker Engine: [27.5.1](#)
 - ▶ NVIDIA Container Toolkit: [1.17.5](#)
 - ▶ MIG Configuration Tool: [0.11.0](#)
 - ▶ DLFW: [25.02](#)

The following table provides information about the matching firmware versions for the NVIDIA DOCA™ Host package with the doca-ofed installation profile [v2.9.1](#).

Table 14: Matching Firmware Versions

OS	DGX A100 ConnectX-6	DGX A100 ConnectX-7	DGX H100/H200 ConnectX-7
OS 7	20.43.2026	28.43.2026	28.43.2026

Firmware for the NVIDIA® BlueField® DPU in NIC mode:

Table 15: BlueField-3 Firmware Versions

OS	DGX H100/H200 BlueField-3	DGX B200 BlueField-3
OS 7	32.43.2402	32.43.2402

3.4.6. DGX OS 7.0.1

Release Date: January 29, 2025

This release introduces the following new features and updates:

- ▶ Introduces support for the NVIDIA DGX™ B200 system.
- ▶ Discontinues support for DGX 1V, DGX2, DGX Station systems.
- ▶ Adds support for the NVIDIA® BlueField®-3 DPU in NIC mode v32.43.2026 LTS on DGX H100/H200 and DGX B200 systems.
 - ▶ [NVIDIA BlueField-3 firmware download](#)
- ▶ Includes support for the NVIDIA DOCA™ Host package with the doca-ofed installation profile [v2.9.1](#) (containing the MLNX_OFED functionality).
 - ▶ [NVIDIA DOCA 2.9.1-2.0.0 download](#)
 - ▶ [MLNX_OFED release transition](#)
- ▶ Migrates the repository sources from one-line-style format to the [Deb822-style format](#).
- ▶ The following changes were made to the repositories and the ISO:
 - ▶ OS base: [Ubuntu 24.04 LTS](#)
 - ▶ Kernel: [6.8.0-51-generic](#)
 - ▶ NVIDIA GPU Drivers:
 - ▶ Release 570 family (default): [570.86.15](#)

- ▶ CUDA Toolkit: 12.8
- ▶ NCCL: 2.25.1
- ▶ cuDNN: 9.7.0
- ▶ DCGM: 4.1.0
- ▶ GPUDirect Storage: 1.13
- ▶ NVSM: 24.09.05
- ▶ Docker Engine: 27.4.1
- ▶ NVIDIA Container Toolkit: 1.17.3
- ▶ GDRCopy: 2.4.4
- ▶ MIG Configuration Tool: 0.10.0
- ▶ DLFW: 25.01

3.5. DGX OS ISO Releases

This section lists all DGX OS ISO releases with the software versions included in the image.

3.5.1. DGX OS ISO 7.3.1

Table 16: DGX OS ISO 7.3.1 for DGX Systems on x86_64 or ARM64 Architecture

Component	Version
Ubuntu	Ubuntu 24.04.3 LTS Base OS 7.3.1
Ubuntu Kernel	<ul style="list-style-type: none"> ▶ x86_64: 6.8.0-87-generic ▶ ARM64 (DGX GBx00): 6.14.0-1013-nvidia-64k ▶ ARM64 (DGX Spark): 6.14.0-1013-nvidia
GPU Driver	<ul style="list-style-type: none"> ▶ x86_64 (DGX B300): 580.105.08 ▶ x86_64 (excluding DGX B300): 570.195.03 and 580.105.08 (optional) ▶ ARM64 (DGX GBx00): 580.105.08 ▶ ARM64 (DGX Spark): 580.95.05
CUDA Toolkit ¹	<ul style="list-style-type: none"> ▶ 13.0 Update 2 for GPU drivers 580.105.08 and 580.95.05 ▶ 12.8 Update 1 for GPU drivers 570.195.03
NCCL	2.28.7
cuDNN	9.14.0
DCGM	4.4.1
Inbox OFED	50.0-2
GPUDirect Storage (GDS)	<ul style="list-style-type: none"> ▶ 1.15 for CUDA 13.0 ▶ 1.13 for CUDA 12.8
NVIDIA Container Toolkit	1.18.0
NVSM	25.09.03
Docker Engine	28.5.1
GDRCopy	2.5.1
MIG Configuration Tool ²	0.13.0
DLFW	25.10
ISO	x86_64: DGXOS-7.3.1-2025-11-12-02-34-35-amd64.iso ARM64: DGXOS-7.3.1-2025-11-12-09-12-21-arm64.iso
MD5 Checksum	x86_64: 7e9aa6b3995d48d80bbe5c7e75d17f74 ARM64: 41c55c8d90234dc400790793322b1398

¹ CUDA Toolkit is only installed by default for DGX stations and is optional for DGX servers. Refer to the [CUDA Release Notes](#) for driver compatibility information.

² Refer to [NVIDIA mig-parted github pages](#) and [deployments](#).

3.5.2. DGX OS ISO 7.3.0

Table 17: DGX OS ISO 7.3.0 for DGX Systems on x86_64 Architecture

Component	Version
Ubuntu	Ubuntu 24.04.3 LTS Base OS 7.3.0
Ubuntu Kernel	6.8.0-86-generic
GPU Driver	<ul style="list-style-type: none"> ▶ x86_64 (DGX B300): 580.95.05 ▶ x86_64 (excluding DGX B300): 570.195.03 and 580.95.05 (optional)
CUDA Toolkit	<ul style="list-style-type: none"> ▶ 13.0 Update 2 for GPU driver 580.95.05 ▶ 12.8 Update 1 for GPU driver 570.195.03
NCCL	2.28.3
cuDNN	9.14.0
DCGM	4.4.1
Inbox OFED	50.0-2
GPUDirect Storage (GDS)	<ul style="list-style-type: none"> ▶ 1.15 for CUDA 13.0 ▶ 1.13 for CUDA 12.8
NVIDIA Container Toolkit	1.17.8
NVSM	25.09.03
Docker Engine	28.5.1
GDRCopy	2.5.1
MIG Configuration Tool	0.12.3
DLFW	25.09
ISO	x86_64: DGXOS-7.3.0-2025-10-14-11-13-07-amd64.iso
MD5 Checksum	x86_64: 10e48709440b7d762a99825e9a409c41

3.5.3. DGX OS ISO 7.2.3

Table 18: DGX OS ISO 7.2.3 for DGX Systems on x86_64 or ARM64 Architecture

Component	Version
Ubuntu	Ubuntu 24.04.3 LTS Base OS 7.2.3
Ubuntu Kernel	<ul style="list-style-type: none"> ▶ x86_64: 6.8.0-85-generic ▶ ARM64 (DGX GBx00): 6.14.0-1008-nvidia-64k ▶ ARM64 (DGX Spark): 6.11.0-1016-nvidia
GPU Driver	<ul style="list-style-type: none"> ▶ x86_64: 570.195.03 and 580.95.05 (optional) ▶ ARM64: 580.95.05
CUDA Toolkit	<ul style="list-style-type: none"> ▶ 13.0 Update 1 for GPU driver 580.95.05 ▶ 12.8 Update 1 for GPU driver 570.195.03
NCCL	2.28.3
cuDNN	9.13.0
DCGM	4.4.1
Inbox OFED	50.0-2
GPUDirect Storage (GDS)	<ul style="list-style-type: none"> ▶ 1.15 for CUDA 13.0 ▶ 1.13 for CUDA 12.8
NVIDIA Container Toolkit	1.17.8
NVSM	25.03.09
Docker Engine	28.3.3
GDRCopy	2.5.1
MIG Configuration Tool	0.12.3
DLFW	25.08
ISO	x86_64: DGXOS-7.2.3-2025-10-04-05-22-49-amd64.iso ARM64: DGXOS-7.2.3-2025-10-04-06-28-28-arm64.iso
MD5 Checksum	x86_64: 3e96d69fc428951f9f891556f5b39058 ARM64: 4993184b420d8c3b1609fa6fad56b17c

3.5.4. DGX OS ISO 7.1.0

Table 19: DGX OS ISO 7.1.0 for DGX Systems on x86_64 or ARM64 Architecture

Component	Version
Ubuntu	Ubuntu 24.04.1 Base OS 7.1.0
Ubuntu Kernel	x86_64: 6.8.0-57-generic ARM64: 6.8.0-1025-nvidia-64k
GPU Driver	570.124.06
CUDA Toolkit	12.8.1
NCCL	2.26.2
cuDNN	9.8.0
DCGM	4.2.2
GPUDirect Storage (GDS)	<ul style="list-style-type: none"> ▶ 1.13 for CUDA 12.8 ▶ 1.9 for CUDA 12.4
NVIDIA Container Toolkit	1.17.5
NVSM	25.03.03
Docker Engine	28.0.4
GDRCopy	2.5
MIG Configuration Tool	0.12.1
DLFW	25.03
ISO	x86_64: DGXOS-7.1.0-2025-04-14-18-49-33-amd64.iso ARM64: DGXOS-7.1.0-2025-04-14-21-14-06-arm64.iso
MD5 Checksum	x86_64: b7058643d2509f62955f554cdaa2e270 ARM64: 3d694f362296e935d59f48ec29dbe1bd

3.5.5. DGX OS ISO 7.0.2

Table 20: DGX OS ISO 7.0.2 for DGX Systems on x86_64 Architecture

Component		Version	Notes
Ubuntu		Ubuntu 24.04.1 Base OS 7.0.2	
Ubuntu Kernel		6.8.0-55-generic	
GPU Driver		570.124.06	
CUDA Toolkit		12.8.1	CUDA Toolkit is only installed by default for DGX stations and is optional for DGX servers. Refer to the CUDA Release Notes for driver compatibility information.
NCCL		2.25.1	
cuDNN		9.8.0	
DCGM		4.1.1	
GPUDirect (GDS)	Storage	<ul style="list-style-type: none"> ▶ 1.13 for CUDA 12.8 ▶ 1.9 for CUDA 12.4 ▶ 1.7 for CUDA 12.2 	
NVIDIA Toolkit	Container	1.17.5	NVIDIA Container Toolkit includes the following packages: <ul style="list-style-type: none"> ▶ libnvidia-container-tools: 1.17.5 ▶ libnvidia-container1: 1.17.5 ▶ nvidia-container-toolkit: 1.17.5
NVSM		24.09.05	
Docker Engine		27.5.1	
GDRCopy		2.4.4	
MIG Configuration Tool		0.11.0	Refer to NVIDIA mig-parted github pages and deployments .
DLFW		25.02	
ISO		DGXOS-7.0.2-2025-03-18-11-47-30.iso	
MD5 Checksum		80966b400d5815be14e8931bdb7f06a	

Chapter 4. Initial Setup

This topic describes the setup process when the DGX system is powered on for the first time after delivery or after the system is reimaged.

To start the process, you need to accept the End User License Agreements (EULA) and set up your username and password. To preview the EULA, visit the [Enterprise Software Agreements](#) page and click [NVIDIA Software License Agreement](#) under **NVIDIA AI, DGX and NVIDIA AI Enterprise**.

4.1. Connecting to the DGX System

During the installation and initial configuration steps, you need to connect to the console of the DGX system. To connect to the DGX system, use one of the following methods:

- ▶ Through a virtual keyboard, video, and mouse (KVM) in the BMC.
- ▶ A direct connection with a local monitor and keyboard.

Refer to the appropriate DGX product user guide for a list of supported connection methods and specific product instructions:

- ▶ [DGX B300 System User Guide](#)
- ▶ [DGX B200 System User Guide](#)
- ▶ [DGX H100/H200 System User Guide](#)
- ▶ [DGX A100 System User Guide](#)
- ▶ [DGX Station A100 User Guide](#)
- ▶ [DGX GB200 System User Guide](#)

4.2. First Boot Setup Wizard

Here are the steps to complete the first boot process. It differs between the DGX systems:

- ▶ [First Boot Process for DGX Servers](#)
- ▶ [First Boot Process for DGX Station](#)

4.2.1. First Boot Process for DGX Servers

Here are the steps to complete the first boot process for DGX servers.

Note

The first boot dialogs are presented over a display. This can be a physical display or the virtual one provided by the remote BMC. These dialogs will not be visible using a serial console.

1. If the DGX OS was installed with an encrypted root filesystem, you will be prompted to unlock the drive. For more information, refer to [Advanced Installation Options \(Encrypted Root\)](#). When you select this menu item, you can encrypt the root filesystem of the DGX system.

Enter `nvidia3d` at the crypt prompt.

2. Accept the EULA to proceed with the DGX system setup.
3. Complete the following steps:
 - a. Select your language and locale preferences.
 - b. Select the country for your keyboard.
 - c. Select your time zone.
 - d. Confirm the UTC clock setting.
 - e. Create an administrative user account with your name, username, and password.
 - ▶ This username is also used as the BMC and GRUB username.
 - ▶ The BMC software does not accept `sysadmin` for a username, and you cannot log in to the BMC with that username.

Note

During this step in the procedure, the default BMC administrator user will be disabled, and a new BMC administrator user will be created.

- ▶ The username must be composed of lower-case letters.
- ▶ The username will be used for administrative activities instead of the root account.
- ▶ Ensure you enter a strong password.

If the password that you entered is weak, a warning message appears.

- f. Create a BMC administrator password. The allowed character length for the BMC password depends on the specific DGX product:
 - ▶ DGX A100: 13 - 20 characters
 - ▶ DGX Station A100: 13 - 20 characters
 - ▶ DGX H100/H200: 13 - 20 characters
 - ▶ DGX B200: 13 - 20 characters
 - ▶ DGX B300: 13 - 20 characters
 - ▶ DGX GB200: 13 - 20 characters

- ▶ DGX GB300: 13 - 20 characters

After you create your login credentials, the default credentials will no longer work.

g. Create a GRUB password.

- ▶ Your GRUB password must have at least 8 characters. If it has less than 8 characters, you cannot click **Continue**.
- ▶ If you continue without entering a password, the GRUB protection will be disabled. For added security, NVIDIA recommends that you set the GRUB password.

h. Create a root filesystem passphrase. This dialog only appears if root filesystem encryption was selected at the time of the DGX OS installation. For more information, refer to [Advanced Installation Options \(Encrypted Root\)](#). When you select this menu item, you can encrypt the root filesystem of the DGX system.

i. Select a primary network interface for the DGX system. This should typically be the interface that you will use for subsequent system configuration or in-band management. For example:

- ▶ DGX A100: enp226s0
- ▶ DGX H100/H200: eno3
- ▶ DGX B200: eno3
- ▶ DGX B300: eno1
- ▶ DGX GB200: enP5p9s0
- ▶ DGX GB300: enP5p9s0

Do not select `enp37s0f3u1u3c2`, `bmc_redfish0`, or similar names because this interface is intended only for out-of-band management or future support of in-band tools that will access the Redfish APIs.

After you select the primary network interface, the system attempts to configure the interface for DHCP and prompts you to enter the name server addresses.

- ▶ If no DHCP is available, click **OK** in the Network autoconfiguration failed dialog box and manually configure the network.
- ▶ To configure a static address, click **Cancel** in the dialog box after the DHCP configuration completes to restart the network configuration steps.
- ▶ To select a different network interface, after the DHCP configuration completes, click **Cancel** in the dialog box to restart the network configuration steps.

j. If prompted, enter the requested networking information, such as the name server or the domain name.

k. Select a host name for the DGX system.

After you complete the first boot process, the DGX system configures the operating system, starts the system services, and displays a login prompt on the console. If the IP of the configured network interface is known, you can log in by using the console or secure shell (SSH).

Caution

Before issuing a reboot, ensure the NVIDIA RAID configuration service has completed by issuing `sudo systemctl status nvidia-raid-config` until the Finished NVIDIA RAID Configuration message appears. The configuration service typically takes about 10 minutes to complete.

4.2.2. First Boot Process for DGX Station

Important

You can connect to the system through a remote BMC console using a virtual keyboard, video, and mouse. To use the remote BMC console, ensure that the OnBrd/Ext VGA Select SBIOS configuration is set to OnBoard. If the SBIOS is not configured this way, the system starts the user interface on the display port instead of the remote BMC console. Refer to [Using DGX Station A100 as a Server Without a Monitor](#) for details on changing the SBIOS configuration.

When you power on your DGX Station or DGX Spark for the first time, you are prompted to accept end user license agreements for NVIDIA software. You are then guided through the process to complete the initial Ubuntu OS configuration.

During the configuration process, to prevent unauthorized users from using non-default boot entries and modifying boot parameters, you need to enter a GRUB password.

1. Accept the EULA and click **Continue**.
2. Select your language, for example, English – English, and click **Continue**.
3. Select your keyboard, for example, English (US) and click **Continue**.
4. Select your location, for example, Los Angeles, and click **Continue**.
5. Enter your username and password, enter the password again to confirm it, and click **Continue**.

Here are some requirements to remember:

- ▶ The username must be composed of lower-case letters.
- ▶ The username will be used instead of the root account for administrative activities.
- ▶ It is also used as the GRUB username.
- ▶ Ensure you enter a strong password.

If the password that you entered is weak, a warning appears.

6. Enter the GRUB password and click **OK**.
 - ▶ Your GRUB password must have at least 8 characters.
If it has less than 8 characters, you cannot click **Continue**.
 - ▶ If you do not enter a password, GRUB password protection will be disabled.
7. If you perform the automated encryption install, you will be prompted to create a new passphrase for your root filesystem.
 - ▶ The default password was seeded with `nvidia3d` which will be disabled after you complete this step.
 - ▶ This new passphrase will unlock your root filesystem when the system boots.

⚠ Caution

Following a reboot, ensure that the NVIDIA RAID configuration service has completed by issuing `sudo systemctl status nvidia-raid-config` until the Finished NVIDIA RAID Configuration message appears. The configuration service typically takes about 10 minutes to complete.

4.3. Post-Installation Tasks

After installing the DGX system, you can proceed with the following tasks.

4.3.1. Performing Package Updates for DGX Servers

After the DGX Server (A100, A800, H100, H200, H800, or B200) has been installed, the packages in the DGX OS ISO 7.0 image used on the system must be updated immediately.

The NVIDIA DGX™ B200 Systems ship from the factory with DGX OS 7.0.0 already installed. In most cases, this should be the starting point for performing DGX OS 7.0.x update with the general availability version of drivers.

i Note

To perform DGX OS updates on systems without an internet connection, refer to [Air-Gapped Installations](#).

Follow these steps for the DGX OS updates:

1. *Add the DOCA Driver to the DGX Server.*
2. *Update Remaining Drivers and Packages.*
3. *Verify the DGX OS Update.*

4.3.1.1 Add the DOCA Driver to the DGX Server

The update from DGX OS 7.0.0 to 7.0.x uses an updated location for the DOCA repository. Add this repository and update the DOCA repository following these steps:

```
$ sudo apt update
$ sudo apt install -y doca-repo
$ sudo apt update
```

4.3.1.2 Update Remaining Drivers and Packages

After the doca-repo installation, install the updates needed for other packages in DGX OS by following the standard DGX OS update step:

1. Upgrade all software packages with the latest versions.

```
$ sudo apt full-upgrade -y
```

2. Reboot the system.

```
$ sudo reboot
```

Note

If an error about the `nvidia_peermem` version mismatch occurs, refer to [The `nvidia_peermem` Module Does Not Load After an OTA Update](#) to resolve the issue.

4.3.1.3 Verify the DGX OS Update

After the update, confirm if the update is completed by reviewing the `/etc/dgx-release` file.

The file should indicate that DGX OS 7.0.0 was initially used with two OTA lines at the end containing the DGX OS 7.0.x version and installation date. For example,

```
$ cat /etc/dgx-release
DGX_NAME="DGX Server"
DGX_PRETTY_NAME="NVIDIA DGX Server"
DGX_SWBUILD_DATE="2025-01-13-10-40-54"
DGX_SWBUILD_VERSION="7.0.0"
DGX_COMMIT_ID="a610e60"
DGX_PLATFORM="DGX Server for DGX B200"
DGX_SERIAL_NUMBER="1663724000002"

DGX_OTA_VERSION="7.0.2"
DGX_OTA_DATE="Wed Mar 19 09:16:58 AM PST 2025"
```

4.3.2. Performing Package Updates for DGX Stations

After the DGX Station (Station A100 or Station A800) has been installed, the packages in the DGX OS ISO 7.0 image used on the system must be updated immediately.

```
$ sudo apt update
$ sudo apt full-upgrade
$ sudo reboot
```

For more information, refer to [Performing Package Upgrades](#).

4.3.3. Adding Support for Additional Languages to the DGX Station

During the initial Ubuntu OS configuration, you are prompted to select the default language on the *DGX Station*. If the language that you select is in the DGX OS 7 software image, it is installed in addition to English, and you will see that language after you log in to access your desktop. If the language that you select is not included, you will still see English after logging in, and you will need to install the language separately.

The following languages are included in the DGX OS 7 software image:

- English

- ▶ Chinese (Simplified)
- ▶ French
- ▶ German
- ▶ Italian
- ▶ Portuguese
- ▶ Spanish

For information about how to install languages, refer to [Install languages](#).

4.3.4. Configuring Your DGX Station

The DGX Display Adaptor card provides DGX OS with multiple display outputs, which allow you to connect multiple monitors to the DGX Station A100 or DGX Spark. If you plan to use more than one display, configure the DGX Station A100 or DGX Spark to use multiple displays after you complete the initial DGX OS configuration. See [First Boot Process for DGX Station](#).

When you power on your DGX station or DGX Spark for the first time, you are prompted to accept end user license agreements for NVIDIA software. You are then guided through the process to complete the initial Ubuntu OS configuration.

1. Connect the displays that you want to use to the mini DisplayPort (DP) connectors at the back of the unit.

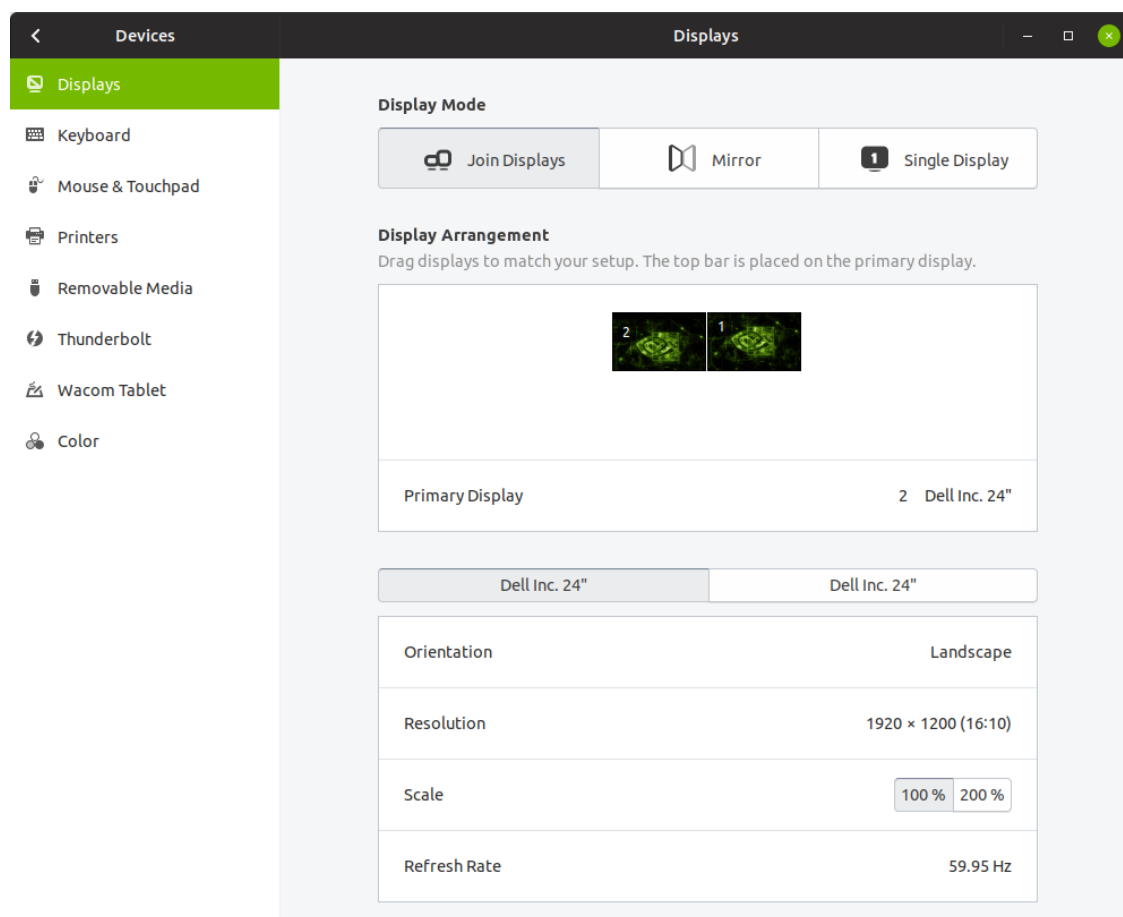
Note

DGX Station A100 also supplies two mini DP to DP adapters if your monitors do not natively support mini DP input.

Each display is automatically detected when you connect it.



2. (Optional) If necessary, adjust the display configuration, such as switching the primary display or changing monitor positions or orientation.
 - a. Open the Displays window.
 - b. In the Displays window, update the necessary display settings and click **Apply**.



4.3.5. Enabling Multiple Users to Remotely Access the DGX System

To enable multiple users to remotely access the DGX system, an SSH server is installed and enabled on the DGX system.

Add other Ubuntu OS users to the DGX system to allow them to remotely log in to the DGX system through SSH. Refer to [Add a new user account](#) for more information.

For information about logging in remotely through SSH, see [Connecting to an OpenSSH Server](#) on the Ubuntu Community Help Wiki.

Important

The DGX system does not provide any additional isolation guarantees between users beyond the guarantees that the Ubuntu OS offers. For guidelines about secure access to the DGX system over SSH, see [Configuring an OpenSSH Server](#) on the Ubuntu Community Help Wiki.

Chapter 5. Reimaging the System

This section provides information about installing the DGX OS by reimaging the system from the DGX OS ISO image.

DGX OS is already preinstalled on new DGX systems and only requires reimaging in limited cases. If your system is already running DGX OS 7, you can skip to [Initial Setup](#) for instructions about the initial setup of the system. To upgrade a system from DGX OS 5 or DGX OS 6, refer to [Upgrading the OS](#).

You also have the option to install Ubuntu and the DGX software manually, for example, if you require custom installation options, such as a specific drive partition scheme. For more information, refer to [Customizing Ubuntu Installation with DGX Software](#). It also describes automating the installation process, such as cluster deployments.

The following situations require you to reimage a DGX system:

- ▶ Install the latest version on a new system.
- ▶ Install an older version.
- ▶ The OS becomes corrupted.
- ▶ The OS drive is replaced or both drives in a RAID-1 configuration are replaced.
- ▶ Encrypt the root filesystem.
- ▶ Revert the DGX system to the originally installed DGX OS.

Caution

Reimaging the system erases all data stored on the OS drives. This includes the `/home` partition, where all users' documents, software settings, and other personal files are stored. If you need to preserve data through the reimaging, you can move the files and documents to the `/raid` directory and install the DGX OS software with the option to preserve the RAID array content.

The reimage process does not change persistent hardware configurations such as MIG settings or data drive encryption.

Important

After completing the installation, refer to [Upgrading the OS](#) to perform a package upgrade to the latest available software versions available since the DGX OS ISO release, including security updates.

5.1. Obtaining the DGX OS ISO Image

Note

Before you begin, ensure that you have an active NVIDIA Enterprise Support account.

To ensure that you install the latest available version of DGX OS, obtain the latest ISO image file from **NVIDIA Enterprise Support**:

1. Go to the [Download Center](#).
2. Choose **Server/Workstation** -> **DGX** and select **All Downloads** for your system.
3. Click the download link for the latest ISO release to go to the announcement.
4. Download the ISO image that is referenced in the announcement and save it to your local disk.
5. Run the `md5sum` command to print the MD5 hash and compare it with the value in the announcement. For example:

```
$ md5sum DGXOS-7.3.1-2025-11-12-02-34-35-amd64.iso
```

Example output:

```
7e9aa6b3995d48d80bbe5c7e75d17f74 DGXOS-7.3.1-2025-11-12-02-34-35-amd64.  
→iso
```

5.2. Installing the DGX OS Image

Install the DGX OS ISO image in one of the following ways:

- ▶ Remotely through the BMC for systems that provide a BMC
For instructions, refer to [Installing the DGX OS Image Remotely Through the BMC](#).
- ▶ Locally from a UEFI-bootable USB flash drive or DVD-ROM
For instructions, refer to [Installing the DGX OS Image from a USB Flash Drive or DVD-ROM](#). After obtaining the DGX OS 7 ISO image from NVIDIA Enterprise Support, create a bootable installation medium, such as a USB flash drive or DVD-ROM that contains the image.

5.2.1. Installing the DGX OS Image Remotely Through the BMC

These instructions describe how to re-image the system remotely through the BMC.

After obtaining the DGX OS 7 ISO image from NVIDIA Enterprise Support, ensure the host that you use for your web browser can access the ISO image file.

1. Log in to the BMC.
For more information, refer to [Connecting to the DGX System](#).
2. Select **Remote Control** and then click **Launch KVM**.

3. Set up the ISO image as virtual media.
 1. From the top bar, select **Browse File**, locate and select the DGX OS ISO file, and click **Open**.
 2. Click **Start Media**.
4. Reset the system and boot the virtual media image.
 1. From the top menu, choose **Power > Hard Reset**, and then click **Perform Action**.
 2. Click **Yes** and then **OK** at the Power Control dialogs.
Wait for the system to power down and then come back online.
 3. Refer to *DGX OS ISO Boot Options* for a description of the GRUB menu options and for instructions on completing the installation process.

5.2.2. Installing the DGX OS Image from a USB Flash Drive or DVD-ROM

After obtaining the DGX OS 7 ISO image from NVIDIA Enterprise Support, create a bootable installation medium, such as a USB flash drive or DVD-ROM, that contains the image.

- ▶ To create a bootable USB flash drive, refer to one of the following links for more information:
 - ▶ On Linux, refer to *Creating a Bootable USB Flash Drive by Using the dd Command*.
 - ▶ On Windows, refer to *Creating a Bootable USB Flash Drive by Using Akeo Rufus*.
- ▶ To create a bootable DVD ROM, refer to *Burning the ISO on to a DVD-ROM* on the Ubuntu Community Help Wiki for more information about the available methods.

5.2.2.1 Creating a Bootable USB Flash Drive by Using the dd Command

On a Linux system, you can use the `dd` command to create a bootable USB flash drive that contains the DGX OS software image.

Note

To ensure that the resulting flash drive is bootable, use the `dd` command to perform a device bit copy of the image. If you use other commands to perform a simple file copy of the image, the resulting flash drive may not be bootable.

Ensure that the following prerequisites are met:

- ▶ The correct DGX OS software image is saved to your local disk.
For more information, refer to *Obtaining the DGX OS ISO Image*.
- ▶ The USB flash drive meets the following requirement:
 - ▶ The USB flash drive has a capacity of at least 16 GB.
 - ▶ The partition scheme on the USB flash drive is a GPT partition for UEFI.

Create the bootable USB Flash drive:

1. Plug the USB flash drive into one of the USB ports of your Linux host. Obtain the device name of the USB flash drive by running the `lsblk` command.

```
lsblk
```

You can identify the USB flash drive from its size, which is much smaller than the size of the SSDs in the DGX software, and from the mount points of any partitions on the drive, which are under `/media`.

In the following example output, the device name of the USB flash drive is `sde`.

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPPOINT
sda	8:0	0	1.8T	0	disk	
_sda1	8:1	0	121M	0	part	/boot/efi
_sda2	8:2	0	1.8T	0	part	/
sdb	8:16	0	1.8T	0	disk	
_sdb1	8:17	0	1.8T	0	part	
sdc	8:32	0	1.8T	0	disk	
sdd	8:48	0	1.8T	0	disk	
sde	8:64	1	7.6G	0	disk	
_sde1	8:65	1	7.6G	0	part	/media/deeplearner/DGXSTATION

2. As root, convert and copy the image to the USB flash drive.

```
sudo dd if=<path-to-ISO-image> bs=2048 of=<usb-drive-device-name>
```

Warning

The `dd` command erases all data on the device that you specify in the `of` argument. To avoid losing data, ensure that you specify the correct path to the USB flash drive.

5.2.2.2 Creating a Bootable USB Flash Drive by Using Akeo Rufus

On a Windows system, you can use the [Akeo Reliable USB Formatting Utility \(Rufus\)](#) to create a bootable USB flash drive that contains the DGX OS software image.

Ensure that the following prerequisites are met:

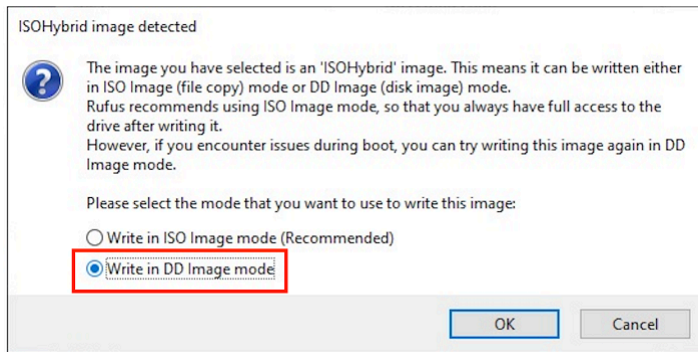
- The correct DGX OS software image is saved to your local disk.
For more information, refer to [Obtaining the DGX OS ISO Image](#).
- The USB flash drive has a capacity of at least 16 GB.

Follow these steps to create the bootable USB Flash drive:

1. Plug the USB flash drive into one of the USB ports of your Windows system.
2. Download and launch the [Akeo Reliable USB Formatting Utility \(Rufus\)](#).
3. In **Drive Properties**, select the following options:
 1. In **Device**, select your USB flash drive.
 2. In **Boot selection**, click **SELECT**, locate, and select the DGX OS software image.

You can leave the other settings at the default.

4. Click **Start**. This step prompts you to select whether to write the image in ISO Image mode (file copy) or DD Image mode (disk image).



5. Select **Write in DD Image mode** and click **OK**.

5.2.3. Booting the DGX OS ISO image

These instructions describe how to boot the DGX OS ISO image locally.

1. Plug the USB flash drive containing the OS image into the DGX system.
2. Connect a monitor and keyboard directly to the DGX system.
3. Boot the system and then press **F11** when the NVIDIA logo appears to access the boot menu.
4. Select the USB volume name that corresponds to the inserted USB flash drive and boot the system from it.

Refer to *DGX OS ISO Boot Options* for a description of the GRUB menu options and for information about completing the installation process.

5.3. DGX OS ISO Boot Options

This section provides information about the available installation and boot options of the DGX OS ISO installer.

These instructions assume that you have booted the DGX OS ISO, either remotely through the BMC or locally from a USB flash drive.

- ▶ When the system boots up, select one of the following options from the GRUB menu:
 - ▶ Install DGX OS <version>
 - ▶ Install DGX OS <version>: Without Reformatting Data RAID (does not mount /raid)
 - ▶ Advanced Installation Options - Install DGX OS <version> Without NVIDIA Drivers - Install DGX OS <version> With Encrypted Root - Install DGX OS <version> With Encrypted Root and Without Reformatting Data RAID
 - ▶ Boot Into Live Environment
 - ▶ Check Media for Defects

See the following sections for more information about these options.

- ▶ Verify that the DGX system booted up and that the image is being installed.

This process will iterate through the software components and copy and install them showing the executed commands. This process generally takes between 15 and 60 minutes, depending

on DGX platform, and how the system is being imaged (for example, BMC over a slow network or locally with a fast USB flash drive).

Note

On DGX servers, the NVIDIA InfiniBand driver is installed and the firmware on the ConnectX cards is updated. This process can take up to 5 minutes for each card. Other system firmware is not updated.

After the installation is complete, the system reboots into the OS, and prompts for configuration information. Refer to [Initial Setup](#) for more information about how to boot up the DGX system for the first time after reimaging the system.

5.3.1. Install DGX OS with Reformatting the Data RAID

Here are the steps to install your DGX system and reformat the data RAID.

When you accept this option, the installation process repartitions all drives, including the OS and the data drives. The data drives are configured as a RAID array and mounted under the `/raid` directory. This process overwrites all the data and file systems that might exist on the OS and data drives. The RAID array on the DGX data disks is intended to be used as a cache and not for long-term data storage, so reformatting the data RAID should not be disruptive.

These changes are preserved across system reboots.

5.3.2. Install DGX OS Without Reformatting the Data RAID

Here are the steps to install your DGX system without reformatting the data RAID.

The RAID array on the DGX data disks is intended for use as a cache and not for long-term data storage, so this should not be disruptive. However, if you are an advanced user and have set up the disks for a non-cache purpose and want to keep the data on those drives, select **Install DGX system without Reformatting the Data RAID** option at the boot menu during the boot installation. This option retains data on the RAID disks, and the following tasks are completed:

- ▶ Installs the cache daemon but leaves it disabled by commenting out the `RUN=yes` line in `/etc/default/cachefilesd`
- ▶ Creates a `/raid` directory, leaves it out of the file system table by commenting out the entry containing `/raid` in `/etc/fstab`
- ▶ Does not format the RAID disks.

When the installation is completed, you can repeat any configuration steps that you had performed to use the RAID disks as other than cache disks. You can always choose to use the RAID disks as cache disks later by enabling `cachefilesd` and adding `/raid` to the file system table:

1. Uncomment the `#RUN=yes` line in `/etc/default/cachefilesd`.
2. To mount a block device or RAID array at `/raid`, uncomment the `/raid` line in `/etc/fstab`.
Ensure to replace `<device>` with the proper value for your environment.
3. Run the following:
 1. Mount `/raid`.

```
sudo mount /raid
```

2. Reload the systemd manager configuration.

```
systemctl daemon-reload
```

3. Start the cache daemon.

```
systemctl start cachefilesd
```

These changes are preserved across system reboots.

5.3.3. Advanced Installation Options (Encrypted Root)

When you select this menu item, you can encrypt the root filesystem of the DGX system.

Note

Select this option only if you want to encrypt the root filesystem.

Aside from the encrypted root filesystem, the behavior is identical to the default installation.

Selecting **Encrypted Root** instructs the installer to encrypt the root filesystem. The encryption is fully automated, and you will be required to manually unlock the root partition by entering a passphrase at the console (through a direct keyboard and mouse connection or through the BMC) each time the system boots.

When you power on your DGX system for the first time, you are prompted to accept end user license agreements for NVIDIA software. You are then guided through the process to complete the initial Ubuntu OS configuration, you can create your passphrase for the drive. If necessary, you can change this passphrase later. For more details see [First Boot Process for DGX Servers](#) or [First Boot Process for DGX Station](#).

Note

Encryption cannot be enabled or disabled after the installation. To change the encryption state again, you need to reimage the drives.

5.3.4. Boot Into a Live Environment

The DGX OS installer image can also be used as a Live image, which means that the image boots up and runs a minimal DGX OS in system memory and does not overwrite anything on the disks in the system.

Live mode does not load drivers and is essentially a simple Ubuntu Server configuration. This mode can be used as a tool to debug a system when the disks on the system are not accessible or should not be touched.

In a typical operation, this option should not be selected.

5.3.5. Check Disc for Defects

Here is some information about how you can check the disc for defects.

If you are experiencing anomalies when you install the DGX OS and suspect the installation media might have an issue, select this item to complete an extensive test of the installation media contents.

The process is time-consuming, and the installation media is usually not the source of the problem. In a typical operation, this option should not be selected.

Chapter 6. Customizing Ubuntu Installation with DGX Software

This section explains the steps for installing and configuring Ubuntu and the NVIDIA DGX Software Stack on DGX systems.

DGX OS provides a customized Ubuntu installation with additional software from NVIDIA to provide a turnkey solution for running AI and analytics workloads. The additional software, the NVIDIA DGX Software Stack, comprises platform-specific configurations, diagnostic and monitoring tools, and drivers that are required for a stable, tested, and supported OS to run AI, machine learning, and analytics applications on DGX systems.

You also have the option to install the NVIDIA DGX Software Stack on top of a vanilla Ubuntu distribution while still benefiting from the advanced DGX features. This installation method supports more flexibility, such as custom partition schemes.

Cluster deployments also benefit from this installation method through Ubuntu's standardized, automated, and non-interactive installation process. Starting with Ubuntu 20.04, the installer introduced a new mechanism for automating the installation, allowing system administrators to install a system unattended and non-interactively. You can find information for creating such a `cloud-init` configuration file in [Cloud-init Configuration File](#). For more information, refer to [Ubuntu Automated Server Installation](#).

The intended audience is IT professionals managing a cluster of DGX systems and integration partners.

6.1. Prerequisites

The following prerequisites are required or recommended, where indicated.

6.1.1. Ubuntu Software Requirements

The DGX Software Stack requires the following software versions:

- ▶ Ubuntu 24.04
- ▶ Linux Kernel 6.8

6.1.2. Access to Software Repositories

The DGX Software Stack is available from repositories that can be accessed from the internet. If your installation does not allow connection to the internet, see [Air-Gapped Installations](#) for information about installing and upgrading software on “air-gapped” systems.

If you are using a proxy server, follow the instructions in the section [Network Configuration](#) for setting up a proxy configuration.

6.2. Installation Considerations

Installing the NVIDIA DGX Software Stack on Ubuntu allows you to select from additional configuration options that would otherwise not be available with the preconfigured DGX OS installer. This includes drive partitioning, filesystem choices, and software selection.

Before you start installing Ubuntu and the NVIDIA DGX Software Stack, you should evaluate the following options. The installation and configuration instructions will be covered in the respective section of this document.

6.2.1. System Drive Mirroring (RAID-1) [recommended]

The DGX B300, DGX B200, DGX H100/H200, DGX A100, DGX GB300, and DGX GB200 systems embed two system drives for mirroring the OS partitions (RAID-1). This ensures data resiliency if one drive fails. To enable mirroring, you need to enable it during the drive configuration of the Ubuntu installation. It cannot be enabled after the installation.

6.2.2. Data Drive RAID-0 or RAID-5

DGX systems are equipped with multiple data drives that can be configured as RAID-0 for performance or RAID-5 for resiliency. RAID-0 provides the maximum storage capacity and performance but does not provide any redundancy. If a single SSD in the array fails, all data stored on the array is lost.

RAID-0 is recommended for data caching. You can use `cachefilesd` to provide caching for NFS shares. The network file system (NFS) is required to take advantage of the cache file system. RAID-5 should be used for persistent data storage.

You have the option to configure RAID and data caching after the initial Ubuntu installation using the `nvidia-config-raid` tool or during the Ubuntu installation. The `nvidia-config-raid` tool is recommended for manual installation.

6.2.3. System Drive Encryption [optional]

Root filesystem encryption is a software-based method to protect the content stored in the system partition(s) from unauthorized access by encrypting the data in real time. It requires users to unlock the filesystem on every boot, either manually by entering a passphrase or automatically using a centralized key server.

System drive encryption can only be enabled during the installation of Ubuntu.

6.2.4. Data Drive Encryption [optional]

Data drive encryption is only supported on DGX B300, DGX B200, DGX H100/H200, DGX A100, DGX GB300, and DGX GB200 systems that are equipped with self-encrypting drives (SED). It can be enabled after Ubuntu is installed using the `nv-disk-encrypt` tool. It requires either storing the keys in the TPM or external key-management.

6.2.5. System Drive Partitioning

Ubuntu uses only a single partition for the entire filesystem by default. This can be configured during the Ubuntu installation for deployments that require a more faceted partition scheme for security reasons. The recommended partitioning scheme is to use only a single partition for the Linux root partition with the ext4 filesystem.

6.3. Installing Ubuntu

There are several methods for installing Ubuntu as described in the [Ubuntu Server Guide](#).

For convenience, this section provides additional instructions that are specific to DGX for installing Ubuntu following the [Basic Installation](#). If you have a preferred method for installing Ubuntu, skip this section.

Steps that are covered in this section:

- ▶ Connecting to the DGX system
- ▶ Booting from the installation media
- ▶ Running the Ubuntu installer (including network and storage configuration steps)

6.3.1. Booting from the Installation Media

During the installation and configuration steps, you need to connect to the console of the DGX system. Refer to [Connecting to the DGX System](#) for more details.

Boot the Ubuntu ISO image in one of the following ways:

- ▶ Remotely through the BMC for systems that provide a BMC
For instructions, refer to [Installing the DGX OS Image Remotely Through the BMC](#).
- ▶ Locally from a UEFI-bootable USB flash drive or DVD-ROM
For instructions, refer to [Installing the DGX OS Image from a USB Flash Drive or DVD-ROM](#).

6.3.2. Running the Ubuntu Installer

After booting the ISO image, the Ubuntu installer should start and guide you through the installation process.

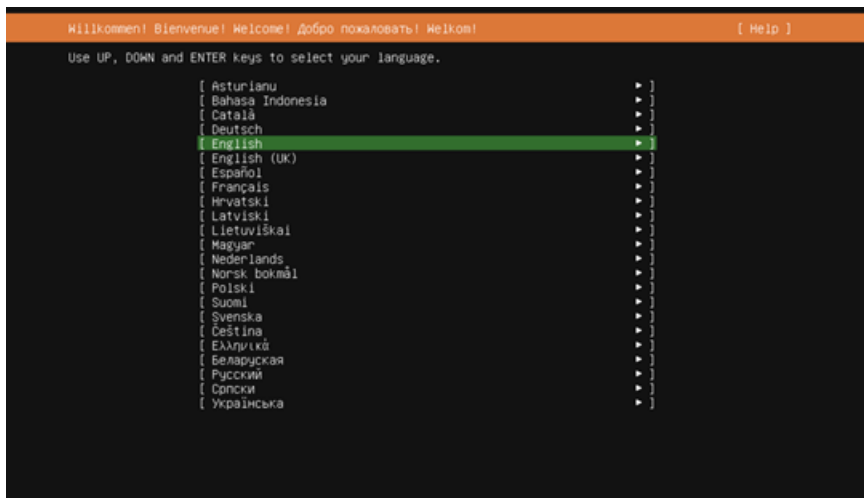
Note

The screenshots in the following steps are taken from a DGX A100. Other DGX systems have differences in drive partitioning and networking.

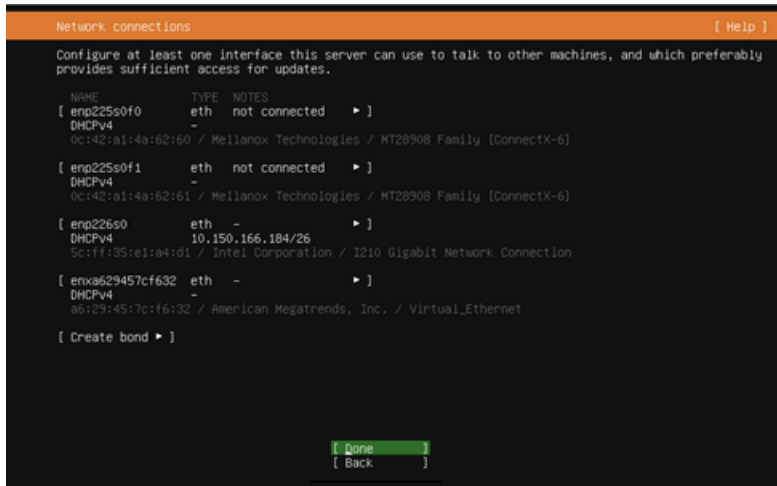
During the boot process of the ISO image, you might encounter some error messages due to older drivers, but they can be safely disregarded.

```
[ 12.951827] nouveau 0000:07:00.0: unknown chipset (170000a1)
[ 12.974418] nouveau 0000:0f:00.0: unknown chipset (170000a1)
[ 12.974927] nouveau 0000:47:00.0: unknown chipset (170000a1)
[ 12.975132] nouveau 0000:4e:00.0: unknown chipset (170000a1)
[ 12.975405] nouveau 0000:87:00.0: unknown chipset (170000a1)
[ 12.977712] nouveau 0000:90:00.0: unknown chipset (170000a1)
[ 12.978439] nouveau 0000:b7:00.0: unknown chipset (170000a1)
[ 12.978756] nouveau 0000:bd:00.0: unknown chipset (170000a1)
stdin: Invalid argument
stdin: Invalid argument
.
Checking integrity, this may take some time
```

1. Select your language at the welcome screen, and then follow the instructions to select whether to update the installer (if offered) and to choose your keyboard.



2. At the Network connections screen, configure your network.



The installer tries to automatically retrieve a DHCP address for all network interfaces, so you should be able to continue without any changes. However, you also have the option to manually configure the interface(s).

3. At the **Guided storage configuration** screen, configure the partitioning and file systems. All DGX systems are shipped preinstalled with DGX OS. The drives are, therefore, already partitioned and formatted. The DGX OS installer configures a single ext4 partition for the root partition in addition to the EFI boot partition. You have the following options:

- ▶ Keep the current partition layout and formatting [recommended].
- ▶ Create a custom partition scheme [advanced].
- ▶ Use a single disk with the default Ubuntu partition scheme.

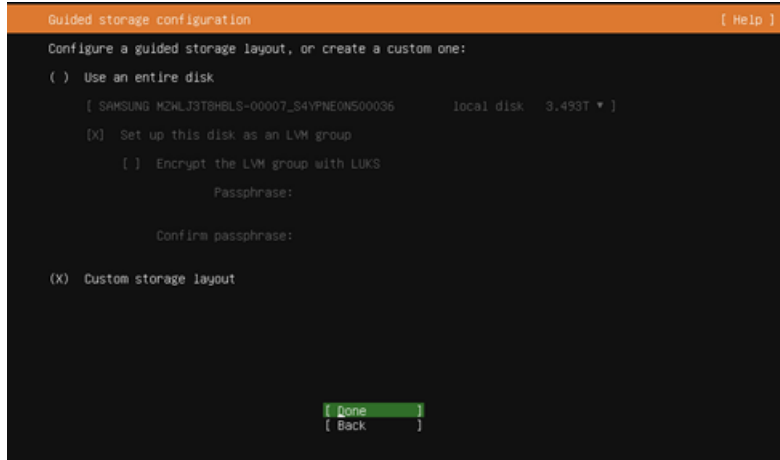
Creating a new custom partition scheme with a RAID configuration is a more involved process and is out of the scope of this document. Refer to the Ubuntu installation guide for more information. When you choose the option to use an entire disk, Ubuntu will only use one of the two redundant boot drives.

Note

The RAID level for the data drive can be changed after the installation of Ubuntu.

The following instructions describe the steps for keeping the current partition layout. It still requires you to re-create and reformat the partitions.

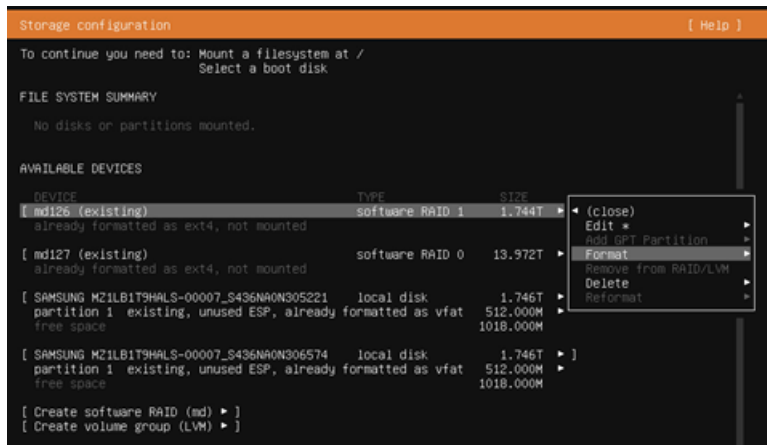
- a. Select Custom storage layout, and then click **Done**.



- b. Identify the system drive.

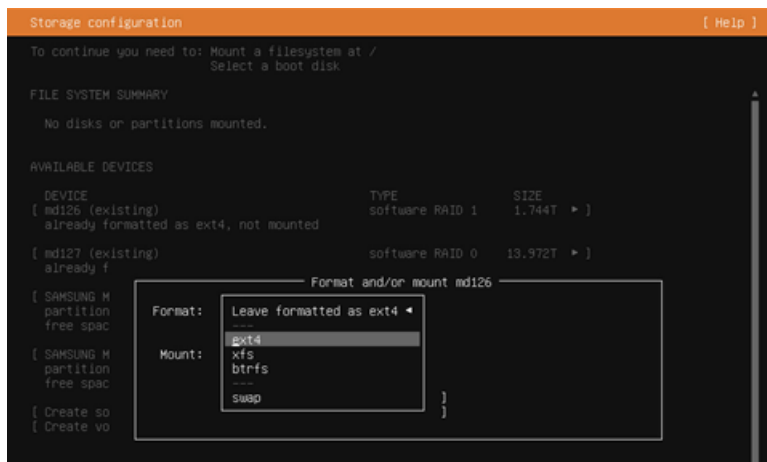
The system drive on the DGX A100, DGX H100/H200, DGX B200, DGX B300, DGX GB200, and DGX GB300 is a RAID 1 array and you should find it easily.

- c. Select the system drive, and then click **Format**.

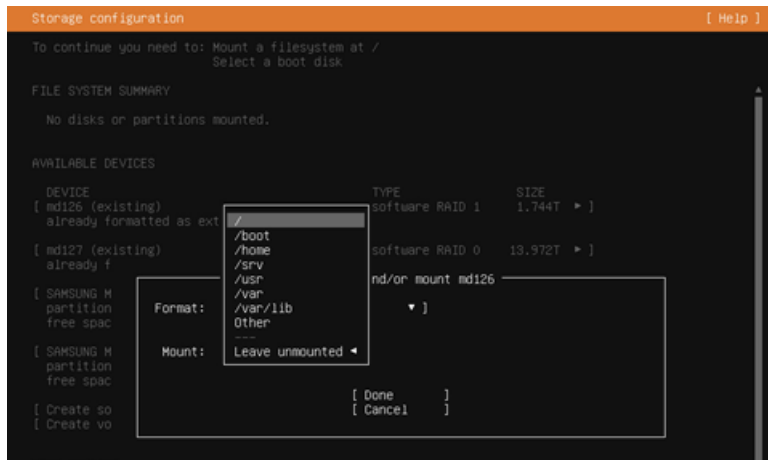


- d. Set **Format** to ext4.

Do not select Leave formatted as <filesystem>.

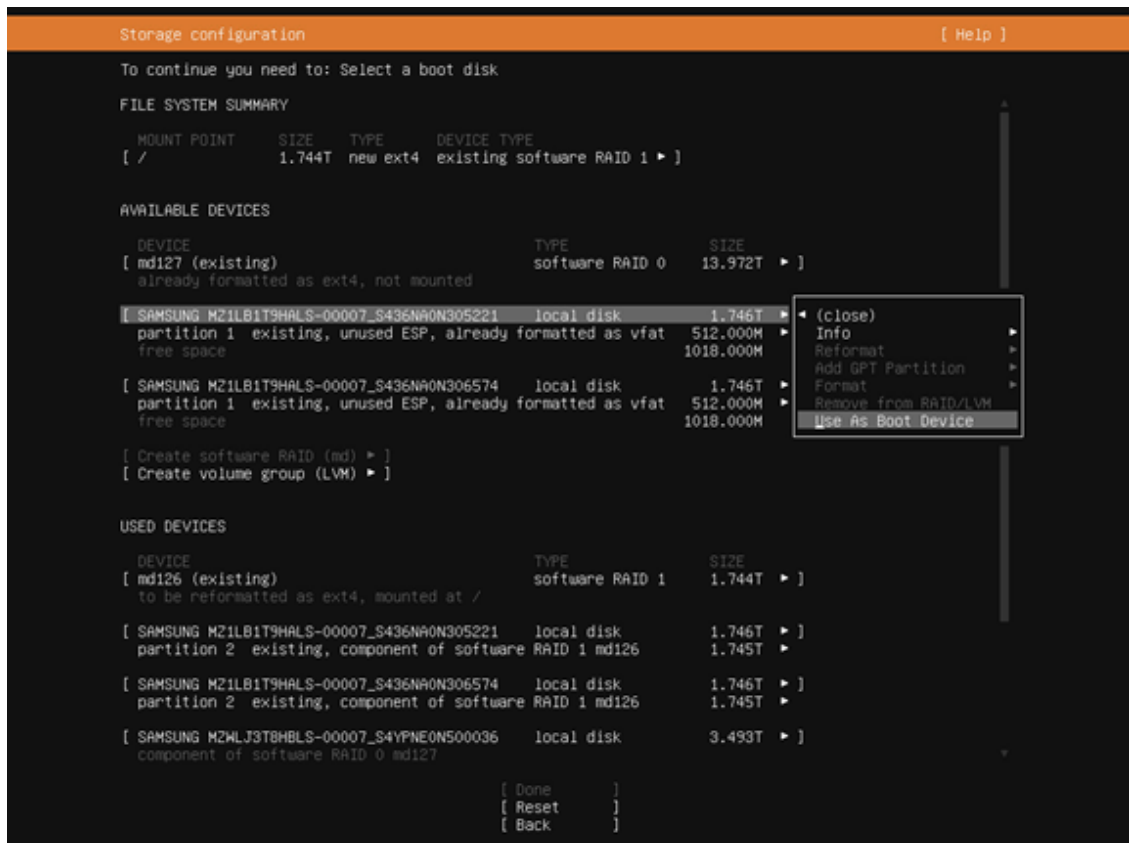


- e. Set **Mount** to **/**.



- f. Set the boot flag on the raw devices.

Identify the system drives under **AVAILABLE DEVICES** (not the RAID array) and select **Use as Boot Device** for the first device. On DGX A100, DGX H100/H200, DGX B200, DGX B300, DGX GB200, and DGX GB300 that have two drives, repeat this process for the second drive and select **Use as Another Boot Device**.



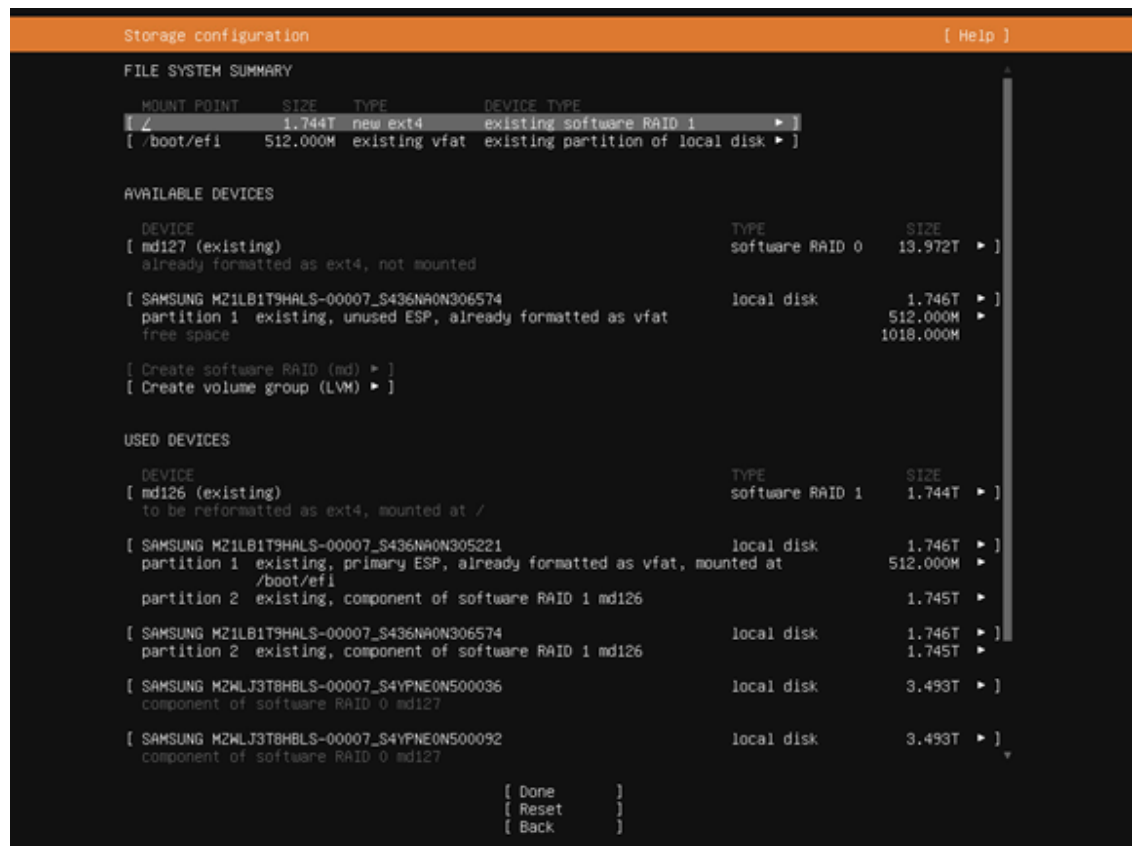
- g. Complete the configuration.

- ▶ **RAID 0 array:** In most cases, the RAID 0 array for the data drives will have already been created from the factory. If it has not been created, you can either create the array in the **Storage Configurations** dialog or by using the `config_raid_array` tool after

completing the Ubuntu installation.

- **Enable drive encryption (Optional):** Encryption can only be enabled during the Storage configuration; it cannot be changed after the installation. To change the encryption state again, you need to reinstall the OS. To enable drive encryption, you have to create a virtual group and volume. This falls outside the scope of this document. Refer to the Ubuntu documentation for more details.
- **Swap partition:** The default installation does not define a swap partition. Linux uses any configured swap partition for temporarily storing data when the system memory is full, incurring a performance hit. With the large memory of DGX systems swapping is not recommended.

The **FILE SYSTEM SUMMARY** at the top of the page should display the root partition on the RAID 1 drive for and a boot/efi partition (the two drives will only show up as a single entry).



Select **Done** and accept all changes.

- Follow the instructions for the remaining tasks.

Create a default user in the **Profile setup** dialog and choose any additional SNAP package you want to install in the **Featured Server Snaps** screen.

Note

Do not install Docker as a snap package. Because Docker CE is installed as part of the `nvidia-system-extra` metapackage, installing it separately as a snap might cause conflicts with the `nvidia-container-toolkit` package.

5. Wait for the installation to complete.

Log messages are presented while the installation is running.

6. Select **Reboot Now** when the installation is complete to restart the system.

After reboot, you can log in using the username and password for the user you have created above.

6.4. Installing the DGX Software Stack

This section requires that you have already installed Ubuntu on the DGX system and rebooted the system.

Attention

By installing the DGX Software Stack you are confirming that you have read and agree to be bound by the [NVIDIA Software License Agreement](#). You are also confirming that you understand that any pre-release software and materials available that you elect to install in a DGX may not be fully functional, may contain errors or design flaws, and may have reduced or different security, privacy, availability, and reliability standards relative to commercial versions of NVIDIA software and materials, and that you use pre-release versions at your risk.

6.4.1. Installing DGX System Configurations and Tools

The NVIDIA DGX Software Stack includes system-specific configurations and tools to leverage the advanced DGX features. They are provided from NVIDIA repositories in the form of software packages that can be installed on top of a typical Ubuntu installation. All system-specific software components are bundled into metapackages specific to a system:

- ▶ system-configurations
- ▶ system-tools
- ▶ system-tools-extra

For details about the content of these packages, refer to the [Release Notes](#).

The following steps enable the NVIDIA repositories and install the system-specific packages.

1. Enable the NVIDIA repositories by extracting the repository information.

This step adds the URIs and configuration preferences to control the package versions that will be installed to the `/etc/apt` directory and the GPG keys for the NVIDIA repositories in the `/usr/share/keyrings` directory.

- ▶ For x86_64 systems:

```
curl https://repo.download.nvidia.com/baseos/ubuntu/noble/x86_64/dgx-  
→repo-files.tgz | sudo tar xzf - -C /
```

- ▶ For ARM64 systems:

```
curl https://repo.download.nvidia.com/baseos/ubuntu/noble/arm64/dgx-  
→repo-files.tgz | sudo tar xzf - -C /
```

2. Update the internal APT database with the latest version information of all packages.

```
sudo apt update
```

3. **Recommended:** Upgrade all software packages to the latest versions.

```
sudo apt upgrade
```

4. Install the DGX system tools and configurations.

- ▶ Install Core DGX system packages required for performance.

```
sudo apt install nvidia-system-core
```

- ▶ Install DGX system utility packages, such as NVSM.

```
sudo apt install nvidia-system-utils
```

- ▶ Install DGX system packages, such as automake, build-essential, and vim for development.

```
sudo apt install nvidia-system-extra
```

- ▶ For DGX Station A100, DGX Station A800, and DGX Spark, install the nvidia-system-station package for a complete desktop, including gnome desktop, xorg, and other desktop packages.

```
sudo apt install nvidia-system-station
```

5. Install the linux-tools package for your kernel.

- ▶ For x86_64 systems:

```
sudo apt install -y linux-tools-generic
```

- ▶ For ARM64 systems like DGX GB200 and DGX GB300:

```
sudo apt install -y linux-tools-nvidia-64k
```

- ▶ For ARM64 systems like DGX Spark:

```
sudo apt install -y linux-tools-nvidia-hwe-24.04
```

6. Install the NVIDIA peermem loader package.

```
sudo apt install -y nvidia-peermem-loader
```

7. **Recommended:** Disable unattended upgrades.

Ubuntu periodically checks for security and other bug fixes and automatically installs updates software packages typically overnight. Because this may be disruptive, you should regularly check for updates and install them manually.

```
sudo apt purge -y unattended-upgrades
```

6.4.2. Configuring Data Drives

The data drives in the DGX systems can be configured as RAID 0 or RAID 5. RAID 0 provides the maximum storage capacity and performance but does not provide any redundancy.

RAID 0 is often used for data caching. You can use `cachefilesd` to provide a cache for NFS shares.

Important

You can change the RAID level later but this will destroy the data on those drives.

The RAID configuration can be configured during the Ubuntu installations. If you have already configured the RAID array during the Ubuntu installation, you can skip the first step and go to step 2.

1. Configure the `/raid` partition.

All DGX systems support RAID 0 and RAID 5 arrays.

- ▶ To create a RAID 0 array:

```
sudo /usr/bin/configure_raid_array.py -c -f
```

- ▶ To create a RAID 5 array:

```
sudo /usr/bin/configure_raid_array.py -c -f -5
```

The command creates the `/raid` mount point and RAID array and adds a corresponding entry in `/etc/fstab`.

2. (Optional) Install tools for managing the self-encrypting drives (SED) for the data drives on the DGX A100, DGX H100/H200, DGX B200, DGX B300, DGX GB200, or DGX GB300.

Refer to [Managing Self-Encrypting Drives](#) for more information.

3. (Optional) To use your RAID array for read caching of NFS mounts, you can install `cachefilesd` and set the `cacheofs` option for an NFS share.

- a. Install `cachefilesd` and `nvidia-conf-cachefilesd`.

This will update the `cachefilesd` configuration to use the `/raid` partition.

```
sudo apt install -y cachefilesd nvidia-conf-cachefilesd
```

Both `cachefilesd` and `nvidia-conf-cachefilesd` might already be installed with the `nvidia-system-utils` metapackage.

- b. Enable caching on all NFS shares you want to cache by setting the `fsc` flag.

Edit `/etc/fstab` and add the `fsc` flag to the mount options as shown in this example.

```
<nfs_server>:<export_path> /mnt nfs rw,noatime,rsz=32768,  
↪wsz=32768,nolock,tcp,intr,fsc,nofail 0 0
```

- c. Mount the NFS share.

If the share is already mounted, use the `remount` option.

```
mount <mount-point> -o,remount
```

- d. To validate that caching is enabled, issue the following.

```
cat /proc/fs/nfsfs/volumes
```

Look for the text FSC=yes in the output of the command. The NFS will be mounted with caching enabled upon subsequent reboot cycles.

6.4.3. Installing the GPU Driver

You can choose different GPU driver branches for your DGX system. The latest driver release includes new features but might not provide the same support duration as an older release. Consult the [Data Center Driver Release Notes](#) for more details and the minimum required driver release for the GPU architecture.

The DGX B300 and DGX B200 systems include the fifth generation of NVIDIA NVLink® and the NVLink Switch technology. With this version of NVlink, additional packages are included with Base OS 7 to enable the full NVLink functionality. These packages include `nvlsdm` and `libnvsdmm` among others. When performing GPU driver updates, it is required to update the driver and the corresponding NVLink stack packages simultaneously. Updating the DGX B300 and DGX B200 systems is described in the following steps of the NVIDIA open GPU kernel modules. When updating DGX B300 and DGX B200, you should also update the DOCA packages.

Use the following command to display a list of installed drivers.

1. Ensure to have the latest version of the package database.

```
sudo apt update
```

2. Display a list of all available drivers.

```
sudo apt list nvidia-driver*open
```

Example output:

```
nvidia-driver-570-open/unknown, now 570.195.03-0ubuntu1 amd64
```

The following steps install the NVIDIA CUDA driver and configure the system. Replace the release version used as an example (570) with the release you want to install. Ensure that the driver release you intend to install is supported by the GPU in the system.

3. Ensure you have the latest version of the package database.

```
sudo apt update
```

By default, the NVIDIA open GPU kernel modules and generic Linux kernel should be installed.

4. Ensure you have the latest kernel version installed.

The driver package has a dependency on the kernel, and updating the database might have updated the version information.

- For x86_64 systems:

```
sudo apt install -y linux-generic
```

- For ARM64 systems:

```
sudo apt install -y linux-nvidia-64k
```

5. Install the driver pinning package.

580 and newer releases have introduced significant packaging changes. When installing one of these versions, a new pinning package must be installed before installing the driver to ensure that installed versions of all related packages are consistent.

The pinning package is named `nvidia-driver-pinning-XXX` where XXX is the driver version. For example, before installing the 580 family driver, install `nvidia-driver-pinning-580`.

```
sudo apt install nvidia-driver-pinning-580
```

After the driver pinning package is installed proceed with the installation of the driver packages.

6. The NVIDIA open GPU kernel modules are supported and should be installed.

Note

- ▶ The DGX B300, DGX Spark, and DGX GB300 systems require the Release 580 family of the NVIDIA open GPU kernel modules.
- ▶ For the Release 580 family, the release branch has been removed from the names of the following packages:

Release 570

- ▶ `nvidia-fabricmanager-570`
- ▶ `libnvidia-nscq-570`
- ▶ `libnvdsdm-570`
- ▶ `nvidia-imex-570`

Release 580

- ▶ `nvidia-fabricmanager`
- ▶ `libnvidia-nscq`
- ▶ `libnvdsdm`
- ▶ `nvidia-imex`

- ▶ For NVSwitch systems with the fifth-generation NVLinks, such as DGX B300 and DGX B200: DGX B300:

```
sudo apt install nvidia-driver-580-open libnvidia-nscq nvidia-
↪modprobe nvidia-fabricmanager datacenter-gpu-manager-4-cuda13 nv-
↪persistence-mode nvlsdm libnvdsdm
```

The Release 580 GPU driver is optional for the DGX B200.

DGX B200:

```
sudo apt install nvidia-driver-570-open libnvidia-nscq-570 nvidia-
↪modprobe nvidia-fabricmanager-570 datacenter-gpu-manager-4-cuda12
↪nv-persistence-mode nvlsdm libnvdsdm-570
```

- ▶ For NVSwitch systems without the fifth-generation NVLinks, such as DGX A100 and DGX H100/H200:

```
sudo apt install nvidia-driver-570-open libnvidia-nscq-570 nvidia-
↪modprobe nvidia-fabricmanager-570 datacenter-gpu-manager-4-cuda12
↪nv-persistence-mode
```

- ▶ For non-NVSwitch systems, such as DGX Station A100, DGX Station A800, and DGX Spark, do not install the NVIDIA Fabric Manager service:

DGX Station A100 and A800:

```
sudo apt install nvidia-driver-570-open libnvidia-nscq-570 nvidia-
↪modprobe datacenter-gpu-manager-4-cuda12 nv-persistence-mode
```

DGX Spark:

```
sudo apt install nvidia-driver-580-open libnvidia-nscq nvidia-
↪modprobe datacenter-gpu-manager-4-cuda13 nv-persistence-mode
```

- For multinode NVLink systems, such as DGX GB200 and DGX GB300, install the NVIDIA IMEX service:

```
sudo apt install nvidia-driver-580-open libnvidia-nscq nvidia-
↪modprobe nvidia-imex datacenter-gpu-manager-4-cuda13 nv-persistence-
↪mode
```

7. Enable the persisted daemon and other services:

- For non-NVswitch systems, such as DGX Station A100, DGX Station A800, and DGX Spark:

```
sudo systemctl enable nvidia-persistenced nvidia-dcgm
```

- For NVswitch systems, such as DGX A100, DGX H100/H200, DGX B200, and DGX B300, make sure also to enable the NVIDIA Fabric Manager service:

```
sudo systemctl enable nvidia-fabricmanager nvidia-persistenced nvidia-
↪dcgm
```

- For multinode NVLink systems, such as DGX GB200 and DGX GB300, enable the NVIDIA IMEX service:

```
sudo systemctl enable nvidia-persistenced nvidia-dcgm nvidia-imex
```

8. Reboot the system to load the drivers and to update system configurations.

- a. Issue reboot.

```
sudo reboot
```

- b. After the system has rebooted, verify that the drivers have been loaded and are managing the NVIDIA devices.

```
nvidia-smi
```

The output should show all available GPUs and the Persistence-Mode On:

```
+-----+
| NVIDIA-SMI 570.124.06  Driver Version: 570.124.06  CUDA Version: 12.
↪8      |
|-----+-----+-----+-----+
| GPU Name          Persistence-M| Bus-Id        Disp.A | Volatile Uncorr.
↪ ECC |
| Fan Temp Perf Pwr:Usage/Cap|      Memory-Usage | GPU-Util
|-----+-----+-----+-----+
(continues on next page)
```


DOCA-OFED Transition Guide.

Note

The DOCA-OFED package is not required for DGX Station A100, DGX Station A800, and DGX Spark.

1. To install DOCA-OFED driver and associate packages, install the metapackage `nvidia-system-mlnx-drivers`, which installs `doca-ofed`, `mlnx-pxe-setup`, `nvidia-mlnx-config`, `nvidia-mstflint-loader`, and `nvidia-ib-umad-loader`.

```
sudo apt install doca-repo
sudo apt install nvidia-system-mlnx-drivers
```

2. Reboot the system.

6.4.5. Installing Docker and the NVIDIA Container Toolkit

Containers provide isolated environments with a full filesystem of the required software for specific applications. To use the NVIDIA provided containers for AI and other frameworks on the DGX and GPUs, you need to install Docker and the NVIDIA Container Toolkit. It takes care of providing access to the GPUs to the software running inside the container.

Note that these tools are also required by the Firmware Update Containers for upgrading the system firmware.

1. Install Docker CE, NVIDIA Container Toolkit, and optimizations for a typical DL workload.

The `docker-ce`, `nvidia-container-toolkit`, and `nv-docker-options` packages might already be installed with the `nvidia-system-utils` and `nvidia-system-extra` metapackages.

```
sudo apt install -y docker-ce nvidia-container-toolkit nv-docker-options
```

2. Restart the docker daemon.

```
sudo systemctl restart docker
```

To validate the installation, run a container and check that it can access the GPUs. The following instructions assume that the NVIDIA GPU driver has been installed and loaded.

Note

This validation downloads a container from the NGC registry and requires that the system has internet access.

1. Execute the following command to start a container and run the `nvidia-smi` tool inside the container:

```
sudo docker run --gpus=all --rm nvcr.io/nvidia/cuda:12.6.2-base-ubuntu24.
  ↳04 nvidia-smi
```

2. Verify that the output shows all available GPUs and has Persistence-Mode set to On.

6.4.6. Installing the NVIDIA System Management (NVSM) Tool [Recommended]

The NVIDIA System Management (NVSM) is a software framework for monitoring NVIDIA DGX nodes in a data center. It allows customers to get a quick system health report and is typically required by the NVIDIA support team to resolve issues.

The `nvsm` and `nvidia-motd` packages might already be installed with the `nvidia-system-utils` metapackage.

The following steps install and configure NVSM.

1. Install the NVIDIA System Management tool (NVSM):

```
sudo apt install -y nvsm
```

2. (Optional) Modify message-of-the-day (MOTD) to display NVSM health monitoring alerts and release information.

```
sudo apt install -y nvidia-motd
```

6.4.7. Additional Software Installed By DGX OS

The Ubuntu and the NVIDIA repositories provide many additional software packages for a variety of applications. The DGX OS Installer, for example, installs several additional software packages to aid system administration and developers that are not installed by default.

The following steps install the additional software packages that get installed by the DGX OS Installer:

Some of these packages might already be installed with the `nvidia-system-extra` metapackage.

1. Install additional software for system administration tasks:

```
sudo apt install -y chrpath cifs-utils fping gdisk iperf ipmitool lsscsi
↪ net-tools nfs-common quota rasdaemon pm-utils samba-common samba-lib
↪ sysstat vlan
```

2. Install additional software for development tasks:

```
sudo apt install -y build-essential automake bison cmake flex gcc-
↪ multilib gdb g++-multilib libelf-dev libltdl-dev m4 swig
```

The NVIDIA CUDA Developer repository provides an easy mechanism to deploy NVIDIA tools and libraries, such as the CUDA Toolkit, cuDNN, or NCCL.

6.5. Next Steps and Additional Information

For further installation and configuration options, refer also to these chapters:

- ▶ *Managing OS and Software Updates* - installing additional software and changing driver branches
- ▶ *Network Configuration* - additional network options and configurations
- ▶ *Data Storage Configuration* - RAID configurations and encryption information
- ▶ *Running NGC Containers* - running NGC containers on the system

Chapter 7. Upgrading the OS

This section provides information about upgrading an existing DGX OS installation.

To reimage the system with DGX OS to a default state, refer to [Reimaging the System](#) for more information.

Important

Before you upgrade a system or any installed software, always consult the [Release Notes](#) for the latest information about available upgrades. You can find out more about the release cadence and release methods for DGX OS in [Release Guidance](#).

This release incorporates the following updates:

- ▶ Ubuntu ConnectX drivers and DOCA-OFED stack
- ▶ Customers are advised to consider these updates and any effect they may have on their application.
- ▶ Best practices support upgrading select systems and verifying that your applications are working as expected before deploying on additional systems.

The following information describes the differences between different types of upgrades:

▶ Release upgrades

When you perform a release upgrade, you currently have the DGX OS 5 or DGX OS 6 installed and want to migrate to DGX OS 7. You can upgrade to DGX OS 7 only from the latest DGX OS 6.x release.

- ▶ For DGX OS 6, refer to [Performing a Release Upgrade from DGX OS 6](#) for the upgrade instructions. The instructions also provide information about completing an over-the-internet upgrade.
- ▶ For DGX OS 5, refer to the [Upgrading the OS](#) section in the *DGX OS 6 User Guide*.

▶ Package upgrades

When you perform package upgrades, you want to install upgrades that are available in the repositories since the initial DGX OS 7 release. The repositories are periodically updated with packages that include bug fixes and security updates. The NVIDIA repository also includes packages with new features that are available with the latest DGX OS minor version release. Refer to [Performing Package Upgrades](#) for instructions.

Note

To change the branch of a driver or CUDA Toolkit, refer to [Managing OS and Software Updates](#) for instructions.

Upgrades are cumulative, meaning your systems will install all available upgrades including upgrades available from Ubuntu, such as the kernel. Performing upgrades will install the latest versions available at the time when the upgrade is performed. **These might be newer than the current DGX OS release.**

Important

The instructions in this chapter upgrade all software for which updates are available from your configured software sources, including applications that you installed yourself. To prevent an application from being upgraded, you can instruct the Ubuntu package manager to keep the current version.

For more information, refer to the Ubuntu Community Help Wiki: [Introduction to Holding Packages](#). It is typically not advised to hold packages as it can disrupt package dependencies.

Important

When you upgrade DGX OS, the system remains on the installed GPU driver branch unless the installed GPU driver branch is end of support. When a GPU driver branch reaches end of support, you will automatically transition to the next supported branch. Refer to [Changing Your GPU Driver Branch](#) for instructions on manually switching GPU driver branches.

7.1. DGX OS 7 Release Upgrade Advisory

You should consider the following points for Ubuntu ConnectX drivers and OFED stack when performing a release upgrade from DGX OS 6 to ensure a successful upgrade:

- ▶ For a release upgrade from DGX OS 6, the Mellanox OFED (MOFED) drivers are replaced with the DOCA OFED drivers from DGX OS 7.
- ▶ You are advised to consider these updates and their effect on their application. For example, some MOFED-dependent applications might be affected.
- ▶ During the release upgrade, the MOFED drivers are replaced with the DOCA OFED drivers.
- ▶ Best practices support upgrading selected systems and verifying that your applications are working as expected before deploying on additional systems.

7.2. Getting Release Information for DGX Systems

Here is some information about how you can determine the release information for your DGX systems.

The `/etc/dgx-release` file provides release information, such as the product name and serial number. This file also tracks the history of the DGX OS software updates by providing the following information:

- ▶ The version number and installation date of the last version to be installed from an ISO image (DGX_SWBUILD_VERSION).
- ▶ The version number and update date of each over-the-network update applied since the software was last installed from an ISO image (DGX_OTA_VERSION).

For DGX OS 7, the DGX_OTA_VERSION file indicates the latest ISO version that was released, and upgrades to the system include the changes that were made in the network repository up to the indicated date. You can use this information to determine whether your DGX system is running the current version of the DGX OS software.

To get release information for the DGX system, view the content of the `/etc/dgx-release` file. For example:

```
more /etc/dgx-release

DGX_NAME="DGX Station A100"
DGX_PRETTY_NAME="NVIDIA DGX Station A100"
DGX_SWBUILD_DATE="2022-10-11-17-49-32"
DGX_SWBUILD_VERSION="5.4.1"
DGX_COMMIT_ID="38d36e8"
DGX_PLATFORM="DGX Station A100"
DGX_SERIAL_NUMBER="1632920000024"

DGX_OTA_VERSION="5.5.0"
DGX_OTA_DATE="Mon 10 Apr 2023 10:11:07 PM PDT"

DGX_OTA_VERSION="6.0.10"
DGX_OTA_DATE="Thu Apr 13 04:55:25 PM PDT 2023"
```

7.3. Preparing to Upgrade the Software

This section provides information about the tasks you need to complete before upgrading your DGX OS software.

7.3.1. Connect to the DGX System Console

Connect to the console of the DGX system using a direct connection or a remote connection through the BMC. See [Connecting to the DGX System](#).

Note

SSH can be used to perform the upgrade. However, if the Ethernet port is configured for DHCP, the IP address might change after the DGX server is rebooted during the upgrade, which results in the loss of connection. A loss of connection might also occur if you are connecting through a VPN. If this happens, connect by using a direct connection or through the BMC to continue the upgrade process. Warning: Connect directly to the DGX server console if the DGX is connected to a 172.17.xx.xx subnet.

DGX OS software installs Docker CE, which uses the 172.17.xx.xx subnet by default for Docker containers. If the DGX server is on the same subnet, you cannot establish a network connection to the DGX server.

See [Configuring Docker IP Addresses](#) To ensure that your DGX system can access the network interfaces for Docker containers, Docker should be configured to use a subnet distinct from other network resources used by the DGX system. for instructions on how to change the default Docker network settings after performing the upgrade.

If you are using a GUI to connect to the console, see [Performing Package Upgrades Using the GUI](#). You can use the graphical Software Updater application to manage package upgrades on the DGX Station.

7.3.2. Verifying the DGX System Connection to the Repositories

Before you attempt to complete the update, you can verify that the network connection for your DGX system can access the public repositories and that the connection is not blocked by a firewall or proxy.

On the DGX system, enter the following:

```
wget -O f1-changelogs http://changelogs.ubuntu.com/meta-release-lts
```

```
wget -O f2-archive http://archive.ubuntu.com/ubuntu/dists/noble/Release
```

```
wget -O f3-security http://security.ubuntu.com/ubuntu/dists/noble/Release
```

```
wget -O f4-nvidia-baseos http://repo.download.nvidia.com/baseos/ubuntu/noble/  
↪x86_64/dists/noble/Release
```

```
wget -O f5-nvidia-cuda https://developer.download.nvidia.com/compute/cuda/  
↪repos/ubuntu2404/x86_64/Release
```

The `wget` commands should be successful, and there should be five files in the directory with non-zero content.

7.4. Performing a Release Upgrade from DGX OS 6

Note

If installed software packages do not have upgrade candidates and you try to upgrade, an error message will be displayed. You need to use the `--force` option and upgrade process. Refer to the [Release Notes](#) for a list of packages that are no longer available in DGX OS 7.

7.4.1. Upgrade DGX OS 6 to the Latest Version

See the Upgrading the OS section in the [DGX OS 6 User Guide](#).

Before you can perform the release upgrade of your system, you need to upgrade the current DGX OS 6 to the latest version. These steps upgrade your system to the latest DGX OS 6 release:

1. Download information from all configured sources about the latest versions of the packages.

```
sudo apt update
```

2. Install all available upgrades for your current DGX OS release.

```
sudo apt -y full-upgrade
```

Note

Depending on which packages were updated when running `sudo apt -y full-upgrade`, you might be prompted to reboot the system before performing `nvidia-release-upgrade`.

7.4.2. Performing the Release Upgrade

Follow these steps to upgrade your system from DGX OS 6 to DGX OS 7:

1. Install the `nvidia-release-upgrade` package for upgrading to the latest DGX OS 6 release.

```
sudo apt install -y nvidia-release-upgrade
```

Note

The next step might install a newer GPU driver. To select a specific driver branch, edit the file `/etc/update-manager/release-upgrades.d/nvidia.cfg` and change the `DriverBranch` setting.

2. Start the DGX OS release upgrade process.

```
sudo nvidia-release-upgrade
```

If you are using a proxy server, add the `-E` option to keep your proxy environment variables. For example:

```
sudo -E nvidia-release-upgrade
```

Note

Some package upgrades require that you reboot the system before completing the upgrade. Ensure that you reboot the system when prompted.

3. Resolve conflicts.

Refer to [Resolving Release Upgrade Conflicts](#) for details and instructions.

4. Wait for the upgrade process to complete and press **y** at the prompt that appears when the system upgrade is completed.

```
System upgrade is complete. Restart is required To finish the upgrade, a
restart is required. If you select 'y' the system will restart.
Continue [yN]
```

The system must be restarted to complete the update process and ensure that any changes are captured by restarted services and runtimes.

Note

If no reboot prompt appeared or if you did not restart the system when prompted, reboot to complete the update process.

```
sudo reboot
```

After the system is restarted, the upgrade process takes several minutes to perform some final installation steps.

7.4.3. Resolving Release Upgrade Conflicts

During the upgrade, the system might encounter conflicts or require other manual intervention.

- ▶ When you are prompted to resolve conflicts in configuration files, evaluate the changes before selecting one of the following options:
 - ▶ Accepting the maintainer's version.
 - ▶ Keeping the local version.
 - ▶ Manually resolving the difference.

Conflicts in some configuration files might be the result of customizations to the Ubuntu Desktop OS made for DGX OS software. For guidance about how to resolve these conflicts, see the chapter in the [Release Notes](#) for the release family to which you are upgrading.

- ▶ `/etc/apt/sources.list.d/dgx.list`. You should install the package maintainer's version.
- ▶ `/etc/ssh/sshd_config`. You can keep the local version that is currently installed.

Conflicts in the following configuration files are the result of customizations to the Ubuntu Desktop OS made for DGX OS 7.

- ▶ `/etc/gdm3/custom.conf.distrib`. You can keep your currently installed version.
 - ▶ `/etc/gdm3/custom.conf`. You can keep your currently installed version.
- ▶ If you are logged in to the DGX system remotely through a secure shell (SSH), you are prompted about whether you want to continue running under SSH.

```
Continue running under SSH?
```

```
This session appears to be running under ssh. It is not recommended to
→perform a upgrade over ssh currently because in case of failure it is
→harder to recover.
```

```
If you continue, an additional ssh daemon will be started at port '1022'.
Do you want to continue?
Continue [yN]
```


- ▶ Enter **y** to continue.
- ▶ An additional sshd daemon is started and the following message is displayed:

```
Starting additional ``sshd`` To make recovery in case of failure
→easier, an
additional sshd will be started on port '1022'. If anything goes wrong
with the running ssh you can still connect to the additional one. If
→you
run a firewall, you may need to temporarily open this port. As this is
potentially dangerous it's not done automatically. You can open the
→port
with e.g.: 'iptables -I INPUT -p tcp --dport 1022 -j ACCEPT' To
→continue
please press [ENTER]
```

- ▶ Press **Enter**.
- ▶ If you are warned that third-party sources are disabled:

```
Third-party sources disabled
Some third-party entries in your sources.list were disabled. You can re-
→enable them after the upgrade with the 'software-properties' tool or
→your package manager.
To continue please press **ENTER**
```

Canonical and DGX repositories are preserved for the upgrade, but any other repositories, for example, Google Chrome or VSCode, will be disabled. After the upgrade, you must manually re-enable any third-party sources that you want to keep.

- ▶ Press **Enter**.
- ▶ You are asked to confirm that you want to start the upgrade.

```
Do you want to start the upgrade?
Installing the upgrade can take several hours. Once the download has
→finished, the process cannot be canceled.
Continue [yN] Details [d]
```

- ▶ Press **Enter**.
- ▶ **(DGX Station only)** In response to the warning that lock screen is disabled, press **Enter** to continue. **Do not** press **Ctrl+C** to respond to this warning, because pressing **Ctrl+C** terminates the upgrade process.
- ▶ If you are prompted to confirm that you want to remove obsolete packages, select one of the options:

```
Remove obsolete packages?
371 packages are going to be removed. Removing the packages can take
→several hours.
Continue [yN] Details [d]
```

- Determine whether to remove obsolete packages and continue with the upgrade.
- Review the list of packages that will be removed.

(continues on next page)

(continued from previous page)

```

    To identify obsolete DGX OS Desktop packages, see the lists of
    ↪obsolete
      packages in the `DGX OS Desktop Release
      Notes <https://docs.nvidia.com/dgx/dgx-os-desktop-release-notes/
    ↪index.html>`__
      for all releases after your current release.

    - If the list contains only packages that you want to remove, enter
      **y** to continue with the upgrade.

```

► Enter **y** to accept the recommended changes, **n** (default) for no, or **d** for more details.

7.4.4. Verifying the Upgrade

Here are steps to verify your upgrade.

1. Confirm the Linux kernel version.

For example, when you upgrade to DGX OS 7.0, the Linux kernel version is at least 6.8.0-48-generic.

2. For the minimum Linux kernel version of the release to which you are upgrading, refer to the release notes for that release.
3. Confirm the NVIDIA Graphics Drivers for Linux version.

```
nvidia-smi
```

For example, for an upgrade to DGX OS 7.0, the NVIDIA Graphics Drivers for Linux version is at least 570:

```

Tu Feb 11 17:00:38 2025
+-----+
↪----+
| NVIDIA-SMI 570.86.05      Driver Version: 570.86.05    CUDA Version: 12.8
↪    |
|-----+-----+-----+
↪----+

```

7.4.5. Recovering from an Interrupted or Failed Update

If the script is interrupted during the update, because of a loss of power or loss of network connection, depending on the issue, you need to restore power or restore the network connection.

If the system encounters a kernel panic after you restore power and reboot the DGX system, you cannot perform the over-the-network update. You need to reinstall DGX OS 6 with the latest image instead. See [Reimaging the System](#).

This section provides information about how to install the DGX OS for instructions and complete the network update.

If you can successfully return to the Linux command line, complete the following steps.

1. Reconfigure the packages.

```
dpkg -a --configure
```

2. Fix the broken package installs.

```
apt -f install -y
```

3. Determine where the release-upgrader was extracted.

```
/tmp/ubuntu-release-upgrader-<random-string>
```

4. Start a bash shell, go to the upgrader, and configure.

```
sudo bash
```

```
cd /tmp/ubuntu-release-upgrader-<random-string>
```

```
RELEASE_UPGRADER_ALLOW_THIRD_PARTY=1 ./noble --  
→frontend=DistUpgradeViewText
```

Do not reboot at this time.

5. Issue the following command and reboot.

```
bash /usr/bin/nvidia-post-release-upgrade
```

```
reboot
```

7.5. Performing Package Upgrades

NVIDIA and Canonical provide updates to the OS in the form of updated software packages between releases with security mitigations and bug fixes. You should evaluate the available updates in regular intervals and update the system that is based on the threat level.

7.5.1. Enabling Extended Security Maintenance Upgrades

This section provides information about Ubuntu's [Extended Security Updates \(ESM\)](#).

As a DGX OS customer, you are entitled to Extended Security Updates from the Ubuntu Universe repository.

You may see the following Ubuntu Pro message from `ubuntu-advantage-tools` during an `apt upgrade` if security updates are available for packages from the Ubuntu Universe repository:

```
Get more security updates through Ubuntu Pro with 'esm-apps' enabled.  
Learn more about Ubuntu Pro at https://ubuntu.com/pro.
```

In addition, DGX users will also get the following NVIDIA message:

```
Your DGX contract entitles you to Extended Security Maintenance updates  
for additional packages in the Ubuntu repository. Please  
contact NVIDIA Support to get your key to enable this capability."
```

After contacting [NVIDIA Enterprise Support](#) to obtain an Ubuntu Pro token, you can use the token with the following command to enable Extended Security Maintenance updates:

```
sudo pro attach XXXXX
```

Ubuntu Pro subscription can be checked with the `sudo pro status` command:

```
sudo pro status
```

7.5.1.1 Renewing Extended Security Maintenance

The Ubuntu Pro Client does not automatically update the Ubuntu Pro subscription for token expiration. If your Ubuntu Pro token expires, refresh the token on the DGX system to renew the Extended Security Maintenance subscription.

To refresh the Ubuntu Pro token:

```
sudo pro refresh
```

To check the new expiration date or the status of ESM:

```
sudo pro status
```

7.5.2. Performing Package Upgrades Using the CLI

You should evaluate the available updates in regular intervals and update the system based on the threat level:

- ▶ Refer to the [Ubuntu Wiki Upgrades](#) for more information about upgrades available for Ubuntu.
- ▶ For a list of the known Common Vulnerabilities and Exposures (CVEs), including those that can be resolved by updating the DGX OS software, refer to the [Ubuntu Security Notices](#)

If updates are available, you can obtain upgraded packages by completing the following steps:

1. Update the internal database with the list of available packages and their versions.

```
sudo apt update
```

2. Review the packages that will be upgraded.

```
sudo apt full-upgrade -s
```

To prevent an application from being upgraded, you can instruct the Ubuntu package manager to “hold packages”. Refer to [Holding Packages](#) for more information.

Note

Holding packages should only be used in extreme rare cases as it can disrupt package dependencies.

3. Upgrade to the latest version.

```
sudo apt full-upgrade
```

When prompted to resolve an issue, answer any questions that appear. Most questions require a **Yes** or **No** response.

- ▶ When prompted to select which the GRUB configuration to use, select the current one on the system.
- ▶ When prompted to select the GRUB install devices, keep the default selection.
- ▶ The other questions will depend on what other packages were installed before the update, and how those packages interact with the update.
- ▶ If a message appears that indicates that the `nvidia-docker.service` failed to start, you can disregard it and continue with the next step. The service will start at that time.

4. When the upgrade is complete, reboot the system.

```
sudo reboot
```

Note

Upgrades to the NVIDIA Graphics Drivers for Linux requires a restart to complete the kernel upgrade. If you upgrade the NVIDIA Graphics Drivers for Linux without restarting the DGX system, when you run the `nvidia-smi` command, an error message is displayed.

```
nvidia-smi
Failed to initialize NVML: Driver/library version mismatch
```

7.5.3. Managing Software Upgrades on DGX Station

This section provides information about managing upgrades between DGX OS releases by using a GUI tool on DGX Station.

7.5.4. Performing Package Upgrades Using the GUI

You can use the graphical Software Updater application to manage package upgrades on the *DGX Station*.

Ensure that you are logged in to your Ubuntu desktop on the **DGX Station** as an administrator user.

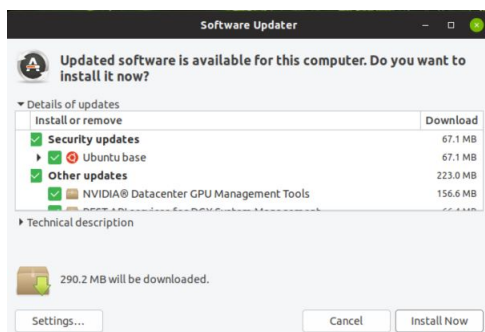
1. Press the **Super** key.

This key is usually found on to the **Alt** key. Refer to [What is the Super key?](#) for more information.

- ▶ If you are using a Windows keyboard, the Super key usually has a Windows logo on it, and it is sometimes called the Windows key or system key.
- ▶ If you are using an Apple keyboard, this key is known as the Apple key.

2. In the search bar, type Software Updater

3. Open the **Software Updater**, review the available updates, and click **[Install Now]**.



Screen capture showing the software updater window.

- ▶ If no updates are available, the *Software Updater* informs you that your software is up to date.
- ▶ If an update requires the removal of obsolete packages, you will be warned that not all updates can be installed.

To continue with the update, complete the following steps:

- a. Click **[Partial Upgrade]**.
- b. Review the list of packages that will be removed. To identify obsolete **DGX Station** packages, see the lists of obsolete packages in the [DGX OS Desktop Release Notes](#) for all releases after your current release.
- c. If the list contains only packages that you want to remove, click **[Start Upgrade]**.
4. When prompted to authenticate, type your password into the **[Password]** field and click **[Authenticate]**.
5. When the update is complete, restart **DGX Station**.

Restart the system even if you are not prompted to restart it to complete the updates. Any update to the NVIDIA Graphics Drivers for Linux requires a restart. If you update the NVIDIA Graphics Drivers for Linux without restarting the **DGX Station**, running the `nvidia-smi` command displays an error message.

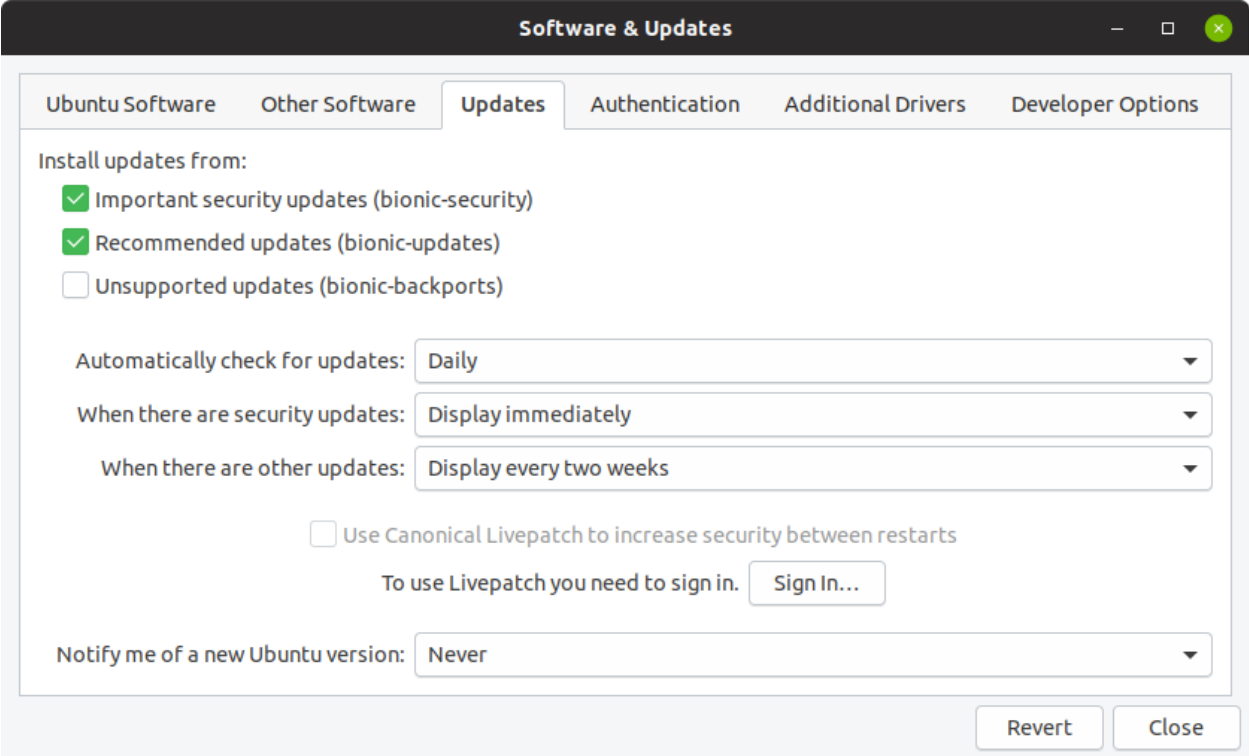
```
nvidia-smi
Failed to initialize NVML: Driver/library version mismatch
```

7.5.5. Checking for Updates to **DGX Station** Software

In **Software & Updates**, you can change your settings to automatically check for package updates and to configure updates from the Ubuntu software repositories. You can also configure your **DGX Station** to notify you of important security updates more frequently than other updates.

In the following example, the **DGX Station** is configured to check for updates daily, to display important security updates immediately, and to display other updates every two weeks.

Screen capture showing the options in the Updates tab of Ubuntu Software & Updates window to check for updates daily, to display important security updates immediately, and to display other updates every two weeks.



Chapter 8. Managing OS and Software Updates

DGX OS 7 is an optimized version of the Ubuntu 24.04 Linux distribution that provides access to an extensive collection of additional software available from the Ubuntu and NVIDIA repositories. For more information about additional software available from Ubuntu, refer to [Install additional applications](#).

Before you install additional software or upgrade installed software, refer to the [Release Notes](#) for the latest release information. To install the additional software, use the `apt` command or the graphical tool. The graphical tool is only available for the DGX Station A100 systems.

In addition, you can change your GPU driver branch and upgrade to a different CUDA Toolkit release to maintain or optimize the OS for your DGX systems.

8.1. Upgrading the System

Before installing any additional software, you should upgrade the system to the latest versions. This ensures you can access new software releases added to the repositories since your last upgrade. Refer to [Upgrading the OS](#) for more information and instructions, including instructions for enabling Ubuntu's [Extended Security Maintenance](#) updates.

Note

- ▶ Before upgrading your system, consult the [Release Notes](#) for the upgrade path and supported DGX systems.
- ▶ You will only see the latest software branches after upgrading the DGX OS.
- ▶ When you switch between software branches, such as the GPU driver or CUDA toolkit, you must install the packages for the new branch. Depending on the software, it will then remove the existing branch or support concurrent branches installed on a system.

8.2. Changing Your GPU Driver Branch

NVIDIA drivers are part of the CUDA repository. For more information about the NVIDIA driver release, refer to the release notes in [NVIDIA Driver Documentation](#).

The DGX B300 and DGX B200 system include the fifth generation of NVIDIA NVLink® and the NVLink Switch technology. With this version of NVlink, additional packages are included with Base OS 7 to

enable the full NVLink functionality. These packages include `nvlsdm` and `libnvsdm` among others. When performing GPU driver updates, it is required to update the driver and the corresponding NVLink stack packages simultaneously. Updating the DGX B300 and DGX B200 systems is listed in the steps of the NVIDIA open GPU kernel modules, as described in [Upgrading Your GPU Driver Branch](#).

8.2.1. Checking the Currently Installed Driver Branch

Before installing a new NVIDIA driver branch, run the following command to check the currently installed driver branch:

```
apt list --installed nvidia-driver*-open
```

8.2.2. Determining the New Available Driver Branches

These steps help you determine which new driver branches are available.

To see the new available NVIDIA driver branches:

1. Update the local database with the latest information from the Ubuntu repository.

```
sudo apt update
```

2. Show the available NVIDIA open GPU kernel module branches.

```
apt list nvidia-driver-*-open
```

8.2.3. Upgrading Your GPU Driver Branch

To manually upgrade your driver to the latest branch:

1. Install the latest kernel.

- For x86_64 systems:

```
sudo apt install -y linux-generic
```

- For ARM64 systems:

```
sudo apt install -y linux-nvidia-64k
```

2. Install the driver pinning package.

580 and newer releases have introduced significant packaging changes. When installing one of these versions, a new pinning package must be installed before installing the driver to ensure that installed versions of all related packages are consistent.

The pinning package is named `nvidia-driver-pinning-XXX` where XXX is the driver version. For example, before installing the 580 family driver, install `nvidia-driver-pinning-580`.

```
sudo apt install nvidia-driver-pinning-580
```

After the driver pinning package is installed proceed with the installation of the driver packages.

3. Upgrade the NVIDIA GPU driver.

Note

- ▶ From the `apt install` examples below, choose the command set appropriate for your environment. Replace the Release 570 GPU driver with the release family you want to install. For DGX systems, the installed GPU driver release must be 570 or greater.
- ▶ The DGX B300, DGX Spark, and DGX GB300 require the Release 580 family of the NVIDIA open GPU kernel modules.
- ▶ For the Release 580 family, the release branch has been removed from the names of the following packages:

Release 570

- ▶ `nvidia-fabricmanager-570`
- ▶ `libnvidia-nscq-570`
- ▶ `libnvdsm-570`
- ▶ `nvidia-imex-570`

Release 580

- ▶ `nvidia-fabricmanager`
- ▶ `libnvidia-nscq`
- ▶ `libnvdsm`
- ▶ `nvidia-imex`

- ▶ To install the NVIDIA open GPU kernel modules of a different release family from the current GPU driver, specify the packages with the `-open` string, for example, `nvidia-driver-570-open`:

Note

In the following commands, the trailing `-` character in `nvidia${GPU_BRANCH}*-` specifies that the currently installed GPU driver will be removed in the same transaction. Because this operation removes packages from the system, it is important to perform a dry run first to ensure that the correct packages will be removed.

- ▶ For non-NVSwitch systems, such as DGX Station A100, DGX Station A800, and DGX Spark, run the following commands:

DGX Station A100 and A800:

```
GPU_BRANCH=$(dpkg -l | grep nvidia-driver | tr -s " " | cut -d' ' -
↪f3 | cut -d'.' -f1)

# Specify --dry-run to check the packages to install.
sudo apt install -y nvidia-driver-570-open libnvidia-nscq-570
↪nvidia-modprobe "*nvidia*${GPU_BRANCH}*-" --dry-run

# Install the packages.
sudo apt install -y nvidia-driver-570-open libnvidia-nscq-570
↪nvidia-modprobe "*nvidia*${GPU_BRANCH}*-"
```

DGX Spark:

```
GPU_BRANCH=$(dpkg -l | grep nvidia-driver | tr -s " " | cut -d' ' -
↪f3 | cut -d'.' -f1)
```

(continues on next page)

(continued from previous page)

```
# Specify --dry-run to check the packages to install.
sudo apt install -y nvidia-driver-580-open libnvidia-nscq nvidia-
↳modprobe "*"nvidia*${GPU_BRANCH}*" --dry-run

# Install the packages.
sudo apt install -y nvidia-driver-580-open libnvidia-nscq nvidia-
↳modprobe "*"nvidia*${GPU_BRANCH}*" --
```

- For multinode NVLink systems, such as DGX GB300 and GB200, run the same commands using the Release 580 GPU driver, but append the `nvidia-imex` package:

```
GPU_BRANCH=$(dpkg -l | grep nvidia-driver | tr -s " " | cut -d' ' -
↳f3 | cut -d'.' -f1)

# Specify --dry-run to check the packages to install.
sudo apt install -y nvidia-driver-580-open libnvidia-nscq nvidia-
↳modprobe nvidia-imex "*"nvidia*${GPU_BRANCH}*" nvidia-imex*- --
↳allow-change-held-packages --dry-run

# Install the packages.
sudo apt install -y nvidia-driver-580-open libnvidia-nscq nvidia-
↳modprobe nvidia-imex "*"nvidia*${GPU_BRANCH}*" nvidia-imex*- --
↳allow-change-held-packages
```

- For NVSwitch systems with the fifth-generation NVLinks, such as DGX B200, run the same commands using the Release 570 GPU driver, but append the `nvidia-fabricmanager-570`, `nvlsdm`, and `libnvdsdm-570` packages:

```
GPU_BRANCH=$(dpkg -l | grep nvidia-driver | tr -s " " | cut -d' ' -
↳f3 | cut -d'.' -f1)

# Specify --dry-run to check the packages to install.
sudo apt install -y doca-ofed --dry-run
sudo apt install -y nvidia-driver-570-open libnvidia-nscq-570
↳nvidia-modprobe nvidia-fabricmanager-570 nvlsdm libnvdsdm-570
↳"nvidia*${GPU_BRANCH}*" libnvdsdm*- --allow-change-held-
↳packages --dry-run

# Install the packages.
sudo apt install -y doca-ofed
sudo apt install -y nvidia-driver-570-open libnvidia-nscq-570
↳nvidia-modprobe nvidia-fabricmanager-570 nvlsdm libnvdsdm-570
↳"nvidia*${GPU_BRANCH}*" libnvdsdm*- --allow-change-held-packages
```

- For NVSwitch systems with the fifth-generation NVLinks, such as DGX B300 and DGX B200, run the same commands using the Release 580 GPU driver, but append the `nvidia-fabricmanager`, `nvlsdm`, and `libnvdsdm` packages:

```
GPU_BRANCH=$(dpkg -l | grep nvidia-driver | tr -s " " | cut -d' ' -
↳f3 | cut -d'.' -f1)

# Specify --dry-run to check the packages to install.
sudo apt install -y doca-ofed --dry-run
```

(continues on next page)

(continued from previous page)

```

sudo apt install -y nvidia-driver-580-open libnvidia-nscq nvidia-
↳modprobe nvidia-fabricmanager nvlsdm libnvsdm "*nvidia*${GPU_
↳BRANCH}*- " libnvsdm*- --allow-change-held-packages --dry-run

# Install the packages.
sudo apt install -y doca-ufed
sudo apt install -y nvidia-driver-580-open libnvidia-nscq nvidia-
↳modprobe nvidia-fabricmanager nvlsdm libnvsdm "*nvidia*${GPU_
↳BRANCH}*- " libnvsdm*- --allow-change-held-packages

```

- For NVSwitch systems without the fifth-generation NVLinks, such as DGX A100, DGX A800, DGX H800, and DGX H100/H200, run the same commands using the Release 570 GPU driver, but append the `nvidia-fabricmanager-570` package:

```

GPU_BRANCH=$(dpkg -l | grep nvidia-driver | tr -s " " | cut -d' ' -
↳f3 | cut -d'.' -f1)

# Specify --dry-run to check the packages to install.
sudo apt install -y doca-ufed --dry-run
sudo apt install -y nvidia-driver-570-open libnvidia-nscq-570
↳nvidia-modprobe nvidia-fabricmanager-570 "*nvidia*${GPU_BRANCH}*-
↳" --dry-run

# Install the packages.
sudo apt install -y doca-ufed
sudo apt install -y nvidia-driver-570-open libnvidia-nscq-570
↳nvidia-modprobe nvidia-fabricmanager-570 "*nvidia*${GPU_BRANCH}*-
↳"

```

- To install the NVIDIA open GPU kernel modules of the same release family as the current GPU driver, such as the Release 570,
 - For non-NVSwitch systems, such as DGX Station A100, DGX Station A800, and DGX Spark, first remove the current driver and then install the package:

DGX Station A100 and A800:

```

# Remove the current driver.
sudo apt-get purge "*nvidia*570*"

# Install the packages.
sudo apt install -y nvidia-driver-570-open libnvidia-nscq-570
↳nvidia-modprobe

```

DGX Spark:

```

# Remove the current driver.
sudo apt-get purge "*nvidia*580*"

# Install the packages.
sudo apt install -y nvidia-driver-580-open libnvidia-nscq nvidia-
↳modprobe

```

- For multinode NVLink systems, such as DGX GB300 and DGX GB200, run the same commands using the Release 580 GPU driver, but append the `nvidia-imex` package:

```
# Remove the current driver.
sudo apt-get purge "*nvidia*580*"

# Install the packages.
sudo apt install -y nvidia-driver-580-open libnvidia-nscq nvidia-
↳ modprobe nvidia-imex
```

- For NVSwitch systems with the fifth-generation NVLinks, such as DGX B200, run the same commands using the Release 570 GPU driver, but append the `nvidia-fabricmanager-570`, `nvls`, and `libnvsdm-570` packages:

```
#Remove the current driver.
sudo apt-get purge "*nvidia*570*"

#Install the packages.
sudo apt install -y doca-ufed
sudo apt install -y nvidia-driver-570-open libnvidia-nscq-570
↳ nvidia-modprobe nvidia-fabricmanager-570 nvls libnvsdm-570
```

- For NVSwitch systems with the fifth-generation NVLinks, such as DGX B300, run the same commands using the Release 580 GPU driver, but append the `nvidia-fabricmanager`, `nvls`, and `libnvsdm` packages:

```
#Remove the current driver.
sudo apt-get purge "*nvidia*580*"

#Install the packages.
sudo apt install -y doca-ufed
sudo apt install -y nvidia-driver-580-open libnvidia-nscq nvidia-
↳ modprobe nvidia-fabricmanager nvls libnvsdm
```

- For NVSwitch systems without the fifth-generation NVLinks, such as DGX A100, DGX A800, DGX H800, and DGX H100/H200, run the same commands using the Release 570 GPU driver, but append the `nvidia-fabricmanager-570` package:

```
#Remove the current driver.
sudo apt-get purge "*nvidia*570*"

#Install the packages.
sudo apt install -y doca-ufed
sudo apt install -y nvidia-driver-570-open libnvidia-nscq-570
↳ nvidia-modprobe nvidia-fabricmanager-570
```

4. Reboot the system to ensure the new drivers get loaded:

```
sudo reboot
```

8.3. Installing or Upgrading to a Newer CUDA Toolkit Release

Only DGX Station and DGX Station A100 have a CUDA Toolkit release installed by default. DGX servers are intended to be shared resources that use containers and do not have CUDA Toolkit installed by default. However, you have the option to install a qualified CUDA Toolkit release.

Although the DGX OS supports all CUDA Toolkit releases that interoperate with the installed driver, DGX OS releases might include a default CUDA Toolkit release that might not be the most recently released version. Unless you must use a new CUDA Toolkit version that contains the new features, we recommend that you remain on the default version that is included in the DGX OS release. Refer to the [DGX OS Software Release Notes](#) for the default CUDA Toolkit release.

Important

Before you install or upgrade to any CUDA Toolkit release, ensure the release is compatible with the driver that is installed on the system. Refer to [CUDA Compatibility](#) for more information and a compatibility matrix.

8.3.1. CUDA Compatibility Matrix and Forward Compatibility

Each CUDA toolkit requires a minimum GPU driver version. This compatibility matrix is documented in [CUDA Compatibility](#).

Newer CUDA Toolkits may be used with older GPU drivers if the appropriate forward compatibility package is installed. Refer to: [Installing the CUDA Forward Compatibility Package](#).

Example:

CUDA Toolkit 12.8 requires GPU driver version 570.86.15; however, the installed GPU driver is the Release 550 GPU driver. To use CUDA Toolkit 12.8 with the older GPU driver, you must install the cuda-compat-12-8 forward compatibility package:

```
sudo apt install cuda-compat-12-8
```

You can set the LD_LIBRARY_PATH manually:

```
LD_LIBRARY_PATH=/usr/local/cuda/compat:$LD_LIBRARY_PATH
```

Alternatively, you can configure it automatically by modifying the `/etc/ld.so.conf` file or by adding a file under the `/etc/ld.so.conf.d/` directory.

8.3.2. Checking the Currently Installed CUDA Toolkit Release

Here is some information about the prerequisites to determine the CUDA Toolkit release that you currently have installed.

Before installing a new CUDA Toolkit release, run the following command to check the currently installed release:

```
apt list --installed cuda-toolkit-*
```

The following example output shows that CUDA Toolkit 12.8 is installed:

```
$ apt list --installed cuda-toolkit-*
Listing... Done
cuda-toolkit-12-8/unknown,now 12.8.1-1 amd64 [installed]
N: There is 1 additional version. Please use the '-a' switch to see it
```

8.3.3. Installing or Upgrading the CUDA Toolkit

These steps help you determine which new CUDA Toolkit releases are available.

To see the new available CUDA Toolkit releases:

1. Update the local database with the latest information from the Ubuntu repository.

```
sudo apt update
```

2. Show all available CUDA Toolkit releases.

```
apt list cuda-toolkit-*
```

The following output shows that 11.8, 12.0, 12.1, and 12.2 are the possible CUDA Toolkit versions that can be installed:

```
Listing... Done
cuda-toolkit-11-8/unknown 11.8.0-1 amd64
cuda-toolkit-12-0/unknown 12.0.0-1 amd64
cuda-toolkit-12-1/unknown 12.1.0-1 amd64
cuda-toolkit-12-2/unknown 12.2.0-1 amd64
```

3. To install or upgrade the CUDA Toolkit, run the following:

```
sudo apt install cuda-toolkit-<version>
```

8.4. Installing the Latest DOCA-OFED Package

The NVIDIA DOCA™ OFED software provides the same functionality as MLNX_OFED, including kernel drivers, user space libraries, and management tools for NVIDIA networking products. For more information about DOCA-OFED, refer to the [What IS DOCA-OFED](#) section in [MLNX_OFED to DOCA-OFED Transition Guide](#). For installation information, refer to [NVIDIA DOCA Installation Guide for Linux](#).

To install the latest version of NVIDIA DOCA-OFED software:

1. Uninstall any older versions of DOCA-OFED (or MLNX_OFED) software from your system before proceeding.

```
for f in $( dpkg --get-selections | grep -E 'dca|flexio|dpa-gdbserver|dpa-
→stats|dpa-resource-mgmt|dpaeumgmt' | awk '{print $2}' ); do echo $f ;
```

(continues on next page)

(continued from previous page)

```
→ sudo apt remove --purge $f -y ; done
sudo /usr/sbin/ofed_uninstall.sh --force
sudo apt-get autoremove
```

2. Add the NVIDIA DOCA-OFED repository to your system.

For x86_64-based DGX systems:

```
$ sudo dd status=none of=/etc/apt/sources.list.d/doca.sources << EOF
Types: deb
URIs: https://linux.mellanox.com/public/repo/doca/baseos8-latest/ubuntu24.
→04/x86_64/
Suites: /
Signed-By: /usr/share/keyrings/GPG-KEY-Mellanox.gpg
EOF
$ sudo apt update
$ sudo apt install nvidia-system-mlnx-drivers -y
```

For ARM64-based DGX systems:

```
$ sudo dd status=none of=/etc/apt/sources.list.d/doca.sources << EOF
Types: deb
URIs: https://linux.mellanox.com/public/repo/doca/baseos8-latest/ubuntu24.
→04/arm64-sbsa/
Suites: /
Signed-By: /usr/share/keyrings/GPG-KEY-Mellanox.gpg
EOF
$ sudo apt update
$ sudo apt install nvidia-system-mlnx-drivers -y
```

Note

This sequence sets up the repository for the most recent version of NVIDIA DOCA-OFED. If you need a specific version, replace `baseos8-latest` with the version. For example, to get version 3.0.0-058218 for an x86_64 system, use the https://linux.mellanox.com/public/repo/doca/3.0.0-058218/ubuntu24.04/x86_64/ URI.

8.5. Installing GPUDirect Storage Support

NVIDIA Magnum IO GPUDirect Storage (GDS) enables a direct data path for direct memory access (DMA) transfers between GPU memory and storage. This software avoids a bounce buffer through the CPU.

Note

This section only applies to the following situations:

- Intend to use GPUDirect Storage in bare metal, but do not use [Linux PCI P2PDMA](#).

- Intend to upgrade or re-install the DOCA-OFED package and the nvidia-fs module because DGX OS 7 provides the updated versions.

8.5.1. Installing GDS Components

On DGX servers (DGX B300, DGX B200, H100/H200, H800, A100/A800, GB300, and GB200):

- Install the nvidia-gds package.

```
sudo apt install nvidia-gds
```

On DGX stations (DGX Station A800 and A100):

- Install the nvidia-gds package.

```
sudo apt update
sudo apt install doca-repo -y
sudo apt update

sudo apt install nvidia-peermem-loader nvidia-gds mlnx-nvme-dkms mlnx-
↳nfsrdma-dkms -y

MODULE_VERSION=$(dkms status nvidia | cut -d "," -f1)
sudo dkms remove -m ${MODULE_VERSION} -k $(uname -r) && sudo dkms install
↳-m ${MODULE_VERSION} -k $(uname -r)
```

8.5.2. Enabling Relaxed Ordering for NVMe Drives

The Samsung NVMe drives used in the NVIDIA DGX systems support relaxed ordering for I/O operations. Relaxed ordering enables the PCIe bus to complete transactions out of order. NVIDIA recommends enabling this setting when you use GPUDirect Storage to improve performance.

- Run the nvidia-relaxed-ordering-nvme.sh utility.

```
sudo /bin/nvidia-relaxed-ordering-nvme.sh enable
```

8.5.3. Configuring NVMe Interrupt Coalescing

The nvidia-nvme-options package, which is installed on all DGX systems, automatically configures NVMe interrupt coalescing on all Samsung and Kioxia drives at each boot. To disable this setting or manually configure the setting, issue the following commands:

To disable the setting:

```
sudo systemctl stop nvidia-nvme-interrupt-coalescing.service
sudo systemctl disable nvidia-nvme-interrupt-coalescing.service
```

To configure the setting manually:

```
sudo /usr/bin/nvidia-nvme-interrupt-coalescing.sh enable
sudo /usr/bin/nvidia-nvme-interrupt-coalescing.sh disable
```

8.5.4. Next Steps

Refer to [Verifying a Successful GDS Installation](#) in the *NVIDIA GPUDirect Storage Installation and Troubleshooting Guide*.

Chapter 9. Installing Firmware

This topic provides information about installing the network card firmware:

9.1. Installing ConnectX-7 Firmware

Follow these steps to update firmware for the ConnectX®-7 InfiniBand/Ethernet PCI Express Adapter Cards using the NVIDIA Networking Firmware Downloads Page.

1. Navigate to the [NVIDIA Networking Firmware Downloads](#) page.
2. From the **ConnectX Adapter Cards Firmware** table, select version **ConnectX-7** and click the **InfiniBand/Ethernet** network protocol.

ConnectX Adapter Cards Firmware	
Product Line	Network Protocol
ConnectX-7	InfiniBand/Ethernet
ConnectX-6 DE	InfiniBand
ConnectX-6 Lx	Ethernet
ConnectX-6 Dx	Ethernet

The [Firmware for ConnectX®-7 InfiniBand](#) page opens.

3. On the **ConnectX-7 Firmware Download Center** matrix, display the firmware information and documentation.
 1. Select the **CURRENT VERSIONS** tab.
 2. Choose the **Version (Current)**, **OPN** (ordering part number), and **PSID** options you want. For example,

ConnectX-7 Firmware Download Center

CURRENT VERSIONS		ARCHIVE VERSIONS	START OVER
Version (Current)	OPN	PSID	Download/Documentation
28.43.2026-LTS	MCX755206AS-NEA-N	MT_0000000892	ConnectX7IB: fw-ConnectX7-rel-28_43_2026-MCX755206AS-NEA_Ax-UEFI-14.36.21-FlexBoot-3.7.500.signed MD5SUM: d7fb9e3c8a93da082f06a71055abfc30 SHA256: 62f96ab0a70121e3ff6c780c91355edf6f317a2530ed5f1f6daa79c61055fcc1 Release Date: 06-December-2024 Documentation: Release Notes EULA
28.43.1014	MCX75510AAS-NEA		
28.39.3560-LTS	MCX75510AAS-HEA		
28.35.4030-LTS	MCX755106AS-HEA		
	MCX755106AC-HEA		
	MCX75343AMS-NEAC		
	MCX75343AMC-NEAC		

- Click the **ConnectX7IB** link to download the firmware BIN file.
- After downloading the correct ConnectX-7 firmware, proceed with the installation steps.

Alternatively, you can use the following methods to install the ConnectX-7 adapter card firmware:

► Using the mstflint tool

If you have installed the MTNIC Driver on your machine, you can update the firmware using the mstflint tool as described in the [mstflint FW Burning Tool README](#). You can download the mstflint tool from the OpenFabrics site at [mstflint_SW for Linux](#).

► Using the NVIDIA Firmware Tools (MFT)

For details, refer to [Updating Firmware for a Single Network Interface Card \(NIC\)](#).

9.2. Installing ConnectX-8 Firmware

Follow these steps to update firmware for the ConnectX®-8 InfiniBand/Ethernet PCI Express Adapter Cards using the NVIDIA Networking Firmware Downloads Page.

- Navigate to the [NVIDIA Networking Firmware Downloads](#) page.
- From the **ConnectX Adapter Cards Firmware** table, select version **ConnectX-8** and click the **InfiniBand/Ethernet** network protocol.

ConnectX Adapter Cards Firmware	
Product Line	Network Protocol
ConnectX-8	InfiniBand/Ethernet
ConnectX-7	InfiniBand/Ethernet
ConnectX-6 DE	InfiniBand
ConnectX-6 Lx	Ethernet

The [Firmware for ConnectX®-8 InfiniBand](#) page opens.

- On the **ConnectX-8 Firmware Download Center** matrix, display the firmware information and documentation.
 - Select the **CURRENT VERSIONS** tab.
 - Choose the **Version (Current)**, **OPN** (ordering part number), and **PSID** options you want.
- Click the **ConnectX8IB** link to download the firmware BIN file.
- After downloading the correct ConnectX-8 firmware, proceed with the installation steps.

Alternatively, you can use the following methods to install the ConnectX-8 adapter card firmware:

► Using the `mstflint` tool

If you have installed the MTNIC Driver on your machine, you can update the firmware using the `mstflint` tool as described in the [mstflint FW Burning Tool README](#). You can download the `mstflint` tool from the OpenFabrics site at [mstflint_SW for Linux](#).

► Using the NVIDIA Firmware Tools (MFT)

For details, refer to [Updating Firmware for a Single Network Interface Card \(NIC\)](#).

9.3. Installing BlueField-3 NIC Firmware

Follow these steps to update firmware for the NVIDIA® BlueField®-3 DPUs in NIC mode using the NVIDIA Networking Firmware Downloads Page.

- Navigate to the [NVIDIA Networking Firmware Downloads](#) page.
- From the **NVIDIA BlueField DPU Firmware** table, select product line **NVIDIA BlueField-3** and click the **InfiniBand/Ethernet** network protocol.

NVIDIA BlueField DPU Firmware	
Product Line	Network Protocol
NVIDIA BlueField-3	InfiniBand/Ethernet
NVIDIA BlueField-2	InfiniBand/Ethernet

The [Firmware for NVIDIA BlueField-3](#) page opens.

- On the **NVIDIA BlueField-3 DPU Firmware Download Center** matrix, display the firmware information and documentation.
 - Select the **CURRENT VERSIONS** tab.
 - Choose the **Version (Current)**, **OPN** (ordering part number), and **PSID** options you want. For example,

NVIDIA BlueField-3 DPU Firmware Download Center

CURRENT VERSIONS				ARCHIVE VERSIONS	START OVER
Version (Current)	OPN	PSID	Download/Documentation		
32.43.2026-LTS	900-9D3B6-00CN-A	MT_0000000883	BlueField3IB: fw-BlueField-3-rel-32_43_2026-900-9D3B6-00CN-A_Ax-NVME-20.4.1-UEFI-21.4.13-UEFI-22.4.14-UEFI-14.36.21-FlexBoot-3.7.500.signed MD5SUM: a05b5b471a5d487781b3c1af200703da SHA256: 5d85eee6eb605e175ab7d6a34f8de5eb17f3dc61f26ee56dbe5198b1d475d40a Date: 06-December-2024 Product Description: NVIDIA BlueField-3 B3240 P-Series Dual-slot FHHL DPU; 400GbE / NDR IB (default mode); Dual-port QSFP112; PCIe Gen5.0 x16 with x16 PCIe extension option; 16 Arm cores; 32GB on-board DDR; integrated BMC; Crypto Enabled Documentation: Release Notes EULA		
32.43.1014	900-9D3B6-00CC-EA				
32.39.3560-LTS	900-9D3B6-00CC-A				
	900-9D3B4-00SV-EA				
	900-9D3B4-00SC-E				
	900-9D3B4-00PN-E				
	900-9D3B4-00EN-E				

Note

PSID (Parameter-Set Identification) is a 16-ASCII character string embedded in the firmware image, providing a unique identification for the firmware configuration. For more information, refer to [Help in identifying the PSID of your Adapter card](#).

- Click the **BlueField3IB** link to download the firmware BIN file.
- After downloading the correct BlueField-3 NIC firmware, proceed with the installation steps.

For more information on the following topics, refer to

- ▶ [NVIDIA BlueField-3 DPU NIC Firmware Release Notes](#)
- ▶ [NVIDIA BlueField-3 DPU Controller User Manual](#)
- ▶ [BlueField DPUs/SuperNICs and DOCA](#)

Alternatively, you can use the NVIDIA firmware tools (MFT) to install the BlueField-3 NIC firmware:

- ▶ To download MFT, refer to the [MFT download page](#).
- ▶ For installation details, refer to [Updating Firmware for a Single Network Interface Card \(NIC\)](#).

9.3.1. Resetting the BlueField-3 BMC Password

Note

After installing DGX OS 7, you should reset the default password of BlueField-3 BMC. The default password is `0penBmc`.

Follow these steps to change the BMC password,

1. Download a current BFB file from [NVIDIA DOCA Downloads](#).
2. Ensure the `rshim` package is installed and the `rshim` service is running.

```
sudo dpkg -l | grep rshim
sudo systemctl status rshim
```

3. Find the devices for any Bluefield-3 cards present on the system.

```
ls /dev/rshim*
```

4. Create a `bf.cfg` configuration file with the following parameter values.

```
NEW_BMC_PASSWORD="NewPassword"
BMC_USER="root"
BMC_REBOOT="yes"
BMC_PASSWORD="0penBmc"
UPDATE_BMC_FW="no"
UPDATE_CEC="no"
WITH_NIC_FW_UPDATE="no"
UPDATE_DPU_OS="no"
UPDATE_DPU_GOLDEN_IMAGE="no"
UPDATE_NIC_FW_GOLDEN_IMAGE="no"
```

These parameter values only update the password. If you also want to update the firmware, adjust the `UPDATE_` parameter values based on your requirements.

5. Update the firmware for each `rshim` device found in step 3.

```
sudo bfb-install --rshim rshim<N> --bfb <path_to_bfb_image> --config
→<path_to_bf.cfg>
```

For more information, refer to

- [Customizing BlueField Software Deployment Using bf.cfg](#)
- [Changing Default Password](#) in the NVIDIA BlueField BMC Software documentation

Chapter 10. System Configurations

This section provides information about less common configuration options once a system has been installed.

Refer also to *DGX OS Connectivity Requirements* for a list of network ports used by various services.

10.1. Network Configuration

This section provides information about you can configure the network in your DGX system.

10.1.1. Configuring Network Proxies

If your network needs to use a proxy server, you need to set up configuration files to ensure the DGX system communicates through the proxy.

10.1.2. For the OS and Most Applications

Here is some information about configuring the network for the OS and other applications.

Edit the `/etc/environment` file and add the following proxy addresses to the file, below the `PATH` line.

```
http_proxy="http://<username>:<password>@<host>:<port>/"
ftp_proxy="ftp://<username>:<password>@<host>:<port>/"
https_proxy="https://<username>:<password>@<host>:<port>/"
no_proxy="localhost, 127.0.0.1, localaddress, .localdomain.com"
HTTP_PROXY="http://<username>:<password>@<host>:<port>/"
FTP_PROXY="ftp://<username>:<password>@<host>:<port>/"
HTTPS_PROXY="https://<username>:<password>@<host>:<port>/"
NO_PROXY="localhost, 127.0.0.1, localaddress, .localdomain.com"
```

Where `username` and `password` are optional.

For example, for the HTTP proxy (both, upper and lower case versions must be changed):

```
http_proxy="http://myproxy.server.com:8080/"
HTTP_PROXY="http://myproxy.server.com:8080/"
```

10.1.3. For the apt Package Manager

Here is some information about configuring the network for the apt package manager.

Edit or create the `/etc/apt/apt.conf.d/myproxy` proxy configuration file and include the following lines:

```
Acquire::http::proxy "http://<username>:<password>@<host>:<port>/" ;
Acquire::ftp::proxy "ftp://<username>:<password>@<host>:<port>/" ;
Acquire::https::proxy "https://<username>:<password>@<host>:<port>/" ;
```

For example:

```
Acquire::http::proxy "http://myproxy.server.com:8080/" ;
Acquire::ftp::proxy "ftp://myproxy.server.com:8080/" ;
Acquire::https::proxy "https://myproxy.server.com:8080/" ;
```

10.2. Configuring ConnectX from InfiniBand to Ethernet

Many DGX Systems are equipped with NVIDIA ConnectX network controllers and are typically used for cluster communications. By default, the controllers are configured as InfiniBand ports. You can optionally configure the ports for Ethernet.

Before or after you reconfigure the port, make sure that the network switch that is connected to the port is also reconfigured to Ethernet or that the port is connected to a different switch configured for Ethernet.

The code samples in the following sections use the `mlxconfig` command.

10.2.1. Determining the Current Port Configuration

Perform the following steps to determine the current configuration for the port.

- Query the devices:

```
sudo mlxconfig -e -d aa:00.0 q |grep -e 'Device\|Configurations\|LINK_TYPE'
```

The following example shows the output for one of the port devices on an NVIDIA DGX B200 system. It shows the device path and the default, current, and next boot configurations, which are all set to IB(1).

```
...
Device #1:
Device type:      BlueField3
Device:          aa:00.0
Configurations:
  ↳ Next Boot
*   LINK_TYPE_P1
  ↳ IB(1)
```

	Default	Current
IB(1)	IB(1)	IB(1)

(continues on next page)

(continued from previous page)

*	LINK_TYPE_P2	IB(1)	IB(1)
→	IB(1)		

- ▶ IB(1) indicates the port is configured for InfiniBand.
- ▶ ETH(2) indicates the port is configured for Ethernet.

Determine the device path bus numbers for the slot number of the port that you want to configure. Refer to the following documents for more information:

- ▶ [DGX B300 Port Description](#) in the *NVIDIA DGX B300 System User Guide*
- ▶ [DGX B200 Network Ports](#) in the *NVIDIA DGX B200 System User Guide*
- ▶ [DGX H100/H200 Network Ports](#) in the *NVIDIA DGX H100/H200 System User Guide*
- ▶ [DGX A100 Network Ports](#) in the *NVIDIA DGX A100 System User Guide*

10.2.2. Configuring the Port

1. Use the `mlxconfig` command with the `set LINK_TYPE_P<x>` argument for each port you want to configure.

The following sample command sets port `LINK_TYPE_P1` of the controller with PCI ID `aa:00.0` to Ethernet (2):

```
sudo mlxconfig -y -d aa:00.0 set LINK_TYPE_P1=2
```

The following example output is from an NVIDIA DGX B200 system.

```
Device #1:
-----

Device type:      BlueField3
Name:             900-9D3B6-00CN-A_Ax
Description:      NVIDIA BlueField-3 B3240 P-Series Dual-slot FHHL DPU;
→400GbE / NDR IB (default mode); Dual-port QSFP112; PCIe Gen5.0 x16 with
→x16 PCIe extension option; 16 Arm cores; 32GB on-board DDR; integrated
→BMC; Crypto Enabled
Device:           aa:00.0

Configurations:                                     Next Boot
→New                                                    ETH(2)
    LINK_TYPE_P1
→IB(1)

Apply new Configuration? (y/n) [n] : y
Applying... Done!
-I- Please reboot machine to load new configurations.
```

Here is an example that sets port `LINK_TYPE_P2` to Ethernet:

```
sudo mlxconfig -y -d aa:00.0 set LINK_TYPE_P2=2
```

2. (Optional) Run `mlxconfig` again to confirm the change:

```
sudo mlxconfig -e -d aa:00.0 q |grep -e 'Device\|Configurations\|LINK_TYPE'
```

In the following example output, port LINK_TYPE_1 is set to ETH(2) for the next boot. The output shows the device path and the default, current, and next boot configurations.

```
...
Device #1:
Device type:      BlueField3
Device:           aa:00.0
Configurations:
Next Boot
* LINK_TYPE_P1    IB(1)    IB(1)
  ETH(2)
* LINK_TYPE_P2    IB(1)    IB(1)
  IB(1)
```

3. Power off and then power on the system for the change to take effect.

Wait for the operating system to boot.

10.2.3. Configuring the ConnectX-8 Port for Splitting

To change the ConnectX-8 interface from a single 800 Gbps InfiniBand port to two 400 Gb Ethernet ports:

1. Set the link type for Port 1 to Ethernet (2).

```
mlxconfig -d <device> set LINK_TYPE_P1=2
```

Replace <device> with the actual PCI device ID of your ConnectX-8 device.

2. Set the number of planes for Port 1 to 0.

```
mlxconfig -d <device> set NUM_OF_PLANES_P1=0
```

3. Define the physical module split configuration.

```
mlxconfig -d <device> set MODULE_SPLIT_M0[0..3]=1 MODULE_SPLIT_M0[4..7]=2
→MODULE_SPLIT_M0[8..15]=FF
```

This specific set of values maps the internal physical lanes of the 800 Gbps port into two separate 400 GbE logical ports.

4. Set the number of Physical Functions (PFs) to 2, allowing the operating system to recognize two 400 GbE network interfaces.

```
mlxconfig -d <device> set NUM_OF_PF=2
```

5. Power off and then power on the system for the change to take effect.

Wait for the operating system to boot.

To revert the device interface to its original single 800 Gbps InfiniBand (IB) port configuration:

```
mlxconfig -d <device> reset
```

Note

The reset action will revert all settings on the device, including any other specific settings you have modified.

For more detailed information on configuring ConnectX-8 devices, refer to [Port Splitting Configurations](#).

10.3. Docker Configuration

To ensure that Docker can access the NGC container registry through a proxy, Docker uses environment variables.

For best practice recommendations on configuring proxy environment variables for Docker, refer to [Control Docker with systemd](#).

10.3.1. Preparing the DGX System to be Used With Docker

Some initial setup of the DGX system is required to ensure that users have the required privileges to run Docker containers and to prevent IP address conflicts between Docker and the DGX system.

10.3.2. Enabling Users To Run Docker Containers

To prevent the docker daemon from running without protection against escalation of privileges, the Docker software requires sudo privileges to run containers. Meeting this requirement involves enabling users who will run Docker containers to run commands with sudo privileges.

You should ensure that only users whom you trust and who are aware of the potential risks to the DGX system of running commands with sudo privileges can run Docker containers.

Before you allow multiple users to run commands with sudo privileges, consult your IT department to determine whether you might be violating your organization's security policies. For the security implications of enabling users to run Docker containers, see [Docker daemon attack surface](#).

You can enable users to run the Docker containers in one of the following ways:

- ▶ Add each user as an administrator user with sudo privileges.
- ▶ Add each user as a standard user without sudo privileges and then add the user to the docker group.

This approach is inherently insecure because any user who can send commands to the docker engine can escalate privilege and run root-user operations.

To add an existing user to the docker group, run this command:

```
sudo usermod -aG docker user-login-id
```

Where `user-login-id` is the user login ID of the existing user that you are adding to the *docker* group.

10.3.3. Configuring Docker IP Addresses

To ensure that your DGX system can access the network interfaces for Docker containers, Docker should be configured to use a subnet distinct from other network resources used by the DGX system.

By default, Docker uses the 172.17.0.0/16 subnet. Consult your network administrator to find out which IP addresses are used by your network. If your network does not conflict with the default Docker IP address range, no changes are needed and you can skip this section. However, if your network uses the addresses in this range for the DGX system, you should change the default Docker network addresses.

You can change the default Docker network addresses by modifying the `/etc/docker/daemon.json` file or modifying the `/etc/systemd/system/docker.service.d/docker-override.conf` file. These instructions provide an example of modifying the `/etc/systemd/system/docker.service.d/docker-override.conf` to override the default Docker network addresses.

1. Open the `docker-override.conf` file for editing.

```
sudo vi /etc/systemd/system/docker.service.d/docker-override.conf
[Service]
ExecStart=
ExecStart=/usr/bin/dockerd -H fd:// -s overlay2
LimitMEMLOCK=infinity
LimitSTACK=67108864
```

2. Make the changes indicated in bold below, setting the correct bridge IP address and IP address ranges for your network.

Consult your IT administrator for the correct addresses.

```
[Service]
ExecStart=
ExecStart=/usr/bin/dockerd -H fd:// -s overlay2 --bip=192.168.127.1/24
--fixed-cidr=192.168.127.128/25
LimitMEMLOCK=infinity
LimitSTACK=67108864
```

3. Save and close the `/etc/systemd/system/docker.service.d/docker-override.conf` file.
4. Reload the `systemctl` daemon.

```
sudo systemctl daemon-reload
```

5. Restart Docker.

```
sudo systemctl restart docker
```

10.3.4. Connectivity Requirements for NGC Containers

To run NVIDIA NGC containers from the NGC container registry, your network must be able to access the following URLs:

For x86_64 systems:

- <http://archive.ubuntu.com/ubuntu/>

- ▶ <http://security.ubuntu.com/ubuntu/>
- ▶ http://repo.download.nvidia.com/baseos/ubuntu/noble/x86_64/ (You can access this URL only by using `apt-get` but not in the browser)
- ▶ https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2404/x86_64/
- ▶ `nvcr.io` (NVIDIA Container Registry)

For ARM64 systems:

- ▶ <http://ports.ubuntu.com/ubuntu-ports>
- ▶ <http://repo.download.nvidia.com/baseos/ubuntu/noble/arm64/> (You can access this URL only by using `apt-get` but not in the browser)
- ▶ <https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2404/sbsa/>
- ▶ `nvcr.io` (NVIDIA Container Registry)

To verify connection to `nvcr.io`, run

```
wget https://nvcr.io/v2
```

You should see connecting verification followed by a 401 error:

```
--2018-08-01 19:42:58-- https://nvcr.io/v2
Resolving nvcr.io (nvcr.io) --> 52.8.131.152, 52.9.8.8
Connecting to nvcr.io (nvcr.io)|52.8.131.152|:443. --> connected.
HTTP request sent, awaiting response. --> 401 Unauthorized
```

10.3.5. Configuring Static IP Addresses for the Network Ports

Here are the steps to configure static IP addresses for network ports.

During the initial boot set up process for your DGX system, one of the steps was to configure static IP addresses for a network interface. If you did not configure the addresses at that time, you can configure the static IP addresses from the Ubuntu command line using the following instructions.

Note

If you are connecting to the DGX console remotely, connect by using the BMC remote console. If you connect using SSH, your connection will be lost when you complete the final step. Also, if you encounter issues with the configuration file, the BMC connection will help with troubleshooting.

If you cannot remotely access the DGX system, connect a display with a 1440x900 or lower resolution, and a keyboard directly to the DGX system.

1. Determine the port designation that you want to configure, based on the physical Ethernet port that you have connected to your network.

If your network needs to use a proxy server, you need to set up configuration files to ensure the DGX system communicates through the proxy.

Refer to [Configuring Network Proxies](#) for the port designation of the connection that you want to configure.

2. Edit the network configuration YAML file, `/etc/netplan/01-netcfg.yaml`, and make the following edits.

Note

Ensure that your file is identical to the following sample and use spaces and not tabs.

```
# This file describes the network interfaces available on your system
# For more information, see netplan(5).
```

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp226s0:
      dhcp4: no
      addresses:
        - 10.10.10.2/24
      routes:
        - to: default
          via: 10.10.10.1
      nameservers:
        addresses: [ 8.8.8.8 ]
```

Consult your network administrator for your site-specific values such as network, gateway, and nameserver addresses. Replace `enp226s0` with the designations that you determined in the preceding step.

3. Save the file.
4. Apply the changes.

```
sudo netplan apply
```

Note

If you are not returned to the command line prompt after a information, see [Changes, errors, and bugs](#) in the *Ubuntu Server Guide*.

10.4. Managing CPU Mitigations

DGX OS software includes security updates to mitigate CPU speculative side-channel vulnerabilities. These mitigations can decrease the performance of deep learning and machine learning workloads.

If your DGX system installation incorporates other measures to mitigate these vulnerabilities, such as measures at the cluster level, you can disable the CPU mitigations for individual DGX nodes and increase performance.

10.4.1. Determining the CPU Mitigation State of the DGX System

Here is information about how you can determine the CPU mitigation state of your DGX system.

If you do not know whether CPU mitigations are enabled or disabled, issue the following.

```
cat /sys/devices/system/cpu/vulnerabilities/*
```

CPU mitigations are **enabled** when the output consists of multiple lines prefixed with **Mitigation:**.

For example:

```
KVM: Mitigation: Split huge pages
Mitigation: PTE Inversion; VMX: conditional cache flushes, SMT vulnerable
→Mitigation: Clear CPU buffers; SMT vulnerable
Mitigation: PTI
Mitigation: Speculative Store Bypass disabled via prctl and seccomp
→Mitigation: usercopy/swapgs barriers and __user pointer sanitization
→Mitigation: Full generic retpoline, IBPB: conditional, IBRS_FW, STIBP:
conditional, RSB filling
Mitigation: Clear CPU buffers; SMT vulnerable
```

CPU mitigations are **disabled** if the output consists of multiple lines prefixed with **Vulnerable**.

```
KVM: Vulnerable
Mitigation: PTE Inversion; VMX: vulnerable Vulnerable; SMT vulnerable
Vulnerable
Vulnerable
Vulnerable: user pointer sanitization and usercopy barriers only; no swapgs
→barriers
Vulnerable, IBPB: disabled, STIBP: disabled Vulnerable
```

10.4.2. Disabling CPU Mitigations

Here are the steps to disable CPU mitigations.

Caution

Performing the following instructions will disable the CPU mitigations provided by the DGX OS software.

1. Install the `nv-mitigations-off` package.

```
sudo apt install nv-mitigations-off -y
```

2. Reboot the system.
3. Verify that the CPU mitigations are disabled.

```
cat /sys/devices/system/cpu/vulnerabilities/*
```

The output should include several vulnerable lines. See [Determining the CPU Mitigation State of the DGX System](#) Here is information about how you can determine the CPU mitigation state of your DGX system example output.

10.4.3. Re-enable CPU Mitigations

Here are the steps to enable CPU mitigations again.

1. Remove the nv-mitigations-off package.

```
sudo apt purge nv-mitigations-off
```

2. Reboot the system.
3. Verify that the CPU mitigations are enabled.

```
cat /sys/devices/system/cpu/vulnerabilities/*
```

The output should include several Mitigations lines. See [Determining the CPU Mitigation State of the DGX System](#) for example output.

10.5. Performance Impact of Enabling init_on_alloc

The CONFIG_INIT_ON_ALLOC_DEFAULT_ON Linux kernel configuration option controls whether the kernel fills newly allocated pages and heap objects with zeroes by default. You can override this setting with the `init_on_alloc=<0|1>` kernel parameter. The DGX OS preinstalled on NVIDIA DGX System sets `init_on_alloc=1` because this setting is the recommended default by Ubuntu for kernel hardening. However, this setting can have a performance impact on the network interface controller performance on DGX systems because zeroing every buffer page upon allocation is frequent and requires time to complete. This option can impact the performance with the inbox OFED driver more than Mellanox OFED (MOFED) driver. The MOFED driver allocates a much larger page cache, which tolerates the increased kernel cost of zeroing pages better. NVIDIA recommends keeping the default setting, `init_on_alloc=1`, for best security. If your deployment permits less strict security and the network interface controller is underperforming, you can disable the security feature.

1. Edit the `/etc/default/grub` file and add `init_on_alloc=0` to the `GRUB_CMDLINE_LINUX_DEFAULT` variable.
2. Generate the GRUB bootloader.

```
sudo update-grub
sudo reboot
```

3. Optional: After the system reboots, verify the change took effect.

```
cat /proc/cmdline
```

Example output:

```
BOOT_IMAGE=/boot/vmlinuz-... init_on_alloc=0
```

10.6. Managing the DGX Crash Dump Feature

This section provides information about managing the DGX Crash Dump feature. You can use the script that is included in the DGX OS to manage this feature.

10.6.1. Using the Script

Here are commands that help you complete the necessary tasks with the script.

- To enable only dmesg crash dumps, run:

```
/usr/sbin/nvidia-kdump-config enable-dmesg-dump
```

This option reserves memory for the crash kernel.

- To enable both dmesg and vmcore crash dumps, run:

```
/usr/sbin/nvidia-kdump-config enable-vmcore-dump
```

This option reserves memory for the crash kernel.

- To disable crash dumps, run:

```
/usr/sbin/nvidia-kdump-config disable
```

This option disables the use of kdump and ensures that no memory is reserved for the crash kernel.

10.7. Connecting to Serial Over LAN

You can connect to serial over a LAN.

Note

This feature applies only to DGX systems that have the BMC.

While dumping vmcore, the BMC screen console goes blank approximately 11 minutes after the crash dump is started. To view the console output during the crash dump, connect to serial over LAN as follows:

```
ipmitool -I lanplus -H <BMC-IP> -U <username> -P <password> sol activate
```

10.8. Filesystem Quotas

Here is some information about filesystem quotas.

When running NGC containers you might need to limit the amount of disk space that is used on a filesystem to avoid filling up the partition. Refer to [How to Set Filesystem Quotas on Ubuntu 18.04](#) about how to set filesystem quotas on Ubuntu 18.04 and later.

10.9. Running Workloads on Systems with Mixed Types of GPUs

The DGX Station A100 comes equipped with four high-performance NVIDIA A100 GPUs and one DGX Display GPU. The NVIDIA A100 GPU is used to run high performance and AI workloads, and the DGX Display card is used to drive a high-quality display on a monitor.

When running applications on this system, it is important to identify the best method to launch applications and workloads to make sure the high performance NVIDIA A100 GPUs are used. You can achieve this in one of the following ways:

- ▶ *Running with Docker Containers*
- ▶ *Running on Bare Metal*
- ▶ *Using Multi-Instance GPUs*

When you log into the system and check which GPUs are available, you find the following:

```
nvidia-smi -L
GPU 0: Graphics Device (UUID: GPU-269d95f8-328a-08a7-5985-ab09e6e2b751)
GPU 1: Graphics Device (UUID: GPU-0f2dff15-7c85-4320-da52-d3d54755d182)
GPU 2: Graphics Device (UUID: GPU-dc598de6-dd4d-2f43-549f-f7b4847865a5)
GPU 3: DGX Display (UUID: GPU-91b9d8c8-e2b9-6264-99e0-b47351964c52)
GPU 4: Graphics Device (UUID: GPU-e32263f2-ae07-f1db-37dc-17d1169b09bf)
```

A total of five GPUs are listed by `nvidia-smi`. This is because `nvidia-smi` is including the DGX Display GPU that is used to drive the monitor and high-quality graphics output.

When running an application or workload, the DGX Display GPU can get in the way because it does not have direct NVlink connectivity, sufficient memory, or the performance characteristics of the NVIDIA A100 GPUs that are installed on the system. As a result you should ensure that the correct GPUs are being used.

10.9.1. Running with Docker Containers

On the DGX OS, because Docker has already been configured to identify the high performance NVIDIA A100 GPUs and assign the GPUs to the container, this method is the simplest.

A simple test is to run a small container with the `[-gpus all]` flag in the command and once in the container that is running `nvidia-smi`. The output shows that only the high-performance GPUs are available to the container:

```
docker run --gpus all --rm -it ubuntu nvidia-smi -L
GPU 0: Graphics Device (UUID: GPU-269d95f8-328a-08a7-5985-ab09e6e2b751)
GPU 1: Graphics Device (UUID: GPU-0f2dff15-7c85-4320-da52-d3d54755d182)
GPU 2: Graphics Device (UUID: GPU-dc598de6-dd4d-2f43-549f-f7b4847865a5)
GPU 3: Graphics Device (UUID: GPU-e32263f2-ae07-f1db-37dc-17d1169b09bf)
```

This step will also work when the `--gpus n` flag is used, where `n` can be 1, 2, 3, or 4. These values represent the number of GPUs that should be assigned to that container. For example:

```
docker run --gpus 2 --rm -it ubuntu nvidia-smi -L
GPU 0: Graphics Device (UUID: GPU-269d95f8-328a-08a7-5985-ab09e6e2b751)
GPU 1: Graphics Device (UUID: GPU-0f2dff15-7c85-4320-da52-d3d54755d182)
```

In this example, Docker selected the first two GPUs to run the container, but if the device option is used, you can specify which GPUs to use:

```
docker run --gpus '"device=GPU-dc598de6-dd4d-2f43-549f-f7b4847865a5,GPU-
↪e32263f2-ae07-f1db-37dc-17d1169b09bf"' --rm -it ubuntu nvidia-smi -L
GPU 0: Graphics Device (UUID: GPU-dc598de6-dd4d-2f43-549f-f7b4847865a5)
GPU 1: Graphics Device (UUID: GPU-e32263f2-ae07-f1db-37dc-17d1169b09bf)
```

In this example, the two GPUs that were not used earlier are now assigned to run on the container.

10.9.2. Running on Bare Metal

To run applications by using the four high performance GPUs, the `CUDA_VISIBLE_DEVICES` variable must be specified **before** you run the application.

Note

This method does not use containers.

CUDA orders the GPUs by performance, so GPU 0 will be the highest performing GPU, and the last GPU will be the slowest GPU.

Warning

`CUDA_DEVICE_ORDER` variable is set to `PCI_BUS_ID`, this ordering will be overridden.

NVIDIA provides an open-source utility, `nvbandwidth`, to measure bandwidth on NVIDIA GPUs.

For information on downloading, building, and running `nvbandwidth`, refer to [NVIDIA/nvbandwidth](#).

10.10. Using Multi-Instance GPUs

Multi-Instance GPUs (MIG) is available on NVIDIA A100 GPUs. If MIG is enabled on the GPUs, and if the GPUs have already been partitioned, then applications can be limited to run on these devices.

This works for both Docker containers and for bare metal using the `[CUDA_VISIBLE_DEVICES]` as shown in the examples below. For instructions on how to configure and use MIG, refer to the [NVIDIA Multi-Instance GPU User Guide](#).

Identify the MIG instances that will be used. Here is the output from a system that has GPU 0 partitioned into 7 MIGs:

```
nvidia-smi -L
GPU 0: Graphics Device (UUID: GPU-269d95f8-328a-08a7-5985-ab09e6e2b751)
  MIG 1g.10gb Device 0: (UUID: MIG-GPU-269d95f8-328a-08a7-5985-ab09e6e2b751/7/
↪0)
  MIG 1g.10gb Device 1: (UUID: MIG-GPU-269d95f8-328a-08a7-5985-ab09e6e2b751/8/
↪0)
  MIG 1g.10gb Device 2: (UUID: MIG-GPU-269d95f8-328a-08a7-5985-ab09e6e2b751/9/
↪0)
```

(continues on next page)

(continued from previous page)

```

MIG 1g.10gb Device 3: (UUID: MIG-GPU-269d95f8-328a-08a7-5985-ab09e6e2b751 /
→11/0)
MIG 1g.10gb Device 4: (UUID: MIG-GPU-269d95f8-328a-08a7-5985-ab09e6e2b751 /
→12/0)
MIG 1g.10gb Device 5: (UUID: MIG-GPU-269d95f8-328a-08a7-5985-ab09e6e2b751 /
→13/0)
MIG 1g.10gb Device 6: (UUID: MIG-GPU-269d95f8-328a-08a7-5985-ab09e6e2b751 /
→14/0)
GPU 1: Graphics Device (UUID: GPU-0f2dff15-7c85-4320-da52-d3d54755d182)
GPU 2: Graphics Device (UUID: GPU-dc598de6-dd4d-2f43-549f-f7b4847865a5)
GPU 3: DGX Display (UUID: GPU-91b9d8c8-e2b9-6264-99e0-b47351964c52)
GPU 4: Graphics Device (UUID: GPU-e32263f2-ae07-f1db-37dc-17d1169b09bf)

```

In Docker, enter the MIG UUID from this output, in which GPU 0 and Device 0 have been selected.

If you are running on DGX Station A100, restart the `nv-docker-gpus` and `docker` system services any time MIG instances are created, destroyed or modified by running the following:

```
sudo systemctl restart nv-docker-gpus; sudo systemctl restart docker
```

`nv-docker-gpus` has to be restarted on DGX Station A100 because this service is used to mask the available GPUs that can be used by Docker. When the GPU architecture changes, the service needs to be refreshed.

```

docker run --gpus '"device=MIG-GPU-269d95f8-328a-08a7-5985-ab09e6e2b751/7/0"'
→--rm -it ubuntu nvidia-smi -L
GPU 0: Graphics Device (UUID: GPU-269d95f8-328a-08a7-5985-ab09e6e2b751)
MIG 1g.10gb Device 0: (UUID: MIG-GPU-269d95f8-328a-08a7-5985-ab09e6e2b751/7/
→0)

```

On bare metal, specify the MIG instances:

Note

This application measures the communication across GPUs, and it is not relevant to read the bandwidth and latency with only one GPU MIG.

The purpose of this example is to illustrate how to use specific GPUs with applications, which is illustrated below.

1. Go to the following directory:

```

cd /path/to/cuda-samples/build/Samples/5_Domain_Specific/
→p2pBandwidthLatencyTest/

```

Note

You can find the source code for CUDA samples in [CUDA Samples](#) on GitHub.

2. Run the `p2pBandwidthLatencyTest`:


```
CUDA_VISIBLE_DEVICES=MIG-GPU-269d95f8-328a-08a7-5985-ab09e6e2b751/7/0 ./
→p2pBandwidthLatencyTest
[P2P (Peer-to-Peer) GPU Bandwidth Latency Test]
Device: 0, Graphics Device MIG 1g.10gb, pciBusID: 1, pciDeviceID: 0,
→pciDomainID:0
```

Note

If a device does not have P2P access to other one, it falls back to normal memcopy procedure. Therefore, you can see lesser Bandwidth (GB/s) and unstable Latency (us) in those cases.

P2P Connectivity Matrix

```
D\D    0
0      1
```

Unidirectional P2P=Disabled Bandwidth Matrix (GB/s)

```
D\D    0
0 176.20
```

Unidirectional P2P=Enabled Bandwidth (P2P Writes) Matrix (GB/s)

```
D\D    0
0 187.87
```

Bidirectional P2P=Disabled Bandwidth Matrix (GB/s)

```
D\D    0
0 190.77
```

Bidirectional P2P=Enabled Bandwidth Matrix (GB/s)

```
D\D    0
0 190.53
```

P2P=Disabled Latency Matrix (us)

```
GPU    0
0      3.57
```

```
CPU    0
0      4.07
```

P2P=Enabled Latency (P2P Writes) Matrix (us)

```
GPU    0
0      3.55
```

```
CPU    0
0      4.07
```

Note

The CUDA Samples are not meant for performance measurements. Results may vary when GPU Boost is enabled.

10.11. Updating the containerd Override File for MIG configurations

When you add MIG instances, the containerd override file does not automatically get updated, and the new MIG instances that you add will not be added to the allow file. When DGX Station A100 starts, after the `nv-docker-gpus` service runs, a containerd override file is created in the `/etc/systemd/system/containerd.service.d/` directory.

Note

This file blocks Docker from using the display GPU on the DGX Station A100.

Here is an example of an override file:

```
[Service]
DeviceAllow=/dev/nvidia1
DeviceAllow=/dev/nvidia2
DeviceAllow=/dev/nvidia3
DeviceAllow=/dev/nvidia4
DeviceAllow=/dev/nvidia-caps/nvidia-cap1
DeviceAllow=/dev/nvidia-caps/nvidia-cap2
DeviceAllow=/dev/nvidiactl
DeviceAllow=/dev/nvidia-modeset
DeviceAllow=/dev/nvidia-uvm
DeviceAllow=/dev/nvidia-uvm-tools
```

The service can only add devices of which it is aware. To ensure that your new MIG instances are added to the allow list, complete the following steps:

1. To refresh the override file, run the following commands:

```
sudo systemctl restart nv-docker-gpus
```

```
sudo systemctl restart docker
```

2. Verify that your new MIG instances are now allowed in the containers. Here is an example of an updated override file:

```
| ID ID Dev | BAR1-Usage | SM  Unc| CE ENC DEC
→OFA JPG | | | ECC|
| | | | |
→ | | | | |
+-----+-----+-----+-----+
| 0  0  0  0 | 0MiB / 81252MiB | 98  0| 7  0  5  1
→1  | | | | |
| | | | | 1MiB / 13107... | |
→ | | | | |
+-----+-----+-----+-----+
→----+
| Processes:
→ |
```

(continues on next page)

(continued from previous page)

GPU	GI	CI	PID	Type	Process name	GPU Memory
↪						
	ID	ID	Usage			
↪						
=====						
No running processes found						
↪						

10.12. Data Storage Configuration

By default, the DGX system includes several drives in a RAID 0 configuration. These drives are intended for application caching, so you

10.12.1. Using Data Storage for NFS Caching

This section provides information about how you can use data storage for NFS caching.

The DGX systems use `cachefilesd` to manage NFS caching.

- ▶ Ensure that you have an NFS server with one or more exports with data that will be accessed by the DGX system
- ▶ Ensure that there is network access between the DGX system and the NFS server.

10.12.2. Using `cachefilesd`

Here are the steps that describe how you can mount the NFS on the DGX system, and how you can cache the NFS by using the DGX SSDs for improved performance.

1. Configure an NFS mount for the DGX system.
 - a. Edit the filesystem tables configuration.

```
sudo vi /etc/fstab
```

- b. Add a new line for the NFS mount by using the local `/mnt` local mount point.

```
<nfs_server>:<export_path> /mnt nfs rw,noatime,rsize=32768,  
↪wsize=32768,nolock,tcp,intr,fsc,nofail 0 0
```

Here, `/mnt` is used as an example mount point.

- ▶ Contact your Network Administrator for the correct values for `<nfs_server>` and `<export_path>`.
 - ▶ The `nfs` arguments presented here are a list of recommended values based on typical use cases. However, `fsc` must always be included because that argument specifies using FS-Cache
- c. Save the changes.
 2. Verify that the NFS server is reachable.

```
ping <nfs_server>
```

Use the server IP address or the server name that was provided by your network administrator.

3. Mount the NFS export.

```
sudo mount /mnt
```

/mnt is an example mount point.

4. Verify that caching is enabled.

```
cat /proc/fs/nfsfs/volumes
```

5. In the output, find FSC=yes.

The NFS will be automatically mounted and cached on the DGX system in subsequent reboot cycles.

10.12.3. Disabling cachefilesd

Here is some information about how to disable cachefilesd.

If you do not want to enable cachefilesd,

- Stop the cachefilesd service.

```
sudo systemctl stop cachefilesd
```

- Disable the cachefilesd service permanently.

```
sudo systemctl disable cachefilesd
```

10.12.4. Changing the RAID Configuration for Data Drives

Here is information that describes how to change the RAID configuration for your data drives.

Caution

You must have a minimum of two drives to complete these tasks. If you have less than two drives, you cannot complete the tasks.

From the factory, the RAID level of the DGX RAID array is RAID 0. This level provides the maximum storage capacity, but it does not provide redundancy. If one SSD in the array fails, the data that is stored on the array is lost. If you are willing to accept reduced capacity in return for a level of protection against drive failure, you can change the level of the RAID array to RAID 5.

Note

If you change the RAID level from RAID 0 to RAID 5, the total storage capacity of the RAID array is reduced.

Before you change the RAID level of the DGX RAID array, back up the data on the array that you want to preserve. When you change the RAID level of the DGX RAID array, the data that is stored on the array is erased.

You can use the `configure_raid_array.py` custom script, which is installed on the system to change the level of the RAID array without unmounting the RAID volume.

- To change the RAID level to RAID 5, run the following command:

```
sudo configure_raid_array.py -m raid5
```

After you change the RAID level to RAID 5, the RAID array is rebuilt. Although a RAID array that is being rebuilt is online and ready to be used, a check on the health of the DGX system reports the status of the RAID volume as unhealthy. The time required to rebuild the RAID array depends on the workload on the system. For example, on an idle system, the rebuild might be completed in 30 minutes.

- To change the RAID level to RAID 0, run the following command:

```
sudo configure_raid_array.py -m raid0
```

To confirm that the RAID level was changed, run the `lsblk` command. The entry in the TYPE column for each drive in the RAID array indicates the RAID level of the array.

10.13. Running NGC Containers

This section provides information about how to run NGC containers with your DGX system.

10.13.1. Obtaining an NGC Account

Here is some information about how you can obtain an NGC account.

NVIDIA NGC provides simple access to GPU-optimized software for deep learning, machine learning, and high-performance computing (HPC). An NGC account grants you access to these tools and gives you the ability to set up a private registry to manage your customized software.

If you are the organization administrator for your DGX system purchase, work with NVIDIA Enterprise Support to set up an NGC enterprise account. Refer to the [NGC Private Registry User Guide](#) for more information about getting an NGC enterprise account.

10.13.2. Running NGC Containers with GPU Support

To obtain the best performance when running NGC containers on DGX systems, you can use one of the following methods to provide GPU support for Docker containers:

- Native GPU support (included in Docker 19.03 and later, installed)
- NVIDIA Container Runtime for Docker

This is in the `nvidia-docker2` package.

The recommended method for DGX OS 7 is native GPU support. To run GPU-enabled containers, run `docker run --gpus`.

Here is an example that uses all GPUs:

```
docker run --gpus all ...
```

Here is an example that uses 2 GPUs:

```
docker run --gpus 2 ...
```

Here is an example that uses specific GPUs:

```
docker run --gpus '"device=1,2"' ...
```

```
docker run --gpus '"device=UUID-ABCDEF-
```

Refer to [Running Containers](#) for more information about running NGC containers on MIG devices.

Chapter 11. Managing Self-Encrypting Drives

The NVIDIA DGX OS software supports the ability to manage self-encrypting drives (SEDs), including setting an Authentication Key for locking and unlocking the drives on NVIDIA DGX B300, DGX B200, DGX H100/H200, DGX A100, and DGX A800 platforms.

You can manage only the SED data drives. The software cannot be used to manage OS drives even if they are SED-capable.

11.1. Overview

The SED management software is in the `nv-disk-encrypt` package.

The software supports the following configurations:

- ▶ NVIDIA DGX B300, DGX B200, DGX H100/H200, DGX A100, and DGX A800 systems where all data drives are self-encrypting drives.
- ▶ Only SEDs used as data drives are supported.

The software will not manage SEDs that are OS drives.

The software provides the following functionality:

- ▶ Identifies eligible drives on the system.
- ▶ Allows you to assign Authentication Keys (passwords) for each SED as part of the initialization process.
 - ▶ Alternatively, the software can generate random passwords for each drive.
 - ▶ The passwords are stored in a password-protected vault on the system.
- ▶ Once initialized, SEDs are locked upon power loss, such as a system shutdown or drive removal. Locked drives get unlocked after power is restored and the root file system is mounted.
- ▶ Provides functionality to export the vault.
- ▶ Provides functionality for erasing the drives.
- ▶ Provides the ability to revert the initialization.

11.2. Installing the Software

Use the package manager to install the `nv-disk-encrypt` package and, optionally, the TPM2 tools package, and reboot the system. You need the TPM tools package if you plan to use the TPM2 to store security keys.

1. Update the packages.

```
sudo apt update
```

2. Install `nv-disk-encrypt`.

```
sudo apt install -y nv-disk-encrypt
```

3. (Optional) Install the TPM tools package.

- For DGX A100, DGX Station A100, DGX H100/H200, DGX B200, or DGX B300, install the `tpm2-tools` package.

```
sudo apt install -y tpm2-tools
```

- For DGX-2, install the `tpm-tools` package.

```
sudo apt install -y tpm-tools
```

4. Reboot.

```
sudo reboot
```

If you plan to use TPM2, enable it. Refer to [Configuring Trusted Computing](#) for more information.

11.3. Configuring Trusted Computing

Here is some information about the controls that are required to configure Trusted Computing (TC).

The DGX system BIOS provides setup controls for configuring the following TC features:

- Trusted Platform Module

NVIDIA DGX B300, DGX B200, DGX H100/H200, DGX A100, and DGX A800 platforms incorporate Trusted Platform Module 2.0 (TPM 2.0). These modules can be enabled from the system BIOS and used in conjunction with the `nv-disk-encrypt` tool. After being enabled, the `nv-disk-encrypt` tool uses the TPM for encryption and stores the vault and SED authentication keys on the TPM instead of on the file system. Using the TPM is preferred because this allows the vault data to persist even if the system is reimaged.

- Block SID

Certain drives shipped with the DGX systems support the Block SID authentication feature. Block SID authentication prevents malicious actors from taking ownership of drives and blocks others from using the drives. By default, the DGX BIOS will send the Block SID request. On such setups, you will need to enable the Disable Block Sid feature in the BIOS before proceeding with the initialization steps.

Note

Enabling the “Disable Block SID” option is only valid for one reboot, so if drive encryption needs to be enabled again, then the feature needs to be enabled in BIOS as well.

11.3.1. Determining Whether Drives Support SID

The drive model is a good indicator of whether the drive supports this feature. Issue the following and look for one of the following model strings:

- ▶ KCM6DRUL3T84
- ▶ KCM6DRUL7T68
- ▶ MZQLB7T6HMLA-00007

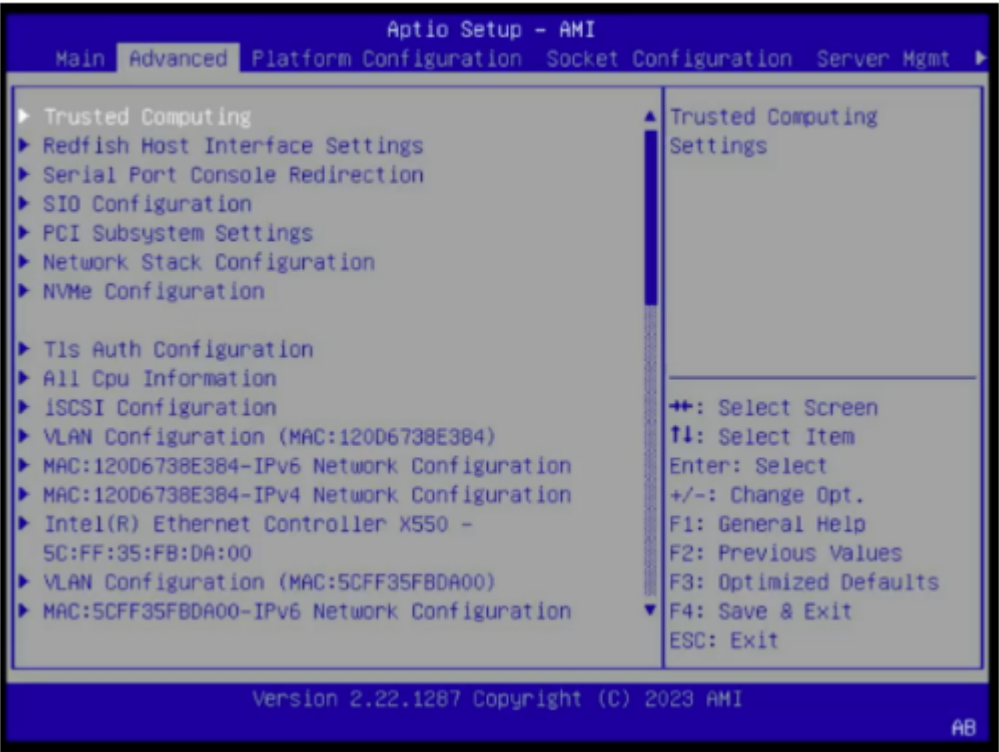
```
sudo nvme list
```

Node	SN	Model
/dev/nvme0n1	70H0A0AHTTHR	KCM6DRUL3T84 ...
/dev/nvme1n1	70H0A007TTHR	KCM6DRUL3T84

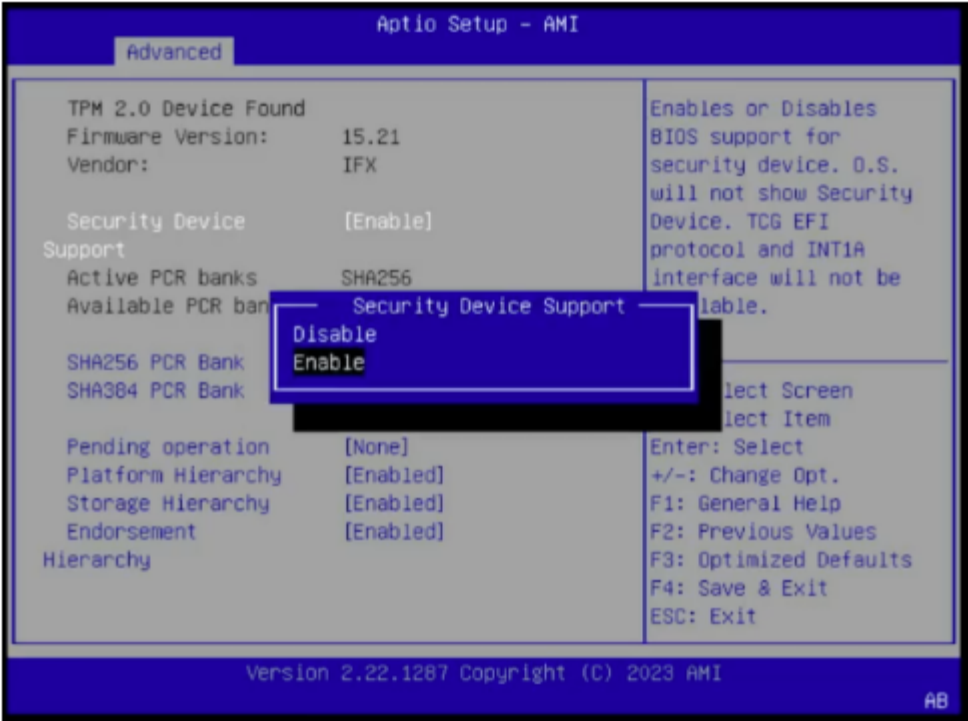
11.3.2. Enabling the TPM and Preventing the BIOS from Sending Block SID Requests

This section provides instructions to enable the TPM and prevent the SBIOS from sending Block SID requests. Each task is independent, so you can select which task to complete.

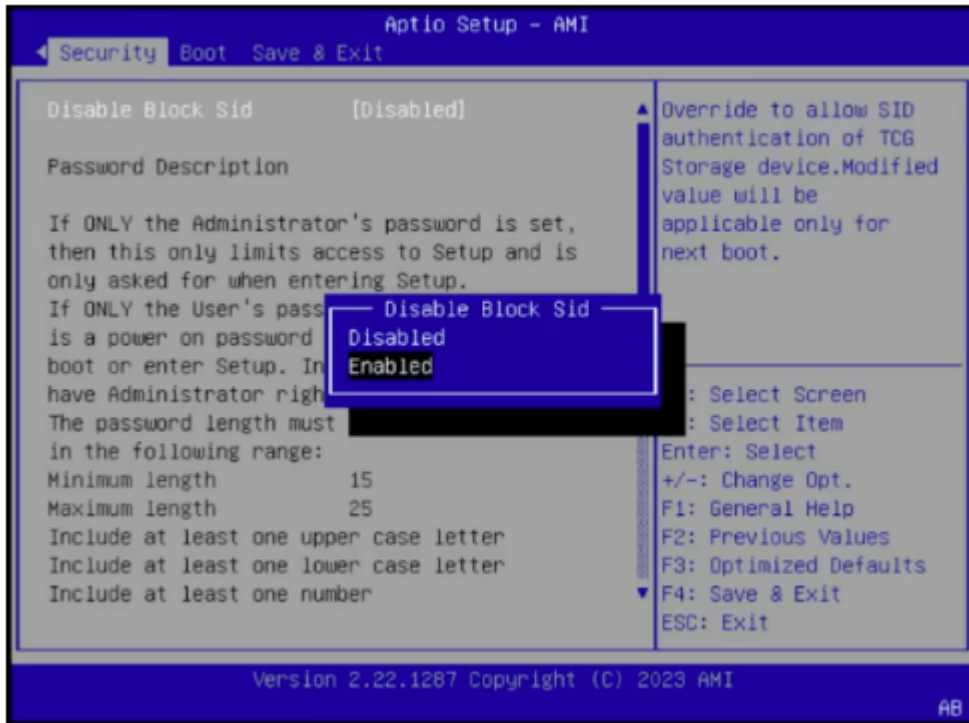
1. Reboot the system, then press [Del] or [F2] at the NVIDIA splash screen to enter the BIOS Setup.
2. Navigate to the **Advanced** tab on the top menu, then scroll to **Trusted Computing** and press [Enter].



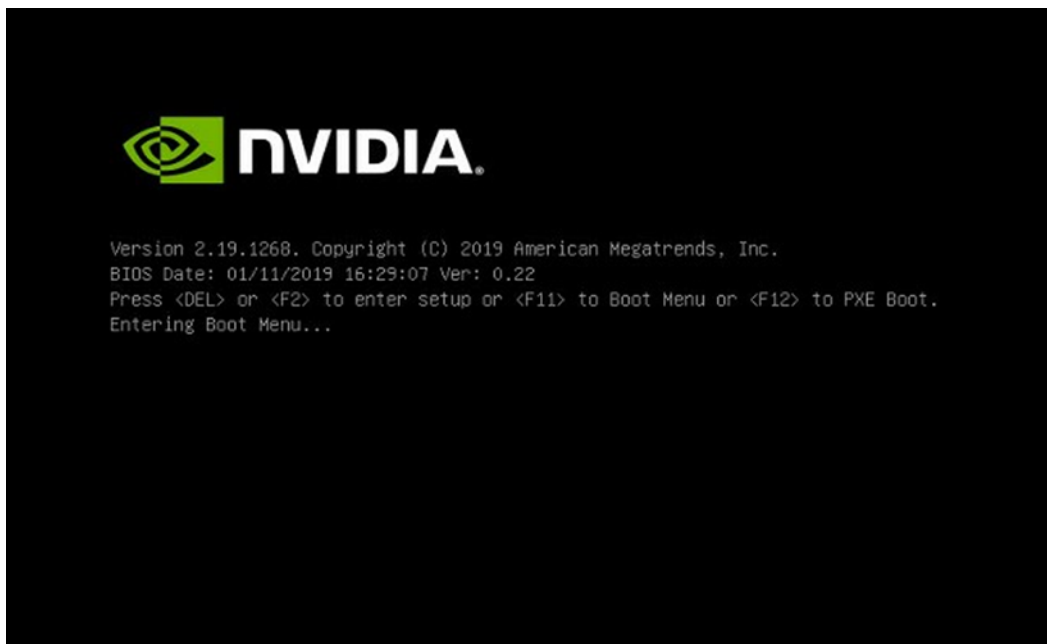
- ▶ To enable TPM, scroll to **Security Device Support** and switch the setting to **Enabled**.



- ▶ To disable Block SID, navigate to the **Security** tab and scroll to **Disable Block Sid**, then switch to **Enabled**.



- When changing the BIOS settings, make sure to select the **Save & Exit** option to continue the boot process.



- If the BIOS is modified to the **Enabled** setting for the **Security Device Support** or **Disable Block Sid** option and the following messages do not appear automatically after reboot:

Press F10 to accept this request
Press ESC to reject this request and continue

Press F10 at the prompt: Press or <F2> to enter setup or <F11> to Boot Menu

or <F12> to PXE Boot.

The messages as mentioned above appear.



5. Press F10 to accept the BIOS settings and continue to configure SED or disk encryption.

11.4. Initializing the System for Drive Encryption

Here is some information about how to initialize the system for drive encryption.

Note

Before initializing drive encryption, review the information in [Configuring Trusted Computing](#) and follow the configuration instructions if needed.

Initialize the system for drive encryption using the `nv-disk-encrypt` command.

```
sudo nv-disk-encrypt init [-k <your-vault-password>] [-f <path/to/json-file>]
→ [-g] [-r]
```

Here is a list of the options:

- **-k:** Lets you create the vault password in the command.
Otherwise, the software will prompt you to create a password before proceeding.
- **-f:** Lets you specify a JSON file that contains a mapping of passwords to drives.
Refer to “Example 1: Passing in the JSON File” for further instructions.

- **-g**: Generates random salt values (stored in `/etc/nv-disk-encrypt/.dgxenc.salt`) for each drive password.

Salt values are characters added to a password for enhanced password security. NVIDIA strongly recommends using this option for best security, otherwise the software will use a default salt value instead of a randomly generated one.

- **-r**: Generates random passwords for each drive.

This avoids the need to create a JSON file or the need to enter a password one by one during the initialization.

11.5. Enabling Drive Locking

After initializing the system for SED management, issue the following command, which uses the `nv-disk-encrypt` command to enable drive locking.

```
sudo nv-disk-encrypt lock
```

After initializing the system and enabling drive locking, the drives will become locked when they lose power. The system will automatically unlock each drive when power is restored to the system and the system is rebooted.

11.6. Initialization Examples

This section provides some initialization examples.

11.6.1. Example 1: Passing in the JSON File

The following instructions in this section describe a method to specify the drive/password mapping ahead of time. This method is useful for initializing several drives at a time and avoids the need to enter the password for each drive after issuing the initialization command, or if you want control of the passwords.

Refer to the following for more information:

- *Determining Which Drives Can be Managed as Self-Encrypting*
- *Creating the Drive/Password Mapping JSON Files and Using it to Initialize the System*

11.6.1.1 Determining Which Drives Can be Managed as Self-Encrypting

Here is some information about how you can determine which drives can be managed as self-encrypting.

Review the storage layout of the DGX system to determine which drives are eligible to be managed as SEDs.

```
sudo nv-disk-encrypt info
```

The default output shows which drives can be used for encryption and which drives cannot. The following status information is provided:

- SED capable: Is this a self-encrypting drive?

- ▶ **Boot disk:** Is this drive currently being used as a boot drive? Does it contain the root filesystem?
- ▶ **Locked:** Is this drive currently in the locked state? Is it able to accept I/O?. It can only be in this state after the following conditions have been met:
 - ▶ Locking has been enabled (`nv-disk-encrypt init`, followed by `nv-disk-encrypt lock`)
 - ▶ The drive is coming back from power-off.
 - ▶ The user queries this state prior to it being (automatically) unlocked.
- ▶ **Lock Enabled:** Are locks enabled on this drive? It will be in this state after initialization (`nv-disk-encrypt init`).
- ▶ **MBR done:** This setting is only relevant for drives that support MBR shadowing. On drives that support this feature, this will report 'Y' after initialization (`nv-disk-encrypt init`)

The following example output snippet shows drives that can be used for encryption. Notice SED capable = Y and Boot disk = N.

```
+-----+
| .Name | .Serial | .Status |
+-----+
| ./dev/nvme3n1 | .xxxxx1 | .SED.capable := Y, .Boot.disk := N, .Locked := N, .Lock.Enabled := N, .MBR.done := N |
| ./dev/nvme6n1 | .xxxxx2 | .SED.capable := Y, .Boot.disk := N, .Locked := N, .Lock.Enabled := N, .MBR.done := N |
| ./dev/nvme9n1 | .xxxxx3 | .SED.capable := Y, .Boot.disk := N, .Locked := N, .Lock.Enabled := N, .MBR.done := N |
```

The following example output snippet shows drives that cannot be used for encryption. Notice SED capable = Y and Boot disk = Y, or SED capable = N.

```
+-----+
| .Name | .Serial | .Status |
+-----+
| ./dev/nvme0n1 | .xxxxx1 | .SED.capable := Y, .Boot.disk := Y, .Locked := N, .Lock.Enabled := N, .MBR.done := N |
| ./dev/sr0 | .xxxxx2 | .SED.capable := N, .Boot.disk := N, .Locked := N, .Lock.Enabled := N, .MBR.done := N |
| ./dev/nvme1n1 | .xxxxx3 | .SED.capable := Y, .Boot.disk := Y, .Locked := N, .Lock.Enabled := N, .MBR.done := N |
| ./dev/sda | .unknown | .SED.capable := N, .Boot.disk := N, .Locked := N, .Lock.Enabled := N, .MBR.done := N |
```

Alternatively, you can specify the output be presented in JSON format by using the `-j` option.

```
sudo nv-disk-encrypt info -j
```

In this case, drives that can be used for encryption are indicated by the following:

```
"sed_capable": true "used_for_boot": false
```

And drives that cannot be used for encryption are indicated by one of the following:

```
"sed_capable": true "used_for_boot": true
```

Or

```
"sed_capable": false
```

11.6.1.2 Creating the Drive/Password Mapping JSON Files and Using it to Initialize the System

You can initialize the system by creating the drive and password map the JSON files.

1. Create a JSON file that lists all the eligible SED-capable drives that you want to manage.

Note

These are the list of drives that you obtained from *Determining Which Drives Can be Managed as Self-Encrypting*

The following example shows the format of the JSON file.

```
{
  "/dev/nvme2n1": "<your-password>",
  "/dev/nvme3n1": "<your-password>",
  "/dev/nvme4n1": "<your-password>",
  "/dev/nvme5n1": "<your-password>",
}
```

- Ensure that you follow the syntax exactly.
- Passwords must consist of only upper-case letters, lower-case letters, digits, and/or the following special characters: ~ : @ % ^ + = _ ,

2. Initialize the system and then enable locking.

The following command assumes you have placed the JSON file in the /tmp directory.

```
sudo nv-disk-encrypt init -f /tmp/<your-file>.json -g
sudo nv-disk-encrypt lock
```

When prompted, enter a password for the vault.

Passwords must consist of only upper-case letters, lower-case letters, digits, and/or the following special characters: ~ : @ % ^ + = _ ,

11.6.2. Example 2: Generating Random Passwords

The commands in this topic use the -k and -r options so that you are not prompted to enter passwords. You pass the vault password into the command and then the command instructs the tool to generate random passwords for each drive.

The vault password must consist of only upper-case letters, lower-case letters, digits, and/or the following special-characters: ~ : @ % ^ + = _ ,

```
sudo nv-disk-encrypt init -k <your-vault-password> -g -r
sudo nv-disk-encrypt lock
```

11.6.3. Example 3: Specifying Passwords One at a Time When Prompted

If there are a small number of drives, or you do not want to create a JSON file, issue the following command.

```
sudo nv-disk-encrypt init -g
sudo nv-disk-encrypt lock
```

The software prompts you to enter a password for the vault and then a password for each eligible SED. Passwords must consist of only upper-case letters, lower-case letters, digits, and/or the following special characters: `~ : @ % ^ + = _ ,`

11.7. Disabling Drive Locking

To disable drive locking at any time after you initialize, run the following command: `$ sudo nv-disk-encrypt disable`

This command disables locking on all drives. You can run the initial set up again at any time after this process is complete.

11.8. Enabling Drive Locking

After initializing the system for SED management, issue the following command, which uses the `nv-disk-encrypt` command to enable drive locking.

```
sudo nv-disk-encrypt lock
```

After initializing the system and enabling drive locking, the drives will become locked when they lose power. The system will automatically unlock each drive when power is restored to the system and the system is rebooted.

11.9. Exporting the Vault

Here is some information about how to export the vault.

To export all drive keys out to a file, use the export function. This requires that you pass in the vault password.

```
sudo nv-disk-encrypt export -k yourvaultpassword  
Writing vault data to /tmp/secrets.out
```

11.10. Erasing Your Data

Here is some information about how you can erase your data.

Caution

Be aware that all data will be lost when executing this. On DGX B300, DGX B200, and DGX H100/H200 systems, these drives generally form a RAID 0 array, and this array will be destroyed when you perform an erase.

After initializing the system for SED management, use the `nv-disk-encrypt` command to erase data on your drives after stopping `cache/filesd` and unmounting the RAID array as follows.

1. Completely stop the RAID.

```
systemctl stop cachefilesd
sudo umount /raid
sudo mdadm --stop /dev/md1
```

2. Perform the erase.

```
sudo nv-disk-encrypt erase
```

This command does the following:

- ▶ Sets the drives in an unlocked state.
- ▶ Disables locking on the drives.
- ▶ Removes the RAID 0 array configuration.

To rebuild the RAID array, issue the following command:

```
sudo /usr/bin/configure RAID array.py -c -f
```

11.11. Clearing the TPM

If you've lost the password to your TPM, you will not be able to access its contents. In this case, the only way to regain access to the TPM is to clear the TPM's contents. After clearing the TPM, you will need to re-initialize the vault and SED authentication keys.

1. Reboot the system, then press [Del] or [F2] at the NVIDIA splash screen to enter the BIOS Setup.
2. Navigate to the Advanced tab on the top menu, scroll to Trusted Computing, and press [Enter].
3. Clear TPM2.
 1. Scroll to Trusted Computing and press [Enter].
 2. Scroll to Pending Operation and press [Enter].
 3. Select TPM Clear at the Pending Operation popup and press [Enter].
4. Save and exit the BIOS Setup.

11.12. Changing Disk Passwords, Adding Disks, or Replacing Disks

The same steps are needed for changing or rotating passwords, adding disks, or replacing disks.

1. Disable SED management.

```
sudo nv-disk-encrypt disable
```

2. Add or replace drives as needed and then rebuild the RAID array. Refer to your system's Service Manual for more information.
3. Enable SED management and assign passwords per the instructions in *Initializing the System for Drive Encryption*.

11.13. Recovering From Lost Keys

NVIDIA recommends backing up your keys and storing them in a secure location. If you lost the key used to initialize and lock your drives, you will not be able to unlock the drive again. If this happens, the only way to recover is to perform a factory-reset, which will result in a loss of data.

SED drives come with a PSID printed on the label; this value can only be obtained by physically examining the drive as exemplified in the following image.



Specify the PSID to reset the drive using the following `sedutil-cli` command:

```
sudo sedutil-cli ----yesIreallywanttoERASEALLmydatausingthePSID  
↪yourdrivesPSID /dev/nvme3n1
```

Chapter 12. Resolved Issues

The following issues that were previously identified as known issues have been resolved.

12.1. DGX Station A100 Fails to Boot After Applying MIG Configurations

12.1.1. Issue

After MIG configurations were successfully applied to a DGX station A100 system running DGX OS 7.0.2, the system failed to boot when you ran the `sudo reboot` command. Resetting the GPUs by performing a DC power cycle could not recover the system.

12.1.2. Workaround

The DGX OS 7.0.2 release does not support the DGX Station A100 system with MIG enabled. To resolve the boot failure, install DGX OS 6.3.2 on the system and then apply MIG configurations.

12.1.3. Status

Resolved in GPU driver versions 570.117 and later, as well as 575.20 and later.

12.2. Update the MLNX Firmware for the Connect-X and Bluefield-3 Adapters

12.2.1. Issue

The online network repository for DOCA 2.9.1/Ubuntu 24.04 does not contain the `mlnx_fw_updater` tool, which is needed to update the Connect-X and Bluefield-3 adapters to their latest firmware versions.

12.2.2. Workaround

Install the `mlnx_fw_updater` tool.

```
$ wget https://linux.mellanox.com/public/repo/mlnx_ofed/latest-24.10/ubuntu24.
  ↳ 04/x86_64/mlnx-fw-updater_24.10-1.1.4.0_amd64.deb
$ sudo apt install mlnx-fw-updater_24.10-1.1.4.0_amd64.deb
$ sudo /opt/mellanox/mlnx-fw-updater/mlnx_fw_updater.pl
```

12.2.3. Status

Resolved in version 7.1.0.

12.3. Errors Occur When Loading Mirrored Repositories on Air-Gapped Systems

12.3.1. Issue

When you run the `apt update` command to load mirrored repositories on an air-gapped system, the following error messages appear:

```
File not found - /media/repository/mirror/security.ubuntu.com/ubuntu/dists/
  ↳ jammy-security/main/cnf/Commands-amd64 (2: No such file or directory)
```

```
Failed to fetch file:/media/repository/mirror/security.ubuntu.com/ubuntu/
  ↳ dists/jammy-security/main/cnf/Commands-amd64 File not found - /media/
  ↳ repository/mirror/security.ubuntu.com/ubuntu/dists/jammy-security/main/cnf/
  ↳ Commands-amd64 (2: No such file or directory)
```

12.3.2. Explanation

This issue occurs because a fix for the `apt-mirror` package, which is available in Ubuntu 23.10, has yet to be implemented in the Ubuntu 22.04 repositories. If you are using an `apt-mirror` package

- ▶ Version later than 0.5.4-1: Contact NVIDIA Enterprise Services by filing a support case.
- ▶ Version 0.5.4-1: Use the following workaround to mirror the repositories.

You can run the following command to determine the version of your `apt-mirror` package:

```
$ dpkg -l | grep apt-mirror

ii apt-mirror          0.5.4-1          all          APT
  ↳ sources mirroring tool
```

12.3.3. Status

Resolved in version 7.1.0.

Chapter 13. Known Issues

13.1. NVMe Opcode 0x2 (I/O Cmd) QID 256 Timeout

13.1.1. Issue

On DGX B300 systems with Micron E1.S drives, the following non-destructive issue appears on every boot. This occurs because the NVSM tool periodically checks drive health by running the `mdadm --examine` command, which triggers an error message similar to the following example approximately every 30 seconds. Over time, this might fill up the kernel log.

```
2025-12-22T21:42:38.413108+08:00 53A kernel: nvme nvme2: I/O tag 641 (3281)
↳opcode 0x2 (I/O Cmd) QID 256 timeout, aborting req_op:DRV_IN(34) size:512
2025-12-22T21:42:38.413129+08:00 53A kernel: nvme nvme2: Abort status: 0x4002
2025-12-22T21:43:08.621095+08:00 53A kernel: nvme nvme2: I/O tag 641 (3281)
↳opcode 0x2 (I/O Cmd) QID 256 timeout, reset controller
2025-12-22T21:43:08.734094+08:00 53A kernel: nvme nvme2: D3 entry latency set
↳to 10 seconds
2025-12-22T21:43:08.755075+08:00 53A kernel: nvme nvme2: 256/0/0 default/read/
↳poll queues
```

This issue is related to the Micron E1.S drive firmware. Currently, there is no temporary solution available.

13.2. Upgrade Error During Package Configuration

13.2.1. Issue

When upgrading DGX OS, the upgrade process might fail during package configuration. This failure occurs because the `nvidia-fs` DKMS module has a dependency on the `nvidia` module, which might not be built until later in the upgrade process.

Error message:

```
dpkg: error processing package linux-nvidia-64k-6.14 (--configure):
dependency problems - leaving unconfigured
```

13.2.2. Workaround

To resolve this issue and complete the upgrade, run the following command immediately after the error occurs:

```
sudo apt -f install -y
```

This command will complete the configuration of the `linux-nvidia-64k-6.14` package.

13.3. Error Messages Reported by GPU Driver

13.3.1. Issue

The GPU driver emits error messages about failing to allocate an `NvKmsKapiDevice`:

```
[ 41.633844] [drm:nv_drm_dev_load [nvidia_drm]] *ERROR* [nvidia-drm] [GPU
→ID 0x00080100] Failed to allocate NvKmsKapiDevice
[ 41.656934] [drm:nv_drm_register_drm_device [nvidia_drm]] *ERROR* [nvidia-
→drm] [GPU ID 0x00080100] Failed to load device
```

These messages can be safely disregarded. A resolution will be provided in a future release.

13.4. DGX GB200 System Failure During Upgrade

13.4.1. Issue

Performing a DGX OS upgrade to version 7.2.1 and later can result in DOCA DKMS build errors on DGX GB200 systems. After attempting to reboot, the system cannot boot to the new kernel.

13.4.2. Workaround

Note

If you have rebooted before applying this workaround, the system might not boot to the new kernel. In this situation, use the GRUB menu's **Advanced Options** to select and boot to an older kernel, and then apply the workaround.

To successfully upgrade to DGX OS 7.2.1 on DGX GB200 systems, run the following commands:

1. Re-run DKMS build with the `--force` option against the newly installed kernel.

Be sure to specify your new kernel version, as the one in the example might be outdated if you have recently upgraded. To help identify the new kernel version, use the `ls -v /boot/vmlinuz-*` command to list installed kernels.


```
sudo dkms autoinstall --force -k 6.8.0-1032-nvidia-64k
```

2. Re-configure broken packages.

```
sudo apt -f install -y
```

13.5. RShim Devices Not Created After Starting rshim.service

13.5.1. Issue

RShim devices might not appear under `/dev/rshim*` because the RShim driver might be used by another entity.

13.5.2. Workaround

1. Edit `/etc/rshim.conf`, and uncomment the line that says `#FORCE_MODE 1`
2. Restart the RShim service.

```
sudo systemctl restart rshim.service
```

13.6. Firmware Crash on DGX GB200 During Boot

13.6.1. Issue

On DGX GB200 systems, you might observe a firmware crash during some reboots, resulting in an extra reboot. This can be visible over the serial console.

Log of events seen for this issue:

```
INFO:      th500_ras_intr_handler: External Abort reason=0 syndrome=0xbe000411
↳ flags=0x1
ERROR:      spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:      spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:      spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:      spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:      spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:      spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:      spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:      spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:      spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:      spmd_ffa_direct_message failed (4294967292) on CPU121
```

(continues on next page)

(continued from previous page)

```

ERROR:  spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:  spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:  spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:  spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:  spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:  spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:  spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:  spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:  spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:  spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:  spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:  spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:  spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:  spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:  spmd_ffa_direct_message failed (4294967292) on CPU121
ERROR:  spmd_ffa_direct_message failed (4294967292) on CPU121
ASSERT: plat/nvidia/tegra/soc/th500/plat_ras.c:408
BACKTRACE: START: assert
0: EL3: 0x78732b2648
1: EL3: 0x78732b0294
2: EL3: 0x78732c45fc
3: EL3: 0x78732c481c
4: EL3: 0x78732bda18
5: EL3: 0x78732b9f18
6: EL3: 0x78732b1258
BACKTRACE: END: assert

```

13.6.2. Workaround

Currently, there is no temporary solution available.

13.7. The nvidia_peermem Module Does Not Load After an OTA Update

13.7.1. Issue

After you perform an OTA update, the `nvidia-peermem-loader` package is installed, but the `nvidia_peermem` module is not loaded. As a result, the following error message appears in the `dmesg` log:

```

nvidia_peermem: disagrees about version of symbol ib_register_peer_memory_
↪client
nvidia_peermem: Unknown symbol ib_register_peer_memory_client (err -22)

```

13.7.2. Workaround

This issue occurs because updating the `mlnx-ofed-kernel` package requires rebuilding the `nvidia-peermem` DKMS module to use the new version of the `mlnx-ofed-kernel` package.

The following commands will rebuild the necessary modules:

```
MODULE_VERSION=`dkms status nvidia -k $(uname -r) | cut -d "," -f1` || true
if [ ! -z "${MODULE_VERSION}" ]; then
    sudo dkms remove -m ${MODULE_VERSION} -k $(uname -r) || true
    sudo dkms install -m ${MODULE_VERSION} -k $(uname -r) || true
fi
```

13.8. The systemd-modules-load Service Failed to Insert the nvidia_peermem Module

13.8.1. Issue

On a DGX Station A100 or DGX Station A800, when you install a Base OS version earlier than 7.0.2, the `nvidia-peermem-loader` package might have been installed on the system. As a result, the following error message occurs:

```
$ sudo systemctl status systemd-modules-load.service
(code=exited, status=1/FAILURE)
...
systemd-modules-load[2143]: Failed to insert module 'nvidia_peermem': Invalid
↪ argument
```

13.8.2. Workaround

To avoid this failure, remove the `nvidia-peermem-loader` package.

```
sudo apt purge nvidia-peermem-loader
```

13.9. BMC Redfish Interface Not Active on the First Boot After Installation

13.9.1. Issue

Reported in DGX OS 7.0.0.

When the DGX system was booted the first time after the DGX OS 7.0.0 installation, the BMC Redfish network interface was not renamed or autoconfigured correctly, as reported by the `ip a` command.

13.9.2. Workaround

During the installation of DGX OS 7.0.x, the Redfish interface might not be configured correctly with the proper interface name or IP address. To resolve this issue, run the following command to reconfigure the interface:

```
sudo /usr/sbin/configure-redfish-intf.bash
```

13.10. No Permissions to Access /var/run/nvidia-fabricmanager for Non-Root Users on DGX B200

13.10.1. Issue

After DGX OS 7.0.x is installed on the DGX B200 system, the required access to the /var/run/nvidia-fabricmanager directory by the Fabric Manager service is not set for a non-root account. This can cause failures running the HPL benchmark and NCCL test as a non-root user.

13.10.2. Workaround

Change the permissions on the /var/run/nvidia-fabricmanager/ directory by running the following commands:

```
# If Fabric Manager is running, stop it.
sudo systemctl stop nvidia-fabricmanager.service

# Change the permission setting to 755.
sudo chmod 755 /var/run/nvidia-fabricmanager/

# Start Fabric Manager.
sudo systemctl start nvidia-fabricmanager.service
```

13.11. Kernel OOPS When Activating VFs from SR-IOV Network Operator

13.11.1. Issue

A Kernel OOPS (Out-of-Memory Panic) occurs when activating 16 or more Virtual Functions (VFs) for a single parent interface or Physical Function (PF).

Primary issue impacts:

- ▶ Shell user – Shell users cannot use standard `ip` for network interface troubleshooting on interfaces with 16 VFs. There are no stats, status, etc.
- ▶ CPU cycles/log space - Resource consumption is a concern. The `dmesg` logs roll in minutes or seconds, preventing proper debugging.

Two failures on the network-operator side:

- ▶ `ib-sriov` process failure - This happened once and is reflected in the log of events below.
- ▶ `sriov-network-config-daemon` failure - Happens continuously.

Both failures produce almost similar symptoms. The following failure from `sriov-network-config-daemon` was observed in the `sriov-network-config-daemon` logs:

```
DiscoverSriovDevices(): unable to get Link for device, skipping {"device":
"0000:81:00.0", "error": "message too long"}
```

13.11.2. Workaround

Create fewer than 16 VFs for a PF.

Log of events seen for this issue:

```
[128435.409604] WARNING: CPU: 113 PID: 1061323 at net/core/rtnetlink.c:3867
↪rtnl_getlink+0x43a/0x470
[128435.409610] Modules linked in: xt_set xt_multiport ipt_rpfilter ip_set_
↪hash_net ip_set_hash_ip ip_set veth ipip tunnel4 ip_tunnel wireguard
↪curve25519_x86_64 libchacha20poly1305 chacha_x86_64 poly1305_x86_64
↪libcurve25519_generic libchacha ip6_udp_tunnel udp_tunnel nf_conntrack_
↪netlink xt_addrtype xt_statistic xt_nat xt_MASQUERADE xt_mark xt_nfacct ipt_
↪REJECT nf_reject_ipv4 xt_tcpudp nft_chain_nat nfnetlink_acct nvidia_uvm(OE)
↪overlay xt_conntrack xt_comment nft_compat nf_tables rpcsec_gss_krb5 qrtr
↪auth_rpcgss rdma_ucm(OE) rdma_cm(OE) iw_cm(OE) ib_ipoib(OE) ib_cm(OE) ib_
↪umad(OE) binfmt_misc intel_rapl_msr intel_rapl_common intel_uncore_
↪frequency intel_uncore_frequency_common intel_ifs i10nm_edac skx_edac_
↪common nfit x86_pkg_temp_thermal intel_powerclamp coretemp kvm_intel nvidia_
↪drm(OE) iaa_crypto nvidia_modeset(OE) cmdlinepart kvm spi_nor pmt_telemetry
↪irqbypass mtd intel_sdspi pmt_class nvidia(OE) rapl intel_cstate video dax_
↪hmem ecc intel_th_gth idxd mei_me isst_if_mmio isst_if_mbox_pci i2c_i801
↪intel_th_pci
[128435.409640] spi_intel_pci isst_if_common idxd_bus intel_vsec ast mei
↪intel_th i2c_smbus i2c_ismt spi_intel cxl_acpi cxl_core input_leds joydev
↪mac_hid knem(OE) ipmi_devintf ipmi_msghandler dm_multipath msr iptable_
↪filter iptable_mangle iptable_nat xt_owner xt_REDIRECT nf_nat nf_conntrack
↪nf_defrag_ipv6 nf_defrag_ipv4 nvme_fabrics efi_pstore nfnetlink dmi_sysfs
↪ip_tables x_tables autofs4 nls_iso8859_1 raid10 raid456 async_raid6_recov
↪async_memcpy async_pq async_xor async_tx raid1 raid0 bonding e1000 mpt3sas
↪raid_class sata_sil i40e ahci libahci xfs nfsv4 mptsas forcedeth udf crc_
↪itu_t aacraid dm_thin_pool dm_persistent_data dm_bufio megaraid_sas isofs
↪mptspi bnx2 mptscsih mptbase hpilo igb i2c_algo_bit bnx2_en e1000e reiserfs
```

(continues on next page)

(continued from previous page)

```

→br_netfilter bridge stp llc megaraid arcmsr dm_bio_prison hpsa scsi_
→transport_sas btrfs blake2b_generic xor raid6_pq ixgbevf bnx2x libcrc32c
→tg3 sata_svw jfs nls_ucs2_utils aic7xxx sata_nv nfsv3 nfs_acl nfs lockd
→grace sunrpc netfs igbvf aic79xx scsi_transport_spi mlx5_ib(OE) ib_
→uverbs(OE) macsec
[128435.409680] ib_core(OE) hid_generic usbhid hid cdc_ether usbnet mii uas
→usb_storage mlx5_core(OE) crct10dif_pclmul crc32_pclmul polyval_clmulni
→polyval_generic mlxfw(OE) ice ghash_clmulni_intel sha256_ssse3 psample sha1_
→ssse3 nvme mlxdevm(OE) ixgbe aesni_intel crypto_simd nvme_core tls xfrm_
→algo cryptd xhci_pci dca mlx_compat(OE) gnss nvme_auth mdio pci_hyperv_intf
→wmi xhci_pci_renesas pinctrl_emmitsburg [last unloaded: ipmi_msghandler]
[128435.409695] CPU: 113 PID: 1061323 Comm: ib-sriov Tainted: G W OE 6.8.0-51-
→generic #52-Ubuntu
[128435.409698] Hardware name: , BIOS 5.32 10/24/2024
[128435.409699] RIP: 0010:rtnl_getlink+0x43a/0x470
[128435.409701] Code: c7 a0 f2 76 8c e8 c6 c5 06 00 4d 85 ed 0f 84 e2 fc ff
→ff 49 c7 45 00 a0 f2 76 8c e9 d5 fc ff ff b8 ea ff ff ff e9 c4 fe ff ff <0f>
→0b e9 a4 fe ff ff 48 c7 c7 d0 f2 76 8c e8 93 c5 06 00 4d 85 ed
[128435.409703] RSP: 0018:ff29f566a62bb700 EFLAGS: 00010246
[128435.409704] RAX: 00000000ffffffa6 RBX: ff24d605ed223480 RCX:
→0000000000000000
[128435.409706] RDX: 0000000000000000 RSI: 0000000000000000 RDI:
→0000000000000000
[128435.409706] RBP: ff29f566a62bb9d8 R08: 0000000000000000 R09:
→0000000000000000
[128435.409707] R10: 0000000000000000 R11: ff24d60420219e00 R12:
→ff24d6042021a200
[128435.409708] R13: ffffffff8df7ecc0 R14: ffffffff8df7ecc0 R15:
→00000000ffffffff
[128435.409709] FS: 00000000007a9d10(0000) GS:ff24d6fdb480000(0000)
→knlgS:0000000000000000
[128435.409710] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
[128435.409711] CR2: 000000c000428000 CR3: 00000004a559a006 CR4:
→0000000000f71ef0
[128435.409713] DR0: 0000000000000000 DR1: 0000000000000000 DR2:
→0000000000000000
[128435.409713] DR3: 0000000000000000 DR6: 00000000fffe07f0 DR7:
→00000000000000400
[128435.409715] PKRU: 55555554
[128435.409715] Call Trace:
[128435.409716] <TASK>
[128435.409718] ? show_regs+0x6d/0x80
[128435.409720] ? __warn+0x89/0x160
[128435.409722] ? rtnl_getlink+0x43a/0x470
[128435.409724] ? report_bug+0x17e/0x1b0
[128435.409727] ? handle_bug+0x51/0xa0
[128435.409730] ? exc_invalid_op+0x18/0x80
[128435.409732] ? asm_exc_invalid_op+0x1b/0x20
[128435.409735] ? rtnl_getlink+0x43a/0x470
[128435.409740] rtnetlink_rcv_msg+0x16d/0x430
[128435.409742] ? apparmor_file_alloc_security+0x43/0x1f0
[128435.409747] ? __pfx_rtnetlink_rcv_msg+0x10/0x10

```

(continues on next page)

(continued from previous page)

```

[128435.409749] netlink_rcv_skb+0x5a/0x110
[128435.409752] rtnetlink_rcv+0x15/0x30
[128435.409754] netlink_unicast+0x24a/0x390
[128435.409756] netlink_sendmsg+0x214/0x470
[128435.409758] __sys_sendto+0x21b/0x230
[128435.409761] __x64_sys_sendto+0x24/0x40
[128435.409763] x64_sys_call+0x1b2d/0x25a0
[128435.409765] do_syscall_64+0x7f/0x180
[128435.409767] ? handle_pte_fault+0x1cb/0x1d0
[128435.409770] ? __handle_mm_fault+0x653/0x790
[128435.409772] ? __count_memcg_events+0x6b/0x120
[128435.409776] ? count_memcg_events.constprop.0+0x2a/0x50
[128435.409779] ? handle_mm_fault+0xad/0x380
[128435.409781] ? do_user_addr_fault+0x333/0x670
[128435.409783] ? irqentry_exit_to_user_mode+0x7b/0x260
[128435.409786] ? irqentry_exit+0x43/0x50
[128435.409787] ? exc_page_fault+0x94/0x1b0
[128435.409789] entry_SYSCALL_64_after_hwframe+0x78/0x80
[128435.409791] RIP: 0033:0x40328e
[128435.409812] Code: 48 89 6c 24 38 48 8d 6c 24 38 e8 0d 00 00 00 48 8b 6c
→24 38 48 83 c4 40 c3 cc cc cc 49 89 f2 48 89 fa 48 89 ce 48 89 df 0f 05 <48>
→ 3d 01 f0 ff ff 76 15 48 f7 d8 48 89 c1 48 c7 c0 ff ff ff ff 48
[128435.409813] RSP: 002b:000000c00036b1b8 EFLAGS: 00000206 ORIG_RAX:
→000000000000002c

```

13.12. ACCESS_REG Command Failure with Err(-22)

13.12.1. Issue

After the initial installation of DGX OS 7.0.1 on a DGX B200 system, the following non-destructive issue has been seen on every boot. This is due to the `node_exporter` attempting to get telemetry from PF0 and PF1, causing a `dmesg` error message similar to the example below to be written to the kernel log once every 30 seconds. This might cause the kernel log to fill up.

```

[11176.517416] mlx5_core 0000:05:00.0: mlx5_cmd_out_err:835:(pid 18360):
→ACCESS_REG(0x805) op_mod(0x1) failed, status bad operation(0x2), syndrome
→(0x9a6171), err(-22)
[11176.534892] mlx5_core 0000:05:00.0: mlx5_cmd_out_err:835:(pid 18360):
→ACCESS_REG(0x805) op_mod(0x1) failed, status bad operation(0x2), syndrome
→(0x9a6171), err(-22)
[11176.589308] mlx5_core 0000:05:00.1: mlx5_cmd_out_err:835:(pid 10354):
→ACCESS_REG(0x805) op_mod(0x1) failed, status bad operation(0x2), syndrome
→(0x9a6171), err(-22)
[11176.607052] mlx5_core 0000:05:00.1: mlx5_cmd_out_err:835:(pid 10354):
→ACCESS_REG(0x805) op_mod(0x1) failed, status bad operation(0x2), syndrome
→(0x9a6171), err(-22)

```

13.12.2. Workaround

Accessing PF0 and PF1 is restricted. Currently, there is no temporary solution available.

13.13. DGX B200 Display Goes Blank Shortly After GRUB Menu

13.13.1. Issue

During the DGX B200 system boot process, the NVIDIA splash screen and the POST display properly. However, when the DGX OS Linux messages start to appear, the screen goes blank. The system is connected to a standard HD monitor with a resolution of 1920 x 1080 pixels through the VGA port. This issue occurs due to an incompatibility between the DGX B200 monitor and the supported resolutions of the connected monitor.

13.13.2. Workaround

When a low-resolution monitor is attached to the system being installed, the GRUB `video=` parameter should be set to the resolution of the monitor that is attached. For this specific issue, where a monitor with a resolution of 1920 x 1080 pixels is connected through the VGA port, add the `video=1920x1080` parameter to the GRUB menu entry at boot time.

1. On the GRUB boot screen, select the **e** key to edit the boot entry.
2. Using the down arrow key, navigate to the `linux` line that contains the boot parameters.
3. Append the following parameter to the `linux` line.

```
video=1920x1080
```

4. Select **Ctrl+x** to boot the system with the added parameter.

The system finishes booting and remains resolution at 1920x1080 pixels.

13.14. nv-disk-encrypt Failed on Pre-Owned NVMe Drives

13.14.1. Issue

Using the `nv-disk-encrypt` tool to initialize the system for NVMe drive encryption failed with the following error messages:

```
takeOwnership failed
SED takeownership failed on /dev/nvme0n1
```


13.14.2. Workaround

To resolve the issue, recover from lost keys and erase the drives as shown in the following single-drive example:

1. Take ownership of all drives one at a time.

```
sudo sedutil-cli --takeownership <your-sid-password> /dev/nvme6n1
```

2. If step 1 fails, specify the PSID to reset the drive using the sedutil-cli command.

Caution

Before performing this step, back up your important data to another location because it will delete everything on the drive.

You can obtain the PSID, which is printed on the label, by physically examining the drive.

For example,

```
sudo sedutil-cli --yesIreallywanttoERASEALLmydatausingthePSID <your-drive-  
→psid> /dev/nvme6n1
```

3. Check taking ownership.

```
sudo sedutil-cli --takeownership <your-sid-password> /dev/nvme6n1
```

4. Revert ownership before the initialization process for drive encryption.

```
sudo sedutil-cli --reverttper <your-sid-password> /dev/nvme6n1
```

5. Initialize the system for drive encryption using the `nv-disk-encrypt init` command.

```
sudo nv-disk-encrypt init [-k <your-vault-password>] [-f <path/to/json-  
→file>] [-g] [-r]
```

13.15. GPUs Cannot Be Reset During MIG Configurations on DGX A100 and A800 Systems

13.15.1. Issue

When you run the `nvidia-mig-parted` tool to apply the MIG configurations on DGX A100 or DGX A800 systems, the following error message might occur:

```
The following GPUs could not be reset:
GPU 00000000:01:00.0: In use by another client
GPU 00000000:47:00.0: In use by another client
GPU 00000000:81:00.0: In use by another client
GPU 00000000:C2:00.0: In use by another client
```

13.15.2. Workaround

To recover from this error, reboot the server to apply the most recent `nvidia-mig-parted` configuration.

For future updates using the `nvidia-mig-parted` command, ensure to run the following command before any additional `nvidia-mig-parted` commands:

```
$ sudo rmmod nvidia_drm nvidia_modeset
```

After the `nvidia-mig-parted` command is complete, reload the `nvidia_drm` and `nvidia_modeset` modules.

13.16. DGX System Device ID Not Found in /usr/share/misc/pci.ids

13.16.1. Issue

When you run the following command to apply the default `mig-parted` configuration, the `nvidia-mig-parted` tool issues warnings about failing to find the device ID for the DGX system:

```
$ sudo nvidia-mig-parted apply -f /etc/nvidia-mig-manager/config-default.yaml
↪ -c all-balanced -k /etc/nvidia-mig-manager/hooks-default.yaml

2024/09/05 01:00:00 WARNING: unable to get device name: [failed to find
↪ device with id '22a3']
2024/09/05 01:00:00 WARNING: unable to get device name: [failed to find
↪ device with id '22a3']
2024/09/05 01:00:00 WARNING: unable to get device name: [failed to find
↪ device with id '22a3']
```

13.16.2. Workaround

Update the system with the current version of the PCI ID list by running the `update-pciids` command:

```
sudo update-pciids
```

13.17. Missing the `nvidia-system-station` Metapackage on DGX Station A100 and DGX Station A800

13.17.1. Issue

During the installation using the DGX OS 7.0.0 ISO on a DGX Station A100 or DGX Station A800, the `nvidia-system-station` metapackage was incorrectly removed.

13.17.2. Workaround

This issue occurs only on the DGX Station A100 and DGX Station A800. To resolve this issue, install the `nvidia-system-station` metapackage manually:

```
sudo apt install nvidia-system-station
```

13.18. Virtualization Not Supported

13.18.1. Issue

Virtualization technology, such as ESXi hypervisors or kernel-based virtual machines (KVM), is not an intended use case on DGX systems and has not been tested.

Chapter 14. DGX OS Connectivity Requirements

In a typical operation, DGX OS runs services to support typical usage of the DGX system.

Some of these services require network communication. The table below describes the port, protocol, direction, and communication purpose for the services. DGX administrators should consider their site-specific access needs and allow or disallow communication with the services as necessary.

14.1. In-Band Management, Storage, and Compute Networks

The following table provides information about the in-band management, storage, and compute networks.

Table 1: In-Band Management, Storage, and Compute Networks

Port (Protocol)	Direction	Use
22 (TCP)	Inbound	SSH
53 (UDP)	Outbound	DNS
80 (TCP)	Outbound	HTTP, package updates
111 (TCP)	In-bound/Outbound	RPCBIND, required by NFS
273 (TCP)		NVIDIA System Management Management
443 (TCP)	Outbound	For internet (HTTP/HTTPS) connection to NVIDIA GPU Cloud. If port 443 is proxied through a corporate firewall, then WebSocket protocol traffic must be supported
1883 (TCP)		Mosquitto Database (used by NVIDIA System Management)

14.2. Out-of-Band Management

The following table provides information about out-of-band management for your DGX system.

Table 2: Out-of-Band Management

Port (Pro- tocol)	Direc- tion	Use
443 (TCP)	In- bound	For BMC web services, remote console services, and CD-media service. If port 443 is proxied through a corporate firewall, then WebSocket protocol traffic must be supported.
623 (UDP)	In- bound	IPMI

Chapter 15. DGX Software Stack

15.1. NVIDIA DGX Software Packages

The following tables list the packages installed as part of the DGX Software Stack, categorized by metapackage names.

15.1.1. nvidia-system-core

Package Name	Description
cuda-compute-repo	CUDA compute repository configuration files.
dgx-release	Package updates the DGX OS release information.
dgx-repo	DGX repository configuration files.
hpc-sdk-repo	NVIDIA HPC SDK repository configuration files.
msecli	Micron Storage Executive CLI.
nv-common-apis	Install commonly used scripts used by Nvidia systems.
nv-cpu-governor	Set CPU governor to performance.
nv-env-paths	Configure the PATH variable.
nv-grubmenu	Make Grub menu visible.
nv-grubserial	Display GRUB menu over a serial console.
nv-iommu	Enable iommu in passthrough mode; enable intel_iommu on systems with Emerald.
nv-ipmi-devintf	Load the ipmi_devintf module.
nv-limits	Increase the file limit.
nv-update-disable	Disable OS update prompt.
nvgpu-services-list	List all GPU-related services.
nvidia-acsc-disable	Disable the PCIe ACS capability.
nvidia-crashdump	NVIDIA crash dump policy.
nvidia-disable-init-on-alloc	Disable heap memory zeroing on allocation.

continues

Table 1 – continued from previous page

Package Name	Description
nvidia-disable-numa-balancing	Disable automatic page fault NUMA memory balancing.
nvidia-earlycon	Set up the early console with no options.
nvidia-enable-power-meter-cap	Enable power capping functionality in ACPI power meter.
nvidia-esm-hook-epilogue	NVIDIA package to clarify ESM policy.
nvidia-fs-loader	Load the nvidia-fs module.
nvidia-ipmisol	Enable IPMI Serial-over-LAN.
nvidia-kernel-defaults	The sysctl default kernel settings for DGX.
nvidia-mig-manager	NVIDIA MIG Partition Editor and Systemd Service.
nvidia-nvme-options	Automatically enables NVMe Interrupt Coalescing at bootup on all Samsung and K
nvidia-pci-bridge-power	Set the PCI bridge power control to on.
nvidia-pci-realloc	Force PCI resource reallocation.
nvidia-raid-config	DGX RAID Configuration.
nvidia-redfish-config	Configure Redfish Host Interface.
nvidia-relaxed-ordering-gpu	Configure PCIe Relaxed Ordering.
nvidia-relaxed-ordering-nvme	Configure PCIe Relaxed Ordering.
nvidia-repo-keys	Add keys to apt trusted.gpg database.

15.1.2. nvidia-system-utils

Package Name	Description
nv-persistence-mode	Enable persistence mode.
nvidia-conf-cachefilesd	Systemd settings for cachefilesd.
nvidia-fs-loader	Load the nvidia-fs module.
nvidia-logrotate	NVIDIA logrotate policy.
nvidia-motd	Custom motd files for NVIDIA platforms.
nvsm	REST API services for DGX System Management.

15.1.3. nvidia-system-mlnx-drivers

Package Name	Description
doca-ofed	The doca-ofed metapackage.
doca-repo	DOCA repository configuration files.
mlnx-nfsrdma-dkms	DKMS support for NFS RDMA kernel module.
mlnx-nvme-dkms	DKMS support for nvme kernel module.
mlnx-pxe-setup	Provide a script to enable PXE booting using Mellanox cards.
nvidia-ib-umad-loader	Load the ib_umad module.
nvidia-mlnx-config	Configure the MLNX devices.
mlnx-fw-updater	Firmware update binaries and utility

15.2. DGX Kernel Parameters, System Configuration Settings, and Runtime Commands

15.2.1. Kernel Parameters

Parameter Name	Description	Package	Location
crashkernel	Amount of memory to use for crash dumps.	nvidia-crashdump	/etc/default/grub.d/kdump-tools.cfg
console=ttyS[0-1],115200n8	Set the console to serial port 0 or 1, using 115200 baud, no parity, and 8 data bits. For dgx-h100 and dgx-h800: console=ttyS0,115200 n8. Other system types: console=ttyS1,115200 n8.	nvidia-ipmisol	/etc/default/grub.d/ipmisol.cfg
iommu=pt	Enable pass through mode only and disable DMA translations. This enables optimizations for the CPU inside the DGX A100.	nv-iommu	/etc/default/grub.d/iommu.cfg
pci=realloc=on	Allow kernel to reallocate PCI resources if allocations done by BIOS are insufficient. This and pcie_ports=native are both required for NVME hot-plug on DGX2.	nv-enable-nvme-hot-plug	/etc/default/grub.d/enable-nvme-hot-plug.cfg

15.2.2. System Configuration Settings

Parameter Name	Description	Package	Location
net.ipv4.conf.all.ar p_announce = 2	Always use the best local address for this target.	nvidia- kernel- defaults	/etc/sysctl.d/20-nvidia- defaults.conf
net.ipv4.conf.default arp_announce = 2	Always use the best local address for this target.	nvidia- kernel- defaults	/etc/sysctl.d/20-nvidia- defaults.conf
net.ipv4.conf.all.ar p_ignore = 1	Only reply to ARP requests on the interface that contains the target IP address.	nvidia- kernel- defaults	/etc/sysctl.d/20-nvidia- defaults.conf
net.ipv4.conf.default.arp = 1	Only reply to ARP requests on the interface that contains the target IP address.	nvidia- kernel- defaults	/etc/sysctl.d/20-nvidia- defaults.conf
NVreg_EnablePCIERela xedOrderingMode=1	Set a reg-key to enable PCIe relaxed-ordering in the GPUs.	nvidia- relaxed- ordering- gpu	/etc/modprobe.d/ nvidia-relaxed- ordering.conf
ker- nel.panic_on_unrecovere = 1	Configure the system to panic on unrecoverable NMI (Non-Maskable Interrupt).	nvidia- crashdump	/etc/sysctl.d/90-dgx- crashdump.conf
ker- nel.unknown_nmi_panic = 1	Configure the system to panic on an unknown NMI.	nvidia- crashdump	/etc/sysctl.d/90-dgx- crashdump.conf
ker- nel.hardlockup_panic = 1	Configure the system to panic when a hard lockup is detected.	nvidia- crashdump	/etc/sysctl.d/90-dgx- crashdump.conf
ker- nel.panic_on_io_nmi = 1	Configure the system to panic on I/O NMI.	nvidia- crashdump	/etc/sysctl.d/90-dgx- crashdump.conf
ker- nel.softlockup_panic = 1	Configure the system to panic when a soft lockup is detected.	nvidia- crashdump	/etc/sysctl.d/90-dgx- crashdump.conf
kernel.panic_on_oops = 1	Configure the system to panic when an Oops occurs.	nvidia- crashdump	/etc/sysctl.d/90-dgx- crashdump.conf
ker- nel.hung_task_panic = 1	Configure the system to panic when a hung task is detected.	nvidia- crashdump	/etc/sysctl.d/90-dgx- crashdump.conf

continues on next page

Table 5 – continued from previous page

Parameter Name	Description	Package	Location
kernel.panic_on_rcu_stall = 1	Configure the system to panic when an RCU stall is detected.	nvidia-crashdump	/etc/sysctl.d/90-dgx-crashdump.conf
kernel.panic = 30	Configure the system to reboot after 30 seconds if a panic occurs.	nvidia-crashdump	/etc/sysctl.d/90-dgx-crashdump.conf

15.2.3. Runtime Commands

Command	Description	Package	Location
setpci -d ::207 68.w=5000:f000	Set MaxReadReq size to 4KB for ConnectX-6 Network (2) Infiniband (07) devices. Only needed for DGX A100/A800	nvidia-mlnx-config	/etc/systemd/system/nvidia-mlnx-config.service
mlxconfig -y -d <device> set ADVANCED_PCI_SETTINGS	Configure Mellanox network device PCI settings. Enables advanced PCI settings. Only needed for DGX A100/A800	nvidia-mlnx-config	/usr/bin/nvidia-mlnx-config.sh
mlxconfig -y -d <device> set MAX_ACC_OUT_READ=4	Configure Mellanox network device PCI settings. Sets maximum accumulated outbound read requests to 44. Only needed for DGX A100/A800	nvidia-mlnx-config	/usr/bin/nvidia-mlnx-config.sh

15.3. DGX Platform JSON Configuration

```
{
  "dgx_a800": {
    "PlatformType": "DGX A800",
    "GrepStr": "^. *920-23687-2535-.*",
    "UsesDmiSystemFamilyForDetection": "False",
    "ConfigureDGXA100Raid": "True",
    "ConfigureDGXStationA100Raid": "False",
    "NVMERelaxedOrdering": "True",
    "GpuRelaxedOrdering": "True",
    "EnablePowerMeterCap": "False",
    "BMCPasswdMinLength": "13",
    "BMCPasswdMaxLength": "20",
    "BMCPasswdSupportsZerofill": "True",
    "BMCPasswdComplexityReq": "False",
```

(continues on next page)

(continued from previous page)

```

    "NVSMAlertsSupported": "True",
    "NeedsMRRSConfig": "True",
    "NeedsAccBytesTuning": "True",
    "IPMIDefaultSerialTTY": "ttyS1",
    "NeedsOEMXconfigOverride": "False",
    "NeedsInitialNvidiaXconfig": "False",
    "NeedsAdaptiveNvidiaXconfig": "False",
    "NeedsContainerdOverride": "False",
    "NeedsOemConfigPostActNetplanApply": "False",
    "UsesFabricManager": "True",
    "IsDgxServerType": "True",
    "IsDgxDesktopType": "False",
    "NeedsDisableNumaBalance": "False",
    "NeedsIommuPt": "True",
    "NeedsDisableInitOnAlloc": "True",
    "NeedsEarlycon": "False",
    "PciRealloc": "",
    "CrashdumpMem": "1G-:2048M",
    "UsePciDisableAcsRedir": "True",
    "ConsoleSetting": "",
    "SerialBaudRate": "115200"
  },
  "dgx_a100":
  {
    "PlatformType": "DGX A100",
    "GrepStr": "^. *920-23687-.*\\|^.*675-23287-.*\\|^.*920-23287-.*",
    "UsesDmiSystemFamilyForDetection": "False",
    "ConfigureDGXA100Raid": "True",
    "ConfigureDGXStationA100Raid": "False",
    "NVMERelaxedOrdering": "True",
    "GpuRelaxedOrdering": "True",
    "EnablePowerMeterCap": "False",
    "BMCPasswdMinLength": "13",
    "BMCPasswdMaxLength": "20",
    "BMCPasswdSupportsZerofill": "True",
    "BMCPasswdComplexityReq": "False",
    "NVSMAlertsSupported": "True",
    "NeedsMRRSConfig": "True",
    "NeedsAccBytesTuning": "True",
    "IPMIDefaultSerialTTY": "ttyS1",
    "NeedsOEMXconfigOverride": "False",
    "NeedsInitialNvidiaXconfig": "False",
    "NeedsAdaptiveNvidiaXconfig": "False",
    "NeedsContainerdOverride": "False",
    "NeedsOemConfigPostActNetplanApply": "False",
    "UsesFabricManager": "True",
    "IsDgxServerType": "True",
    "IsDgxDesktopType": "False",
    "NeedsDisableNumaBalance": "False",
    "NeedsIommuPt": "True",
    "NeedsDisableInitOnAlloc": "True",
    "NeedsEarlycon": "False",

```

(continues on next page)

(continued from previous page)

```

    "PciRealloc": "",
    "CrashdumpMem": "1G-:2048M",
    "UsePciDisableAcsRedir": "True",
    "ConsoleSetting": "",
    "SerialBaudRate": "115200"
  },
  "dgx_h100":
  {
    "PlatformType": "DGX H100",
    "GrepStr": "^.*DGXH100.*\\|^.*DGX H100.*",
    "UsesDmiSystemFamilyForDetection": "False",
    "ConfigureDGXA100Raid": "True",
    "ConfigureDGXStationA100Raid": "False",
    "NVMERelaxedOrdering": "True",
    "GpuRelaxedOrdering": "False",
    "EnablePowerMeterCap": "False",
    "BMCPasswdMinLength": "13",
    "BMCPasswdMaxLength": "20",
    "BMCPasswdSupportsZerofill": "True",
    "BMCPasswdComplexityReq": "True",
    "NVSMAlertsSupported": "True",
    "NeedsMRRSConfig": "True",
    "NeedsAccBytesTuning": "False",
    "IPMIDefaultSerialTTY": "ttyS0",
    "NeedsOEMXconfigOverride": "False",
    "NeedsInitialNvidiaXconfig": "False",
    "NeedsAdaptiveNvidiaXconfig": "False",
    "NeedsContainerdOverride": "False",
    "NeedsOemConfigPostActNetplanApply": "False",
    "UsesFabricManager": "True",
    "IsDgxServerType": "True",
    "IsDgxDesktopType": "False",
    "NeedsDisableNumaBalance": "False",
    "NeedsIommuPt": "True",
    "NeedsDisableInitOnAlloc": "True",
    "NeedsEarlycon": "False",
    "PciRealloc": "off",
    "CrashdumpMem": "1G-:2048M",
    "UsePciDisableAcsRedir": "True",
    "ConsoleSetting": "",
    "SerialBaudRate": "115200"
  },
  "dgx_h200":
  {
    "PlatformType": "DGX H200",
    "GrepStr": "^.*DGXH200.*\\|^.*DGX H200.*",
    "UsesDmiSystemFamilyForDetection": "False",
    "ConfigureDGXA100Raid": "True",
    "ConfigureDGXStationA100Raid": "False",
    "NVMERelaxedOrdering": "True",
    "GpuRelaxedOrdering": "False",
    "EnablePowerMeterCap": "False",

```

(continues on next page)

(continued from previous page)

```

    "BMCPasswordMinLength": "13",
    "BMCPasswordMaxLength": "20",
    "BMCPasswordSupportsZeroFill": "True",
    "BMCPasswordComplexityReq": "True",
    "NVSMAlertsSupported": "True",
    "NeedsMRRSConfig": "True",
    "NeedsAccBytesTuning": "False",
    "IPMIDefaultSerialTTY": "ttyS0",
    "NeedsOEMXconfigOverride": "False",
    "NeedsInitialNvidiaXconfig": "False",
    "NeedsAdaptiveNvidiaXconfig": "False",
    "NeedsContainerdOverride": "False",
    "NeedsOemConfigPostActNetplanApply": "False",
    "UsesFabricManager": "True",
    "IsDgxServerType": "True",
    "IsDgxDesktopType": "False",
    "NeedsDisableNumaBalance": "False",
    "NeedsIommuPt": "True",
    "NeedsDisableInitOnAlloc": "True",
    "NeedsEarlycon": "False",
    "PciRealloc": "off",
    "CrashdumpMem": "1G-:2048M",
    "UsePciDisableAcsRedir": "True",
    "ConsoleSetting": "",
    "SerialBaudRate": "115200"
  },
  "dgx_h800":
  {
    "PlatformType": "DGX H100",
    "GrepStr": "^.*DGXH800.*\\|^.*DGX H800.*",
    "UsesDmiSystemFamilyForDetection": "False",
    "ConfigureDGXA100Raid": "True",
    "ConfigureDGXStationA100Raid": "False",
    "NVMERelaxedOrdering": "True",
    "GpuRelaxedOrdering": "False",
    "EnablePowerMeterCap": "False",
    "BMCPasswordMinLength": "13",
    "BMCPasswordMaxLength": "20",
    "BMCPasswordSupportsZeroFill": "True",
    "BMCPasswordComplexityReq": "True",
    "NVSMAlertsSupported": "True",
    "NeedsMRRSConfig": "True",
    "NeedsAccBytesTuning": "False",
    "IPMIDefaultSerialTTY": "ttyS0",
    "NeedsOEMXconfigOverride": "False",
    "NeedsInitialNvidiaXconfig": "False",
    "NeedsAdaptiveNvidiaXconfig": "False",
    "NeedsContainerdOverride": "False",
    "NeedsOemConfigPostActNetplanApply": "False",
    "UsesFabricManager": "True",
    "IsDgxServerType": "True",
    "IsDgxDesktopType": "False",

```

(continues on next page)

(continued from previous page)

```

    "NeedsDisableNumaBalance": "False",
    "NeedsIommuPt": "True",
    "NeedsDisableInitOnAlloc": "True",
    "NeedsEarlycon": "False",
    "PciRealloc": "off",
    "CrashdumpMem": "1G-:2048M",
    "UsePciDisableAcsRedir": "True",
    "ConsoleSetting": "",
    "SerialBaudRate": "115200"
  },
  "dgx_b200":
  {
    "PlatformType": "DGX B200",
    "GrepStr": "^.*DGXB200.*\\|^.*DGX B200.*",
    "UsesDmiSystemFamilyForDetection": "False",
    "ConfigureDGXA100Raid": "True",
    "ConfigureDGXStationA100Raid": "False",
    "NVMERelaxedOrdering": "True",
    "GpuRelaxedOrdering": "False",
    "EnablePowerMeterCap": "False",
    "BMCPasswdMinLength": "13",
    "BMCPasswdMaxLength": "20",
    "BMCPasswdSupportsZerofill": "True",
    "BMCPasswdComplexityReq": "True",
    "NVSMAlertsSupported": "True",
    "NeedsMRRSConfig": "True",
    "NeedsAccBytesTuning": "False",
    "IPMIDefaultSerialTTY": "ttyS0",
    "NeedsOEMXconfigOverride": "False",
    "NeedsInitialNvidiaXconfig": "False",
    "NeedsAdaptiveNvidiaXconfig": "False",
    "NeedsContainerdOverride": "False",
    "NeedsOemConfigPostActNetplanApply": "False",
    "UsesFabricManager": "True",
    "IsDgxServerType": "True",
    "IsDgxDesktopType": "False",
    "NeedsDisableNumaBalance": "False",
    "NeedsIommuPt": "True",
    "NeedsDisableInitOnAlloc": "True",
    "NeedsEarlycon": "False",
    "PciRealloc": "off",
    "CrashdumpMem": "2048M,high",
    "UsePciDisableAcsRedir": "True",
    "ConsoleSetting": "",
    "SerialBaudRate": "115200"
  },
  "dgx_b300":
  {
    "PlatformType": "DGX B300",
    "GrepStr": "^.*DGXB300.*\\|^.*DGX B300.*",
    "UsesDmiSystemFamilyForDetection": "False",
    "ConfigureDGXA100Raid": "True",

```

(continues on next page)

(continued from previous page)

```

    "ConfigureDGXStationA100Raid": "False",
    "NVMERelaxedOrdering": "True",
    "GpuRelaxedOrdering": "False",
    "EnablePowerMeterCap": "False",
    "BMCPasswordMinLength": "13",
    "BMCPasswordMaxLength": "20",
    "BMCPasswordSupportsZerofill": "True",
    "BMCPasswordComplexityReq": "True",
    "NVSMAlertsSupported": "True",
    "NeedsMRRSConfig": "True",
    "NeedsAccBytesTuning": "False",
    "IPMIDefaultSerialTTY": "ttyS0",
    "NeedsOEMXconfigOverride": "False",
    "NeedsInitialNvidiaXconfig": "False",
    "NeedsAdaptiveNvidiaXconfig": "False",
    "NeedsContainerdOverride": "False",
    "NeedsOemConfigPostActNetplanApply": "False",
    "UsesFabricManager": "True",
    "IsDgxServerType": "True",
    "IsDgxDesktopType": "False",
    "NeedsDisableNumaBalance": "False",
    "NeedsIommuPt": "True",
    "NeedsDisableInitOnAlloc": "True",
    "NeedsEarlycon": "False",
    "PciRealloc": "off",
    "CrashdumpMem": "2048M,high",
    "UsePciDisableAcsRedir": "True",
    "ConsoleSetting": "",
    "SerialBaudRate": "115200"
  },
  "dgx_gb200":
  {
    "PlatformType": "DGX GB200",
    "GrepStr": "^.*DGXGB200.*\\|^.*DGX GB200.*\\|^ *GB200 NVL.*",
    "UsesDmiSystemFamilyForDetection": "False",
    "ConfigureDGXA100Raid": "True",
    "ConfigureDGXStationA100Raid": "False",
    "NVMERelaxedOrdering": "True",
    "GpuRelaxedOrdering": "False",
    "EnablePowerMeterCap": "True",
    "BMCPasswordMinLength": "13",
    "BMCPasswordMaxLength": "20",
    "BMCPasswordSupportsZerofill": "True",
    "BMCPasswordComplexityReq": "True",
    "NVSMAlertsSupported": "True",
    "NeedsMRRSConfig": "True",
    "NeedsAccBytesTuning": "False",
    "IPMIDefaultSerialTTY": "ttyS0",
    "NeedsOEMXconfigOverride": "False",
    "NeedsInitialNvidiaXconfig": "False",
    "NeedsAdaptiveNvidiaXconfig": "False",
    "NeedsContainerdOverride": "False",

```

(continues on next page)

(continued from previous page)

```

    "NeedsOemConfigPostActNetplanApply": "False",
    "UsesFabricManager": "False",
    "IsDgxServerType": "True",
    "IsDgxDesktopType": "False",
    "NeedsDisableNumaBalance": "True",
    "NeedsIommuPt": "False",
    "NeedsDisableInitOnAlloc": "True",
    "NeedsEarlycon": "True",
    "PciRealloc": "on",
    "CrashdumpMem": "2048M,high",
    "UsePciDisableAcsRedir": "True",
    "ConsoleSetting": "",
    "SerialBaudRate": "115200"
  },
  "dgx_gb300":
  {
    "PlatformType": "DGX GB300",
    "GrepStr": "^.*DGXGB300.*\\|^.*DGX GB300.*\\|^ *GB300 NVL.*",
    "UsesDmiSystemFamilyForDetection": "False",
    "ConfigureDGXA100Raid": "True",
    "ConfigureDGXStationA100Raid": "False",
    "NVMERelaxedOrdering": "True",
    "GpuRelaxedOrdering": "False",
    "EnablePowerMeterCap": "True",
    "BMCPasswordMinLength": "13",
    "BMCPasswordMaxLength": "20",
    "BMCPasswordSupportsZerofill": "True",
    "BMCPasswordComplexityReq": "True",
    "NVSMAlertsSupported": "True",
    "NeedsMRRSConfig": "True",
    "NeedsAccBytesTuning": "False",
    "IPMIDefaultSerialTTY": "ttyS0",
    "NeedsOEMXconfigOverride": "False",
    "NeedsInitialNvidiaXconfig": "False",
    "NeedsAdaptiveNvidiaXconfig": "False",
    "NeedsContainerdOverride": "False",
    "NeedsOemConfigPostActNetplanApply": "False",
    "UsesFabricManager": "False",
    "IsDgxServerType": "True",
    "IsDgxDesktopType": "False",
    "NeedsDisableNumaBalance": "True",
    "NeedsIommuPt": "False",
    "NeedsDisableInitOnAlloc": "True",
    "NeedsEarlycon": "True",
    "PciRealloc": "",
    "CrashdumpMem": "2048M,high",
    "UsePciDisableAcsRedir": "True",
    "ConsoleSetting": "",
    "SerialBaudRate": "115200"
  },
  "dgx_spark":
  {

```

(continues on next page)

(continued from previous page)

```

    "PlatformType": "DGX Spark",
    "GrepStr": "DGX Spark",
    "UsesDmiSystemFamilyForDetection": "True",
    "ConfigureDGXA100Raid": "False",
    "ConfigureDGXStationA100Raid": "False",
    "NVMERelaxedOrdering": "False",
    "GpuRelaxedOrdering": "False",
    "EnablePowerMeterCap": "False",
    "BMCPasswordMinLength": "1",
    "BMCPasswordMaxLength": "20",
    "BMCPasswordSupportsZerofill": "False",
    "BMCPasswordComplexityReq": "False",
    "NVSMAlertsSupported": "False",
    "NeedsMRRSConfig": "False",
    "NeedsAccBytesTuning": "False",
    "IPMIDefaultSerialTTY": "ttyS1",
    "NeedsOEMXconfigOverride": "False",
    "NeedsInitialNvidiaXconfig": "True",
    "NeedsAdaptiveNvidiaXconfig": "False",
    "NeedsContainerdOverride": "False",
    "NeedsOemConfigPostActNetplanApply": "False",
    "UsesFabricManager": "False",
    "IsDgxServerType": "False",
    "IsDgxDesktopType": "True",
    "NeedsDisableNumaBalance": "False",
    "NeedsIommuPt": "False",
    "NeedsDisableInitOnAlloc": "True",
    "NeedsEarlycon": "uart,mmio32,0x16A00000",
    "PciRealloc": "",
    "CrashdumpMem": "1G-:1024M",
    "UsePciDisableAcsRedir": "True",
    "ConsoleSetting": "tty0 plymouth.ignore-serial-consoles plymouth.use-
→simplifiedrm",
    "SerialBaudRate": "921600"
  },
  "dgxstation_a100":
  {
    "PlatformType": "DGXSTATION A100",
    "GrepStr": "^. *920-23487-.*\\|^.*675-23487-.*\\|^DGX Station A100",
    "UsesDmiSystemFamilyForDetection": "False",
    "ConfigureDGXA100Raid": "False",
    "ConfigureDGXStationA100Raid": "True",
    "NVMERelaxedOrdering": "True",
    "GpuRelaxedOrdering": "True",
    "EnablePowerMeterCap": "False",
    "BMCPasswordMinLength": "13",
    "BMCPasswordMaxLength": "20",
    "BMCPasswordSupportsZerofill": "True",
    "BMCPasswordComplexityReq": "False",
    "NVSMAlertsSupported": "True",
    "NeedsMRRSConfig": "True",
    "NeedsAccBytesTuning": "True",

```

(continues on next page)

(continued from previous page)

```

    "IPMIDefaultSerialTTY": "ttyS1",
    "NeedsOEMXconfigOverride": "True",
    "NeedsInitialNvidiaXconfig": "True",
    "NeedsAdaptiveNvidiaXconfig": "True",
    "NeedsContainerdOverride": "True",
    "UsesFabricManager": "False",
    "IsDgxServerType": "False",
    "IsDgxDesktopType": "True",
    "NeedsDisableNumaBalance": "False",
    "NeedsIommuPt": "True",
    "NeedsDisableInitOnAlloc": "True",
    "NeedsEarlycon": "False",
    "PciRealloc": "",
    "CrashdumpMem": "1G-:2048M",
    "UsePciDisableAcsRedir": "True",
    "ConsoleSetting": "",
    "SerialBaudRate": "115200"
  },
  "dgxstation_a800":
  {
    "PlatformType": "DGXSTATION A800",
    "GrepStr": "^.*920-23487-2535.*\\|^.*675-23487-0200.*",
    "UsesDmiSystemFamilyForDetection": "False",
    "ConfigureDGXA100Raid": "False",
    "ConfigureDGXStationA100Raid": "True",
    "NVMERelaxedOrdering": "True",
    "GpuRelaxedOrdering": "True",
    "EnablePowerMeterCap": "False",
    "BMCPasswdMinLength": "13",
    "BMCPasswdMaxLength": "20",
    "BMCPasswdSupportsZerofill": "True",
    "BMCPasswdComplexityReq": "False",
    "NVSMAlertsSupported": "True",
    "NeedsMRRSConfig": "True",
    "NeedsAccBytesTuning": "True",
    "IPMIDefaultSerialTTY": "ttyS1",
    "NeedsOEMXconfigOverride": "True",
    "NeedsInitialNvidiaXconfig": "True",
    "NeedsAdaptiveNvidiaXconfig": "True",
    "NeedsContainerdOverride": "True",
    "UsesFabricManager": "False",
    "IsDgxServerType": "False",
    "IsDgxDesktopType": "True",
    "NeedsDisableNumaBalance": "False",
    "NeedsIommuPt": "True",
    "NeedsDisableInitOnAlloc": "True",
    "NeedsEarlycon": "False",
    "PciRealloc": "",
    "CrashdumpMem": "1G-:2048M",
    "UsePciDisableAcsRedir": "True",
    "ConsoleSetting": "",
    "SerialBaudRate": "115200"
  }
}

```

(continues on next page)

(continued from previous page)

```
}
}
```

15.4. DGX Platform JSON Configuration Definitions

Name	Definition
PlatformType	Printable string representation of the platform type (for example, DGX H100).
GrepStr	Regex matching the product name as reported by dmidecode
UsesDmiSystemFamilyForDetection	Indicates that the platform uses the “System Family”, as reported by dmidecode <code>--string system-family</code> , for platform detection. DGX platforms typically use the “System Product Name” instead.
ConfigureDGXA100Raid	Used by <code>nvidia-raid-config</code> calls to create RAID array with a DGX A100-like disk arrangement: RAID 0 spanning all available U.2 drives.
ConfigureDGXStationA100Raid	Used to create RAID array with a DGX Station A100-like disk arrangement: single U.2, no RAID.
NVMERelaxedOrdering	Package installs the <code>/usr/bin/nvidia-relaxed-ordering-nvme.sh</code> script. Requires all data disks to be from the same vendor. Currently only supported in Samsung NVMe disks.
GpuRelaxedOrdering	The <code>nvidia-relaxed-ordering-gpus</code> package calls this function to change GPU driver settings based on platform.
EnablePowerMeterCap	The package <code>nvidia-enable-power-meter-cap</code> configures a server to enable power capping in ACPI power meter for Grace based platforms. Setting this to True will configure <code>acpi_power_meter.force_cap_on=y</code> in the grub boot parameters.
BMCPasswordMinLength	The package <code>nvidia-oem-config-plugins</code> creates EULA, BMC, etc oem-config screens that use this attribute to set BMC password requirements during ISO installation.
BMCPasswordMaxLength	The package <code>nvidia-oem-config-plugins</code> creates EULA, BMC, etc oem-config screens that use this attribute to set BMC password requirements during ISO installation.

continues on next page

Table 7 – continued from previous page

Name	Definition
BMCPasswordSupportsZerofill	The package <code>nvidia-oem-config-plugins</code> creates EULA, BMC, etc oem-config screens that use this attribute to set BMC password requirements during ISO installation.
BMCPasswordComplexityReq	The package <code>nvidia-oem-config-plugins</code> creates EULA, BMC, etc oem-config screens that use this attribute to set BMC password requirements during ISO installation.
NVSMAlertsSupported	NVSM is only supported on DGX Platforms. If NVSM is installed, <code>nvidia-motd</code> will change the motto of the day to show NVSM alerts.
NeedsMRRSConfig	<code>Nvidia-mlnx-config</code> uses this attribute to use <code>mlxconfig</code> and sets various PCI settings. Sets <code>MaxReadReq</code> size to 4KB for all Network (2) Infiniband (07) devices on DGX A100, DGX A800 and DGX2 only.
NeedsAccBytesTuning	<code>Nvidia-mlnx-config</code> uses this attribute to use <code>mlxconfig</code> and sets various PCI settings.
IPMIDefaultSerialTTY	Sets a default IPMI serial console port in grub kernel parameters.
NeedsOEMXconfigOverride	For <code>dgxstation_a100</code> or <code>dgxstation_a800</code> , <code>nvidia-conf-xconfig</code> create oem-config override service.
NeedsInitialNvidiaXconfig	For <code>dgxstation_a100</code> or <code>dgxstation_a800</code> , <code>nvidia-conf-xconfig</code> calls <code>nvidia xconfig</code> and creates empty initial configuration.
NeedsAdaptiveNvidiaXconfig	For <code>dgxstation_a100</code> or <code>dgxstation_a800</code> , <code>nvidia-conf-xconfig</code> calls <code>nvidia xconfig</code> and creates empty initial configuration.
NeedsContainerdOverride	The package <code>nv-docker-gpus</code> checks this for <code>dgxstation_a100</code> or <code>dgxstation_a800</code> . In these cases, this package limits <code>nvidia docker</code> to use 3D Controller class GPUs.
NeedsOemConfigPostActNetplanApply	<code>Nvidia-oem-config-postact</code> checks for DCS and DCS legacy platforms, this forces a “netplan apply” after OEM ISO installation.
UsesFabricManager	For platforms DGX2 up to DGX B200, package <code>dgx-release-upgrade</code> checks this to install proper <code>nvidia-fabricmanager</code> package for GPU driver.
IsDgxServerType	During a release upgrade, <code>dgx-release-upgrade</code> checks this to install packages just for DGX servers.
IsDgxDesktopType	During a release upgrade, <code>dgx-release-upgrade</code> checks this to install packages just for DGX Stations (DGX Station A100, etc.)

continues on next page

Table 7 – continued from previous page

Name	Definition
NeedsDisableNumaBalance	In Grace based platforms, it configures the system to disable automatic page fault NUMA memory balancing.
NeedsIommuPt	Set <code>iommu=pt</code> for AMD Rome platforms to enable iommu in passthrough mode on DGX AX00, DGX Station AX00, and DGX HX00 platforms.
NeedsDisableInitOnAlloc	Adds the option <code>init_on_alloc=0</code> to the kernel boot string. Disabling <code>init-on-alloc</code> allows the kernel to allocate a page of memory without initializing it. This improves performance by skipping the initialization of the memory. <code>init-on-alloc</code> is currently disabled for all Grace platforms.
NeedsEarlycon	Adds the option <code>earlycon</code> to the kernel boot string. This sets up the early console with no options. The early console is determined by the <code>stdout-path</code> property in the device tree's chosen node.
PciRealloc	Determine whether to set grub to <code>pci=realloc=on</code> , <code>pci=realloc=off</code> , or not to set <code>pci=realloc</code> at all.
CrashdumpMem	The <code>kdump</code> service uses this value to reserve the crash kernel memory for each kernel. The minimum size of the crash kernel can vary depending on the hardware and machine specifications.
UsePciDisableAcsRedir	The package <code>nvidia-acs-disable</code> configures the system to disable PCIe Access Control Services based on this setting.
ConsoleSetting	This field modifies the kernel command-line console setting.
SerialBaudRate	This value adjusts the console serial port baud rate on the GRUB command line.

Chapter 16. PXE Boot Setup

The dgx-server UEFI BIOS supports PXE boot. Several manual customization steps are required to get PXE to boot the Base OS image.

Caution

This document is meant to be used as a reference. Explicit instructions are not given to configure the DHCP, HTTP, and TFTP servers. The end user's IT team is expected to configure these servers to fit their company's environment and security guidelines.

16.1. Requirements

- ▶ TFTP server
 - ▶ Software that provides TFTP service.
- ▶ HTTP server
 - ▶ An HTTP server is used to transfer large files, such as the iso image and `initrd`. alternatively, TFTP can be used for this purpose. HTTP is used in the example below.
- ▶ DHCP server
 - ▶ Software that provides dynamic host configuration protocol (DHCP) service.

Note

The TFTP server, HTTP server, and DHCP server can all be configured on the same system, or they can each be on different systems.

- ▶ linux bootloader
- ▶ ip address: `<ftp ip>`
- ▶ fully qualified host: `<ftp host>`

This topic provides some guidance concerning how to set up a PXE boot environment for DGX systems. For complete details, refer to online documentation for setting up a PXE boot server. In this example, `xinetd` is used to provide TFTP service; `dnsmasq` is used to provide DHCP service; and `syslinux` is used as the bootloader.

16.2. Overview of the PXE Server

The PXE server requires configuration in the following areas:

- ▶ bootloader (grub)
- ▶ TFTP contents (the kernel and `initrd`)

In this example, TFTP is configured to serve files from `/local/tftp/`. You will need to configure your TFTP server to serve files from `/local/tftp/` or the directory you desire to use.
- ▶ HTTP contents (the iso image)

In this example, HTTP is configured to serve files from `/local/http/`. You will need to configure your HTTP server to serve files from `/local/http/` or the directory you desire to use.
- ▶ DHCP

16.2.1. PXE Server Configuration

x86_64 Instructions

In this example, the directory structure on the HTTP and TFTP server looks like this:

```
/local/
  http/
    base_os_7.0.0/
      base_os_7.0.0.iso
  tftp/
    grub2/
      base_os_7.0.0/
        vmlinuz
        initrd
    grub.cfg
    bootx64.efi
    grubx64.efi
```

Note

The `vmlinuz` and `initrd` files are specified relative to the TFTP root, `/local/tftp/`; and the location of the ISO, `base_os_7.0.0.iso`, is relative to the HTTP root, `/local/http/`.

Here, the DHCP and PXE servers are configured to use the above directory structure. The person responsible for deploying the PXE environment should change the directory names and structure to fit their infrastructure.

You can set up the directory structure on your HTTP and TFTP server similarly.

The contents of the `/local/tftp/grub2/grub.cfg` file should look something like this:

```
set default=0
set timeout=-1
insmod all_video

menuentry 'Install BaseOS 7.0.0' {
```

(continues on next page)

(continued from previous page)

```
linuxefi /grub2/base_os_7.0.0/vmlinuz fsck.mode=skip autoinstall ip=dhcp
→url=http://<Server IP>/base_os_7.0.0/base_os_7.0.0.iso nvme-core.
→multipath=n nouveau.modeset=0
initrdefi /grub2/base_os_7.0.0/initrd
}
```

Note

The kernel boot parameters should match the contents of the corresponding ISO's boot menu found in `/mnt/boot/grub/grub.cfg`.

When the system being installed boots via PXE, boot files located on `/local/tftp` are retrieved from the TFTP server. (In this example, the TFTP server is provided by the `xinetd` service whose configuration file, `/etc/xinetd.d/tftp`, specifies that the boot files are located on `/local/tftp`.) When a system is PXE booted, the `bootx64.efi` file that is designated in the DHCP server's `dhcpd.conf` file is retrieved by TFTP transfer (see [Configure Your DHCP Server](#)). By default, after the `bootx64.efi` is booted, the PXE boot `grub.cfg` file, `grub2/grub.cfg` in this example, provides menu options for booting further. The PXE boot `grub.cfg` config file specifies the locations of the kernel and `initrd` files relative to the `tftp` directory.

► Configure the HTTP Directory:

Configure the HTTP file directory and ISO image by placing a copy of the Base OS 7.0.0 ISO in directory `/local/http/base_os_7.0.0/`. In this example, the full path is `/local/http/base_os_7.0.0/base_os_7.0.0.iso`.

► Configure the TFTP Directory By Using the Following Steps:

Mount the Base OS 7.0.0 ISO. Assume your mount point is `/mnt`:

```
sudo mount -o loop /local/http/base_os_7.0.0/base_os_7.0.0.iso /mnt
```

Copy the kernel and `initrd` from the ISO to the `tftp` directory:

```
cp /mnt/casper/vmlinuz /local/tftp/grub2/base_os_7.0.0/
cp /mnt/casper/initrd /local/tftp/grub2/base_os_7.0.0/
```

Unmount the Base OS 7.0.0 ISO:

```
umount /mnt
```

► Download GRUB Packages For x86_64 and Copy the *.efi shim Binaries for PXE Booting into Place:

```
cd /tmp
wget https://mirror.stream.centos.org/9-stream/BaseOS/x86_64/os/Packages/
→grub2-efi-x64-2.06-94.el9.x86_64.rpm
wget https://mirror.stream.centos.org/9-stream/BaseOS/x86_64/os/Packages/
→shim-x64-15-15.el8_2.x86_64.rpm
```

Note

If either `grub2-efi-x64-2.06-94.el9.x86_64.rpm` or `shim-x64-15-15.el8_2.x86_64.rpm` do not exist on https://mirror.stream.centos.org/9-stream/BaseOS/x86_64/os/

Packages, use the most recent version of either file from the same <https://nvidia.github.io/dgxos/7.0.0/packages.html> location.

Unpack the RPMs with the following commands:

```
rpm2cpio grub2-efi-x64-2.06-94.el9.x86_64.rpm | cpio -idmv
rpm2cpio shim-x64-15-15.el8_2.x86_64.rpm | cpio -idmv
```

Copy the following binaries from the unpacked RPMs to /local/tftp/grub2/:

```
cp -p $(find . -name shimx64.efi) /local/tftp/grub2/shimx64.efi
cp -p $(find . -name grubx64.efi) /local/tftp/grub2/grubx64.efi
cd -
```

Make a copy of shimx64.efi in /local/tftp/grub2/ and name the copy bootx64.efi:

```
cp -p /local/tftp/grub2/shimx64.efi /local/tftp/grub2/bootx64.efi
```

For more information, refer to [How to netboot the server installer on amd64](#).

x86_64 Alternate Instructions If initrd File Transfer Times Out Over TFTP

In some cases, the transfer of the `initrd` can time out over TFTP. A workaround for this is to host the requisite files - `initrd`, `vmlinuz`, and the ISO - over HTTP instead. Hosting these over HTTP makes the transfer faster and more reliable. In this example, assume the HTTP server files are hosted from /local/http. The directory structure on the HTTP and TFTP server looks like this:

```
/local/
  http/
    base_os_7.0.0/
      base_os_7.0.0.iso
      vmlinuz
      initrd
  tftp/
    grub2/
      grub.cfg
      bootx64.efi
      grubx64.efi
```

Note

In this alternate configuration method, the `vmlinuz` and `initrd` file and the ISO, `base_os_7.0.0.iso` are now all specified relative to the HTTP root, /local/http/.

When configured this way, the `grub.cfg` file will contain text that will look something like this:

```
set default=0
set timeout=-1
insmod all_video

menuentry 'Install BaseOS 7.0.0' {
  linuxefi (http,<HTTP Server IP>)/base_os_7.0.0/vmlinuz fsck.mode=skip
  ↪autoinstall ip=dhcp url=http://<Server IP>/base_os_7.0.0/base_os_7.0.0.iso
  ↪nvme-core.multipath=n nouveau.modeset=0
```

(continues on next page)

(continued from previous page)

```
initrdefi (http,<HTTP Server IP>)/base_os_7.0.0/initrd
}
```

Note

The kernel boot parameters should match the contents of the corresponding ISO's boot menu, found in /mnt/boot/grub/grub.cfg.

ARM64 Instructions

Download the relevant grub package with the correct architecture specified:

```
wget https://mirror.stream.centos.org/9-stream/BaseOS/aarch64/os/Packages/
↳ grub2-efi-aa64-2.06-94.el9.aarch64.rpm
```

Unpack the RPMs with the following commands:

```
rpm2cpio grub2-efi-aa64-2.06-94.el9.aarch64.rpm | cpio -idmv
```

Copy the following binary from the unpacked RPM to /local/tftp/grub2/:

```
grubaa64.efi
```

An example of the directory structure on the HTTP and TFTP server could be similar to:

```
/local/
  http/
    base_os_7.0.0/
      base_os_7.0.0.iso
      vmlinuz
      initrd
  tftp/
    grub2/
      grub.cfg
      bootx64.efi
      grubaa64.efi
```

When configured this way, the grub.cfg file would contain lines similar to:

```
set default=0
set timeout=-1
insmod all_video

menuentry 'Install BaseOS 7.0.0' {
  linuxefi (http,<HTTP Server IP>)/base_os_7.0.0/vmlinuz fsck.mode=skip
  ↳ autoinstall ip=dhcp url=http://<Server IP>/base_os_7.0.0/base_os_7.0.0.iso
  ↳ nvme-core.multipath=n nouveau.modeset=0
  initrdefi (http,<HTTP Server IP>)/base_os_7.0.0/initrd
```

Note

The kernel boot parameters should match the contents of the corresponding ISO's boot menu, found in `/mnt/boot/grub/grub.cfg`.

16.3. TFTP and HTTP Server Verification

After you have set up all elements of your PXE server, prior to doing a PXE install, verify that the TFTP and HTTP servers are working properly.

TFTP Server Verification

To verify that the TFTP server has been set up correctly, from a different system on the same subnet, use `tftp` to get one of the files that will be obtained via `tftp` during the PXE boot. In this example, the TFTP server has been set up to serve files from `/local/tftp`. The grub configuration file, `grub.cfg` is located on `/local/tftp/grub2/grub.cfg`; therefore, from the TFTP command prompt, request `grub.cfg` via `get grub2/grub.cfg`.

```
cd /tmp
tftp <TFTP_Server_IP>
get grub2/grub.cfg
quit
```

HTTP Server Verification

To verify that the HTTP server has been set up correctly, use the `wget` command to get one of the files that will be obtained via HTTP during the PXE boot. In this example, the HTTP server has been set up to serve files from `/local/http`. The ISO, `base_os_7.0.0.iso`, is located on `/local/http/base_os_7.0.0/base_os_7.0.0.iso`; therefore, test the HTTP request to retrieve `base_os_7.0.0.iso` by running the following commands:

```
cd /tmp
wget http://<HTTP_Server_IP>/baseos-7/base_os_7.0.0.iso
```

16.4. Useful Parameters for Configuring Your System's Network Interfaces

`ip=dhcp`: tells the initramfs to automatically configure the system's interfaces using DHCP.

- ▶ If only one interface is connected to the network, this should be sufficient.
- ▶ If multiple interfaces are connected to the network, the first interface that receives a reply will be used.

16.5. Parameters Unique to the Base OS Installer

- ▶ `rebuild-raid` tells the installer to rebuild the data RAID if specified. Installing from the factory should always specify this, but it is optional otherwise.

- ▶ `md5checkdisc` will not perform an installation when this is specified. It will simply unpack the ISO and check that its contents match what is described in `md5sum.txt`.
- ▶ `offwhendone` powers off the system after the installation. Otherwise, we reboot when done. Factory installs will specify this.
- ▶ `nooemconfig` skip `oemconfig` and create default user `nvidia`, seeding initial password. Used for touchless install in PXE install or automatic VM creation/installation.
- ▶ `force-ai` allows you to supply your autoinstall file. If the networking is set up, you can provide a URL. Otherwise, this has to be one that exists in the installer.

For example:

```
force-ai=/ai/dgx_a100-ai.yaml
force-ai=http://your-server.com/your-ai.yaml
```

Note

Refer to the note the [Autoinstall Customizations](#) section for special formatting considerations when using custom autoinstall files along with the `force-ai` parameter.

16.6. Configure Your DHCP Server

The DHCP server is responsible for providing the IP address of the TFTP server and the name of the bootloader file in addition to the usual role of providing dynamic IP addresses. The address of the TFTP server is specified in the DHCP configuration file as `next-server`, and the bootloader file is specified as `filename`. The architecture option can be used to detect the architecture of the client system and to serve the correct version of the grub bootloader (x86, IA-32, ARM, and so on).

An example of the PXE portion of `dhcpd.conf` is:

```
class "pxeclients" {
match if substring (option vendor-class-identifier, 0, 9) = "PXEClient";

next-server <TFTP_Server_IP>;

    # x86 UEFI
    if option arch = 00:06 {
        filename "grub2/bootx64.efi";
    # x64 UEFI
    } else if option arch = 00:07 {
        filename "grub2/bootx64.efi";
    # ARM 64-bit UEFI
    } else if option arch = 00:0b {
        filename "grub2/grubaa64.efi";
    } else {
        filename "pxelinux.0";
    }
}
```

An example of the subnet portion for the system of `dhcpd.conf` is shown below. The target network is 10.10.10.0. The DHCP server will assign the specific IP address from the range specified. Specific IP

addresses can be configured. Consult the PXE installation setup documentation instructions for your DHCP server for more detailed information.

```
subnet 10.10.10.0 netmask 255.255.255.0

{ option broadcast-address 10.10.10.255;
  option routers 10.10.10.1;
  option subnet-mask 255.255.255.0;
  option interface-mtu 9000;
  range 10.10.10.200 10.10.10.250; }
```

16.7. (Optional) Configure CX-4/5/6/7 Cards to PXE Boot

DGX-Servers may also PXE boot using the MLNX CX-4/5/6 cards. If you are logged into the DGX-Server host OS, and running DGX Base OS 4.4 or later, you can perform this section's steps using the `/usr/sbin/mlnx_pxe_setup.bash` tool, which will enable the UEFI PXE ROM of every MLNX Infiniband device found.

Otherwise, proceed with the manual steps below.

16.7.1. Enable UEFI for ConnectX Cards

To PXE boot from the MLNX ConnectX-4/5/6/7 cards, you must first enable the UEFI PXE ROM of the card you wish to PXE boot from because it is disabled by default. This needs to be performed from the DGX Server host OS itself, it cannot be done remotely.

Follow these steps using the DOCA OFED software, which allows you to install the same drivers and tools of MLNX_OFED.

1. To determine the device name and current configurations of the MLNX ConnectX cards, run the `sudo mlxconfig query` command:

```
$ sudo mlxconfig query

Device #1:
-----

Device type:    ConnectX4
Name:           MCX455A-ECA_Ax
Description:    ConnectX-4 VPI adapter card; EDR IB (100Gb/s) and 100GbE;
→single-port QSFP28; PCIe3.0 x16; ROHS R6
Device:         /dev/mst/mt4115_pciconf3

Configurations:                                Next Boot
...
...
EXP_ROM_UEFI_ARM_ENABLE                        False(0)
EXP_ROM_UEFI_x86_ENABLE                        False(0)
...
...
```

2. Enable the ConnectX card's UEFI PXE ROM for your system's architecture:

```
user@dgxserver$ sudo mlxconfig -y -d /dev/mst/mt4115_pciconf3 set EXP_ROM_
→UEFI_x86_ENABLE=1
user@dgxserver$ sudo mlxconfig -y -d /dev/mst/mt4115_pciconf3 set EXP_ROM_
→UEFI_ARM_ENABLE=1
```

3. Reboot the server for the change to take effect.

```
user@dgxserver$ sudo reboot
```

4. Upon reboot, confirm the configuration was set.

```
user@dgxserver$ sudo mlxconfig query

Device #1:
-----

Device type:    ConnectX4
Name:           MCX455A-ECA_Ax
Description:     ConnectX-4 VPI adapter card; EDR IB (100Gb/s) and 100GbE;
→single-port QSFP28; PCIe3.0 x16; ROHS R6
Device:         /dev/mst/mt4115_pciconf3

Configurations:                                Next Boot
...
...
EXP_ROM_UEFI_ARM_ENABLE                        True(1)
EXP_ROM_UEFI_x86_ENABLE                        True(1)
...
...
```

16.8. (Optional) Configure the DGX-Server to PXE Boot Automatically

16.8.1. Add PXE to the Top of the UEFI Boot Order

On systems with a BMC, you can specify the DGX-Server to PXE boot by adding it to the top of the UEFI boot order. This may be done out-of-band via IPMI.

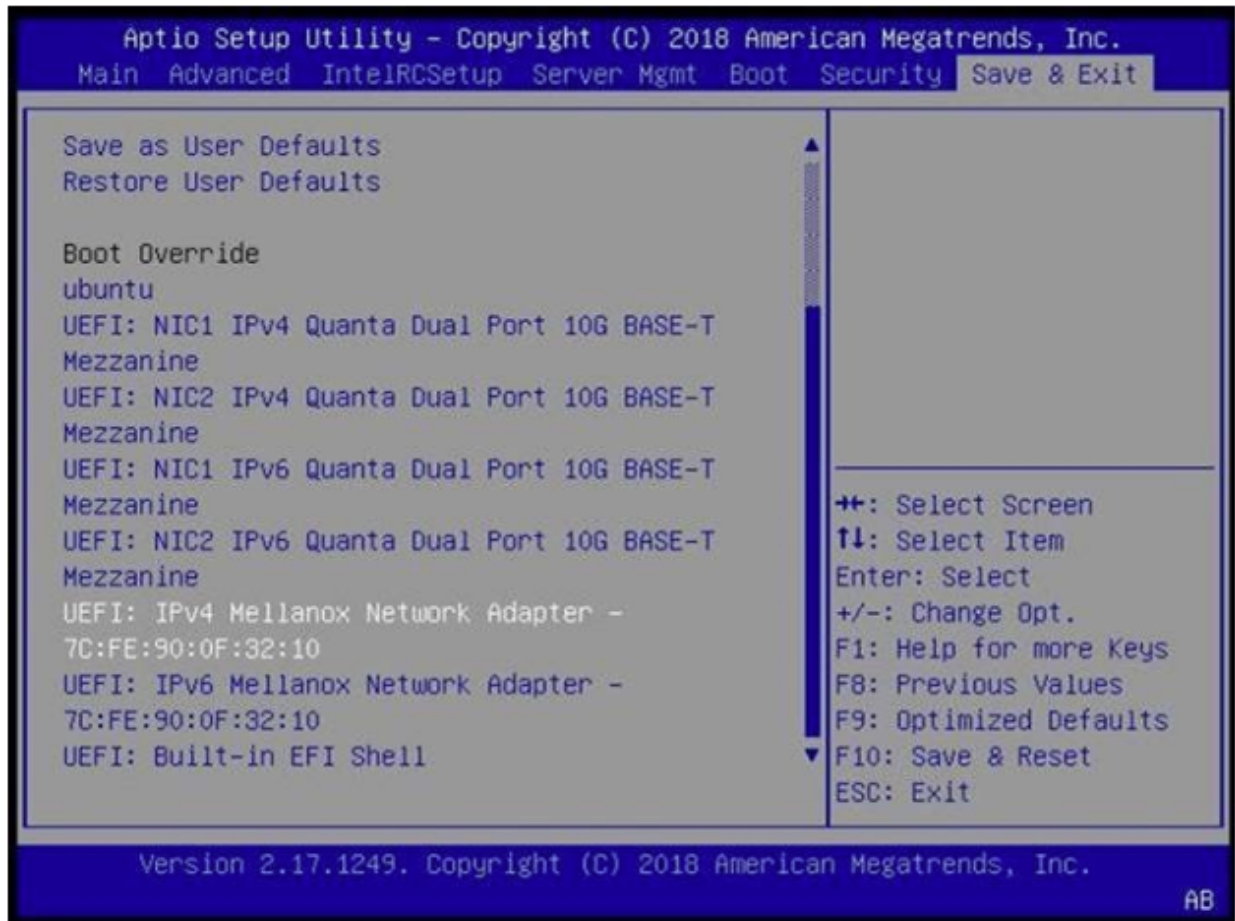
```
ipmitool -I lanplus -H <DGX_BMC_IP> -U <ADMIN> -P <PASSWORD> chassis bootdev
→pxe options=efiboot
```

Note

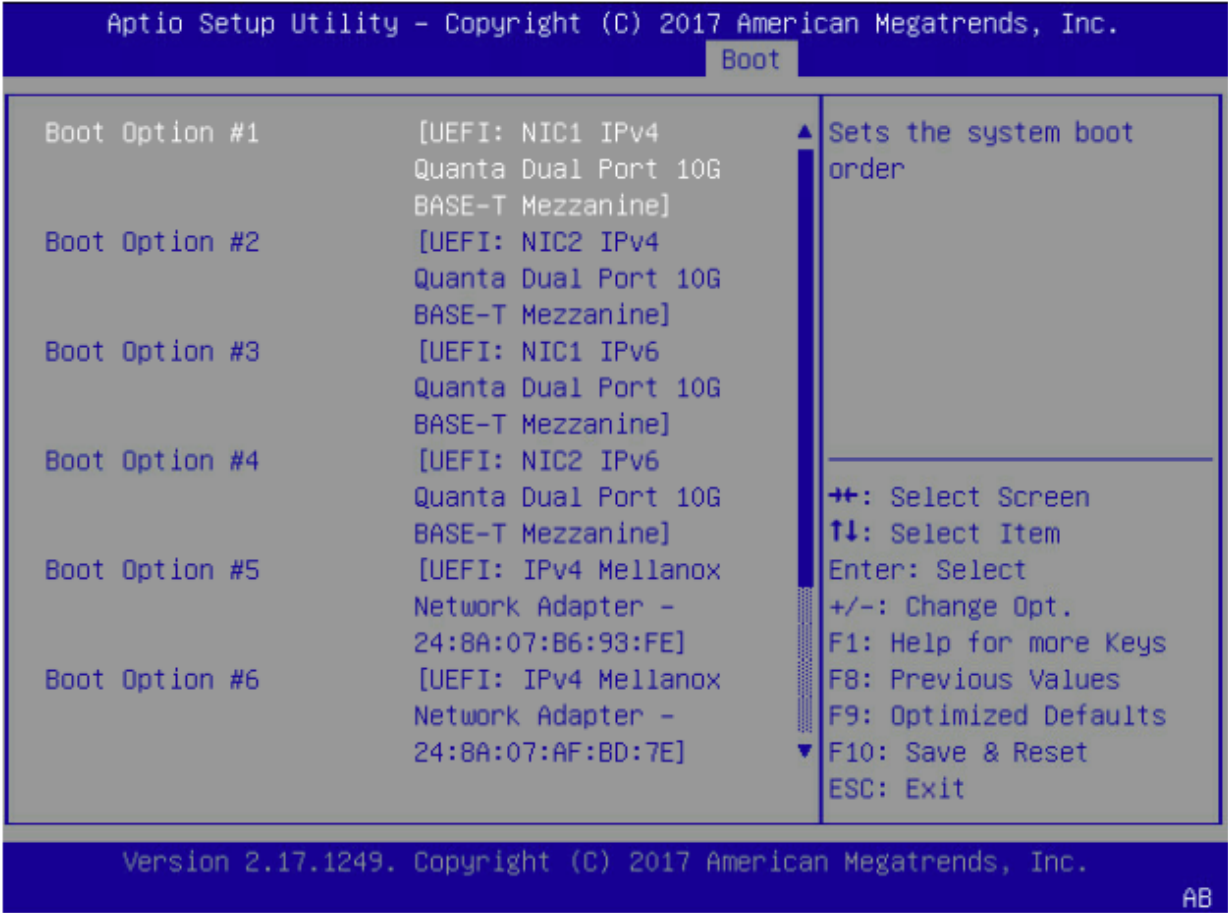
This only sets the DGX-Server to PXE boot, but doesn't specify the order of network devices to attempt PXE from. This is a limitation of our current UEFI and BMC FW. See the following section to specify the network device boot order.

16.9. Configure Network Boot Priorities

The UEFI Network Drive BBS Priorities allows you to specify the order of network devices to PXE boot from. To modify this, you must reboot your DGX-Server and enter the UEFI boot selection menu by pressing **F2** or **Del** when you see the splash screen. Navigate to the **Boot** menu, and then scroll down to **UEFI NETWORK Drive BBS Priorities**.

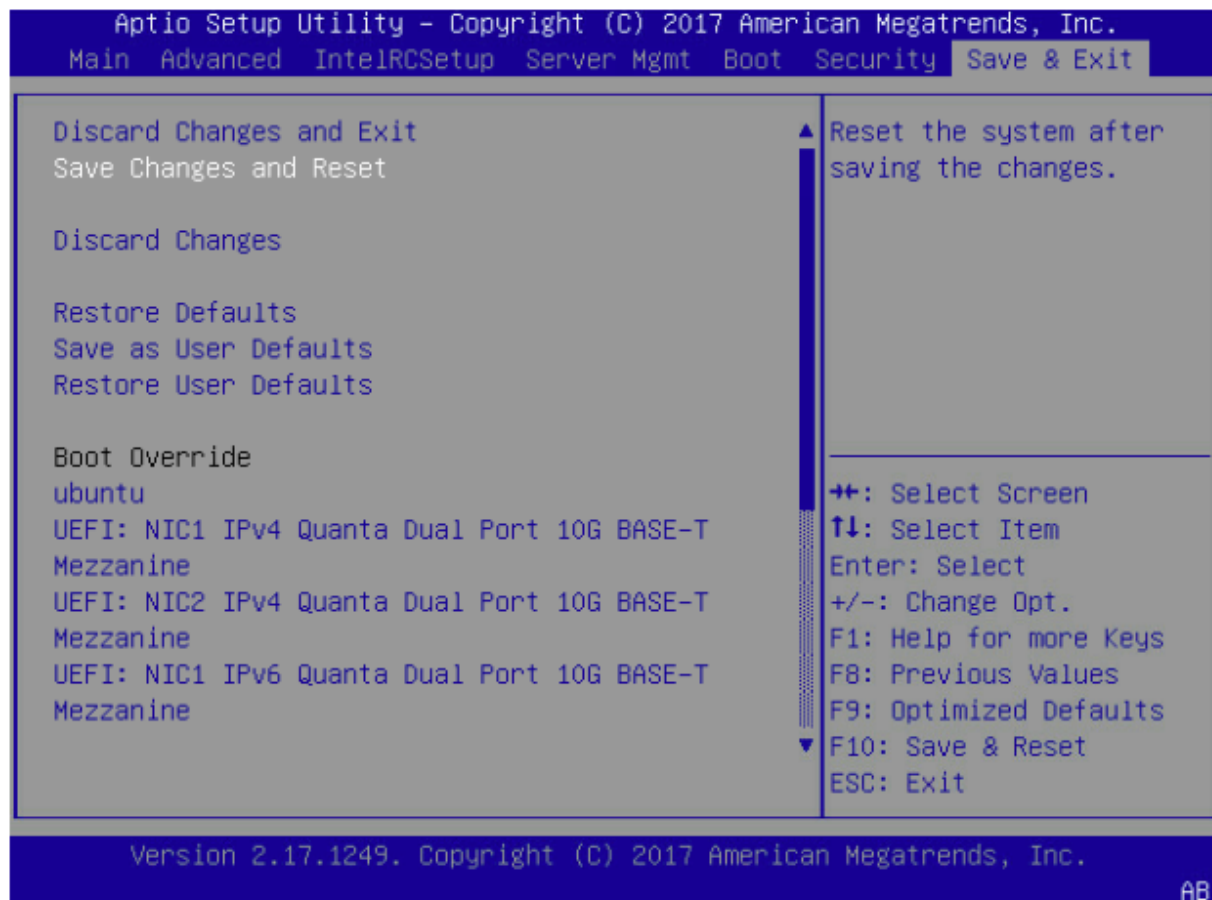


Configure the order of devices to attempt network boots from using this menu.



Save and Exit.

After you finished ordering the network boot priorities, save your changes and reset.



16.10. Make the DGX-Server PXE Boot

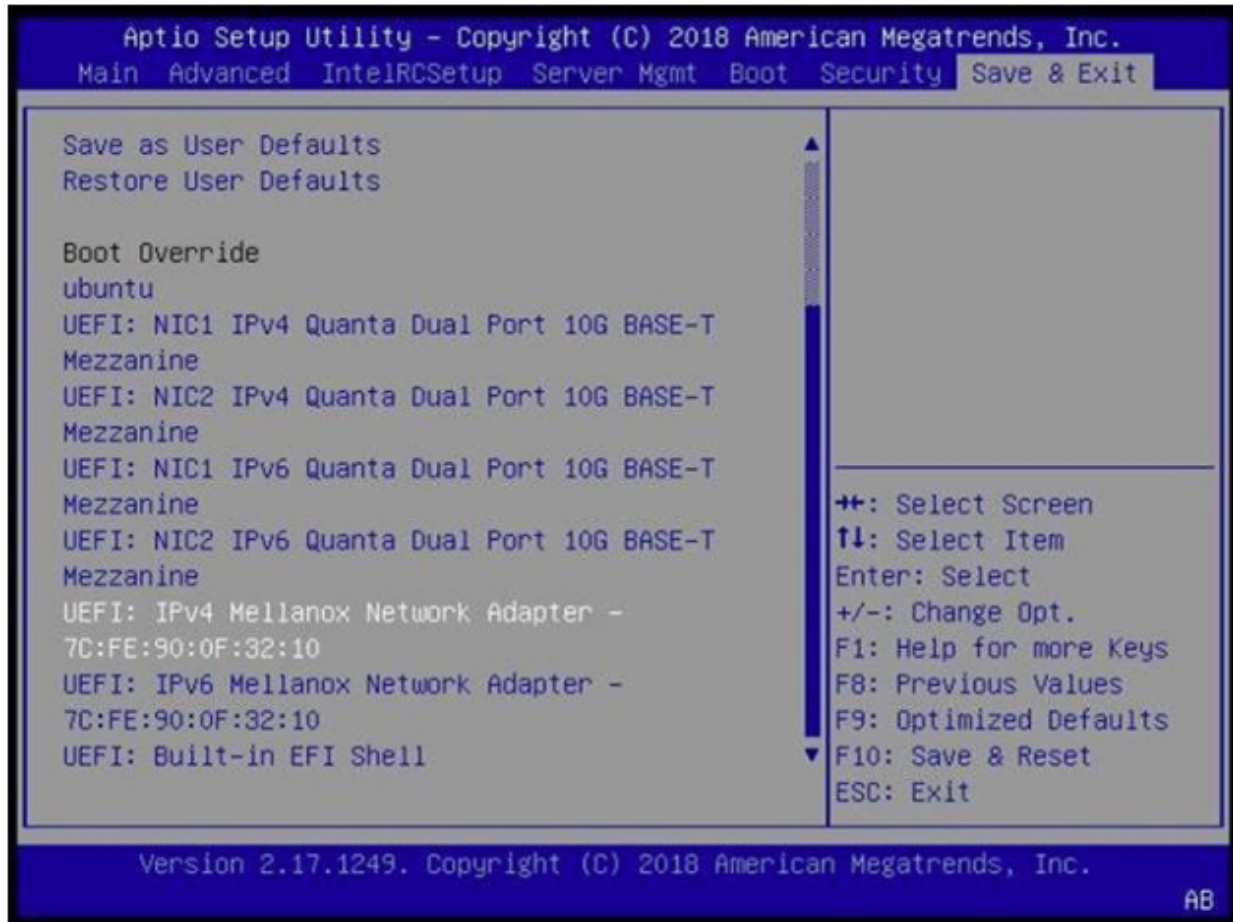
16.10.1. Automated PXE Boot Process

If you followed the optional steps above, you can now reboot and UEFI will attempt PXE boot using the devices in order specified in the Network Drive BBS Priorities list.

16.10.2. Manual PXE Boot Process

If you want to manually trigger the PXE boot, reboot your DGX-Server and enter the UEFI boot selection menu by pressing **F2** or **Del** when you see the splash screen.

Navigate to the **Save & Exit** menu, scroll down to the Boot Override section, and choose the appropriate network port to boot from. The MLNX cards will only appear if you enable the UEFI PXE ROM of that card.



Alternatively, you can press **F12** at the SBIOS splash screen, and SBIOS will iterate thru each NIC and try PXE on each one. The order of the NICs attempted is specified by the Network Drive BBS Priorities.

16.11. Other IPMI Boot Options

For more information about specifying boot order via IPMI, see the `chassis` command and `bootdev` subcommand in the `ipmitool` man page.

For more information about the IPMI specification, refer to [Intelligent Platform Management Interface Specification v2.0 rev. 1.1](#).

16.12. Autoinstall Customizations

There are many autoinstall files that you can reference inside the Base OS 7.x ISO; these are contained in:

```
casper/ubuntu-server-minimal.ubuntu-server.installer.kernel.nvidia.squashfs
```

You can mount the ISO and then mount this squashfs to view the many autoinstall files that are packed within:

```
mkdir -p /tmp/iso_mnt
mkdir -p /tmp/squash_mnt
sudo mount /path/to/DGXOS-<version>-<date>.iso /tmp/iso_mnt/
sudo mount /tmp/iso_mnt/casper/ubuntu-server-minimal.ubuntu-server.installer.
↪kernel.nvidia.squashfs /tmp/squash_mnt/
find /tmp/squash_mnt/ai/ -name '*.yaml'
```

For some deployments, you might want to use their own autoinstall files. This section will describe some sections contained in the built-in autoinstall files as well as how to perform some common customizations.

Note

The installer expects a unified autoinstall file rather than the typical split vendor/user/meta-data format. This means that the user-supplied autoinstall file will need to account for some formatting differences, namely, the `autoinstall:` keyword needs to be dropped and the indentations adjusted accordingly:

```
#
# typical user-data file
#
#cloud-config
autoinstall:
  version: 1
  identity:
    realname: 'DGX User'
    username: dgxuser
    password: '$6$g3vXaGj.MQpP/inN$16.
↪JtAueRAfMtQweK7qASjxXiEX8Vue3CvRcwON81Rt9BJm1EQKtnf0VSnCqHrTsy88PbMDDHq6k.
↪iM6PWfHr1'

#
# unified autoinstall file
#
version: 1
identity:
  realname: 'DGX User'
  username: dgxuser
  password: '$6$g3vXaGj.MQpP/inN$16.
↪JtAueRAfMtQweK7qASjxXiEX8Vue3CvRcwON81Rt9BJm1EQKtnf0VSnCqHrTsy88PbMDDHq6k.
↪iM6PWfHr1'
```

16.13. NVIDIA-Specific Autoinstall Variables

The autoinstall files contained in the ISO are platform-specific, and serve as a good starting point for custom versions. Many of them contain variables, prefixed with `CHANGE_`, which will be substituted by the installer:

- `CHANGE_STORAGE_REG` This gets removed and uncommented when the boot parameter `ai-encrypt-root` is not present. Uncommenting this stanza results in the standard disk partitioning scheme without LUKS encryption.

- **CHANGE_STORAGE_ENC** This gets removed and uncommented when the boot parameter `ai-encrypt-root` is present. Uncommenting this stanza results in an encrypted root partition.
- **CHANGE_BOOT_DISK_NAME_x** This is a disk-name, without the `/dev` prefix. There may be multiple ones (for example, `CHANGE_BOOT_DISK_NAME_1` and `CHANGE_BOOT_DISK_NAME_2`) for platforms that expect a RAIDed boot device as is the case for DGX A100.

Note

The installer will find the appropriate disk name to substitute here. Alternatively, the `force-bootdisk` parameter can be used to specify the disk name(s).

- **CHANGE_BOOT_DISK_PATH_x** This is the same as the `CHANGE_BOOT_DISK_NAME_x` variable above, except that it is prefixed with `/dev/`.
- **CHANGE_DESC_PLATFORM** The installer will substitute this with a platform-specific descriptive name.
- **CHANGE_SERIAL_NUMBER** The installer will substitute this with the serial number reported by `dmidecode`.
- **CHANGE_INSTALL_PKGS** The installer will substitute this value with a list of packages specific to the platform. The lists of packages are specified by the `*-pkgs` files in the squashfs.

Note

The list of packages here will include `oem-config` and its dependencies. When you supply your own autoinstall file, you also want to perform the additional setup steps provided by `oem-config` and have these steps performed during autoinstall instead. For this use case we recommend adding, in the `late-commands` section, a step to remove the `oem-config` and `ubiquity` packages:

```
late-commands:
...
- curtin in-target -- apt-get purge -y oem-config ubiquity
```

- **CHANGE_REBUILD_RAID** This gets replaced with either `true` or `false` based on whether or not the `rebuild-raid` boot parameter is present.
- **CHANGE_IPMISOL** This gets replaced with either `true` or `false` based on whether or not the `ai-encrypt-root` boot parameter is present. When we set the system up with encryption, we also undo the IPMI serial-over-LAN configuration to ensure that the LUKS passphrase prompt shows up on the console rather than the serial-over-LAN interface.

Attention

While it is possible to replace these values on your own, we strongly recommend letting the installer handle this.

16.14. Common Customizations

In this section, we will describe some common customizations that may be useful in more custom deployments.

16.15. Network Configuration

To configure the network at install time, you can add a `network` section to your autoinstall file. In this example we will create a netplan configuration file that sets the `enp1s0f0` interface to use DHCP:

```
network:
  version: 2
  ethernets:
    enp1s0f0:
      dhcp4: yes
```

16.16. Creating a User

To create a user at install time, you can add an `identity` section to your autoinstall file. The following example sets the system hostname to `dgx` and creates a user `nvidia` with the password `nvidia`.

```
# To generate an encrypted password:
#   printf '<plaintext_password>' | openssl passwd -6 -stdin
#
# For example:
#   printf 'nvidia' | openssl passwd -6 -stdin
identity:
  hostname: dgx
  password: $6$8fqF54QDoaLMtDXJ
  ↪$J02iNH1xW9hHtzH6APpUX4X4HkRx2xY2ZKy9DQpG0QhW700uTk3DwHr9FnAAh1JIyqn3L277Jy9MEzW4MyVsV0
  username: nvidia
```

For more examples, refer to the [Autoinstall configuration reference manual](#) in the Ubuntu installation documentation.

Chapter 17. Air-Gapped Installations

For security purposes, some installations require that systems be isolated from the internet or outside networks.

An air-gapped system is not connected to an unsecured network, such as the public Internet, to an unsecured LAN, or to other computers that are connected to an unsecured network. The default mechanisms to update software on DGX systems and loading container images from the NGC Container Registry require an Internet connection. On an air-gapped system, which is isolated from the Internet, you must provide alternative mechanisms to update software and load container images.

Since most DGX software updates are completed through an over-the-network process with NVIDIA servers, this section explains how updates can be made when using an over-the-network method is not an option. It also includes a process to install Docker containers.

Here are the methods you can use:

- Download the ISO image, copy it to removable media and then reimage the DGX System from the media.

This method is available only for software versions that are available as ISO images for download. For details, see [Reimaging the System](#). This section provides information about how to install the DGX OS.

- Update the DGX software by performing a network update from a local repository.

This method is available only for software versions that are available for over-the-network updates.

17.1. Creating a Local Mirror of the NVIDIA and Canonical Repositories

Here are the steps to download the necessary packages to create a mirror of the repositories that are needed to update NVIDIA DGX systems. For more information on DGX OS versions and the release notes available, refer to [Release Notes](#).

Note

These procedures apply only to upgrades in the same major release, such as from 6.x to 6.y. The steps do not support upgrades across major releases, such as from DGX OS 6 to DGX OS 7.

1. Identify the sources that correspond to the public NVIDIA and Canonical repositories that provide updates to the DGX OS.

You can identify these sources from the `/etc/apt/sources.list` file and the contents of the `/etc/apt.sources.list.d/` directory, or by using [System Settings, Software & Updates].

2. Create and maintain a private mirror of the repository sources that you identified in the previous step.
3. Update the sources that provide updates to the DGX system to use your private repository mirror instead of the public repositories.

To update these sources, modify the `/etc/apt/sources.list` file and the contents of `/etc/apt.sources.list.d/` directory.

17.2. Creating the Mirror of the Repositories

The instructions in this section are to be performed on a system with internet access.

- ▶ A system installed with Ubuntu OS is needed to create the mirror because there are several Ubuntu tools that need to be used.
- ▶ You must be logged in to the system installed with Ubuntu OS as an administrator user because this procedure requires `sudo` privileges.
- ▶ The system must contain enough storage space to replicate the repositories to a file system. The space requirement could be as high as 250 GB.
- ▶ An efficient way to move a large amount of data is needed, for example, shared storage in a DMZ, or portable USB drives that can be brought into the air-gapped area.

The data will need to be moved to the systems that need to be updated. Make sure that any portable drives are formatted using ext4 or FAT32.

To create the mirror:

1. Ensure that the storage device is attached to the system with network access and identify the mount point of the device.

Here is a sample mount point that was used in these instructions:

```
/media/usb/repository
```

2. Install the `apt-mirror` package.

```
sudo apt update
```

```
sudo apt install apt-mirror
```

3. Change the ownership of the target directory to the `apt-mirror` user in the `apt-mirror` group.

```
sudo chown apt-mirror:apt-mirror /media/usb/repository
```

The target directory must be owned by the user `apt-mirror` or the replication will not work.

4. Configure the path of the destination directory in `/etc/apt/mirror.list` and use the included list of repositories below to retrieve the packages for both Ubuntu base OS and the NVIDIA DGX OS packages.

For x86_64 systems:


```
##### config #####
#
set base_path /media/usb/repository #/your/path/here
#
# set mirror_path $base_path/mirror
# set skel_path $base_path/skel
# set var_path $base_path/var
# set cleanscript $var_path/clean.sh
# set defaultarch <running host architecture>
# set postmirror_script $var_path/postmirror.sh
set run_postmirror 0
set nthreads 20
set _tilde 0
#
##### end config #####
# Standard Canonical package repositories:
deb http://security.ubuntu.com/ubuntu noble-security main multiverse
→universe restricted
deb http://archive.ubuntu.com/ubuntu/ noble main multiverse universe
→restricted
deb http://archive.ubuntu.com/ubuntu/ noble-updates main multiverse
→universe restricted
#
deb-i386 http://security.ubuntu.com/ubuntu noble-security main multiverse
→universe restricted
deb-i386 http://archive.ubuntu.com/ubuntu/ noble main multiverse universe
→restricted
deb-i386 http://archive.ubuntu.com/ubuntu/ noble-updates main multiverse
→universe restricted
#
# CUDA specific repositories:
deb http://developer.download.nvidia.com/compute/cuda/repos/ubuntu2404/
→x86_64/ /
#
# DGX specific repositories:
deb http://repo.download.nvidia.com/baseos/ubuntu/noble/x86_64/ noble
→common dgx
deb http://repo.download.nvidia.com/baseos/ubuntu/noble/x86_64/ noble-
→updates common dgx
#
deb-i386 http://repo.download.nvidia.com/baseos/ubuntu/noble/x86_64/
→noble common dgx
deb-i386 http://repo.download.nvidia.com/baseos/ubuntu/noble/x86_64/
→noble-updates common dgx
# Clean unused items
clean http://archive.ubuntu.com/ubuntu
clean http://security.ubuntu.com/ubuntu
```

For ARM64 systems:

```
##### config #####
#
set base_path /media/usb/repository #/your/path/here
```

(continues on next page)

(continued from previous page)

```
#
# set mirror_path $base_path/mirror
# set skel_path $base_path/skel
# set var_path $base_path/var
# set cleanscript $var_path/clean.sh
# set defaultarch <running host architecture>
# set postmirror_script $var_path/postmirror.sh
set run_postmirror 0
set nthreads 20
set _tilde 0
#
##### end config #####
# Standard Canonical package repositories:
deb http://ports.ubuntu.com/ubuntu-ports/ noble-security main multiverse
→universe restricted
deb http://ports.ubuntu.com/ubuntu-ports/ noble main multiverse universe
→restricted
deb http://ports.ubuntu.com/ubuntu-ports/ noble-updates main multiverse
→universe restricted
#
# CUDA specific repositories:
deb http://developer.download.nvidia.com/compute/cuda/repos/ubuntu2404/
→sbsa/ /
#
# DGX specific repositories:
deb http://repo.download.nvidia.com/baseos/ubuntu/noble/arm64/ noble
→common dgx
deb http://repo.download.nvidia.com/baseos/ubuntu/noble/arm64/ noble-
→updates common dgx
#
# Clean unused items
clean http://ports.ubuntu.com/ubuntu-ports/
```

5. Run `apt-mirror` and wait for it to finish downloading content.

This will take a long time depending on the network connection speed.

```
sudo apt-mirror
```

6. Eject the removable storage with all packages.

```
sudo eject /media/usb/repository
```

17.3. Configuring the Target Air-Gapped System

Here are the steps that explain how you can configure a target air-gapped DGX OS 7 system.

The instructions in this section are to be performed on the target air-gapped DGX system.

- ▶ The target air-gapped DGX system is installed, has gone through the first boot process, and is ready to be updated with the latest packages.
- ▶ The USB storage device on which the mirrors were created is attached to the target DGX system.

There are other ways to transfer the data that are not covered in this document as they will depend on the data center policies for the air-gapped environment.

1. Mount the storage device on the air-gapped system to `/media/usb/repository` for consistency. 2. Configure the `apt` command to use the file system as the repository in the file `/etc/apt/sources.list` by modifying the following lines.

For x86_64 systems:

```
deb file:///media/usb/repository/mirror/security.ubuntu.com/ubuntu noble-
→security main multiverse universe restricted
deb file:///media/usb/repository/mirror/archive.ubuntu.com/ubuntu/ noble
→main multiverse universe restricted
deb file:///media/usb/repository/mirror/archive.ubuntu.com/ubuntu/ noble-
→updates main multiverse universe restricted
```

For ARM64 systems:

```
deb file:///media/usb/repository/mirror/ports.ubuntu.com/ubuntu-ports/
→noble-security main multiverse universe restricted
deb file:///media/usb/repository/mirror/ports.ubuntu.com/ubuntu-ports/
→noble main multiverse universe restricted
deb file:///media/usb/repository/mirror/ports.ubuntu.com/ubuntu-ports/
→noble-updates main multiverse universe restricted
```

2. Configure `apt` to use the NVIDIA DGX OS packages in the `/etc/apt/sources.list.d/dgx.list` file.

For x86_64 systems:

```
deb file:///media/usb/repository/mirror/repo.download.nvidia.com/baseos/
→ubuntu/noble/x86_64/ noble main dgx
deb file:///media/usb/repository/mirror/repo.download.nvidia.com/baseos/
→ubuntu/noble/x86_64/ noble-updates main dgx
```

For ARM64 systems:

```
deb file:///media/usb/repository/mirror/repo.download.nvidia.com/baseos/
→ubuntu/noble/arm64/ noble main dgx
deb file:///media/usb/repository/mirror/repo.download.nvidia.com/baseos/
→ubuntu/noble/arm64/ noble-updates main dgx
```

3. Configure `[apt]` to use the NVIDIA CUDA packages in the `/etc/apt/sources.list.d/cuda-compute-repo.list` file.

For x86_64 systems:

```
deb file:///media/usb/repository/mirror/developer.download.nvidia.com/
→compute/cuda/repos/ubuntu2404/x86_64/ /
```

For ARM64 systems:

```
deb file:///media/usb/repository/mirror/developer.download.nvidia.com/
→compute/cuda/repos/ubuntu2404/sbsa/ /
```

4. Update the `apt` repository.

```
sudo apt update
```

The output from this command is similar to the following example.

```
Get:1 file:/media/usb/repository/mirror/security.ubuntu.com/ubuntu noble-  
→security InRelease [107 kB]  
Get:2 file:/media/usb/repository/mirror/archive.ubuntu.com/ubuntu noble  
→InRelease [265 kB]  
Get:3 file:/media/usb/repository/mirror/archive.ubuntu.com/ubuntu noble-  
→updates InRelease [111 kB]  
Get:4 file:/media/usb/repository/mirror/developer.download.nvidia.com/  
→compute/cuda/repos/ubuntu2404/x86_64 InRelease  
Get:5 file:/media/usb/repository/mirror/repo.download.nvidia.com/baseos/  
→ubuntu/noble/x86_64 noble InRelease [12.5 kB]  
Get:6 file:/media/usb/repository/mirror/repo.download.nvidia.com/baseos/  
→ubuntu/noble/x86_64 noble-updates InRelease [12.4 kB]  
Get:7 file:/media/usb/repository/mirror/developer.download.nvidia.com/  
→compute/cuda/repos/ubuntu2404/x86_64 Release [697 B]  
Get:8 file:/media/usb/repository/mirror/developer.download.nvidia.com/  
→compute/cuda/repos/ubuntu2404/x86_64 Release.gpg [836 B]  
Reading package lists... Done
```

5. Upgrade the system using the newly configured local repositories.

```
sudo apt full-upgrade
```

Chapter 18. Cloud-init Configuration File

This section provides instructions for creating a cloud-init configuration file for the [Ubuntu Automated Server Installation](#).

18.1. Modifying the Configuration File

The following instructions provide an outline of the example configuration file. It does not work as is but requires additional modifications as described in the sections. For more information, refer to [Ubuntu AUtomed Server Installation](#).

1. Begin the configuration file with the following header:

```
#cloud-config
autoinstall:
  version: 1
```

2. Define a default user (the example uses Ubuntu), localization, and keyboard layout.

```
##
## Set initial system and user information
## use mkpasswd -m sha-512 <password> to create a password
##
identity:
  realname: DGX Ubuntu User
  hostname: dgx-host
  password: <PASSWORD HASH>
  username: ubuntu
locale: en_US
keyboard:
  layout: en
  variant: us
reporting:
  builtin:
  type: print
```

3. The network section describes the network configuration and supports fixed addresses, DHCP, and various other network options. The names of the network interfaces are system-dependent. These are the primary management ports for various DGX systems. For example:
 - DGX A100: enp226s0

- ▶ DGX H100/H200: eno3
- ▶ DGX B200: eno3
- ▶ DGX GB200: enP5p9s0
- ▶ DGX GB300: enP5p9s0

```
##
## Network Configuration
##
network:
  version: 2
  ethernet:
    enp1s0f0:
      dhcp4: yes
```

4. Update the Subiquity installer to the edge channel.

```
refresh-installer:
  channel: edge
  update: yes
```

5. Provide details about the additional NVIDIA repositories. Refer to [Drive Partitioning](#) below for more information.

```
##
## Enable this for using the remote repositories
##
apt:
  <Repository details for the CUDA Compute and DGX Repository>

  conf: |
    Dpkg::Options {
      "--force-confdef";
      "--force-confold";
```

6. Configure storage.

The next section describes the storage configuration, including swap configuration and drive partitioning. By setting the size to 0, we disable the SWAP partition. Refer to [Drive Partitioning](#).

The `reorder_uefi` flag tells the installer not to change the boot order to place the currently booted entry (BootCurrent) to the first option.

```
##
## Storage Configuration
##
storage:
  config:
    <Partition and other configurations>
  swap:
    size: 0
  grub:
    reorder_uefi: false
```

7. Enable the SSH server.

You can also set a default SSH key.

```
##
## SSH Server
##
ssh:
  install-server: yes
  allow-pw: yes
```

8. Provide a list of packages that should be installed.

Refer to the comments in this text for instructions on changing the package names for specific DGX systems and on enabling or disabling features.

```
##
## Packages
##
packages:

##
## NVIDIA DGX system configurations and system tools
## Replace dgx-a100 for other DGX systems:
## dgx1      for DGX-1
## dgx2      for DGX-2
## dgx-a100  for DGX A100
## dgx-h100  for DGX H100
##
- dgx-a100-system-configurations
- dgx-a100-system-tools
- dgx-a100-system-tools-extra

## Remove this if you don't want to use cachefilesd
- nvidia-conf-cachefilesd

## Remove this if boot drive encryption is enabled and you don't
## want the passphrase dialog only visible on the serial console
- nvidia-ipmisol

##
## NVIDIA CUDA driver and tools
## Change the driver version to the branch you want to install
##
- datacenter-gpu-manager
- libnvidia-nscq-525
- linux-modules-nvidia-525-server-generic
- nvidia-driver-525-server
- nvidia-modprobe
- nv-persistence-mode

## Uncomment these to support the NVswitch on DGX A100 and DGX H100/H200
## Ensure that the driver version matches with the versions above
# - libnvidia-nscq-525
# - nvidia-fabricmanager-525
```

(continues on next page)

(continued from previous page)

```
##
## Mellanox drivers and tools
##
- nvidia-mlnx-config
- nvidia-mlnx-names
- nvidia-mlnx-ofed-misc

##
## NVIDIA container support
##
- docker-ce
- nv-docker-options
- nvidia-docker2

##
## NVIDIA system management tools
##
- nvsm
- nvidia-motd
```

9. Add any additional software packages you want to install during autoinstall.
10. Finally, add a list of additional commands to be executed at the end of the installation.
 - ▶ Disable unattended upgrades
 - ▶ Disable the ondemand governor defaulting to performance mode
 - ▶ Enable DCGM and OpenIBD services
 - ▶ Enable nv-peer-mem

```
##
## Commands executed after completion of the installation
##
late-commands:
- curtin in-target --target=/target -- apt purge -y unattended-
→upgrades
- curtin in-target --target=/target -- systemctl disable ondemand
- curtin in-target --target=/target -- systemctl enable dcgm openibd
- curtin in-target --target=/target -- update-rc.d nv_peer_mem
→defaults
# DGX A100 ...
- curtin in-target -- mlnx_pxe_setup.bash
```

18.2. Drive Partitioning

```
storage:
  config:
  - id: disk-sda
    type: disk
    ptable: gpt
```

(continues on next page)

(continued from previous page)

```

path: /dev/sda
name: osdisk
wipe: superblock-recursive
- id: partition-sda1
  type: partition
  device: disk-sda
  number: 1
  size: 512M
  flag: boot
  grub_device: true
- id: partition-sda2
  type: partition
  device: disk-sda
  number: 2
  size: 100G
- id: format-partition-sda1
  type: format
  fstype: fat32
  label: efi
  volume: partition-sda1
- id: format-partition-sda2
  type: format
  fstype: ext4
  label: root
  volume: partition-sda2
- id: root-mount
  type: mount
  path: /
  device: format-partition-sda2
  options: errors=remount-ro
  passno: 1
- id: boot-mount
  type: mount
  path: /boot/efi
  device: format-partition-sda1
  passno: 1
- id: disk-sdb
  type: disk
  ptable: gpt
  path: /dev/sdb
  name: raid
  wipe: superblock-recursive
- id: partition-sdb1
  type: partition
  device: disk-sdb
  number: 1
- id: format-partition-sdb1
  type: format
  fstype: ext4
  label: raid
  volume: partition-sdb1
- id: raid-mount

```

(continues on next page)

(continued from previous page)

```
type: mount
path: /raid
device: format-partition-sdb1
passno: 2
```

Chapter 19. Installing Docker Containers

This method applies to Docker containers hosted on the NVIDIA NGC Container Registry, and requires that you have an active NGC account.

1. On a system with internet access, log in to the NGC Container Registry by entering the following command and credentials.

Username:

Password:

1. Type \$oauthtoken exactly as shown for the Username.
2. This is a special username that enables API key authentication. In place of apikey, paste in the API Key text that you obtained from the NGC website.
3. Enter the docker pull command, specifying the image registry, image repository, and tag:
4. Verify the image is on your system using docker images.
5. Save the Docker image as an archive.
6. Transfer the image to the air-gapped system using removable media such as a USB flash drive.
7. Load the NVIDIA Docker image.

```
docker load -i framework.tar
```

8. Verify the image is on your system.

```
docker images
```

Chapter 20. Third-Party License Notices

This NVIDIA product contains third party software that is being made available to you under their respective open source software licenses. Some of those licenses also require specific legal information to be included in the product. This section provides such information.

20.1. Micron msecli

The `msecli` utility is provided under the following terms:

Micron Technology, Inc. Software License Agreement PLEASE READ THIS LICENSE AGREEMENT ("AGREEMENT") FROM MICRON TECHNOLOGY, INC. ("MTI") CAREFULLY: BY INSTALLING, COPYING OR OTHERWISE USING THIS SOFTWARE AND ANY RELATED PRINTED MATERIALS ("SOFTWARE"), YOU ARE ACCEPTING AND AGREEING TO THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE WITH THE TERMS OF THIS AGREEMENT, DO NOT INSTALL THE SOFTWARE. LICENSE: MTI hereby grants to you the following rights: You may use and make one (1) backup copy the Software subject to the terms of this Agreement. You must maintain all copyright notices on all copies of the Software. You agree not to modify, adapt, decompile, reverse engineer, disassemble, or otherwise translate the Software. MTI may make changes to the Software at any time without notice to you. In addition MTI is under no obligation whatsoever to update, maintain, or provide new versions or other support for the Software. OWNERSHIP OF MATERIALS: You acknowledge and agree that the Software is proprietary property of MTI (and/or its licensors) and is protected by United States copyright law and international treaty provisions. Except as expressly provided herein, MTI does not grant any express or implied right to you under any patents, copyrights, trademarks, or trade secret information. You further acknowledge and agree that all right, title, and interest in and to the Software, including associated proprietary rights, are and shall remain with MTI (and/or its licensors). This Agreement does not convey to you an interest in or to the Software, but only a limited right to use and copy the Software in accordance with the terms of this Agreement. The Software is licensed to you and not sold.

DISCLAIMER OF WARRANTY:

THE SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. MTI EXPRESSLY DISCLAIMS ALL WARRANTIES EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, NONINFRINGEMENT OF THIRD PARTY RIGHTS, AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. MTI DOES NOT WARRANT THAT THE SOFTWARE WILL MEET YOUR REQUIREMENTS, OR THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR-FREE. FURTHERMORE, MTI DOES NOT MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE SOFTWARE IN TERMS OF ITS CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE. THE ENTIRE RISK ARISING OUT OF USE OR PERFORMANCE OF THE SOFTWARE REMAINS WITH YOU. IN NO EVENT SHALL MTI, ITS AFFILIATED COMPANIES OR THEIR SUPPLIERS BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, INCIDENTAL, OR SPECIAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF YOUR USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF MTI

HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Because some jurisdictions prohibit the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

TERMINATION OF THIS LICENSE: MTI may terminate this license at any time if you are in breach of any of the terms of this Agreement. Upon termination, you will immediately destroy all copies the Software.

GENERAL: This Agreement constitutes the entire agreement between MTI and you regarding the subject matter hereof and supersedes all previous oral or written communications between the parties. This Agreement shall be governed by the laws of the State of Idaho without regard to its conflict of laws rules.

CONTACT: If you have any questions about the terms of this Agreement, please contact MTI's legal department at (208) 368-4500. By proceeding with the installation of the Software, you agree to the terms of this Agreement. You must agree to the terms in order to install and use the Software.

20.2. Mellanox (OFED)

MLNX_OFED <<http://www.mellanox.com/>> is provided under the following terms:

Copyright (c) 2006 Mellanox Technologies. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Chapter 21. Notices

21.1. 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 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.

21.2. Trademarks

NVIDIA, the NVIDIA logo, DGX, DGX-1, DGX-2, DGX A100, DGX H100, DGX H200, DGX B200, DGX B300, DGX GB200, DGX GB300, DGX Station, DGX Station A100, and DGX Spark are trademarks and/or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Chapter 22. About DGX OS 7

NVIDIA DGX OS provides a customized installation of Ubuntu Linux with system-specific optimizations and configurations, additional drivers, and diagnostic and monitoring tools. It provides a stable, fully tested, and supported OS to run AI, machine learning, and analytics applications on DGX Supercomputers.

NVIDIA DGX™ systems are shipped preinstalled with DGX OS to provide a turnkey solution for running AI and analytics workloads. Initial system configuration is deferred to a setup wizard that runs after the first boot. The setup wizard offers users a fast onboarding experience for using DGX systems.

The DGX OS installer is released as an ISO image to reimage a DGX system. The additional software, the NVIDIA DGX Software Stack included in DGX OS is provided as packages that are available from software repositories over the internet.

You also have the option to install the NVIDIA DGX Software Stack on a regular Ubuntu 24.04 while still benefiting from the advanced DGX features. This installation method supports more flexibility, such as custom partition schemes. Cluster deployments also benefit from this installation method by taking advantage of Ubuntu's standardized automated and non-interactive installation process.

22.1. DGX OS 7 Features

The following are the key features of the DGX OS 7 release:

- ▶ Based on Ubuntu 24.04 with the Linux kernel version 6.8 for the recent hardware and security updates and updates to software packages, such as Python, GCC, and OpenJDK.
- ▶ Includes the Ubuntu generic kernel (DGX servers based on x86_64) and the NVIDIA-optimized Linux kernel (DGX servers based on ARM64.)
- ▶ Provides access to all NVIDIA GPU driver branches and CUDA toolkit versions.
- ▶ Uses the NVIDIA DOCA™ OFED (OpenFabrics Enterprise Distribution) software, which is the successor to MLNX_OFED.
- ▶ Provides the Ubuntu Pro Client's Extended Security Maintenance (ESM) subscription from the Ubuntu Universe repository.
- ▶ Supports Emerald Rapids CPUs.

22.2. Supported NVIDIA DGX Systems

DGX OS 7 supports the following systems:

Table 1: Supported DGX Systems

Architecture x86_64	DGX Systems	Minimum DGX OS ISO Release
	DGX B300 2.3 TB	7.3.0
	DGX B200 1,440 GB	7.0.2
	DGX H200 1,128 GB	6.3.1
	DGX H100 640 GB	6.0.11
	DGX H800 640 GB	6.1.0
	DGX A100 640 GB	5.5.1
	DGX A100 320 GB	5.5.1
	DGX A800 640 GB	5.5.1
	DGX Station A100 320 GB	5.0.2
	DGX Station A100 160 GB	5.0.2
	DGX Station A800 320 GB	5.0.2
	DGX Systems	
Architecture ARM64		Minimum DGX OS ISO Release
	DGX GB300	7.2.3
	DGX GB200	7.1.0
	DGX Spark	7.2.3

DGX OS 7 does not support the following systems:

Table 2: Unsupported DGX Systems

Support	DGX Systems	End-of-Support Date
End of support	<ul style="list-style-type: none"> ▶ DGX-2 ▶ DGX-1 (V100) ▶ DGX Station (V100) 	August 2025

22.3. Installation and Upgrade

This document covers installation and upgrade options for DGX OS. It also provides instructions for setting up the system and installing additional software.

Initial Setup

If DGX OS 7 is already installed on your DGX system, refer to [Initial Setup](#) for information about setting up the system on first boot.

After initial setup, refer to [Upgrading the OS](#) to perform a *package upgrade* to the latest software package versions.

Upgrading the OS

To upgrade your DGX OS to the latest software package versions or for information about performing a *release upgrade* from DGX OS 5 or DGX OS 6 to DGX OS 7, refer to [Upgrading the OS](#).

Reimaging the System

To restore a DGX system to a default DGX OS installation and erase all data, you can use the ISO image that includes an autonomous installer. Refer to [Reimaging the System](#) for more information.

Customizing Ubuntu Installation with DGX Software

To install Ubuntu and the DGX Software Stack, refer to [Customizing Ubuntu Installation with DGX Software](#) for information about automating the installation process, such as a cluster deployment.

Managing OS and Software Updates

DGX OS and Ubuntu provide additional software packages, including additional NVIDIA software and driver options. Refer to [Managing OS and Software Updates](#) for more information and installation instructions.

Important

Before you upgrade or install any new software, always consult the [Release Notes](#) for the latest information about available upgrades. You can find out more about the release cadence and release methods for DGX OS in [Release Guidance](#).

22.4. Related Documentation

Refer to the following documents that are related to DGX OS 7:

- ▶ [DGX Documentation](#)

All documentation for DGX products, including product user guides, software release notes, and firmware update container information

- ▶ [MIG User Guide](#)

The Multi-Instance GPU (MIG) feature allows the NVIDIA A100 GPU to be securely partitioned into up to seven discrete GPU instances.

- ▶ [NGC Private Registry](#)

How to access the NGC container registry for using containerized deep learning GPU.

- ▶ [NVSM Software User Guide](#)

Contains instructions for using the NVIDIA System Manager software.

- ▶ [DCGM Software User Guide](#)

Contains instructions for using the Data Center GPU Manager software.

22.5. NVIDIA Enterprise Support

NVIDIA Enterprise Support is the support resource for DGX customers and can assist with hardware, software, or NGC application issues. For more information about how to obtain support, visit [NVIDIA Enterprise Support](#).

Copyright

©2022-2025, NVIDIA Corporation