NVIDIA DGX OS 5.0

User Guide

DU-10211-001 _v5.0.0 | December 2022

Table of Contents

Chapter 1. Introduction to the DGX OS 5 User Guide

The NVIDIA® DGX™ OS is a customized, Linux distribution that is based on Ubuntu Linux. It includes platform-specific configurations, diagnostic and monitoring tools, and the drivers that are required to provide the stable, tested, and supported OS to run AI, machine learning, and analytics applications on DGX systems.

Important:

- If plan to reimage your DGX system, proceed to Preparing for Operation and read through the guide.
- If plan to complete the first boot process and install DGX OS, continue to Initial DGX OS Set Up.

DGX OS 5 includes the following features:

- A Ubuntu 20.04 LTS distribution
- One ISO for all DGX systems
- NVIDIA System Management (NVSM)

  NVSM provides active health monitoring and system alerts for NVIDIA DGX nodes in a data center. It also provides simple commands to check the health of the DGX systems from the command line.
- Data Center GPU Management (DCGM)

  This software enables node-wide administration of GPUs and can be used for cluster and data-center level management.
- DGX system-specific support packages
- NVIDIA GPU driver, CUDA toolkit, and domain specific libraries
- Docker Engine
- NVIDIA Container Toolkit
- Cachefiles Daemon for caching NFS reads
- Tools to convert data disks between RAID levels
- Disk drive encryption and root filesystem encryption (optional)

- Mellanox OpenFabrics Enterprise Distribution for Linux (MOFED) and Mellanox Software Tools (MST) for systems with Mellanox network cards

For more information, refer to the Release Notes section in the DGX documentation and locate the release notes for your DGX OS 5.x release.

1.1. Additional Documentation

Here are links to some additional DGX documentation.

- DGX Documentation

  All documentation for DGX products, including product user guides, software release notes, firmware update container information, and best practices documentation.
- MIG User Guide

  The new Multi-Instance GPU (MIG) feature allows the NVIDIA A100 GPU to be securely partitioned into up to seven separate GPU Instances for CUDA applications.
- NGC Private Registry

  How to access the NGC container registry for using containerized deep learning GPU-accelerated applications on your DGX system.
- NVSM Software User Guide

  Contains instructions for using the NVIDIA System Management software.
- DCGM Software User Guide

  Contains instructions for using the Data Center GPU Manager software.

1.2. Customer 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 the NVIDIA Enterprise Support website.

Chapter 2. Preparing for Operation

2.1. Software Installation and Setup

DGX OS 5 is preinstalled on new DGX systems. A setup wizard in the First Boot procedure requires you to create a user, set locales and keyboard layout, set passwords, and perform basic network configuration.

For systems that are running DGX OS version 4, you can upgrade the system to DGX OS 5 from network repositories (distribution upgrade) or reimage the system from the DGX OS 5 ISO image. The reimaging process installs the OS but defers the initial setup to the First Boot Process for DGX Servers or First Boot Process for DGX Station.

Note: If your system is already installed with DGX OS 5, you can continue to Initial DGX OS Setup.

There might be other situations where you need to reimage a system, such as the following:

- When the OS becomes corrupt.
- When the OS drive is replaced or both drives in a RAID-1 configuration are replaced.
- When you want to encrypt the root filesystem.
- When you want a fresh installation of DGX OS 5.

Important:

When you upgrade the OS, the configurations and data is preserved. Reimaging wipes the drive, and consequently, all configurations and data on the system.

- To reimage a system, see Installing the DGX OS (Reimaging the System).
- To upgrade a system, see Upgrading Your DGX OS Release.

2.2. Connecting to the DGX System

During the initial installation and configuration steps, you need to connect to the console of the DGX system.

There are several ways to connect to the DGX system, including the following:

- 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 A100 System User Guide
- DGX-2 System User Guide
- DGX-1 User Guide
- DGX Station User Guide
- DGX Station A100 User Guide

Chapter 3. Installing the DGX OS (Reimaging the System)

This section provides information about how to install the DGX OS.

Important: Installing DGX OS 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.

1. Obtain the latest DGX OS ISO image from NVIDIA Enterprise Support.
   See Obtaining the DGX OS ISO Image for more information.
2. Install the DGX OS ISO image in one of the following ways:
   - Remotely through the BMC for systems that provide a BMC.

     Refer to Reimaging the System Remotely for instructions.

     This method is not available for DGX Station.
   - Locally from a UEFI-bootable USB flash drive or DVD-ROM.

     Refer to Installing the DGX OS Image from a USB Flash Drive or DVD-ROM for instructions.

3.1. Obtaining the DGX OS ISO

To ensure that you install the latest available version of DGX OS, obtain the current ISO image file from NVIDIA Enterprise Support.
Before you begin, ensure that you have an NVIDIA Enterprise Support account.

1. Go to the DGX Software Firmware Download Matrix, locate, and click the announcement for the latest DGX OS 5 release for your system.
2. Download the ISO image that is referenced in the release notification and save it to your local disk.

3. To verify the integrity and authenticity of the image, write down the MD5 value in the announcement .
4. Run the `md5sum` command to print the MD5 hash and compare it with the value in the announcement.

```
md5sum DGXOS-5.0.0-2020-09-21-15-40-02.iso
e4c77338ed35d7a34e772d8552e9d080 --> DGXOS-5.0.0-2020-09-21-15-40-02.iso
```

3.2. Installing the DGX OS Image Remotely through the BMC

These instructions describe how to re-image the system remotely through the BMC. For information about how to restore the system locally, see Installing the DGX OS Image from a USB Flash Drive or DVD-ROM.

Before re-imaging the system remotely, ensure that the correct software image is saved to your local disk. For more information, see Obtaining the DGX OS ISO.

1. Log in to the BMC.
2. Click **Remote Control** and then click **Launch KVM**.
3. Set up the ISO image as virtual media.
   a). From the top bar, click **Browse File** and then locate and select the DGX OS ISO file and click **Open**.
   b). Click **Start Media**.
4. Reboot, install the image, and complete the DGX OS setup.
   a). From the top menu, click **Power** and select **Hard Reset**, then click **Perform Action**.
   b). Click **Yes** and then **OK** at the Power Control dialogs, then wait for the system to power down snd then come back online.
   c). Refer to Installation Options for a description of the GRUB menu options and for instructions on completing the installation process.

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

After obtaining the DGX OS 5 ISO image from NVIDIA Enterprise Support, create a bootable installation medium, such as a USB flash drive or DVD-ROM, that contains the image.

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

1. Plug the USB flash drive containing the OS image into the DGX system.
2. If installing on a DGX server, connect a monitor and keyboard directly to the DGX system.
3. Boot the system and then press **F11** when the NVIDIA logo appears to get to the boot menu.
4. Select the USB volume name that corresponds to the inserted USB flash drive and boot the system from it.
5. Refer to Installation Options for a description of the GRUB menu options and for instructions on completing the installation process.

3.3.1. Creating a Bootable USB Flash Drive by Using the `dd` Command

On a Linux system, you can use the dd (http://manpages.ubuntu.com/manpages/bionic/en/man1/dd.1.html) 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, see Obtaining the Software ISO Image and Checksum File.
- The USB flash drive meets the following requirement:
  - The USB flash drive has a capacity of at least 16 GB.
  - **This requirement applies only to DGX A100**: The partition scheme on the USD flash drive is a CPT partition scheme for UEFI.

1. Plug the USB flash drive into one of the USB ports of your Linux system.
2. Obtain the device name of the USB flash drive by running the lsblk (http://manpages.ubuntu.com/manpages/bionic/man8/lsblk.8.html) 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, the device name of the USB flash drive is `sde`.

   ```
   ~$ lsblk
   NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
   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
~$
```

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

```
$ sudo dd if=path-to-software-image bs=2048 of=usb-drive-device-name
```

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

3.3.2. Creating a Bootable USB Flash Drive by Using Akeo Rufus

On a Windows system, you can use the Akeo Reliable USB Formatting Utility (Rufus) (https://rufus.akeo.ie/) to create a bootable USB flash drive that contains the DGX OS software image.

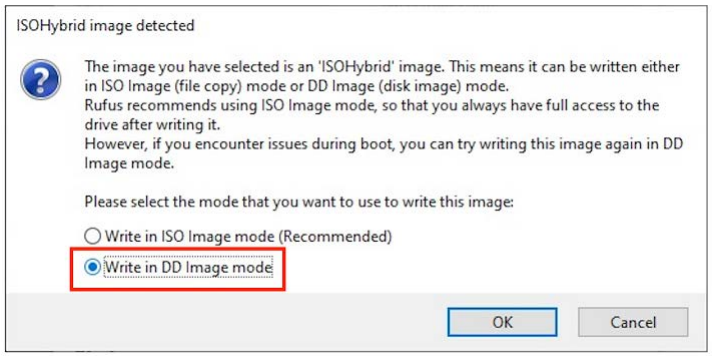
Ensure that the following prerequisites are met:

- The correct DGX OS software image is saved to your local disk.

  For more information, see Obtaining the Software ISO Image and Checksum File.
- The USB flash drive has a capacity of at least 16 GB.

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) (https://rufus.akeo.ie/).
3. In **Drive Properties**, select the following options:
   a). In **Device**, select your USB flash drive.
   b). 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**.

3.4. Installation Options

This section provides information about the available installation options.
These instructions assumes that you have booted the DGX system from the DGX OS, either remotely through the BMC or locally from a USB flash drive.

1. When the system boots up, select one of the following options from the GRUB menu:
   - **Install DGX OS <version>: Install DGX OS and reformat data RAID**
   - **Install DGX OS <version>: Without Reformatting Data RAID**
   - **Advanced Installation Options: Select to install with an encrypted root filesystem**
     - **Install DGX OS <version> With Encrypted Root**
     - **Install DGX OS <version> With Encrypted Root and Without Reformatting Data RAID**
   - **Boot Into Live Environment**
   - **Check Disc for Defects**

   See the subsections below for more information about these options.
2. 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 Mellanox InfiniBand driver is installed and the Mellanox card firmware is updated. This process can take up to 5 minutes for each card. Other system firmware is not updated.

After the installation is completed, the system reboots into the OS, and prompts for configuration information. See Initial DGX OS Setup for more information about how to boot up the DGX system for the first time after a fresh installation.

3.4.1. Install DGX OS and Reformat 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.

3.4.2. 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 to be used 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 formatting 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. Uncomment the `/raid` line in `etc/fstab`.
3. Run the following:

   a). Mount `/raid`.

   ```
   sudo mount /raid
   ```

   b). Reload the systemd manager configuration.

   ```
   systemctl daemon-reload
   ```

   c). Start the cache daemon.

   ```
   systemctl start cachefilesd.server
   ```

These changes are preserved across system reboots.

3.4.3. Advanced Installation Options (Encrypted Root)

When you select this menu item, you have the ability to encrypt the root filesystem of the DGX system.

Important: This option should **only** be selected when you want to encrypt the root filesystem.

Aside from the encrypted root filesystem, the behavior is identical. See Install DGX OS and Install DGX OS Without Reformatting Data RAID for more information.

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.

During the First Boot Process for DGX Servers or the First Boot Process for DGX Station, you can create your passphrase for the drive. If necessary, you can change this passphrase later.

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

3.4.4. 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.

3.4.5. 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, selecting this item to complete an extensive test of the install media contents.

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

Chapter 4. Initial DGX OS Set Up

This section describes the set up process when the DGX system is powered on for the first time after delivery or after the server is reimaged.

To start the process, you need to accept the End User License Agreements (EULA) and to set up your username and password. To preview the EULA, visit https://www.nvidia.com/en-us/data-center/dgx-systems/support/ and click the **DGX EULA** link.

4.1. First Boot Process for DGX Servers

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

1. If the DGX OS was installed with an encrypted root filesystem, you will be prompted to unlock the drive.

   See Advanced Installation Options (Encrypted Root) for more information.
2. Enter **`nvidia3d`** at the crypt prompt.
3. Accept the EULA to proceed with the DGX system set up.
4. 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 **will not** accept **`sysadmin`** for a username, and you will not be able to log in to the BMC with that username.
   - 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 appears.

   f). Create a BMC admin password.

   The allowed character length for the BMC password depends on the specific DGX product:

- DGX-1: 1 - 20 characters
- DGX-2: 1 - 20 characters
- DGX A100: 13 - 20 characters
- DGX Station A100: 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. See Advanced Installation Options (Encrypted Root) for more information.

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-1: `enp1s0f0`
- DGX-2: `enp6s0`
- DGX A100: `enp226s0`

Do not select `enp37s0f3u1u3c2`, `bmc_redfish0`, or something similar, as 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.

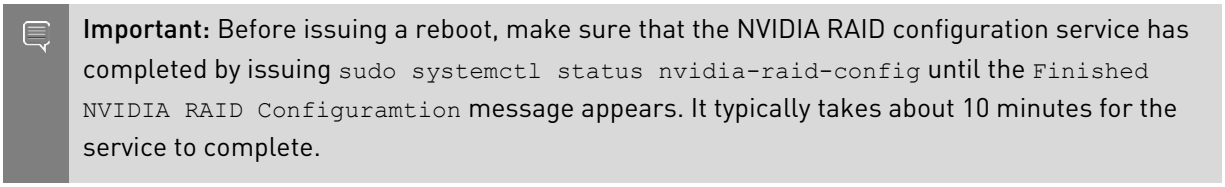
- If no DHCP is available, click **OK** at the Network autoconfiguration failed dialog and manually configure the network.
- To configure a static address, then click **Cancel** at the dialog after the DHCP configuration completes to restart the network configuration steps.
- To select a different network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



Important: Before issuing a reboot, make sure that the NVIDIA RAID configuration service has completed by issuing `sudo systemctl status nvidia-raid-config` until the `Finished NVIDIA RAID Configuramtion` message appears. It typically takes about 10 minutes for the service to complete.

4.2. First Boot Process for DGX Station

When you power on your DGX Station 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**.

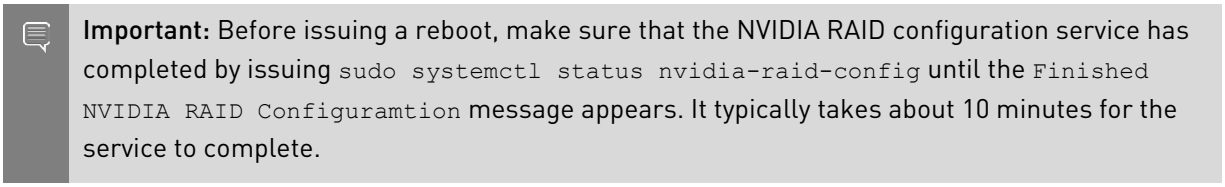
   Here are some requirements to remember:

   - 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 performed the automated encryption install, you will also 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 be used to unlock your root filesystem when the system boots.



Important: Before issuing a reboot, make sure that the NVIDIA RAID configuration service has completed by issuing `sudo systemctl status nvidia-raid-config` until the `Finished NVIDIA RAID Configuramtion` message appears. It typically takes about 10 minutes for the service to complete.

Chapter 5. Post-Installation Tasks

You can complete the following tasks after you install your DGX sytem.

5.1. 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 5 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 5 software image:

- English
- Chinese (Simplified)
- French
- German
- Italian
- Portuguese
- Russian
- Spanish

For information about how to install languages, see Install languages.

5.2. Configuring your DGX Station To Use Multiple Displays

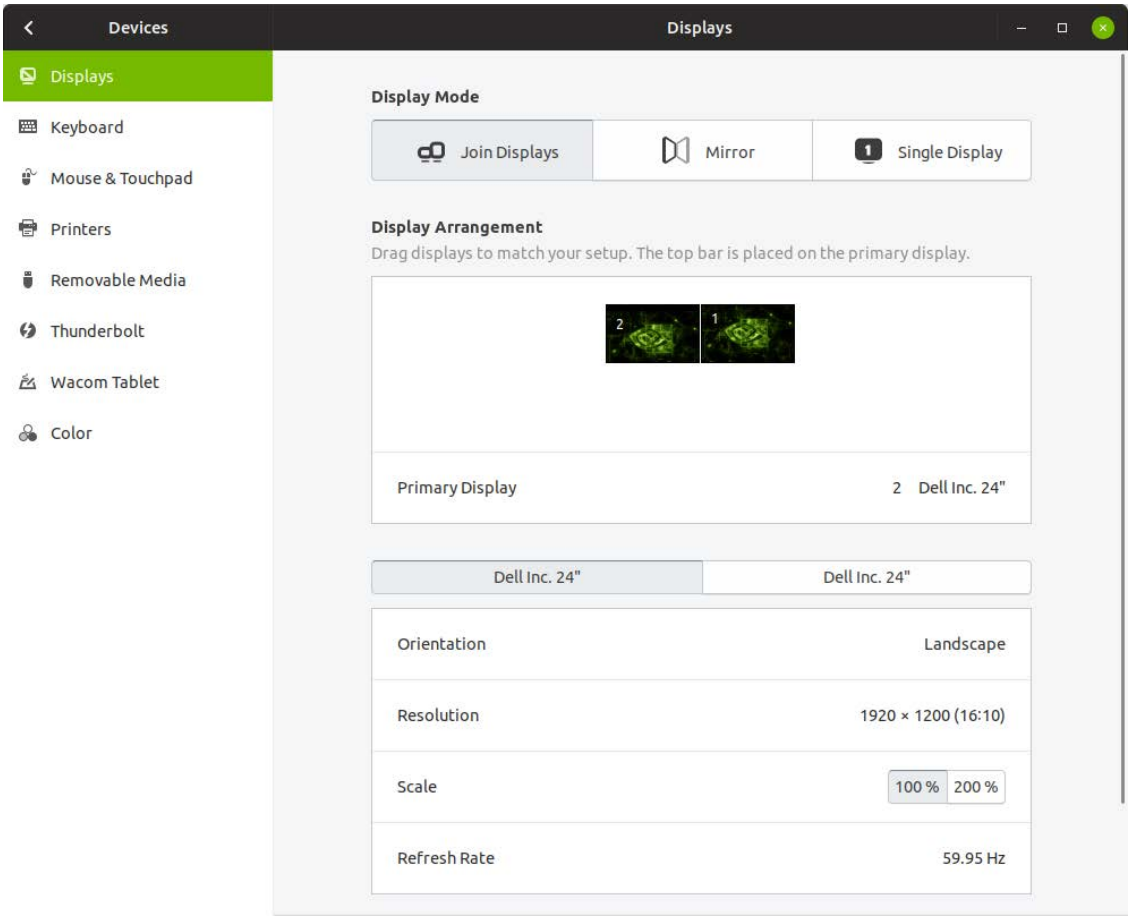
The DGX Display Adaptor card provides DGX OS with multiple display outputs, which allow you to connect multiple monitors to the DGX Station A100. If you plan to use more than one display, configure the DGX Station A100 to use multiple displays **after** you complete the initial DGX OS configuration.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

1. Connect the displays that you want to use to the mini DisplayPort (DP) connectors (or the DisplayPort connectors DGX Station V100) 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 as 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**.



5.2.1. DGX Station V100

This information **only** applies to DGX Station V100.

High-resolution displays consume a large quantity of GPU memory. If you connected three 4K displays to the DGX Station V100, the displays might consume most of the GPU memory on the NVIDIA Tesla V100 GPU card to which these displays are connected, especially if you are running graphics-intensive applications.

If you are running memory-intensive compute workloads on the DGX Station V100, and are experiencing performance issues, consider conserving GPU memory by reducing or minimizing the graphics workload.

- To reduce the graphics workload, disconnect any additional displays you connected and use only one display with the DGX Station V100.
- If you disconnect a display from the DGX Station V100, the disconnection is automatically detected, and the display settings are automatically adjusted for the remaining displays.
- To minimize the graphics workload, shut down the display manager and use secure shell (SSH) to remotely log in to the DGX Station.
- In DGX OS 5.x, log in to the DGX Station remotely, run the following commands:
  - To start the GNOME Display Manager (GDM3):

```
$ sudo systemctl start gdm3
```

- To stop the GDM3, run the following command:

```
$ sudo systemctl stop gdm3
```

5.3. 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 how to log in remotely through SSH, see Connecting to an OpenSSH Server (https://help.ubuntu.com/community/SSH/OpenSSH/ConnectingTo) 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 how to secure access to the DGX system over SSH, see Configuring an OpenSSH Server (https://help.ubuntu.com/community/SSH/OpenSSH/Configuring) on the Ubuntu Community Help Wiki.

Chapter 6. Upgrading Your DGX OS Release

This section provides information about upgrading your DGX system.

ATTENTION: Before you upgrade a system, refer to the DGX OS 5 Software Release Notes for more information.

Here is some information that describes the difference between the different types of upgrades:

- When you perform a **release upgrade**, you currently have the DGX OS 4.x installed, and you want to move to DGX OS 5.

  You can upgrade to DGX OS 5 only from the latest DGX OS 4.x (for DGX Station, DGX-2, or DGX-1 systems) or from the latest DGX OS 4.99.x release (for DGX A100 systems). Refer to Upgrading to DGX OS 5 for the upgrade instructions. The instructions also provide information about completing an over-the-internet upgrade.

  Important: If your 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 the packages will be removed as part of the upgrade process. Refer to the DGX OS Software Release Notes for a list of packages that are no longer available in DGX OS 5.

- When you perform a **package upgrade**, you want to install upgrades that have been made available in the network repository since the initial DGX OS release.

  The network repositories are periodically updated with package upgrades and will include new features that are available with the latest DGX OS minor version release.

WARNING: The instructions in this section upgrade all software for which updates are available from your configured software sources, including applications that you installed yourself. If you want to prevent an application from being upgraded, you can instruct the Ubuntu package manager to keep the current version. For more information, see Introduction to Holding Packages on the Ubuntu Community Help Wiki.

6.1. 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 5, 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"
DGX_PRETTY_NAME="NVIDIA DGX Station"
DGX_SWBUILD_DATE="2017-09-18"
DGX_SWBUILD_VERSION="3.1.2"
DGX_COMMIT_ID="15cd1f473bb53d9b64503e06c5fee8d2e3738ece"
DGX_SERIAL_NUMBER=XXXXXXXXXXXXX

DGX_OTA_VERSION="3.1.3"
DGX_OTA_DATE="Wed Nov 15 15:35:25 PST 2017"

DGX_OTA_VERSION="4.7.0"
DGX_OTA_DATE="Fri Dec 19 13:49:06 PST 2020"

DGX_OTA_VERSION="5.0.0"
DGX_OTA_DATE="Tue Jan 19 14:23:18 PDT 2021"

DGX_OTA_VERSION="5.0.0"
DGX_OTA_DATE="Tue Feb 23 17:45:30 PST 2021"
```

6.2. Preparing to Upgrade the Software

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

6.2.1. 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 for more information.

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 will not be able to establish a network connection to the DGX server.

See Configuring Docker IP Addresses 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 by Using the GUI .

6.2.2. Verifying the DGX System Connection to the Repositories

Before you attempt to complete the update, 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/focal/Release
$ wget -O f3-security \
http://security.ubuntu.com/ubuntu/dists/focal/Release
$ wget -O f4-nvidia-baseos \
http://repo.download.nvidia.com/baseos/ubuntu/focal/x86_64/dists/focal/Release
$ wget -O f5-nvidia-cuda \
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64/Release
```

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

6.3. Upgrading to DGX OS 5

This section provides information about how to upgrade to DGX OS 5 from a DGX OS 4.x (for DGX Station, DGX-1, or DGX-2) or a DGX OS 4.99.x release.

Refer to Connecting to the DGX System for guidance on connecting to the console to perform the upgrade.

Important: If you are updating from an earlier DGX OS 5 version, **do not** perform these steps. Follow the instructions at Performing Package Upgrades by Using the CLI.

1. If you have **DGX OS 4.12 or earlier** installed or version 4.99.x on DGX A100, ensure you have the correct GPG signing keys installed on your system.

   Before you continue the upgrade, refer to the following release notes for instructions and details:

   - DGX OS 4 Server Release Notes for DGX-1, DGX-2, and DGX A100
   - DGX OS 4 Desktop Release Notes for DGX Station

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

   ```
   $ sudo apt update
   ```

3. Install all available upgrades for your **current** DGX OS release.

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

4. Install the `nvidia-release-upgrade` package for upgrading to the next major DGX OS release.

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

5. Issue the following only if upgrading from DGX OS 4.99.

   ```
   $ sudo apt install -y nvidia-fabricmanager-450/bionic-updates --allow-downgrades
   ```

6. Start the DGX OS release upgrade process.

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

   Note: This step upgrades all driver versions that are currently supported in DGX OS 4.x (R410, R418, and R450), to version R470.

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

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

   Tip: 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. Complete the following tasks:

Note: [Only for DGX Station]

During an upgrade to a DGX OS 5 release from an earlier release, you are prompted to resolve conflicts in configuration files. When prompted, evaluate the changes before accepting the maintainer's version, keeping the local version, or manually resolving the difference.

Conflicts in the following configuration files are the result of customizations to the Ubuntu Desktop OS made for DGX OS 5.

- `/etc/ssh/sshd_config`. You can keep the local version that is currently installed.
- `/etc/gdm3/custom.conf.distrib`. You can keep your currently installed version.
- `/etc/gdm3/custom.conf`. You can keep your currently installed version.
- `/etc/apt/sources.list.d/dgx.list`. You should install the package maintainer's version.

Tip: Some package updates require that you reboot the system before completing the upgrade. Ensure that you reboot the system when prompted.

a). If you have packages that do not have upgrade candidates, you will see the following message:

```
WARNING: The following packages are installed, but have no 20.04
upgrade path.
They will be uninstalled during the release upgrade process. libnccl2 libnccl-
dev libcudnn7 libcudnn7-dev libcudnn7-doc libcudnn8 libcudnn8-dev libcudnn8-
samples
The --force option must be used to proceed.
```

If you see this message, run the `nvidia-release-upgrade` command with the `--force` option.

b). If you are logged in to the DGX system remotely through 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]
```

c). Enter **y** to continue.

d). 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]
```

e). Press **Enter**.

f). 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.

g). Press **Enter**.

h). 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]
```

i). Press **Enter**.

j). **(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.

k). 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 DGX OS Desktop Release Notes for the release family to which you are upgrading.

l). When prompted to confirm that you want to remove obsolete packages, enter **y**, **N**, or **d**.

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

m).Determine whether to remove obsolete packages and continue with the upgrade.

i. Review the list of packages that will be removed.

To identify obsolete DGX OS Desktop packages, see the lists of obsolete packages in the DGX OS Desktop Release Notes for all releases after your current release.

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

n). Press **y** at the prompt that appears when the system upgrade is completed.

```
System upgrade is complete.
Restart required
To finish the upgrade, a restart is required.
```

```
If you select 'y' the system will be restarted. Continue [yN]
```

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

8. If no reboot prompt appeared or if you did not restart the system when prompted, then 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.

6.3.1. Verifying the Upgrade

Here is are steps to verify your upgrade.

1. Confirm the Linux kernel version.

   For example, when you upgrade to DGX OS 5.0, the Linux kernel version is at least `5.4.0-52-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 Desktop 5.0, the NVIDIA Graphics Drivers for Linux version is at least 450.80.02:

```
Tue Oct 13 09:02:14 2020
+-----------------------------+------------------------------------------------+
| NVIDIA-SMI 450.80.02     Driver Version: 450.80.02     CUDA Version: 11.0

|-------------------------+----------------------+--------------------------+
```

6.3.2. 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 5 with the latest image instead. See Installing the DGX OS (Reimaging the System) 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 \
./focal --frontend=DistUpgradeViewText
```

   Do not reboot at this time.

5. Issue the following command and reboot.

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

6.4. Performing Package Upgrades by Using the CLI

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 by using the `apt full upgrade` command that is based on the threat level.

- For more information about upgrading to a supported version of Ubuntu, refer to the Ubuntu Wiki Upgrades.
- 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.

Important: You are responsible for upgrading the software on the DGX system to install the updates from these sources.

If updates are available, you can obtain the package upgrades by completing the following steps:

1. If you have a DGX OS version **earlier than 5.3**, ensure you have the correct GPG signing keys installed on your system.

   Before you continue the upgrade, refer to DGX OS 5 Release Notes for instructions and details.

2. Update the list of available packages and their versions.

```
$ sudo apt update
```

3. Review the packages that will be upgraded.

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

   To prevent an application from being graded, instruct the Ubuntu package manager to keep the current version. Refer to Introduction to Holding Packages for more information.

4. Install the updated CUDA repo preferences package to ensure the Fabric Manager and NSCQ library are installed from the Canonical repo.

```
$ sudo apt install cuda-compute-repo
```

5. Upgrade to the latest version.

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

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.

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

```
$ sudo reboot
```

Note: If you are running a DGX -2 KVM system, you can wait until any currently running VMs are terminated before rebooting the DGX-2.

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
```

6.5. Managing Software Upgrades on the Desktop

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

6.5.1. Performing Package Upgrades by 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 the bottom-left of your keyboard, next 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**.

- 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 your 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.

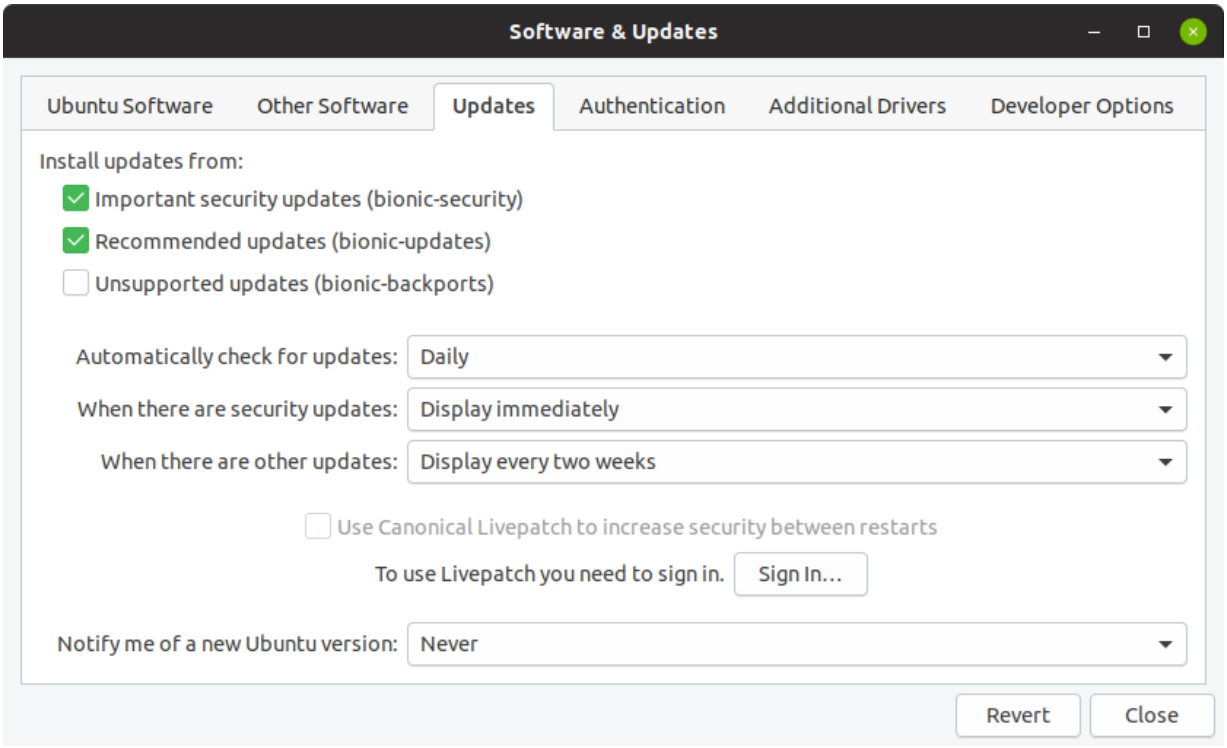
```
$ nvidia-smi
Failed to initialize NVML: Driver/library version mismatch
```

6.5.2. 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.



Chapter 7. Installing Additional Software

DGX OS 5 is an optimized version of the Ubuntu 20.04 Linux distribution with access to a large collection of additional software that is available from the Ubuntu repositories. You can install the additional software using the `apt` command or through a graphical tool.

CAUTION: Driver Branch Upgrade: Before upgrading to a different driver branch, upgrade the system to the latest versions by running

```
$ sudo apt update
$ sudo apt upgrade
```

Tip: The graphical tool is only available in DGX Station.

ATTENTION: Before you upgrade a system, refer to the DGX OS 5 Release Notes for additional information.

If your system has a DGX OS version **earlier than version 5.3**, ensure that you have correct GPG signing keys installed on your system. Refer to Rotating the GPG Keys for instructions to update the keys and additional information.

For more information, refer to the following topics in the Ubuntu documentation for Desktop:

- Package Management
- Install additional applications

7.1. Changing Your GPU Branch

NVIDIA drivers are released as precompiled and signed kernel modules by Canonical and are available directly from the Ubuntu repository. Signed drivers are required to verify the integrity of driver packages and identity of the vendor.

However, the verification process requires that Canonical build and release the drivers with Ubuntu kernel updates after their release cycle is complete, and this process might

sometimes delay new driver branch releases and updates. For more information about the NVIDIA driver release, refer to the NVIDIA Driver Documentation.

Important: The Ubuntu repositories provide the following versions of the signed and precompiled NVIDIA drivers:

- The general NVIDIA display drivers
- The NVIDIA Data Center GPU drivers

On your DGX system, you should only install the packages that include the NVIDIA Data Center GPU drivers. The metapackages for the NVIDIA Data Center GPU driver have the `-server` suffix .

7.1.1. Checking the Currently Installed Driver Branch

Here is some information about the prerequisite to determining the driver branch that you currently have installed.
Before you install a new NVIDIA driver branch, to check the currently installed driver branch, run the following command:

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

7.1.2. 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. If you have a DGX OS version **earlier than 5.3**, ensure you have the correct GPG signing keys installed on your system.

   Before you continue the upgrade, refer to DGX OS 5 Release Notes for instructions and details.
2. Update the local database with the latest information from the Ubuntu repository.

   ```
   apt update
   ```
3. Show all available driver branches.

   ```
   apt list nvidia-driver*server
   ```

7.1.3. Upgrading Your GPU Branch

To manually upgrade your driver to the latest branch:

WARNING: If you are upgrading from an older branch than R515 to a driver branch R515 or newer, or if you are downgrading from a branch R515 or newer to an older branch than R515, ensure to follow step 6. Note that R510 is a transitional package for R515 and should not be installed.

1. Purge the existing driver.

   In this example, the R450 driver packages will be removed first. Whether you upgrade or downgrade the NVIDIA GPU driver, the old drivers should be removed.

   ```
   $ sudo apt-get purge "*nvidia*450*"
   ```

2. Install the latest kernel.

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

3. To install the latest NVIDIA GPU driver, for example, R470, complete one of the following tasks:

   - On **Non-Fabric Manager** systems, such as DGX-1, DGX Station V100 (Volta), and DGX Station A100, run the following command:

     ```
     $ sudo apt install -y linux-modules-nvidia-470-server-generic nvidia-driver-470-server libnvidia-nscq-470 nvidia-modprobe nvidia-conf-xconfig nv-docker-gpus
     ```

   - On **Fabric Manager** systems, such as DGX-2 and DGX A100, run the same command, but append the `nvidia-fabricmanager-470` package:

     ```
     $ sudo apt install -y linux-modules-nvidia-470-server-generic nvidia-driver-470-server libnvidia-nscq-470 nvidia-modprobe nvidia-fabricmanager-470
     ```

   Important: The version numbers are only examples, and you should replace this value with the version that you want to install.

4. **Before** you reboot your DGX-2 or DGX A100 system, enable the `nvidia-fabricmanager` service.

   ```
   $ sudo systemctl unmask nvidia-fabricmanager
   $ sudo systemctl enable nvidia-fabricmanager
   ```

5. If you are using a DGX-1, DGX-2, or DGX A100, run the following command:

   ```
   $ sudo apt install -y --reinstall nvidia-peer-memory-dkms
   $ sudo /usr/sbin/update-rc.d nv_peer_mem defaults
   ```

6. If you are upgrading from a branch older than R515 to a driver branch R515 or newer, or if you are downgrading from a branch R515 or newer to an older branch than R515, install the correct DCGM version. You can skip this step, otherwise.

   a). If you are upgrading to a branch R515 or newer from a branch older than R515, identify the latest DCGM 3.x version:

   ```
   $ apt-cache policy datacenter-gpu-manager

   Version table:
        1:3.1.3 580
           580 https://developer.download.nvidia.com/...
        1:3.0.4 580
           580 https://developer.download.nvidia.com/...
        1:2.4.7 600
           580 https://developer.download.nvidia.com/...
           600 https://repo.download.nvidia.com/baseos/...
   ```

   Identify the latest DCGM 3.x version. In the case above, this would be "1:3.1.3". Install the latest DCGM 3.x version:

   ```
   $ sudo apt install datacenter-gpu-manager=1:3.1.3
   ```

b). If you are downgrading to a branch older than R510 from R515 or a newer branch (note that R510 is a transitory package for R515) then install DCGM version 2:

```
$ sudo apt install datacenter-gpu-manager/$(lsb_release -cs)-updates -y
--allow-downgrades
```

7.2. 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.

7.2.1. Checking the Currently Installed CUDA Toolkit Release

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

Important: The CUDA Toolkit is **not** installed on DGX servers by default, and if you try to run the following command, no installed package will be listed.

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

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

For example, the following output shows that CUDA Toolkit 11.0 is installed:

```
$ apt list --installed cuda-toolkit-*
Listing... Done
cuda-toolkit-11-0/unknown,unknown,now 11.0.3-1 amd64 [installed]
N: There is 1 additional version. Please use the '-a' switch to see it
```

7.2.2. Determining the New Available CUDA Toolkit Releases

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

To see the new available CUDA Toolkit releases:

1. If you have a DGX OS version **earlier than 5.3**, ensure you have the correct GPG signing keys installed on your system.

   Before you continue the upgrade, refer to DGX OS 5 Release Notes for instructions and details.
2. Update the local database with the latest information from the Ubuntu repository.

   ```
   apt update
   ```
3. Show all available CUDA Toolkit releases.

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

The following output shows that 11.0, 11.1, and 11.2 are the possible CUDA Toolkit versions that can be installed:

```
$ apt list cuda-toolkit-*
Listing... Done
cuda-toolkit-11-0/unknown,unknown,now 11.0.3-1 amd64 [installed]
cuda-toolkit-11-1/unknown,unknown 11.1.1-1 amd64
cuda-toolkit-11-2/unknown,unknown 11.2.1-1 amd64
```

7.2.3. Installing the CUDA Toolkit or Upgrading Your CUDA Toolkit to a Newer Release

You can install or upgrade your CUDA Toolkit to a newer release.

To install or upgrade the CUDA Toolkit, run the following command:

```
apt install cuda-toolkit-11-2
```

Important: Here, version *11.2* is an example, and you should replace this value with the actual version that you want to install.

7.3. 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, which avoids a bounce buffer through the CPU.

7.3.1. Prerequisites to Installing GPUDirect Storage Support

7.3.1.1. Using the MLNX_OFED Driver

If using the MLNX_OFED driver, be sure to enable GDS support as explained in https://docs.nvidia.com/gpudirect-storage/troubleshooting-guide/index.html#mofed-req-install

7.3.1.2. Installing nvidia_peermem

For CUDA 11.5.1 and later, if you plan to use Weka FS or IBM SpectrumScale then you need to run:

```
modprobe nvidia_peermem
```

This will load the module that supports peerdirect capabilities. It is necessary to run this command after reboot of the system.

In order to load the module automatically after every reboot, run the following command:

```
echo "nvidia-peermem" | sudo tee /etc/modules-load.d/nvidia-peermem.conf
```

Note:

`nvidia_peer_memory` module not loading:

DGX OS 5.1.1 provides nv_peer_mem 1.2 and MLNX_OFED 5.4-3.1.0.0 to resolve an issue discovered in MLNX_OFED 5.4-1.0.3.0. `nv_peer_mem` 1.2 isn't compatible with MLNX_OFED <= 5.4-1.0.3.0, and attempting to use `nv_peer_mem` 1.2 with MLNX_OFED <= 5.4-1.0.3.0 will result in a error such as the one below:

```
root@dgx-02:~# cat /var/lib/dkms/nv_peer_mem/1.2/build/make.log
DKMS make.log for nv_peer_mem-1.2 for kernel 5.4.0-92-generic (x86_64)
Wed Jan 5 20:36:09 UTC 2022
INFO: Building with MLNX_OFED from: /usr/src/ofa_kernel/default
```

If you must use MLNX_OFED <= 5.4-1.0.3.0 and have encountered this issue, then it is recommended to downgrade to `nv_peer_mem` 1.1.

Ubuntu:

```
sudo apt install --reinstall nvidia-peer-memory-dkms=1.1-0-nvidia2
```

RHEL8:

```
sudo dnf downgrade nvidia-peer-memory-dkms-21.03-0.el
```

```
modprobe nv_peer_mem
```

7.3.2. Installing nvidia-gds

To use GDS, perform the following.install `nvidia-gds` with the correct dependencies.

1. Populate the `${NVIDIA_DRV_VERSION}` variable.

   ```
   $ NVIDIA_DRV_VERSION=$(cat /proc/driver/nvidia/version | grep Module | awk '{print $8}' |
    cut -d '.' -f 1)
   ```

2. Install `nvidia-gds` with the correct dependencies.

   ```
   $ sudo apt install nvidia-gds-<ver> nvidia-dkms-${NVIDIA_DRV_VERSION}-server
   ```

Use the CUDA Toolkit version number in place of `<ver>`; for example, `11-4`.

Chapter 8. Network Configuration

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

8.1. Configuration 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.

8.1.1. 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:

```
http_proxy="http://myproxy.server.com:8080/"
ftp_proxy="ftp://myproxy.server.com:8080/";
https_proxy="https://myproxy.server.com:8080/";
```

8.1.2. 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/";
```

8.1.3. For Docker

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.

8.2. 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.

8.2.1. 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
  ```

  user-login-id

  The user login ID of the existing user that you are adding to the `docker` group.

8.2.2. 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/dockeroverride.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/dockeroverride.conf` file.
4. Reload the `systemctl` daemon.

```
$ sudo systemctl daemon-reload
```

5. Restart Docker.

```
$ sudo systemctl restart docker
```

8.3. 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.

8.3.1. In-Band Management, Storage, and Compute Networks

This 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)	Inbound/Outbound	RPCBIND, required by NFS
273 (TCP)		NVIDIA System 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)

8.3.2. Out-of-Band Management

This table provides information about out-of-band management for your DGX system.

Table 2. Out-of-Band Management

Port (Protocol)	Direction	Use
443 (TCP)	Inbound	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)	Inbound	IPMI

8.4. 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:

- http://archive.ubuntu.com/ubuntu/
- http://security.ubuntu.com/ubuntu/
- http://repo.download.nvidia.com/baseos/ubuntu/focal/x86_64/

  You can access this URL by using `apt-get` and not in a browser.
- https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64/
- https://apt.dockerproject.org/repo/
- https://nvcr.io/

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
```

8.5. 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.

See Configuring Network Proxies for the port designation of the connection that you want to configure.

2. Edit the network configuration yaml file.

Note: Ensure that your file identical to the following sample and use spaces and not tabs.

```
$ sudo vi /etc/netplan/01-netcfg.yaml

network: version: 2
renderer: networkd Ethernets:
<port-designation>: dhcp4: no
dhcp6: no
addresses: [10.10.10.2/24] gateway4: 10.10.10.1 nameservers:
search: [<mydomain>, <other-domain>]
 addresses: [10.10.10.1, 1.1.1.1]
```

Consult your network administrator for the appropriate information for the items in bold, such as network, gateway, and nameserver addresses, and use the port designations that you determined in step 1.

3. After you complete your edits, press **ESC** to switch to the command mode.
4. Save the file to the disk and exit the editor.
5. Apply the changes.

```
$ sudo netplan apply
```

Note: If you are not returned to the command line prompt after a minute, then reboot the system.For additional information, see Changes, errors, and bugs in the *Ubuntu Server Guide*.

Chapter 9. Additional Features and Instructions

This section provides information about uncommon configuration and features.

9.1. 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.

9.1.1. 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
```

9.1.2. 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 for example output.

9.1.3. 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.

9.2. 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.

9.2.1. 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.

9.2.2. Connecting to Serial Over LAN

You can connect to serial over a LAN.

Important: This information applies **only** to 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 -U -P sol activate
```

9.3. 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.

9.4. 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 a Multi-Instance GPU

When you log into the system and check which GPUs are available, you find the following:

```
lab@ro-dvt-058-80gb:~$ 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.

9.4.1. 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:

```
lab@ro-dvt-058-80gb:~$ 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:

```
lab@ro-dvt-058-80gb:~ $ 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:

```
lab@ro-dvt-058-80gb:~$ 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.

9.4.2. 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.

Important: If the `CUDA_DEVICE_ORDER` variable is set to `PCI_BUS_ID`, this ordering will be overridden.

In the following example, a CUDA application that comes with CUDA samples is run. In the output, GPU 0 is the fastest in a DGX Station A100, and GPU 4 (DGX Display GPU) is the slowest:

```
lab@ro-dvt-058-80gb:~$ sudo apt install cuda-samples-11-2
lab@ro-dvt-058-80gb:~$ cd /usr/local/cuda-11.2/samples/1_Utilities/p2pBandwidthLatencyTest
lab@ro-dvt-058-80gb:/usr/local/cuda-11.2/samples/1_Utilities/p2pBandwidthLatencyTest
$ sudo make
/usr/local/cuda/bin/nvcc -ccbin g++ -I../../common/inc  -m64    --threads
0 -gencode arch=compute_35,code=sm_35 -gencode arch=compute_37,code=sm_37
-gencode arch=compute_50,code=sm_50 -gencode arch=compute_52,code=sm_52
-gencode arch=compute_60,code=sm_60 -gencode arch=compute_61,code=sm_61
-gencode arch=compute_70,code=sm_70 -gencode arch=compute_75,code=sm_75
-gencode arch=compute_80,code=sm_80 -gencode arch=compute_86,code=sm_86
-gencode arch=compute_86,code=compute_86 -o p2pBandwidthLatencyTest.o -c
 p2pBandwidthLatencyTest.cu
nvcc warning : The 'compute_35', 'compute_37', 'compute_50', 'sm_35', 'sm_37' and
 'sm_50' architectures are deprecated, and may be removed in a future release (Use -
Wno-deprecated-gpu-targets to suppress warning).
/usr/local/cuda/bin/nvcc -ccbin g++   -m64
-gencode arch=compute_35,code=sm_35 -gencode arch=compute_37,code=sm_37
-gencode arch=compute_50,code=sm_50 -gencode arch=compute_52,code=sm_52
-gencode arch=compute_60,code=sm_60 -gencode arch=compute_61,code=sm_61
-gencode arch=compute_70,code=sm_70 -gencode arch=compute_75,code=sm_75
-gencode arch=compute_80,code=sm_80 -gencode arch=compute_86,code=sm_86
-gencode arch=compute_86,code=compute_86 -o p2pBandwidthLatencyTest
 p2pBandwidthLatencyTest.o
nvcc warning : The 'compute_35', 'compute_37', 'compute_50', 'sm_35', 'sm_37' and
 'sm_50' architectures are deprecated, and may be removed in a future release (Use -
Wno-deprecated-gpu-targets to suppress warning).
mkdir -p ../../bin/x86_64/linux/release
cp p2pBandwidthLatencyTest ../../bin/x86_64/linux/release
lab@ro-dvt-058-80gb:/usr/local/cuda-11.2/samples/1_Utilities/p2pBandwidthLatencyTest
 $ cd /usr/local/cuda-11.2/samples/bin/x86_64/linux/release
lab@ro-dvt-058-80gb:/usr/local/cuda-11.2/samples/bin/x86_64/linux/release $ ./
p2pBandwidthLatencyTest
[P2P (Peer-to-Peer) GPU Bandwidth Latency Test]
Device: 0, Graphics Device, pciBusID: 1, pciDeviceID: 0, pciDomainID:0
Device: 1, Graphics Device, pciBusID: 47, pciDeviceID: 0, pciDomainID:0
Device: 2, Graphics Device, pciBusID: 81, pciDeviceID: 0, pciDomainID:0
Device: 3, Graphics Device, pciBusID: c2, pciDeviceID: 0, pciDomainID:0
Device: 4, DGX Display, pciBusID: c1, pciDeviceID: 0, pciDomainID:0
Device=0 CAN Access Peer Device=1
Device=0 CAN Access Peer Device=2
Device=0 CAN Access Peer Device=3
Device=0 CANNOT Access Peer Device=4
Device=1 CAN Access Peer Device=0
Device=1 CAN Access Peer Device=2
Device=1 CAN Access Peer Device=3
Device=1 CANNOT Access Peer Device=4
Device=2 CAN Access Peer Device=0
Device=2 CAN Access Peer Device=1
Device=2 CAN Access Peer Device=3
Device=2 CANNOT Access Peer Device=4
Device=3 CAN Access Peer Device=0
Device=3 CAN Access Peer Device=1
Device=3 CAN Access Peer Device=2
Device=3 CANNOT Access Peer Device=4
Device=4 CANNOT Access Peer Device=0
Device=4 CANNOT Access Peer Device=1
Device=4 CANNOT Access Peer Device=2
Device=4 CANNOT Access Peer Device=3


***NOTE: In case a device doesn't have P2P access to other one, it falls back to
 normal memcopy procedure.
So you can see lesser Bandwidth (GB/s) and unstable Latency (us) in those cases.
```

```
P2P Connectivity Matrix
     D\D     0     1     2     3     4
     0       1     1     1     1     0
     1       1     1     1     1     0
     2       1     1     1     1     0
     3       1     1     1     1     0
     4       0     0     0     0     1
Unidirectional P2P=Disabled Bandwidth Matrix (GB/s)
   D\D     0       1       2       3       4
     0 1323.03  15.71  15.37  16.81  12.04
     1  16.38 1355.16  15.47  15.81  11.93
     2  16.25  15.85 1350.48  15.87  12.06
     3  16.14  15.71  16.80 1568.78  11.75
     4  12.61  12.47  12.68  12.55 140.26
Unidirectional P2P=Enabled Bandwidth (P2P Writes) Matrix (GB/s)
   D\D     0       1       2       3       4
     0 1570.35  93.30  93.59  93.48  12.07
     1  93.26 1583.08  93.55  93.53  11.93
     2  93.44  93.58 1584.69  93.34  12.05
     3  93.51  93.55  93.39 1586.29  11.79
     4  12.68  12.54  12.75  12.51 140.26
Bidirectional P2P=Disabled Bandwidth Matrix (GB/s)
   D\D     0       1       2       3       4
     0 1588.71  19.60  19.26  19.73  16.53
     1  19.59 1582.28  19.85  19.13  16.43
     2  19.53  19.39 1583.88  19.61  16.58
     3  19.51  19.11  19.58 1592.76  15.90
     4  16.36  16.31  16.39  15.80 139.42
Bidirectional P2P=Enabled Bandwidth Matrix (GB/s)
   D\D     0       1       2       3       4
     0 1590.33 184.91 185.37 185.45  16.46
     1 185.04 1587.10 185.19 185.21  16.37
     2 185.15 185.54 1516.25 184.71  16.47
     3 185.55 185.32 184.86 1589.52  15.71
     4  16.26  16.28  16.16  15.69 139.43
P2P=Disabled Latency Matrix (us)
   GPU     0       1       2       3       4
     0   3.53  21.60  22.22  21.38  12.46
     1  21.61   2.62  21.55  21.65  12.34
     2  21.57  21.54   2.61  21.55  12.40
     3  21.57  21.54  21.58   2.51  13.00
     4  13.93  12.41  21.42  21.58   1.14

   CPU     0       1       2       3       4
     0   4.26  11.81  13.11  12.00  11.80
     1  11.98   4.11  11.85  12.19  11.89
     2  12.07  11.72   4.19  11.82  12.49
     3  12.14  11.51  11.85   4.13  12.04
     4  12.21  11.83  12.11  11.78   4.02
P2P=Enabled Latency (P2P Writes) Matrix (us)
   GPU     0       1       2       3       4
     0   3.79   3.34   3.34   3.37  13.85
     1   2.53   2.62   2.54   2.52  12.36
     2   2.55   2.55   2.61   2.56  12.34
     3   2.58   2.51   2.51   2.53  14.39
     4  19.77  12.32  14.75  21.60   1.13

   CPU     0       1       2       3       4
     0   4.27   3.63   3.65   3.59  13.15
     1   3.62   4.22   3.61   3.62  11.96
     2   3.81   3.71   4.35   3.73  12.15
     3   3.64   3.61   3.61   4.22  12.06
     4  12.32  11.92  13.30  12.03   4.05

NOTE: The CUDA Samples are not meant for performance measurements. Results may vary
 when GPU Boost is enabled.
```

The example above shows the peer-to-peer bandwidth and latency test across all five GPUs, including the DGX Display GPU. The application also shows that there is no peer-to-peer connectivity between any GPU and GPU 4. This indicates that GPU 4 should not be used for high-performance workloads.

Run the example one more time by using the CUDA_VISIBLE_DEVICES variable, which limits the number of GPUs that the application can see.

Note: All GPUs can communicate with all other peer devices.

```
lab@ro-dvt-058-80gb: /usr/local/cuda-11.2/samples/bin/x86_64/linux/release$
 CUDA_VISIBLE_DEVICES=0,1,2,3 ./p2pBandwidthLatencyTest
[P2P (Peer-to-Peer) GPU Bandwidth Latency Test]
Device: 0, Graphics Device, pciBusID: 1, pciDeviceID: 0, pciDomainID:0
Device: 1, Graphics Device, pciBusID: 47, pciDeviceID: 0, pciDomainID:0
Device: 2, Graphics Device, pciBusID: 81, pciDeviceID: 0, pciDomainID:0
Device: 3, Graphics Device, pciBusID: c2, pciDeviceID: 0, pciDomainID:0
Device=0 CAN Access Peer Device=1
Device=0 CAN Access Peer Device=2
Device=0 CAN Access Peer Device=3
Device=1 CAN Access Peer Device=0
Device=1 CAN Access Peer Device=2
Device=1 CAN Access Peer Device=3
Device=2 CAN Access Peer Device=0
Device=2 CAN Access Peer Device=1
Device=2 CAN Access Peer Device=3
Device=3 CAN Access Peer Device=0
Device=3 CAN Access Peer Device=1
Device=3 CAN Access Peer Device=2

***NOTE: In case a device doesn't have P2P access to other one, it falls back to
 normal memcopy procedure.
So you can see lesser Bandwidth (GB/s) and unstable Latency (us) in those cases.

P2P Connectivity Matrix
     D\D     0     1     2     3
     0       1     1     1     1
     1       1     1     1     1
     2       1     1     1     1
     3       1     1     1     1
Unidirectional P2P=Disabled Bandwidth Matrix (GB/s)
   D\D     0      1      2      3
     0 1324.15  15.54  15.62  15.47
     1  16.55 1353.99  15.52  16.23
     2  15.87  17.26 1408.93  15.91
     3  16.33  17.31  18.22 1564.06
Unidirectional P2P=Enabled Bandwidth (P2P Writes) Matrix (GB/s)
   D\D     0      1      2      3
     0 1498.08  93.30  93.53  93.48
     1  93.32 1583.08  93.54  93.52
     2  93.55  93.60 1583.08  93.36
     3  93.49  93.55  93.28 1576.69
Bidirectional P2P=Disabled Bandwidth Matrix (GB/s)
   D\D     0      1      2      3
     0 1583.08  19.92  20.47  19.97
     1  20.74 1586.29  20.06  20.22
     2  20.08  20.59 1590.33  20.01
     3  20.44  19.92  20.60 1589.52
Bidirectional P2P=Enabled Bandwidth Matrix (GB/s)
   D\D     0      1      2      3
     0 1592.76 184.88 185.21 185.30
     1 184.99 1589.52 185.19 185.32
     2 185.28 185.30 1585.49 185.01
     3 185.45 185.39 184.84 1587.91
```

```
P2P=Disabled Latency Matrix (us)
   GPU     0      1      2      3
     0   2.38  21.56  21.61  21.56
     1  21.70   2.34  21.54  21.56
     2  21.55  21.56   2.41  21.06
     3  21.57  21.34  21.56   2.39

   CPU     0      1      2      3
     0   4.22  11.99  12.71  12.09
     1  11.86   4.09  12.00  11.71
     2  12.52  11.98   4.27  12.24
     3  12.22  11.75  12.19   4.25
P2P=Enabled Latency (P2P Writes) Matrix (us)
   GPU     0      1      2      3
     0   2.32   2.57   2.55   2.59
     1   2.55   2.32   2.59   2.52
     2   2.59   2.56   2.41   2.59
     3   2.57   2.55   2.56   2.40

   CPU     0      1      2      3
     0   4.24   3.57   3.72   3.81
     1   3.68   4.26   3.75   3.63
     2   3.79   3.75   4.34   3.71
     3   3.72   3.64   3.66   4.32

NOTE: The CUDA Samples are not meant for performance measurements. Results may vary
 when GPU Boost is enabled.
```

For bare metal applications, the UUID can also be specified in the `CUDA_VISIBLE_DEVICES` variable as shown below:

```
lab@ro-dvt-058-80gb:/usr/local/cuda-11.2/samples/bin/x86_64/linux/release
 $ CUDA_VISIBLE_DEVICES=GPU-0f2dff15-7c85-4320-da52-d3d54755d182,GPU-dc598de6-
dd4d-2f43-549f-f7b4847865a5 ./p2pBandwidthLatencyTest
```

The GPU specification is longer because of the nature of UUIDs, but this is the most precise way to pin specific GPUs to the application.

9.4.3. 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:

```
lab@ro-dvt-058-80gb:~$ 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)
  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:

```
lab@ro-dvt-058-80gb:~$ 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.

```
lab@ro-dvt-058-80gb:~$ 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:

Remember: 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 clearly illustrated below.

```
lab@ro-dvt-058-80gb: /usr/local/cuda-11.2/samples/bin/x86_64/linux/release
$ 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: In case a device doesn't have P2P access to other one, it falls back to
 normal memcopy procedure.
So 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.
```

9.5. Updating the containerd Override File

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.

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:

   ```
   colossus@ro-evt-038-80gb:~$ sudo systemctl restart nv-docker-gpus
   colossus@ro-evt-038-80gb:~$ 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:

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 450.80.02    Driver Version: 450.80.02    CUDA Version: 11.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Graphics Device     On   | 00000000:C2:00.0 Off |                   On |
| N/A   32C    P0    65W / 275W |                  N/A |     N/A      Default |
|                               |                      |              Enabled |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| MIG devices:                                                                |
+------------------+----------------------+-----------+----------------------+
| GPU  GI  CI  MIG |         Memory-Usage |        Vol|         Shared       |
|      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:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Chapter 10. 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 must set up your own NFS storage for long-term data storage.

10.1. 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.

10.1.1. 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.

- 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.

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 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.

10.1.2. Disabling cachefilesd

Here is some information about how to disable `cachefilesd`.

If you do not want to enable `cachefilesd` by running:

```
$ sudo systemctl stop cachefilesd
$ sudo systemctl disable cachefilesd
```

10.2. 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 2 drives to complete these tasks. If you have less than 2 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.

Remember: 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.

Chapter 11. Running NGC Containers

This section provides information about how to run NGC containers with your DGX system.

11.1. 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.

11.2. 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 5 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,1"' ...
  ```

- Refer to Running Containers for more information about running NGC containers on MIG devices.
  ```

Appendix A. Installing the Software on Air-Gapped DGX Systems

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 Installing the DGX OS (Reimaging the System).
- 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.

A.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 DGX OS Server Release Number Scheme.

Note: These procedures apply only to upgrades in the same major release, such as from 5.x to 5.y. The steps do not support upgrades across major releases, such as from 4.x to 5.x.

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.

A.2. Creating the Mirror in a DGX OS 5 System

The instructions in this section are to be performed on a system with network access.

The following are the prerequisites.

- 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 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.

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.

```
############# 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 focal-security main multiverse universe
 restricted
deb http://archive.ubuntu.com/ubuntu/ focal main multiverse universe restricted
deb http://archive.ubuntu.com/ubuntu/ focal-updates main multiverse universe
 restricted
#
deb-i386 http://security.ubuntu.com/ubuntu focal-security main multiverse
 universe restricted
deb-i386 http://archive.ubuntu.com/ubuntu/ focal main multiverse universe
 restricted
deb-i386 http://archive.ubuntu.com/ubuntu/ focal-updates main multiverse universe
 restricted
#
# CUDA specific repositories:
deb http://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64/ /
#
# DGX specific repositories:
deb http://repo.download.nvidia.com/baseos/ubuntu/focal/x86_64/ focal common dgx
deb http://repo.download.nvidia.com/baseos/ubuntu/focal/x86_64/ focal-updates
 common dgx
#
deb-i386 http://repo.download.nvidia.com/baseos/ubuntu/focal/x86_64/ focal common
 dgx
deb-i386 http://repo.download.nvidia.com/baseos/ubuntu/focal/x86_64/ focal-
updates common dgx
# Clean unused items
clean http://archive.ubuntu.com/ubuntu
clean http://security.ubuntu.com/ubuntu
```

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
```

A.3. Configuring the Target Air-Gapped DGX OS 5 System

Here are the steps that explain how you can configure a target air-gapped DGX OS 5 system.

The instructions in this section are to be performed on the target air-gapped DGX system.

The following are the prerequisites.

- 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 (http://manpages.ubuntu.com/manpages/bionic/en/man8/apt.8.html) command to use the file system as the repository in the file `/etc/apt/sources.list` by modifying the following lines.

   ```
   deb file:///media/usb/repository/mirror/security.ubuntu.com/ubuntu focal-security
    main multiverse universe restricted
   deb file:///media/usb/repository/mirror/archive.ubuntu.com/ubuntu/ focal main
    multiverse universe restricted
   deb file:///media/usb/repository/mirror/archive.ubuntu.com/ubuntu/ focal-updates
    main multiverse universe restricted
   ```
3. Configure `apt` to use the NVIDIA DGX OS packages in the `/etc/apt/sources.list.d/dgx.list` file.

   ```
   deb file:///media/usb/repository/mirror/repo.download.nvidia.com/baseos/ubuntu/
   focal/x86_64/ focal main dgx
   deb file:///media/usb/repository/mirror/repo.download.nvidia.com/baseos/ubuntu/
   focal/x86_64/ focal-updates main dgx
   ```
4. Configure `apt` to use the NVIDIA CUDA packages in the `/etc/apt/sources.list.d/cuda-compute-repo.list` file.

   ```
   deb file:///media/usb/repository/mirror/developer.download.nvidia.com/compute/
   cuda/repos/ubuntu2004/x86_64/ /
   ```
5. Update the `apt` repository.

   ```
   $ sudo apt update
   ```

   Output from this command is similar to the following example.

   ```
   Get:1 file:/media/usb/repository/mirror/security.ubuntu.com/ubuntu focal-security
    InRelease [107 kB]
   Get:2 file:/media/usb/repository/mirror/archive.ubuntu.com/ubuntu focal InRelease
    [265 kB]
   Get:3 file:/media/usb/repository/mirror/archive.ubuntu.com/ubuntu focal-updates
    InRelease [111 kB]
   Get:4 file:/media/usb/repository/mirror/developer.download.nvidia.com/compute/
   cuda/repos/ubuntu2004/x86_64  InRelease
   Get:5 file:/media/usb/repository/mirror/repo.download.nvidia.com/baseos/ubuntu/
   focal/x86_64 focal InRelease [12.5 kB]
   ```

```
Get:6 file:/media/usb/repository/mirror/repo.download.nvidia.com/baseos/ubuntu/
focal/x86_64 focal-updates InRelease [12.4 kB]
Get:7 file:/media/usb/repository/mirror/developer.download.nvidia.com/compute/
cuda/repos/ubuntu2004/x86_64  Release [697 B]
Get:8 file:/media/usb/repository/mirror/developer.download.nvidia.com/compute/
cuda/repos/ubuntu2004/x86_64  Release.gpg [836 B]
Reading package lists... Done
```

6. Upgrade the system using the newly configured local repositories.

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

Appendix B. Setting Up DGX OS 5 for PXE Boot

This chapter describes how to set up your network and the DGX server so that the server boots from it using the preboot execution environment (PXE). 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, FTP, and TFTP servers. It is expected that the end user's lab admin will configure them to fit within their company's security guidelines.

Assumptions

1. A TFTP and FTP server are setup

   a). IP address is <FTP IP>

   b). Fully qualified host is <FTP host>

This document is intended to provide detailed step-by-step instructions on how to set up a PXE boot environment for DGX systems. The examples are based on a DGX A100. There are several major components of the solution:

- DHCP server: dnsmasq (https://thekelleys.org.uk/dnsmasq/doc.html) is used in this doc.
- TFTP server: dnsmasq is also used as a TFTP server.
- HTTP server: HTTP server is used to transfer large files, such is ISO image and initrd. Alternatively, FTP can also be used for this purpose. HTTP in used this doc.
- Syslinux: Linux bootloader software package.

Overview of the PXE Server

The PXE Server is divided up into the following areas:

- Bootloader (syslinux)
- FTP contents (i.e. the ISO image and its extracted contents)

Note that when comparing the Base OS 5.x contents to previous versions, there are comparatively very few files in the ISO. The rough directory structure on the FTP server will look like (where 5.x.y is the version you're installing)::

```
/local/ftp
```

```
└── dgxbaseos-5.x.y
    ├── base_os_5.x.y.iso
    ├── boot/
    │   └── grub
    │       └── bios.img
    │       └── boot.catdefault
    │       └── grub.cfg
    └── EFI
    │   └── efiboot.imgg
    ├── initrd
    ├── live/
    │       └── filesystem.squashfs
    ├── md5sum.txt
    ├── UBUNTU_LIVE
    ├── vmlinuz
/local/syslinux
└── efi64
    ├── syslinux.efi
    ├── ldlinux.e64
    ├── pxelinux.cfg
    │   └── default
    ├── sdgx-5.x.y/
    │   └── vmlinuz
    │   └── initrd
```

The tftp-server (controlled by the xinetd service and configuration found in /etc/xinetd.d/tftp) points to the syslinux directory for when the system PXE boots. TFTP is what transfers the syslinux.efi file that is designated in the DHCP server's dhcpd.conf file (see Configure your DHCP server). By default after the syslinux.efi is booted it looks for a pxelinux.cfg/default file with the menu options for booting further, i.e. the kernel and initrd in that sdgx/ directory. That config file will look in the same working directory for its files as it did for the pxelinux.cfg directory. In this case it is /local/syslinux/efi64.

The following steps will assume the DHCP and PXE servers are configured to use the above directory structure. The lab admin, or whoever is in charge of deploying the PXE environment, should change the directory names and structure to fit their infrastructure.

B.1. Setting Up dnsmasq for DHCP

The DGX system BIOS does not allow you to manually specify a PXE server. You need to use the DHCP server to determine what it is and from which file to boot.

1. Install dnsmasq.
   ```
   $ apt install dnsmasq
   ```
2. Back up the config file prior to customizing it.
   ```
   $ mv /etc/dnsmasq.conf /etc/dnsmasq.conf.backup
   ```
3. Edit the config file using the appropriate values for your system.
   ```
   $ vi /etc/dnsmasq.conf

   interface=enp226s0
   bind-interfaces
   domain=nvidia.com

   dhcp-range=enp226s0,10.33.254.106,static,1h
   dhcp-host=5c:ff:35:e1:a4:d1,DGX-A100-PM2,10.33.254.106
   dhcp-option=option:router,10.33.254.1
   dhcp-option=option:dns-server,8.8.8.8
   ```

```
enable-tftp
tftp-root=/local/syslinux/efi64/
dhcp-boot=syslinux.efi
```

4. Create and set up the `local/syslinux/efi64/` folder.

```
$ sudo mkdir -p /local/syslinux/efi64
$ systemctl restart dnsmasq
$ systemctl status dnsmasq
```

B.2. Configuring the HTTP File Directory and ISO Image

Here are the steps to configure the http file directory and ISO image for PXE boot.

The instructions produce the following file structure on the PXE server.

```
/local/http
└── dgxbaseos-5.x.y
    ├── base_os_5.x.y.iso
    ├── initrd
    └── vmlinuz
```

You should have obtained and copied the ISO image to "`/tmp/dgxbaseos-5.x.y.iso`".

1. Create the DGX OS directory.

```
$ sudo mkdir -p /local/http/dgxbaseos-5.x.y
```

2. Mount the ISO.

   Your mount point is `/mnt`.

```
$ sudo mount -o loop /tmp/base_os_5.x.y.iso /mnt
```

3. Copy the ISO and its extracted contents to the HTTP directory.

```
$ sudo cp /mnt/{initrd,vmlinuz} /local/http/dgxbaseos-5.x.y/

$ cp /tmp/dgxbaseos-5.0.2.iso /local/http/dgxbaseos-5.x.y/
```

4. Ensure all the files are owned, or are at least readable, by the `http` user.
5. You can now unmount the ISO.

```
$ sudo umount /mnt
```

B.3. Customizing the Curtin YAML

The curtin file is located in the filesystem.squashfs. Creating a touchless install, such as by seeding the initial userid and password, and skipping the oem-config steps, is accomplished by customizing the curtin file. (nooemconfig boot parameter described above does initial userid/password seeding and oem-config skip. For any other customization to curtin yaml, follow the steps below).

1. Get the filesystem.squashfs from the ISO
2. Run "unsquashfs filesystem.squashfs"

3. Copy the curtin file for the platform you're trying to PXE install from the **squashfs-root/curtin/** directory into a webserver or FTP server.
4. Customize the curtin file
5. Use the **force-curtin** parameter in your PXE bootloader entry to point to the URL of your new curtin file

More info about curtin can be found here: https://curtin.readthedocs.io/en/latest/index.html

An example curtin file for performing touchless installs can be found here: https://gitlab-master.nvidia.com/jamien/jamien-work/-/blob/master/curtin/fast-curtin.yaml

B.4. Configure your DHCP server

The DGX-Server BIOS unfortunately does not allow you to specify a PXE server manually. You have to rely on the DHCP server to tell you what it is, and what file to boot from. In the DHCP server's /etc/dhcp/dhcpd.conf file:

- Set the "next-server" to <FTP IP_ADDR>
- Set the "filename" to "efi64/syslinux.efi"

Again, this assumes the directory structure in the Overview.

Optional: Configure CX-4/5/6 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, then 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.

Query UEFI PXE ROM state

In order to PXE boot from the MLNX CX-4/5/6 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 can't be done remotely. The configuration of the MLNX CX devices is handled through the "mlxconfig" command.

To determine the device name and current configurations of the MLNX CX cards, run `$ sudo mlxconfig query`

user@dgx1server$ 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_x86_ENABLE False(0)

...

...

Enable UEFI PXE ROM

The "EXP_ROM_UEFI_x86_ENABLE" configuration must be set to True(1) for the MLNX CX card that you wish to PXE boot from, and reboot.

user@dgx1server$ sudo mlxconfig -y -d /dev/mst/mt4115_pciconf3 set EXP_ROM_UEFI_x86_ENABLE=1

user@dgx1server$ sudo reboot

Upon reboot, confirm the configuration was set.

user@dgx1server$ 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_x86_ENABLE True(1)

...

...

B.5. Configuring the syslinux Bootloader

These instructions install the syslinux 6.04-pre1 package.
The 5.x.y OS image does not include a bootloader for PXE boot. Instead, the syslinux from kernel.org is used. The syslinux 6.04-pre1 package is installed because fixes in 6.04 are required to support PXE booting from the Mellanox ethernet cards as an option. PXE boot from the Mellanox cards also require at least 16.23.1020 level firmware with:

- FW 16.23.1020
- PXE 3.5.0504
- UEFI 14.16.0017

These instructions produce the following file structure on the PXE server.

```
/local/syslinux
└── efi64
    ├── ldlinux.e64
    ├── libcom32.c32
    ├── libutil.c32
    ├── menu.c32
    ├── pxelinux.0
    ├── pxelinux.cfg
    │   └── default
    ├── syslinux.efi
    └── vesamenu.c32
```

1. Install Syslinux 6.04-pre1
   a). Download and extract the syslinux archive.

   ```
   $ cd /local
   $ wget https://mirrors.edge.kernel.org/pub/linux/utils/boot/syslinux/
   Testing/6.04/syslinux-6.04-pre1.tar.gz
   $ sudo tar -xvf syslinux-6.04-pre1.tar.gz
   ```

   b). Copy the syslinux files.

   ```
   $ sudo ln -s /local/syslinux-6.04-pre1 /local/syslinux
   $ sudo mkdir /local/syslinux/efi64/pxelinux.cfg
   $ sudo cp syslinux-6.04-pre1/bios/core/pxelinux.0 syslinux/efi64/
   $ sudo cp syslinux-6.04-pre1/efi64/efi/syslinux.efi syslinux/efi64/
   $ sudo cp syslinux-6.04-pre1/efi64/com32/elflink/ldlinux/ldlinux.e64 syslinux/
   efi64/
   $ sudo cp syslinux-6.04-pre1/efi64/com32/menu/vesamenu.c32 syslinux/efi64/
   $ sudo cp syslinux-6.04-pre1/efi64/com32/menu/menu.c32 syslinux/efi64/
   $ sudo cp syslinux-6.04-pre1/efi64/com32/libutil/libutil.c32 syslinux/efi64
   $ sudo cp syslinux-6.04-pre1/efi64/com32/lib/libcom32.c32 syslinux/efi64/
   ```

2. Edit PXE boot default configuration file as follows and make sure you update the IP address of the http server:

   ```
   $ sudo vi /local/syslinux/efi64/pxelinux.cfg/default

   default menu
   prompt 0
   timeout 300

   menu title DGX Install Images

   label dgxbaseos-5.x.y
           menu label DGXBaseOS 5.x.y (Base OS)
   ```

```
        kernel http://<http-address>/dgxbaseos-5.x.y/vmlinuz
        initrd http://<http-address>/dgxbaseos-5.x.y/initrd
        append boot=live config components union=overlay noswap noeject
 ip=:::::enp226s0:dhcp ethdevice-timeout=60 fetch=http://<http-address>/
dgxbaseos-5.x.y/DGXOS-5.x.y-<release-labels>.iso apparmor=0 elevator=noop
 nvme-core.multipath=n nouveau.modeset=0 rebuild-raid offwhendone console=tty0
 console=ttyS1,115200n8
```

The kernel boot parameters should match the contents of the corresponding ISO's boot menu, found in `/mnt/boot/grub/grub.cfg`.

B.5.1. Live Boot Parameters

Here is some information about the live boot parameters specified in the `pxelinux.cfg/default` file.

Note: The manpages for all of the live-boot parameters used above (and more) can be found here..

The following parameters can be used to configure your system's network interfaces:

ip=

Tells the `initramfs` to automatically configure the system's interfaces by using DHCP. If only one interface is connected to the network, this interface should be enough. If multiple interfaces are connected to the network, then the `initramfs` will go with the first interface that receives a reply.

In this case, you might need to provide a more specific configuration. For instance,`ip=:::::enp1s0f0:dhcp` tells the `initramfs` to configure the `enp1s0f0` interface by using DHCP.

ethdevice-timeout

Allows you to specify how long to wait for the ethernet devices to get configured. The example above gives the system 60 seconds to configure its interfaces before giving up.

The following parameters are unique to the DGX OS Installer:

rebuild-raid

If the data RAID is specified, tells the installer to rebuild the RAID. An installation from the factory should always specify this parameter, but otherwise, it is optional.

curtin-encrypt-root

Sets up LUKS encryption on the root filesystem using a well-known passphrase.

no-mlnx-fw-update

Skips updating the Mellanox cards firmware during the installation.

B.6. Boot the DGX System over PXE

You can power cycle the DGX A100 through BMC GUI, or, alternatively, use "ipmitool" to set pxe boot and reboot the client DGX A100:

```
$ ipmitool -I lanplus -H <DGX-BMC-IP> -U <username> -P <password> chassis bootdev
 pxe options=efiboot
$ ipmitool -I lanplus -H <DGX-BMC-IP> -U <username> -P <password> chassis power
 reset
```

Troubleshooting

During the PXE boot process, you may see following errors from client BMC:

```
BootCurrent: 0018
Timeout: 1 seconds
BootOrder: 0000,0006,0007,0008,000C,000D,000E,000F,0016,0017,0001
Boot0000* ubuntu
Boot0001  UEFI: Built-in EFI Shell
Boot0004* UEFI: PXE IPv4 Intel(R) I210 Gigabit  Network Connection
Boot0005* UEFI: PXE IPv6 Intel(R) I210 Gigabit  Network Connection
Boot0006* UEFI: PXE IPv4 Mellanox Network Adapter - 0C:42:A1:5F:44:7C
Boot0007* UEFI: PXE IPv4 Mellanox Network Adapter - 0C:42:A1:74:74:9E
Boot0008* UEFI: PXE IPv4 Mellanox Network Adapter - 0C:42:A1:74:74:8E
Boot0009* UEFI: PXE IPv4 Mellanox Network Adapter - 0C:42:A1:74:6B:CA
Boot000A* UEFI: PXE IPv4 Mellanox Network Adapter - 0C:42:A1:74:74:9A
Boot000B* UEFI: PXE IPv4 Mellanox Network Adapter - 0C:42:A1:5F:51:D8
Boot000C* UEFI: PXE IPv4 Mellanox Network Adapter - 0C:42:A1:74:6A:FA
Boot000D* UEFI: PXE IPv4 Mellanox Network Adapter - 0C:42:A1:74:74:C2
Boot000E* UEFI: PXE IPv4 Mellanox Network Adapter - 0C:42:A1:4A:62:60
Boot000F* UEFI: PXE IPv4 Mellanox Network Adapter - 0C:42:A1:4A:62:61
Boot0016* UEFI: PXE IPv6 Mellanox Network Adapter - 0C:42:A1:4A:62:60
Boot0017* UEFI: PXE IPv6 Mellanox Network Adapter - 0C:42:A1:4A:62:61
Invalid BootOrder order entry value0018
                                    ^
efibootmgr: entry 0018 does not exist
Unexpected error while running command.
Command: ['unshare', '--fork', '--pid', '--', 'chroot', '/tmp/tmpsgpxpe9k/target',
 'efibootmgr', '-o', '0018,0000,0006,0007,0008,000C,000D,000E,000F,0016,0017,0001']
Exit code: 8
```

The solution is to wait for the Ubuntu console to appear, login as root without password, then run the following commands to make the Intel I210 NIC as the first NIC to try PXE boot:

```
$ efibootmgr -o 0004,0000,0006,0007,0008,0009,000A,000B, 000C,000D,000E,000F
```

B.7. (Reference) Other IPMI Boot Options

Here is some information about other IPMI boot options.

- For more information about specifying boot order by using the `ipmi` command, refer to the ipmitool man page, and search for the following sections:
  - The `chassis` command
  - The `bootdev` subcommand
- For more information about the IPMI specification, see: https://www.intel.com/content/www/us/en/products/docs/servers/ipmi/ipmi-second-gen-interface-spec-v2-rev1-1.html

Appendix C. DGX OS 5 GPU Driver support

Overview

- R510 and earlier GPU drivers depend on NSCQ v1

> **Note:** Canonical R510 packages transition you to R515 automatically

- R515 and later have a dependency on NSCQ v2
- DCGM 2.x only supports NSCQ v1
- DCGM 3.x only supports NSCQ v2
- DCGM 2.x is hosted by the DGX OS 5 and EL8 repositories
- DCGM 2.x is installed by default on DGX OS 5 and EL8
- Both DCGM 2.x and 3.x are hosted by the CUDA repositories
- However, DCGM package metadata (such as Depends/Conflicts/Replaces directives) and DCGM package naming (i.e. not branchified names) means that if you only use the CUDA repo, you will be auto-upgraded to DCGM 3.x.
- NVSM earlier than 22.09.3 only supports DCGM 2.x APIs
- NVSM 22.09.3 and newer supports both DCGM 2.x and 3.x APIs
- DGX OS 5.4.2 and EL8-22.08.2 will provide NVSM 22.09.3

C.1. DGX OS 5 (Ubuntu 20.04)

DGX OS 5 enables both DGX and CUDA repos, but prioritizes the DGX repos higher than the CUDA repos (600 vs 580). DGX users will not be automatically upgraded to DCGM 3.x:

```
$ apt-cache policy datacenter-gpu-manager

  datacenter-gpu-manager:
  Installed: 1:2.4.7
  Candidate: 1:2.4.7
  Version table:
  1:3.1.3 580
  580 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64
 Packages
  1:3.0.4 580
```

```
 580 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64
Packages
 *** 1:2.4.7 600
 580 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64
Packages
 600 https://repo.download.n
```

How to support R510/R515+ and DCGM 3.x

APT will prevent automatic upgrades to DCGM based on the DGX repo preferences as seen above. However, it will not prevent you from directly installing DCGM 3.x from the CUDA repo:

```
$ sudo apt install datacenter-gpu-manager=1:3.0.4

  [sudo] password for lab:
  Reading package lists... Done
  Building dependency tree
  Reading state information... Done
  The following packages were automatically installed and are no longer required:
    apt-clone archdetect-deb bogl-bterm dgx-release dpkg-repack laptop-detect
 libdebian-installer4 libibdm1 libiw30 libmysqlclient21 libopensm9 - libosmcomp5
 libosmvendor5
    libsnmp-base libsnmp35 linux-headers-5.4.0-122 linux-headers-5.4.0-122-generic
 linux-image-5.4.0-122-generic linux-modules-5.4.0-122-generic
    linux-modules-extra-5.4.0-122-generic linux-signatures-nvidia-5.4.0-122-generic
 linux-tools-5.4.0-122 linux-tools-5.4.0-122-generic lldpd - - mysql-common ngc
    nvidia-kernel-defaults nvidia-lldpd-defaults nvidia-mig-manager python3-icu
 python3-pam rdate tasksel tasksel-data

  Use 'sudo apt autoremove' to remove them.

  The following packages will be upgraded:
    - datacenter-gpu-manager
  1 upgraded, 0 newly installed, 0 to remove and 75 not upgraded.
  1 not fully installed or removed.
  Need to get 0 B/497 MB of archives.

  After this operation, 237 MB of additional disk space will be used.
  (Reading database ... 167686 files and directories currently installed.)
  Preparing to unpack .../datacenter-gpu-manager_1%3a3.0.4_amd64.deb ...
  Unpacking datacenter-gpu-manager (1:3.0.4) over (1:2.4.7) ...
```

Once the installed version is greater than the prioritized version, then the APT preferences will no longer be used. Users will then be able to get OTA upgrades of DCGM 3.x from the CUDA repo as part of the usual "apt upgrade" process. To illustrate this behavior, DCGM 3.0.4 was installed in the example above, despite the CUDA repo also providing 3.1.3. APT will now consider 3.1.3 is a valid upgrade candidate despite the CUDA repo's lower priority:

```
$ apt-cache policy datacenter-gpu-manager

  datacenter-gpu-manager:
    Installed: 1:3.0.4
    Candidate: 1:3.1.3
    Version table:
       1:3.1.3 580
          580 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/
x86_64  Packages
   *** 1:3.0.4 580
          580 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/
x86_64  Packages
          100 /var/lib/dpkg/status
       1:2.4.7 600
          580 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/
x86_64  Packages
          600 https://repo.download.nvidia.com/baseos/ubuntu/focal/x86_64 focal-
updates/common amd64 Packages
```

C.2. Instructions

Switching from pre-R515 to R515+

1. Update to the latest DGX OS 5 to get NVSM 22.09.3 or higher

   $ sudo apt update

   $ sudo apt full-upgrade -y
2. Switch to the R515+ driver branch using the following steps:

   https://docs.nvidia.com/dgx/dgx-os-5-user-guide/index.html#install-new-cuda-driver
3. Determine the latest DCGM 3.x

   $ apt-cache policy datacenter-gpu-manager
4. Install the latest DCGM 3.x

   $ sudo apt install datacenter-gpu-manager=<version>

Switching from R515+ to pre-R515

1. Switch to the pre-R515 driver branch using the following steps:

   https://docs.nvidia.com/dgx/dgx-os-5-user-guide/index.html#install-new-cuda-driverNote: Canonical R510 packaging will transition automatically to the R515 branch.

Downgrade to DCGM 2.x $ sudo apt update $ sudo apt install datacenter-gpu-manager/ $(lsb_release -cs)-updates -y --allow-downgrades

Appendix D. 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.

D.1. 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 DONOT 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 ORmIMPLIED, 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 ofliability 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.
```

D.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.Concept definition.
```

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
www.nvidia.com