DGX Software with Red Hat Enterprise Linux 8

Installation Guide

RN-09301-002 _v04 | January 2023

Table of Contents

Chapter 1. Introduction

The NVIDIA® DGX™ systems are shipped with DGX™ OS which incorporates the NVIDIA DGX software stack built upon the Ubuntu Linux distribution. Instead of running the Ubuntu distribution, you can run Red Hat Enterprise Linux on the DGX system and still take advantage of the advanced DGX features.

This document explains how to install and configure the NVIDIA DGX software stack on DGX systems installed with or Rocky Linux 8.

Note: While it may be possible to use other derived Linux distributions besides Red Hat Enterprise Linux, not all have been tested and qualified by NVIDIA. Refer to the DGX Software for Red Hat Enterprise Linux 8 Release Notes for the list of tested and qualified software and Linux distributions.

1.1. Related Documentation

- NVIDIA DGX Software for Red Hat Enterprise Linux 8 - Release Notes
- NVIDIA DGX-1 User Guide
- NVIDIA DGX-2 User Guide
- NVIDIA DGX A100 User Guide
- NVIDIA DGX Station User Guide

1.2. Prerequisites

The following are required (or recommended where indicated).

1.2.1. Red Hat Subscription

You need a Red Hat subscription if you plan to install and use Red Hat Enterprise Linux 8 on the DGX. A subscription also lets you obtain update packages and additional packages for

Red Hat Enterprise Linux. You can either purchase a subscription or obtain a free evaluation subscription from the Red Hat Software & Download Center.

Note: Of the available Red Hat Enterprise Linux platforms, **only** Red Hat Enterprise Linux **Server** is supported on DGX systems (DGX servers and DGX Station workstation). Other Red Hat Enterprise Linux platforms are **not** supported on any DGX system.

1.2.2. Access to Repositories

The repositories can be accessed from the internet.

If your installation does not allow connection to the internet, see the section Installing Software on Air-Gapped NVIDIA DGX Systems for information about updating software on "air-gapped" systems.

If you are using a proxy server, then follow the instructions in the section Configuring a System Proxy to make sure the system can access the necessary URIs.

1.2.2.1. NVIDIA Repositories

After installing Red Hat Enterprise Linux on the DGX system, you must enable the NVIDIA DGX software repository (https://repo.download.nvidia.com). The repository includes the NVIDIA drivers and software for supporting DGX systems.

See the section Enabling the Repositories for instructions on how to enable the repository.

1.2.2.2. Red Hat Repositories

Installation of the DGX Software over Red Hat Enterprise Linux 8 requires access to several additional repositories.

- Red Hat Enterprise BaseOS Repository: `rhel-8-for-x86_64-baseos-rpms`
- Red Hat Enterprise AppStream Repository: `rhel-8-for-x86_64-appstream-rpms`
- Red Hat Enterprise CodeReady Linux Builder Repository: `codeready-builder-for-rhel-8-x86_64-debug-rpms`

1.2.3. Network File System

On DGX servers, the data drives are meant to be used as a cache. DGX Station users can follow the same usage, or can alternatively opt to use these drives for storage. When using the data drives as cache, a network file system (NFS) is recommended to take advantage of the cache file system provided by the DGX software stack.

1.2.4. BMC Password

The DGX BMC comes with default login credentials as specified in Appendix B: Changing the BMC Login.

Important:

NVIDIA recommends disabling the default username and creating a unique BMC username and strong password as soon as possible. Refer to Appendix C: Changing the BMC Login for instructions.

Chapter 2. Installing Red Hat Enterprise Linux 8

There are several methods for installing Red Hat Enterprise Linux as described in the Red Hat Enterprise Linux Installation Guide (https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/installation_guide/index).

See the DGX Software for Red Hat Enterprise Linux Release Notes for the version of Red Hat Enterprise Linux 8 that is qualified and tested for use with the DGX software.

For convenience, this section describes how to install Red Hat Enterprise Linux or Rocky Linux using the Quick Install method, and shows when to reclaim disk space in the process. It describes a minimal installation. If you have a preferred method for installing Red Hat Enterprise Linux, then you can skip this section but be sure to reclaim disk space occupied by the existing Ubuntu installation.

The interactive method described here installs Red Hat Enterprise Linux on DGX using a connected monitor and keyboard and USB stick with the ISO image, or remotely through the remote console of the BMC.

2.1. Obtaining Red Hat Enterprise Linux 8

Obtain the Red Hat Enterprise Linux 8 ISO image and store on your local disk or create a boot USB drive formatted for UEFI. See Downloading Red Hat Enterprise Linux (https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/installation_guide/chap-download-red-hat-enterprise-linux) for instructions.

For Rocky Linux 8, visit Downloading Rocky Linux.

Important: Refer to the release notes for any critical information regarding supported releases or dependencies.

2.2. Booting Red Hat Enterprise Linux 8 ISO Locally

1. Plug the USB flash drive containing the Red Hat Enterprise Linux 8 ISO image into the DGX.
2. Connect a monitor and keyboard directly to the DGX.
3. Boot the system and 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. Follow the instructions at Installing Red Hat Enterprise Linux

2.3. Booting the Red Hat Enterprise Linux 8 ISO Remotely

Skip this chapter if you are using a monitor and keyboard for installing locally. It is also not possible to boot an ISO remotely if you are doing the install on a DGX Station.

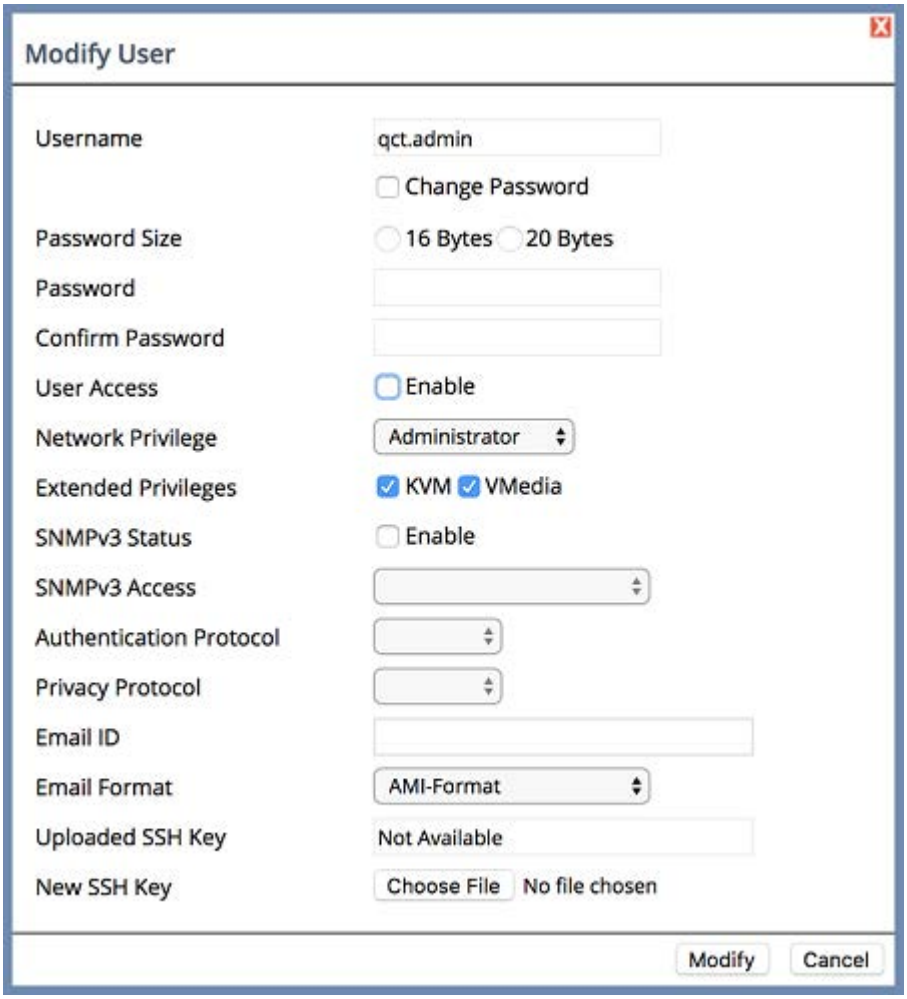
2.3.1. Booting the ISO Image on the DGX-1 Remotely

Skip this chapter if you are using a monitor and keyboard for installing locally.

- For instructions applicable to the NVIDIA DGX-2 or DGX A100, see Booting the ISO Image on the DGX-2 or DGX A100 Remotely

1. Connect to the BMC and change user privileges.
   a). Open a Java-enabled web browser within your LAN and go to `http://<BMC-ip-address>/`, then log in.

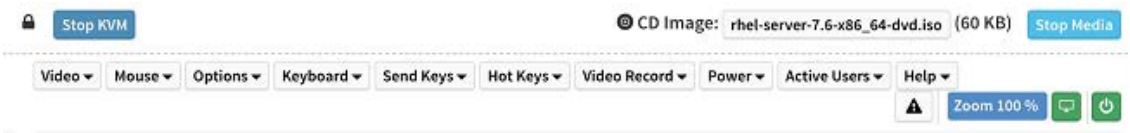
   Use Firefox or Internet Explorer. Google Chrome is not officially supported by the BMC.
   b). From the top menu, click **Configuration** and then select **Users**.
   c). Select the user name that you created for the BMC, then click **Modify User**.
   d). In the **Modify User** dialog, select the **VMedia** checkbox to add it to the extended privileges for the user, then click **Modify**.



2. Set up the ISO image as virtual media and reboot the system.
   a). From the top menu, click **Remote Control** and select **Remote KVM**.

   b). Click **Launch KVM** to open the remote console.
   c). From the top menu bar in the KVM window, click **Browse File** and select the ISO image, then click **Start Media**.

   The CD image should now be connected.

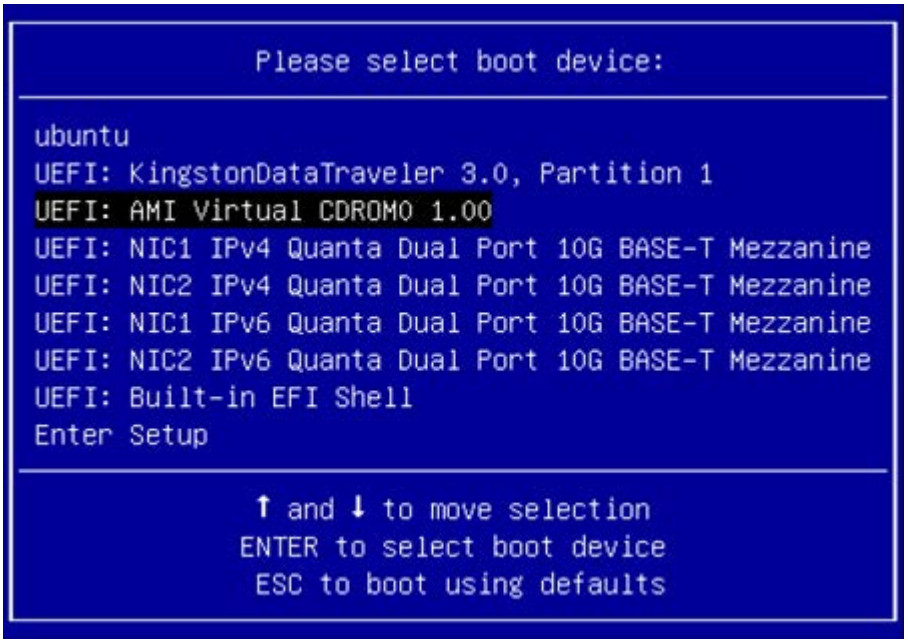


d). From the top menu bar in the KVM window, click **Power** and then select **Reset Server**.

3. Boot the CD ROM image.

   Typically, the default boot order does not boot the CDROM image. This can be changed in the BIOS or as a one-time option in the boot menu. To bring up the boot menu, press F11 at the beginning of the boot process. Pressing **F11** will display Show Boot Options at the top of the virtual display before entering the boot menu. Use the 'soft' keyboard (Menu → Keyboard Layout → SoftKeyboard → <Language>) to bring up a virtual keyboard if pressing the physical key has no effect.

a). In the boot menu, select **UEFI: AMI Virtual CDROM 1.00** as the boot device and then press **ENTER**.

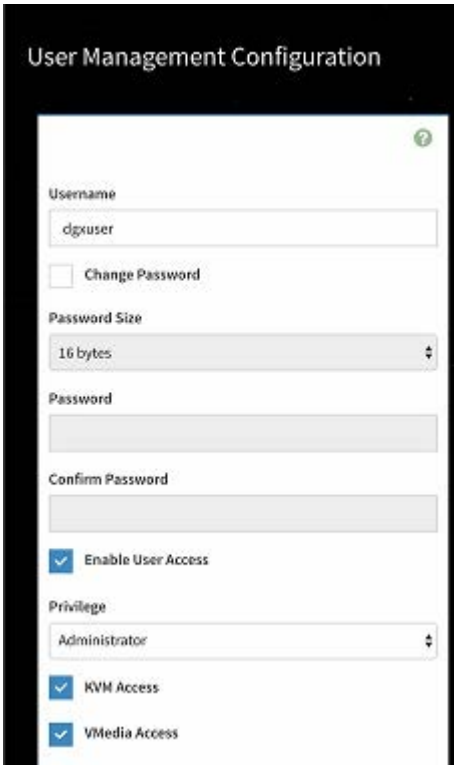


b). Follow the instructions at Installing Red Hat Enterprise Linux.

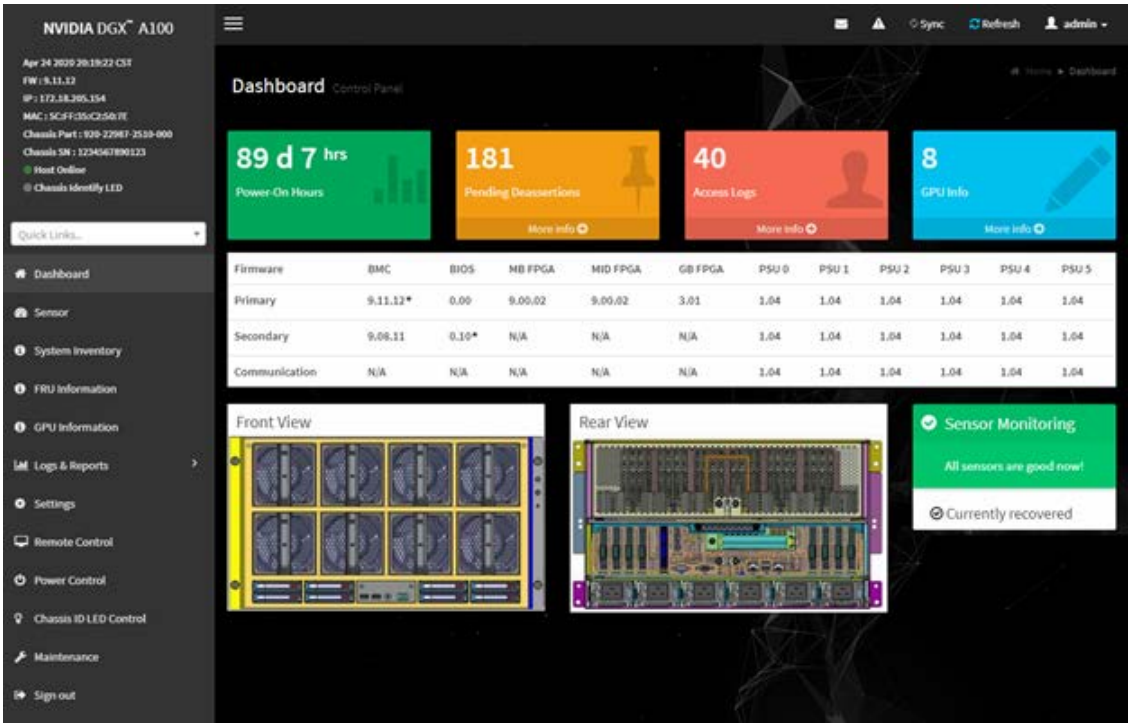
2.3.2. Booting the ISO Image on the DGX-2 or DGX A100 Remotely

Skip this chapter if you are using a monitor and keyboard for installing locally.

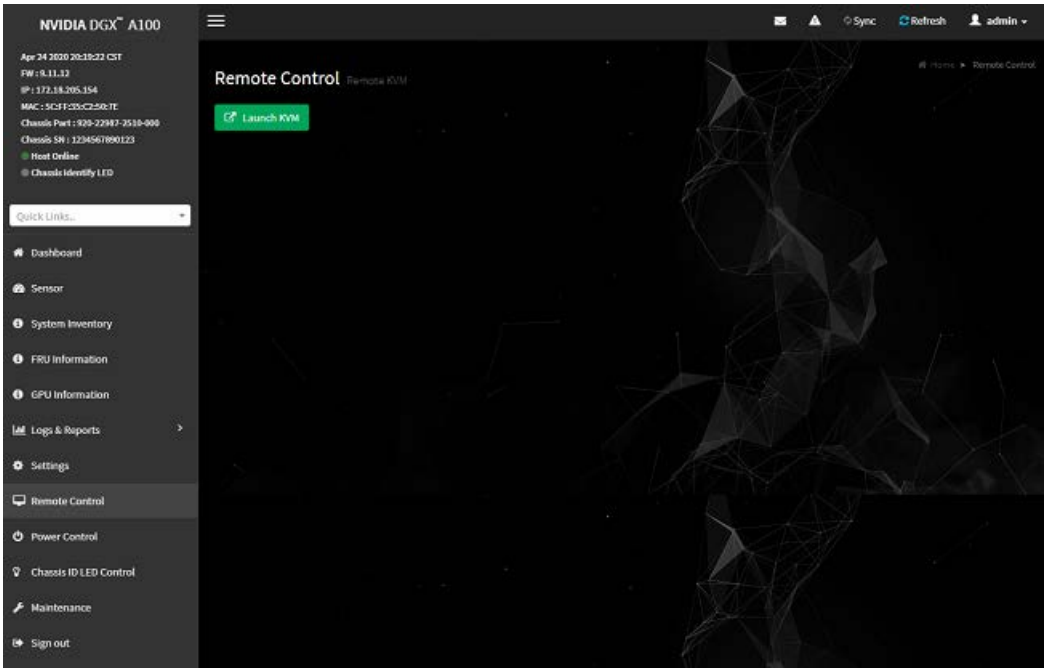
1. Connect to the BMC and ensure the required user privileges are set.
   a). Open a browser within your LAN and go to `https://<BMC-ip-address>/`, then log in.
   b). From the left-side menu, click **Settings** and then select **User Management**.
   c). Click the card with the user name that you created for the BMC.
   d). In the **User Management Configuration** dialog, make sure the **VMedia Access** checkbox is selected, then click **Save**.



2. Set up the ISO image as virtual media.
   a). From the left-side menu, click **Remote Control**.

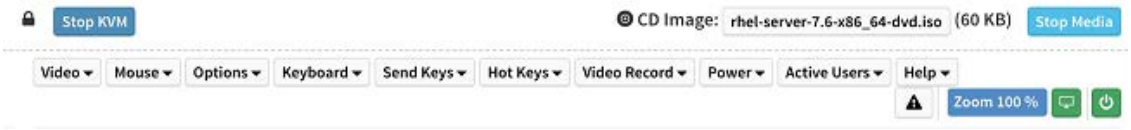


   b). Select **Launch KVM**.



c). From the top menu bar in the KVM window, click **Browse File** and select the ISO image, then click **Start Media**.

The CD image should now be connected.



d). From the top menu bar in the KVM window, click **Power** and then select **Reset Server**.

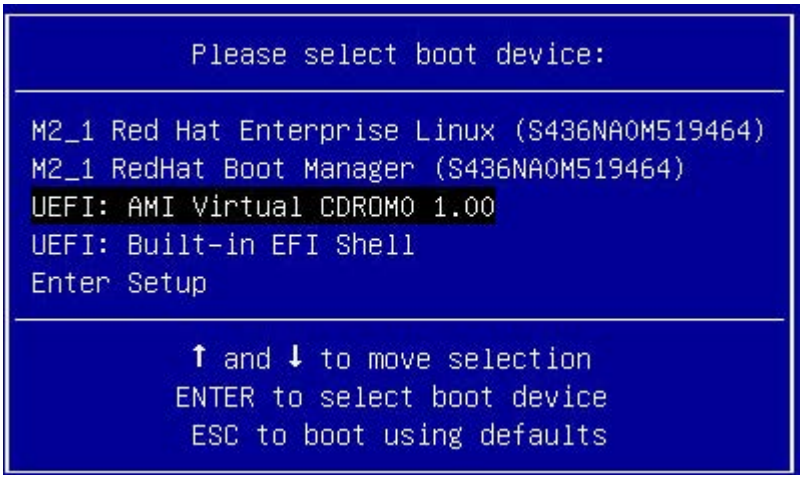
3. Boot from the virtual media.

   Typically, the default boot order does not boot the CDROM image. You can change this in the BIOS or as a one-time option in the boot menu.

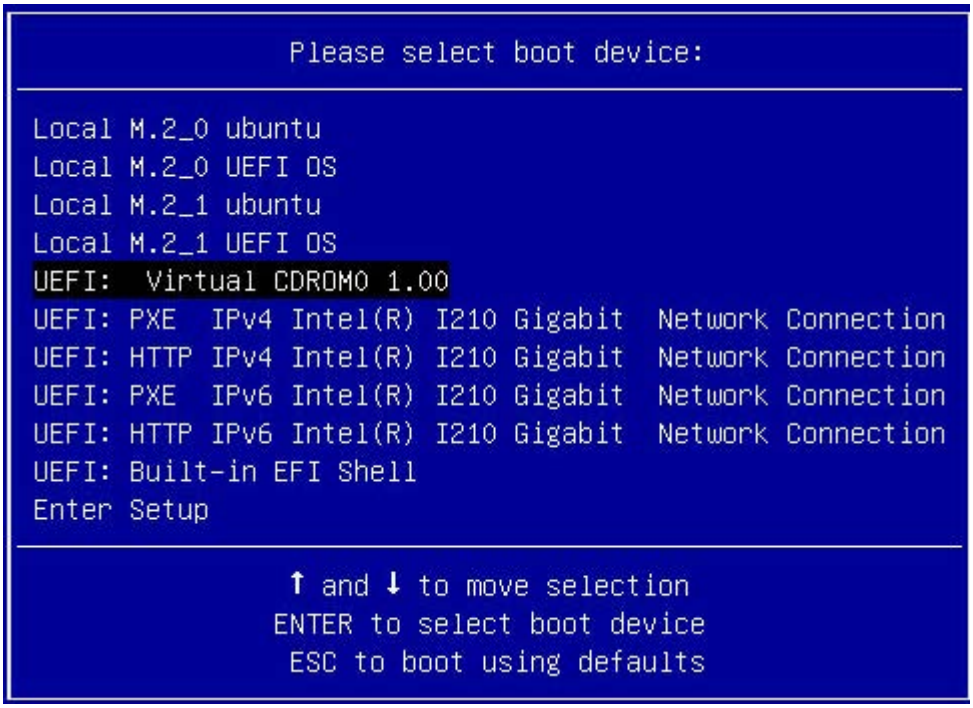
   a). To bring up the boot menu, press **F11** at the beginning of the boot process.

   Pressing **F11** will display **Entering Boot Menu** in the virtual display before entering the boot menu.

b). In the boot menu, select **UEFI: AMI Virtual CDROM0 1.00** as the boot device and then press **ENTER**.



If you are booting from the DGX-2, the boot device is labeled **UEFI: Virtual CDROM0 1.00**.



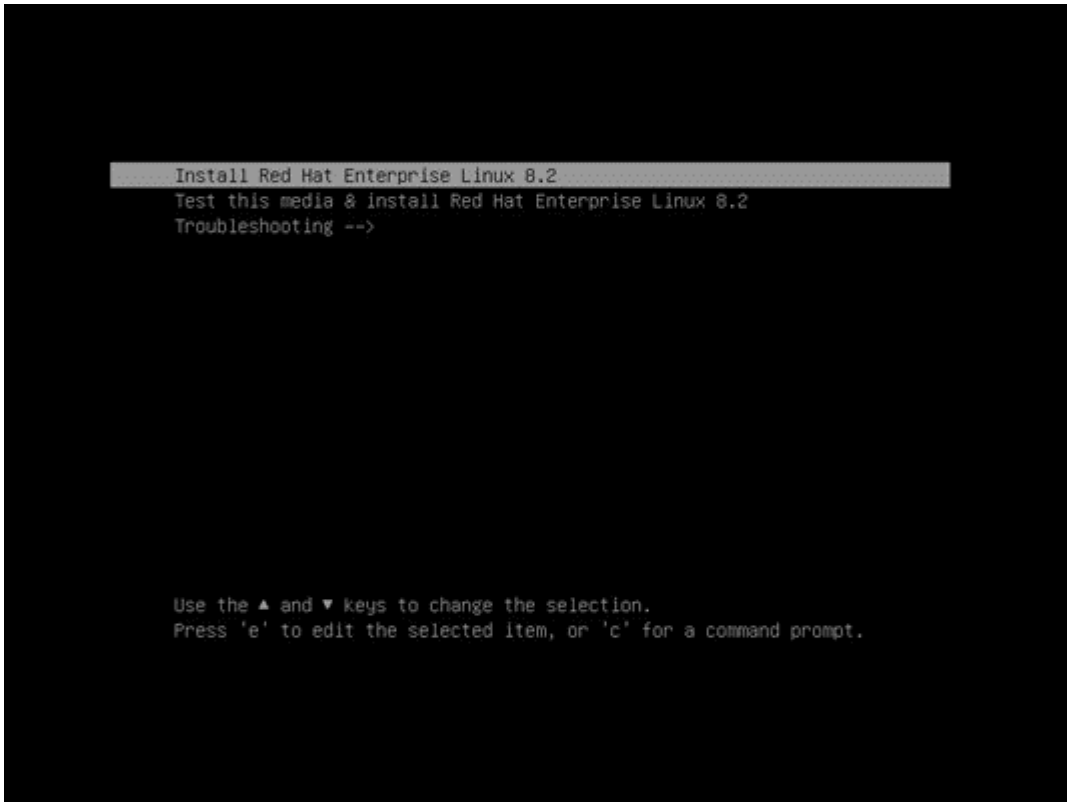
c). Follow the instructions at Installing Red Hat Enterprise Linux

2.4. Installing Red Hat Enterprise Linux

This section assumes you have booted the Red Hat Enterprise Linux ISO image. For DGX Station, this can only be performed locally; for all other platforms, this can be performed locally or remotely.

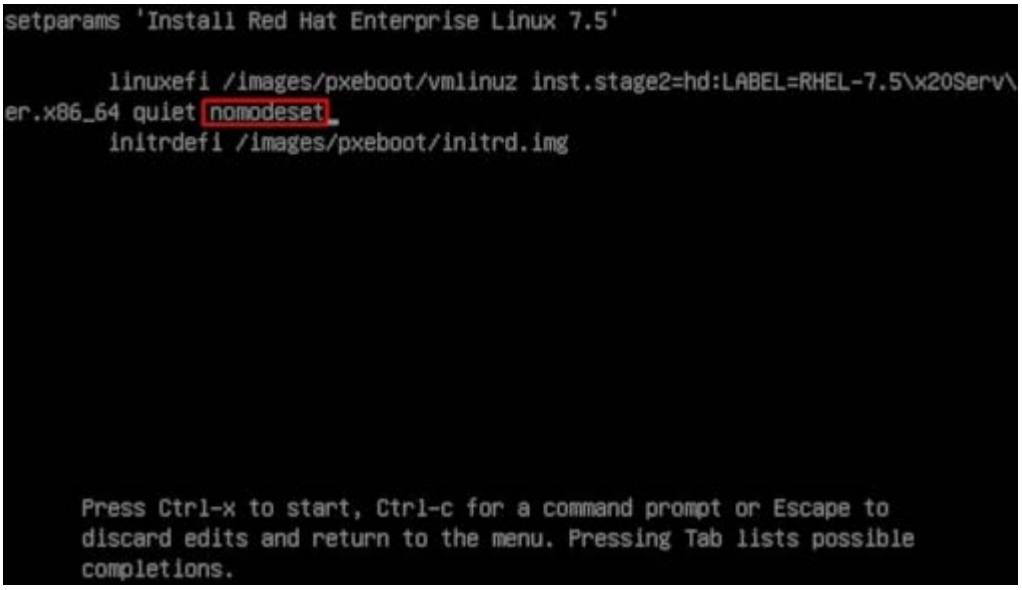
If you wish to automate the installation with a kickstart file, see the section Installing with Kickstart

1. After booting the ISO image, the GRUB menu for the installer appears.



2. Select `Install Red Hat Enterprise Linux` and press 'e' to edit the selected option.
   a). Add the "nomodeset" parameter as shown below.

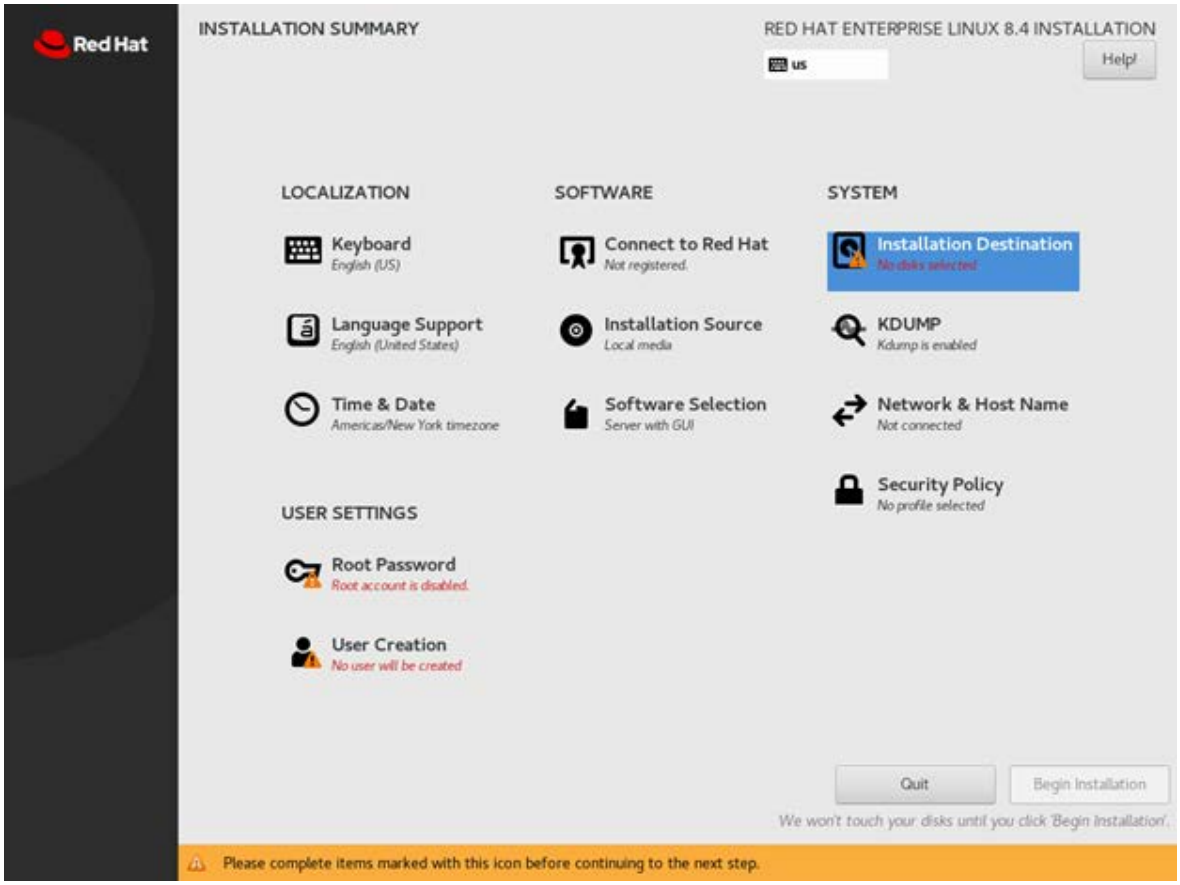
   "nomodeset" prevents the in-box Nouveau driver from loading as it may not properly support the GPUs used in the DGX systems, resulting in display issues.



   b). Press **Ctrl-x** to save your changes and start the installation.

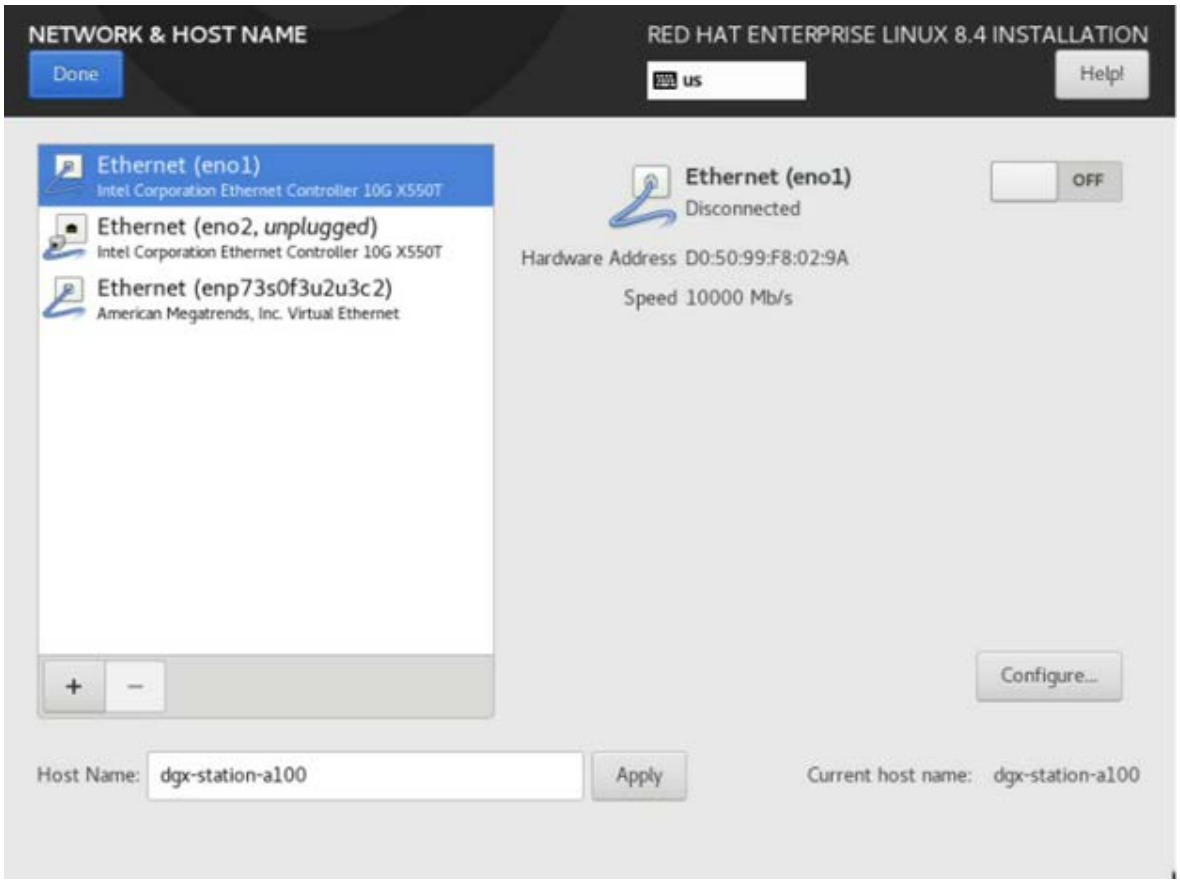
3. Refer to the Red Hat Enterprise Linux Quick Installation Guide (https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/installation_guide/chap-simple-install) for guidance on using the installer.
4. Configure the language, region, date, time, keyboard, and other configuration options you may need from the **Installation Summary** screen.
5. Set the **Software Selection** to the correct value.
   - For DGX Station and DGX Station A100: Set to **Server with GUI**.
   - For DGX Servers (DGX 1-1, DGX-2, DGX A100): Set to **Minimal Install**.

Note: Setting the correct Software Selection is critical for proper operation.

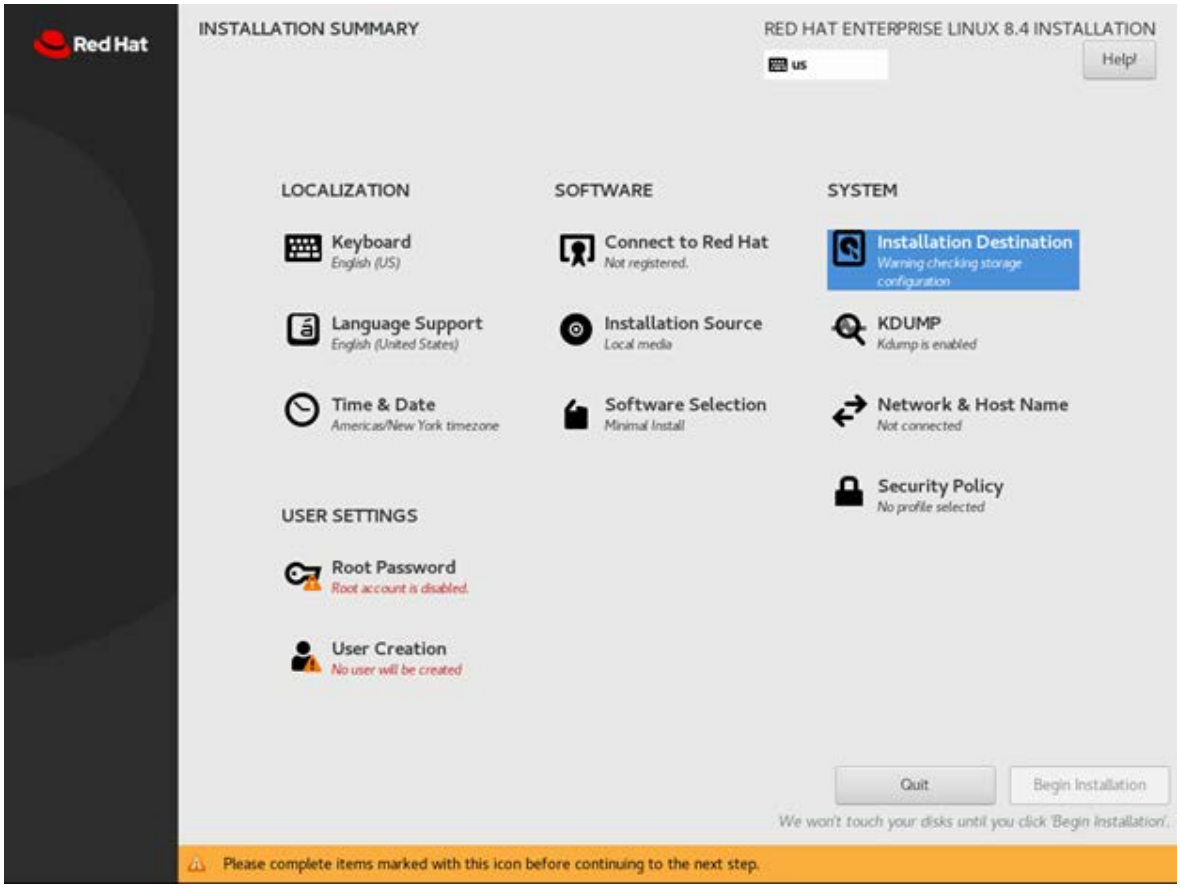


6. Partition your disks based on the platform and whether or not you would like to use encryption:
   - Disk Partitioning for DGX-1, DGX Station, and DGX Station A100
   - Disk Partitioning with Encryption for DGX-1, DGX Station, and DGX Station A100
   - Disk Partitioning for DGX-2 and DGX A100
   - Disk Partitioning with Encryption for DGX-2 and DGX A100
7. Configure Ethernet.

From the **Network & Host Name** section, select and enable the Ethernet device. This defaults to DHCP and can be changed for static IP configurations under **Configure**.



8. From the **INSTALLATION SUMMARY** screen, set your password (at **User Settings > Root Password**) and create a new user (at **User Settings > User Creation**), then click **Begin installation** to start the installation.



When the system reboots, be sure and eject any still-mounted installation discs.

9. Complete the initial Red Hat Enterprise Linux setup.
   - If you installed using the *Server with GUI* base environment, the Initial Setup starts automatically where you can accept the license agreement and register the system. See the Red Hat instructions for details.
   - If you are using CentOS you will need to accept the license agreement.
   - If you installed with any other base environment, log on to the system as root user and then register the system.

     ```
     subscription-manager register --auto-attach --username=user_name --
     password=password
     ```

See How to register and subscribe a system to the Red Hat Customer Portal using Red Hat Subscription-Manager for further information.

2.4.1. Installing with Kickstart

Install using Kickstart

Kickstart provides a method of automating the installation process by providing a file which contains the answers to the questions that would be asked during installation. NVIDIA provides kickstart templates for all supported platforms in the following locations:

- https://repo.download.nvidia.com/baseos/el/el-files/8/el8-ks/dgxstation-ks.cfg
- https://repo.download.nvidia.com/baseos/el/el-files/8/el8-ks/dgx2-ks.cfg
- https://repo.download.nvidia.com/baseos/el/el-files/8/el8-ks/dgx1-ks.cfg
- https://repo.download.nvidia.com/baseos/el/el-files/8/el8-ks/dgx-a100-ks.cfg
- https://repo.download.nvidia.com/baseos/el/el-files/8/el8-ks/dgxstation-a100-ks.cfg

In these files are tags that you must replace with your site specific information: language, keyboard, timezone, hostname, etc. Each tag is of the form `<CHANGE_YOUR_xxxx>`. These tags must be replaced with your specific information.

Once you have a kickstart file customized for your installation, place that file in a location that can be accessed by NFS, FTP, HTTP, or HTTPS. After booting from the installation medium, when the grub menu appears, press 'e' to edit the grub entry as you normally would to add the "`nomodeset`" option. In addition to "`nomodeset`", add an option in the format "`inst.ks=<URL>`". For example: `inst.ks=https://192.168.1.2/kickstart/dgx-a100-cfg.ks`

Lastly, after using these kickstart files -- remember to reboot the system one final time to apply all settings.

For more information on using kickstart files with Red Hat 8, refer to the Starting Kickstart Installations chapter in Red Hat's documentation.

2.4.2. Disk Partitioning for DGX-1, DGX Station, and DGX Station A100

Note: The screenshots in the following section are taken from a DGX-1. The screens for the DGX Station and DGX Station A100 installations can present slightly different information for such things as disk size, disk space available, interface names, etc.

1. At the **Installation Destination** screen, select the first drive (sda) to install on.

   Under **Storage Configuration**, click the **Custom** radio button and then click **Done**.

   This brings up the **Manual Partitioning** window.

2. Expand the drop-down menu for the device you've selected and delete all existing partitions until there are none remaining.
3. Click the **+** button on the bottom left to create a new partition.

   Create the `/boot/efi` partition with a size of 512 MiB.

4. Select **Standard Partition** for the Device Type.

Under **Devices**, ensure only the drive you chose in step 1 is selected. Click Update Settings to confirm your changes.

5. Click the **+** button on the bottom left again to create another new partition.

   Create the / partition, this time leaving the **Desired Capacity** field empty. This lets the installer know to use the remaining capacity of the disks.

6. For the / partition, again select **Standard Partition** for the Device Type.

   Under **Devices**, ensure that only the drive you chose in step 1 is selected. For the File System, select **XFS**. Click **Update Settings** to confirm your changes.

7. Click **Done**.

   This causes a yellow warning bar to appear on the bottom because no swap partition has been created.

8. Click **Done** again and then click **Accept Changes** to write all of our customizations to disk.

9. Return to step 5 in the Installing Red Hat Enterprise Linux section.

2.4.3. Disk Partitioning with Encryption for DGX-1, DGX Station, and DGX Station A100

Note: The screenshots in the following section are taken from a DGX-1. The screens for the DGX Station and DGX Station A100 installations can present slightly different information for such things as disk size, disk space available, interface names, etc.

1. At the **Installation Destination** screen, select the first drive (`sda`) to install on. Under **Storage Configuration**, click the **Custom** radio button and then click **Done**.

   This brings up the **Manual Partitioning** window.

2. Expand the drop-down menu for the device you've selected and delete all existing partitions until there are none remaining.

3. Click the **+** button on the bottom left to create a new partition.

   Create the `/boot/efi` partition with a size of 512 MiB.

4. Select **Standard Partition** for the Device Type.

   Under **Devices**, ensure only the drive you chose in step 1 is selected.

Click **Update Settings** to confirm your changes.

5. Click the **+** button on the bottom left again to create another new partition.

   Create the /boot partition with a size of 2 GiB.

6. For the /boot partition, again select **Standard Partition** for the Device Type.

   Under **Devices**, ensure that only the drive you chose in step 1 is selected. For the File System, select **XFS**.

   Click **Update Settings** to confirm your changes.

7. Click the **+** button on the bottom left again to create another new partition.

   Create the / partition, this time leaving the **Desired Capacity** field empty. This lets the installer know to use the remaining capacity of the disks.

8. For the / partition, again select **Standard Partition** for the Device Type.

   Be sure to check the **Encrypt** checkbox, and choose **luks2** for the **LUKS Version**.

   For the **File System**, select **XFS**. Again confirm that this partition uses both of the devices you selected in step 2.

   Click **Update Settings** to confirm your changes.

   A prompt will now appear, asking for a LUKS password.

9. Be sure to choose a strong and secure password.

   After entering your password twice, click **Save Passphrase**.

10. Click **Done**.

    This causes a yellow warning bar to appear on the bottom because no swap partition has been created.

11. Click **Done** again and then click **Accept Changes** to write all of our customizations to disk.

12. Return to step 5 in the Installing Red Hat Enterprise Linux section.

2.4.4. Disk Partitioning for DGX-2 and DGX A100

Note: The screenshots in the following section are taken from a DGX A100. The screens for the DGX-2 installation can present slightly different information for such things as disk size, disk space available, interface names, etc.

1. At the **Installation Destination** screen, select both M.2 disks (that is, the 894,25GB drives for DGX-2, or the 1.8 TB drives for DGX A100).

   Under **Storage Configuration**, click the **Custom** radio button and then click **Done**.

   This brings up the **Manual Partitioning** window.

2. Expand the drop-down menu for the device you've selected and delete all existing partitions until there are none remaining.
3. Click the **+** button on the bottom left to create a new partition.

   Create the `/boot/efi` partition with a size of 512 MiB.

4. Change the **Device Type** to **RAID**, change the **RAID Level** to **RAID1**, then click **Update Settings** and confirm that this partition covers both of the devices you selected in step 2.

5. Click the **+** button on the bottom left again to create another new partition.

   Create the **/** partition, this time leaving the **Desired Capacity** field empty. This lets the installer know to use the remaining capacity of the disks.

6. For the / partition, again select **RAID** for the **Device Type**, **RAID1**, and the **RAID Level**.

   For the **File System**, select **XFS**.

   Click **Update Settings** to confirm this partition uses both of the devices you selected in step 2.

7. Click **Done**.

   This causes a yellow warning bar to appear on the bottom because no swap partition has been created.

8. Click **Done** again and then click **Accept Changes** to write all of our customizations to disk.

9. Return to the instructions in the Installing Red Hat Enterprise Linux section.

   This should be at the step to configure Ethernet.

2.4.5. Disk Partitioning with Encryption for DGX-2 and DGX A100

Note: The screenshots in the following section are taken from a DGX A100. The screens for the DGX-2 installation can present slightly different information for such things as disk size, disk space available, interface names, etc.

1. At the **Installation Destination** screen, select both M.2 disks (that is, the 894,25GB drives for DGX-2, or the 1.8 TB drives for DGX A100).

   Under **Storage Configuration**, click the **Custom** radio button and then click **Done**.

   This brings up the **Manual Partitioning** window.
2. Expand the drop-down menu for the device you've selected and delete all existing partitions until there are none remaining.

3. Click the **+** button on the bottom left to create a new partition.

   Create the `/boot/efi` partition with a size of 512 MiB.

4. Change the **Device Type** to **RAID**, change the **RAID Level** to **RAID1**, and confirm that this partition covers both of the devices you selected in step 2 by clicking **Modify** in the **Device(s)** section and making sure both disks are still selected.

   Note that the partition information on the left may not reflect both devices yet.

   Click **Update Settings** to confirm your changes, and verify that the `/boot/efi` partition shown lists the partition name and not the individual device.
5. Click the **+** button on the bottom left again to create another new partition.

   Create the /boot partition with a size of **2 GiB**.

6. For the /boot partition, again select **RAID** for the **Device Type**, **RAID1**, and the **RAID Level**.

   For the **File System**, select **XFS**.

   Again confirm that this partition uses both of the devices you selected in step 2 by clicking **Modify** in the **Device(s)** section and making sure both disks are still selected.

   Click **Update Settings** to confirm your changes.
7. Click the **+** button on the bottom left again to create another new partition.

   Create the / partition, this time leaving the **Desired Capacity** field empty. This lets the installer know to use the remaining capacity of the disks.

8. For the / partition, again select RAID for the Device Type, RAID1 and the RAID Level.

   Be sure to check the **Encrypt** checkbox, and choose **luks2** for the **LUKS Version**.

   For the **File System**, select **XFS**. Again confirm that this partition uses both of the devices you selected in step 2.

   Click **Update Settings** to confirm your changes, and **Done** to begin exiting the partition menu.

   A prompt will now appear asking for a LUKS password.
9. Be sure to choose a strong and secure password.

   After entering your password twice, click **Save Passphrase**.
10. Click **Done**.

    This causes a yellow warning bar to appear on the bottom because no swap partition has been created.

11. Click **Done** again and then click **Accept Changes** to write all of our customizations to disk.

12. Return to step 5 in the Installing Red Hat Enterprise Linux section.

Chapter 3. Installing the DGX Software

This section requires that you have already installed Red Hat Enterprise Linux 8 or derived operating system on the DGX system. You can skip this section of you already installed the DGX software stack during a kickstart install.

3.1. Configuring a System Proxy

If your network requires use of a proxy, then

- Edit the file `/etc/dnf/dnf.conf` and make sure the following lines are present in the `[main]` section, using the parameters that apply to your network:

```
proxy=http://<Proxy-Server-IP-Address>:<Proxy-Port>
proxy_username=<Proxy-User-Name>
proxy_password=<Proxy-Password>
```

3.2. Enabling the DGX Software Repository

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

Install the NVIDIA DGX Package for Red Hat Enterprise Linux.

```
$ sudo dnf install -y https://repo.download.nvidia.com/baseos/el/el-files/8/nvidia-repo-setup-21.06-1.el8.x86_64.rpm
```

3.3. Installing Required Components

1. On Red Hat Enterprise Linux, run the following commands to enable additional repositories required by the DGX Software.

```
sudo subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms
sudo subscription-manager repos --enable=rhel-8-for-x86_64-baseos-rpms
sudo subscription-manager repos --enable=codeready-builder-for-rhel-8-x86_64-rpms
```

2. Upgrade to the latest software.

Important: Before performing the upgrade, consult the release notes for additional instructions depending on the specific EL8 release.

```
sudo dnf update -y --nobest
```

3. Install DGX tools and configuration files.

   - For DGX-1, install **DGX-1 Configurations**.

     ```
     sudo dnf group install -y 'DGX-1 Configurations'
     ```

   - For the DGX-2, install **DGX-2 Configurations**.

     ```
     sudo dnf group install -y 'DGX-2 Configurations'
     ```

   - For the DGX A100, install **DGX A100 Configurations**.

     ```
     sudo dnf group install -y 'DGX A100 Configurations'
     ```

   - For the DGX Station, install **DGX Station Configurations**.

     ```
     sudo dnf group install -y 'DGX Station Configurations'
     ```

   - For the DGX Station A100, install **DGX Station A100 Configurations**.

     ```
     sudo dnf group install -y 'DGX Station A100 Configurations'
     ```

   The configuration changes take effect only after rebooting the system, which will be performed after installing the CUDA driver.

4. Configure the `/raid` partition.

   All DGX systems support RAID 0 or RAID 5 arrays.

   The following commands create a RAID array, mount it to `/raid` and create an appropriate entry in `/etc/fstab`.

   - To create a RAID 0 array:

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

   - To create a RAID 5 array:

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

Note: The RAID array must be configured before installing `nvidia-conf-cachefilesd`, which places the proper SELinux label on the `/raid` directory. If you ever need to recreate the RAID array — which will wipe out any labeling on `/raid` — after `nvidia-conf-cachefilesd` has already been installed, be sure to restore the label manually before restarting `cachefilesd`.

```
sudo restorecon /raid
sudo systemctl restart cachefilesd
```

Optional: If you wish to use your RAID array for caching, install `nvidia-conf-cachefilesd`. This will update the `cachefilesd` configuration to use the `/raid` partition.

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

5. Install the NVIDIA CUDA driver.

Important: If you are installing the CUDA driver from a local repository, follow the instructions at Installing the NVIDIA CUDA Driver from the Local Repository instead of this step.

a). For non-NVSwitch systems such as DGX-1, DGX Station, and DGX Station A100, install either the R450 or R470 driver using the default and src profiles:

Installing the R450 drivers

```
sudo dnf module install -y nvidia-driver:450/{default.src}
sudo dnf install -y nv-persistence-mode libnvidia-nscq-450
```

Installing the R470 drivers

```
sudo dnf module install -y nvidia-driver:470/{default,src}
sudo dnf install -y nv-persistence-mode libnvidia-nscq-470
```

b). For NVSwitch systems such as DGX-2 and DGX A100, install either the R450 or R470 driver using the fabric manager (fm) and src profiles:

Installing the R450 drivers

```
sudo dnf module install -y nvidia-driver:450/{fm,src}
sudo dnf install -y nv-persistence-mode nvidia-fm-enable
```

Installing the R470 drivers

```
sudo dnf module install -y nvidia-driver:470/{fm,src}
sudo dnf install -y nv-persistence-mode nvidia-fm-enable
```

6. (DGX Station A100 Only) Install additional packages required for DGX Station A100.

These packages must be installed after installation of the `nvidia-driver` module.

```
sudo dnf install -y nvidia-conf-xconfig nv-docker-gpus
```

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

a). Issue the reboot.

```
sudo reboot
```

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

```
nvidia-smi
```

The output should show all available GPUs.

Example:

```
+-----------------------------------------------------------------------------
+
| 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  Tesla V100-SXM2...  On   | 00000000:06:00.0 Off |                    0
 |
| N/A   35C    P0    42W / 300W |      0MiB / 16160MiB |      0%      Default
 |
|                               |                      |                  N/A
 |
+-------------------------------+----------------------+----------------------
+
|   1  Tesla V100-SXM2...  On   | 00000000:07:00.0 Off |                    0
 |
| N/A   35C    P0    44W / 300W |      0MiB / 16160MiB |      0%      Default
 |
|                               |                      |                  N/A
 |
+-------------------------------+----------------------+----------------------
+
...
+-------------------------------+----------------------+----------------------
+
|   7  Tesla V100-SXM2...  On   | 00000000:8A:00.0 Off |                    0
 |
| N/A   35C    P0    43W / 300W |      0MiB / 16160MiB |      0%      Default
 |
|                               |                      |                  N/A
 |
+-------------------------------+----------------------+----------------------
+


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

8. Install the NVIDIA container device plugin.
   a). Install `docker-ce`.

   As this may conflict with existing packages on the system, specify the `--allowerasing` option:

   ```
   sudo dnf install -y docker-ce --allowerasing
   ```

   b). Install the **NVIDIA Container Runtime** group

   ```
   sudo dnf group install -y 'NVIDIA Container Runtime'
   ```

   c). Restart the docker daemon.

   ```
   sudo systemctl restart docker
   ```

   d). Run the following command to verify the installation.

   ```
   sudo docker run --gpus=all --rm nvcr.io/nvidia/cuda:11.0-base nvidia-smi
   ```

   See the section Running Containers for more information about this command.

   The output should show all available GPUs

```
+-----------------------------------------------------------------------------
+
| NVIDIA-SMI 470.57.02    Driver Version: 470.57.02    CUDA Version: 11.4
 |
|-------------------------------+----------------------+----------------------
+
| 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  Tesla V100-SXM2...  On   | 00000000:06:00.0 Off |                    0
 |
| N/A   35C    P0    42W / 300W |      0MiB / 16160MiB |      0%      Default
 |
|                               |                      |                  N/A
 |
+-------------------------------+----------------------+----------------------
+
|   1  Tesla V100-SXM2...  On   | 00000000:07:00.0 Off |                    0
 |
| N/A   35C    P0    44W / 300W |      0MiB / 16160MiB |      0%      Default
 |
|                               |                      |                  N/A
 |
+-------------------------------+----------------------+----------------------
+
...
+-------------------------------+----------------------+----------------------
+
|   7  Tesla V100-SXM2...  On   | 00000000:8A:00.0 Off |                    0
 |
| N/A   35C    P0    43W / 300W |      0MiB / 16160MiB |      0%      Default
 |
|                               |                      |                  N/A
 |
+-------------------------------+----------------------+----------------------
+


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

Installation of another required software component is explained in Using the NVIDIA Mellanox InfiniBand Drivers.

3.4. Installing Optional Components

The DGX is fully functional after installing the components as described in Installing Required Components. If you intend to launch NGC containers (which incorporate the CUDA toolkit, NCCL, cuDNN, and TensorRT) on the DGX system, which is the expected use case, then you can skip this section.
If you intend to use your DGX system as a development system for running deep learning applications on bare metal, then install the optional components as described in this section.

- To install the CUDA Toolkit 11.2, issue the following.
  ```
  $ sudo dnf install -y cuda-toolkit-11-2 cuda-compat-11-2 nvidia-cuda-compat-setup
  ```
- To administer self-encrypting drives, install the `nv-disk-encrypt` package, issue the following.
  ```
  $ sudo dnf install -y nv-disk-encrypt
  $ sudo reboot
  ```
  Refer to the "Managing Self-Encrypting Drives" section in the DGX A100 User Guide for usage information.
- To install the NVIDIA Collectives Communication Library (NCCL) Runtime, refer to the NCCL:Getting Started documentation.
- To install the CUDA Deep Neural Networks (cuDNN) Library Runtime, refer to the NVIDIA cuDNN page.
- To install NVIDIA TensorRT, refer to the NVIDIA TensorRT page.
- To install NVIDIA GPUDirect Storage (GDS), issue the following to install the GDS packages.
  ```
  $ sudo dnf install nvidia-gds
  ```
  Be sure to enable GDS within the MLNX_OFED driver if you install the driver. Refer to Using the NVIDIA Mellanox InfiniBand Drivers.

3.5. Installing the Optional NVIDIA Desktop Theme

The DGX Software Repository also provides optional theme packages and desktop wallpapers to give the user-interface an NVIDIA look-and-feel. These packages would have been installed as part of the **DGX Station Configurations** group, but users can also manually install this:

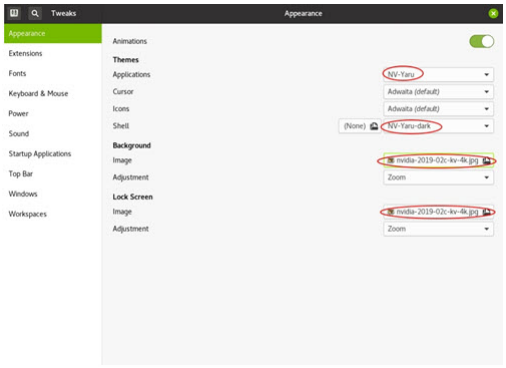
1. To apply the theme and background images, first open gnome-tweaks.
2. Under **Applications**, select one of the **NV-Yaru** themes.
   This comes in default, light, and dark variations.
3. Under **Shell**, select the **NV-Yaru-dark** theme.

   If this field is grayed out, you may need to reboot the system or restart GDM in order to enable the user-themes extension.

To restart GDM, issue the following.

```
sudo systemctl restart gdm
```

4. Select one of the NVIDIA wallpapers for the background image and lock screen.



Chapter 4. Upgrading the OS and DGX Software

NVIDIA and Red Hat 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 `sudo dnf update --nobest` command.

For a list of the known Common Vulnerabilities and Exposures (CVEs), including those that can be resolved by updating the OS software, refer to the Red Hat Security Updates

> **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 issuing the command

$ sudo dnf update -nobest

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

4.1. 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.

4.2. 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:

$ sudo list --installed cuda-toolkit-*

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

$ sudo 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

4.3. 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 Red Hat repository.

$ sudo update

1. Show all available CUDA Toolkit releases.

$ sudo dnf 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

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

4.4. 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:

$ sudo dnf 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.

4.5. Installing GPUDirect Storage Support

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

4.5.1. Prerequisites to Installing GPUDirect Storage Support

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

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

WARNING: nvidia-peer-memory module not loading issue:

DGX os5.1.1 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 is not compatible with MLNX_OFED <= 5.4-1.0.3.0

Attempting to use nv_peer_mem 1.2 with MLNX_OFED <= 5.4-1.0.3.0 will result in the following error message:

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 ...

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

$ sudo dnf downgrade nvidia-peer-memory-dkms-21-03-0.el modprobe nv_peer_mem

4.5.2. Installing nvidia-gds

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

1. Populate the `${NVIDIA_DRV_VERSION}` variable.
1. Install `nvidia-gds` with the correct dependencies.

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

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

Chapter 5. Using the NVIDIA Mellanox InfiniBand Drivers

The DGX software stack for Red Hat-derived operating systems does not include the Mellanox OpenFabrics Enterprise Distribution (MLNX_OFED) for Linux. This is to ensure that the MLNX_OFED driver is in sync with the Red Hat distribution kernel. This section describes how to download, install, and upgrade MLNX_OFED on systems that are running Red Hat Enterprise Linux.

5.1. Determining the MLNX_OFED Version to Install

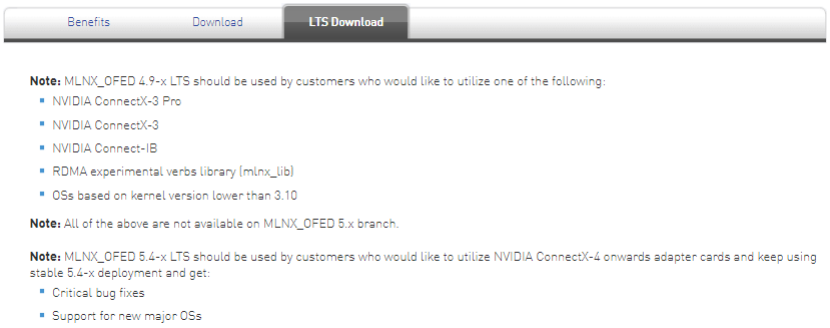
NVIDIA validates each release of NVIDIA EL8 software with a specific MLNX_OFED version. Consult the NVIDIA EL8 release notes for the recommended MLNX_OFED version to install for a particular version of NVIDIA EL8 software.

5.2. Installing the NVIDIA Mellanox InfiniBand Drivers

This section describes how to install MLNX_OFED on systems that do not yet have it installed. It is imperative that a validated MLNX_OFED version is used for the RHEL version that the DGX system is running. Note that the "`dnf update`" command that is run before installing the NVIDIA driver will update the system to the latest Red Hat Enterprise Linux version.

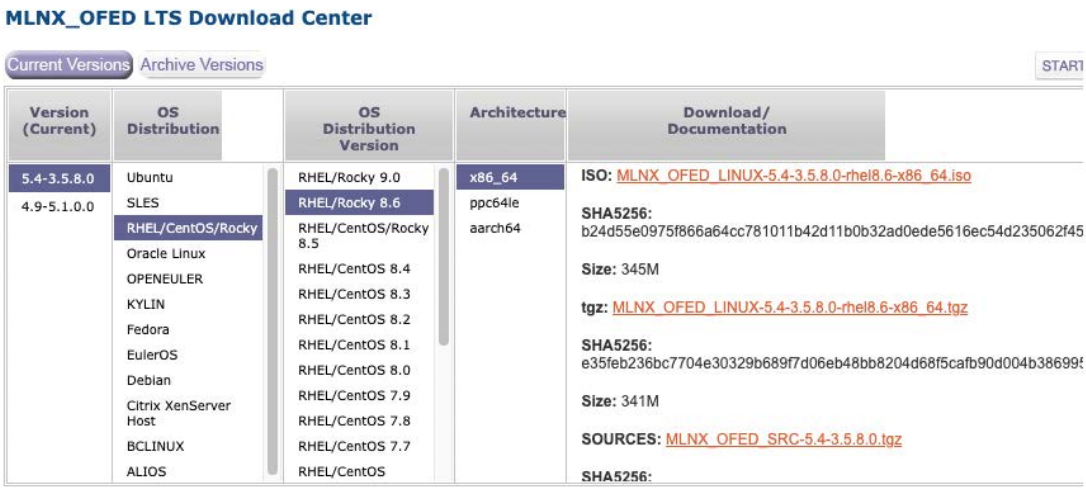
1. Determine which version of Red Hat Enterprise Linux is installed on the DGX system.
   ```
   cat /etc/redhat-release
   ```
2. Determine the appropriate MLNX_OFED software bundle to install..

   Refer to Determining the MLNX_OFED Version to Install.
3. Download the MLNX_OFED software bundle.
   a). Visit the Linux InfiniBand Drivers page, scroll down to the Download wizard, and then click the **LTS Download** tab.



b). At the **MLNX_OFED Download Center** matrix, choose

- The version to install (you may need to select **Archive Versions**),
- **RHEL/CentOS** (under OS Distribution), and
- The relevant OS Distribution Version and Architecture.



c). Click the desired ISO/tgz package.

To obtain the download link, accept the End User License Agreement.

4. After downloading the correct MLNX_OFED software bundle, proceed with the installation steps.
   a). Re-visit the MLNX_OFED Software Releases site and select the MLNX_OFED software version you intend to use.

b). Use the side menu to navigate to **Installation->Installing MLNX_OFED**, and follow the instructions.

> **Note:** The system may report that additional software needs to be installed before performing the installation. If such a message appears, install the software and then retry installing the MLNX_OFED driver.

5. If you intend to use NVIDIA GPUDirect Storage (GDS), enable the driver's GDS support according to the instructions at https://docs.nvidia.com/gpudirect-storage/troubleshooting-guide/index.html#mofed-req-install.
6. Install `nvidia-mlnx-config`.
```
sudo dnf install -y nvidia-mlnx-config
```
7. Install kernel headers and development packages for your kernel.

   These are needed for the ensuing DKMS compilation.
```
sudo dnf install -y kernel-headers-$(uname -r) kernel-devel-$(uname -r)
```
8. After installing the MLNX_OFED drivers, install the NVIDIA peer memory module.
```
sudo dnf install -y nvidia-peer-memory-dkms
```

> **Note:** nvidia-peer-memory-dkms version 1.2 or later, requires MOFED version 5.4-3.0.3.0 or later. Using MOFED versions that do not fit this criteria will result in a build failure of the nv_peer_mem DKMS module. For more information, see:
>
> - https://github.com/Mellanox/nv_peer_memory/issues/94#

> **Note:** While in-box drivers may be available, using the in-box drivers is not recommended as they provide lower performance than the official MLNX OFED drivers and they do not support the GPUDirect™ RDMA feature. For more information on configuring the in-box drivers, see the following Red Hat Enterprise Linux documentation:
>
> - Configuring InfiniBand and RDMA Networks

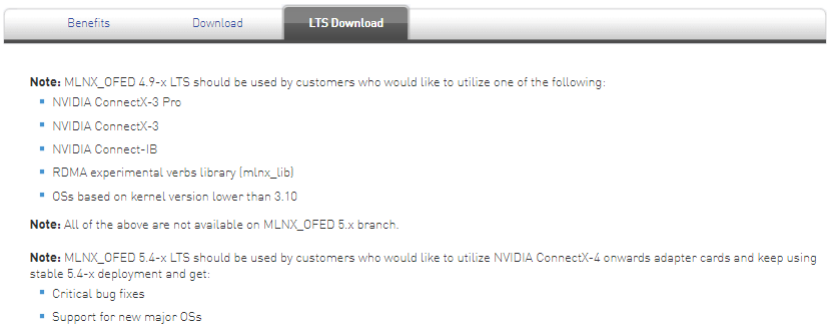
5.3. Updating the NVIDIA Mellanox InfiniBand Drivers

This section describes how to update MOFED on systems that already have it installed. The Mellanox InfiniBand Drivers in RPM packages are precompiled for a specific kernel version. Again -- it is imperative that the correct MOFED version is used for the RHEL version that the DGX system has been updated to. There is no need to uninstall the current MOFED first, because the "mlnxofedinstall" script will automatically uninstall any previously installed versions.

1. Upgrade the Red Hat Enterprise Linux release and kernel version.
```
sudo dnf update --nobest
```

2. Determine which version of Red Hat Enterprise Linux is installed on the DGX system.
   ```
   cat /etc/redhat-release
   ```
3. Determine the appropriate MLNX_OFED software bundle to install..

   Refer to Determining the MLNX_OFED Version to Install.
4. Download the MLNX_OFED software bundle.
   a). Visit the Linux InfiniBand Drivers page, scroll down to the Download wizard, and then click the **LTS Download** tab.



   b). At the **MLNX_OFED Download Center** matrix, choose

   - The version to install (you may need to select **Archive Versions**),
   - **RHEL/CentOS** (under OS Distribution), and
   - The relevant OS Distribution Version and Architecture.

   c). Click the desired ISO/tgz package.

   To obtain the download link, accept the End User License Agreement.
5. Mount the downloaded ISO somewhere on the system.

   The following example shows the ISO being mounted on the `/mnt` directory.

```
sudo mount MLNX_OFED_LINUX-<version>.iso /mnt
```

6. Prepare to install the driver.
   a). Remove `nvidia-mlnx-config` and `nvidia-peer-memory-dkms`.
   ```
   sudo dnf remove -y nvidia-mlnx-config nvidia-peer-memory-dkms
   ```
   The `mlnxofedinstall` step will remove packages prior to installing new ones, causing `nvidia-mlnx-config` and `nvidia-peer-memory-dkms` to fall out because they depend on some of these removed packages. Removing those components ahead of time avoids issues. These will be reinstalled as a final step.
   b). Specify the new kernel version to use when installing the driver.
   ```
   NEXTKERNEL=$(sudo grubby --default-kernel | sed 's/.*vmlinuz\-//g')
   ```
7. Install the driver with the `-k` and `-s` flags to specify the new kernel version and kernel source path.
   ```
   sudo /mnt/mlnxofedinstall -k ${NEXTKERNEL} -s /lib/modules/${NEXTKERNEL}/build --force
   ```

   Note: The system may report that additional software needs to be installed before performing the installation. If such a message appears, install the software and then repeat this step.

8. Reboot.
   ```
   sudo reboot
   ```
9. Reinstall `nvidia-mlnx-config` and `nvidia-peer-memory-dkms`.
   ```
   sudo dnf install -y nvidia-mlnx-config nvidia-peer-memory-dkms
   ```
10. If you intend to use NVIDIA GPUDirect Storage (GDS), enable the driver's GDS support according to the instructions at https://docs.nvidia.com/gpudirect-storage/troubleshooting-guide/index.html#mofed-req-install.

Chapter 6. Running Containers

The following is an example of running the CUDA container from the NGC registry.

```
sudo docker run --gpus=all --rm nvcr.io/nvidia/cuda:11.0-base nvidia-smi
```

Note: While users are encouraged to launch GPU containers via "docker run –-gpus=all", users who still prefer "nvidia-docker" may see an AVC denial from SELinux:

```
/bin/nvidia-docker: line 34: /bin/docker: Permission denied
```

The workaround for this issue is to install the nvidia-container-selinux package:

```
$ sudo dnf install -y nvidia-container-selinux
```

Chapter 7. Configuring Storage - NFS Mount and Cache

By default, the DGX System includes multiple SSDs in a RAID configuration (4 SSDs in the DGX-1, 8 or 16 SSDs in the DGX-2, and 3 SSDs in the DGX Station). These SSDs are intended for application caching, so NVIDIA recommends that you set up your own NFS storage for long term data storage. The following instructions describe how to mount the NFS onto the DGX System, and how to cache the NFS using the DGX SSDs for improved performance.

Make sure that you have an NFS server with one or more exports with data to be accessed by the DGX System, and that there is network access between the DGX System and the NFS server.

1. Install the nfs packages.
   ```
   $ sudo yum install nfs-utils
   ```
2. Configure an NFS mount for the DGX.
   a). Edit the filesystem tables configuration.
   ```
   sudo vi /etc/fstab
   ```
   b). Add a new line for the NFS mount, using the local mount point of `/mnt`.
   ```
   <nfs_server>:<export_path> /mnt nfs
    rw,noatime,rsize=32768,wsize=32768,nolock,tcp,intr,fsc,nofail 0 0
   ```
   `/mnt` is used here as an example mount point.

   Consult 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 as that argument specifies use of FS-Cache.

   c). Save the changes.
3. Verify the NFS server is reachable.
   ```
   ping <nfs_server>
   ```
   Use the server IP address or the server name provided by your network administrator.
4. Mount the NFS export.
   ```
   sudo mount /mnt
   ```
   /mnt is the example mount point used in step 1.
5. Verify caching is enabled.

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

Look for the text `FSC=yes` in the output. The NFS will be mounted and cached on the DGX System automatically upon subsequent reboot cycles.

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

When installing Red Hat Enterprise Linux on DGX systems, many of the packages that the DGX system requires are downloaded over the internet from both Red Hat and NVIDIA servers. This is not suitable for tightly secured systems which must be "Air-Gapped"; isolated from the internet or outside networks.

Since most DGX software updates are accomplished 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 includes a process for installing containers as well.

.

A.1. Registering Your System

See the Red Hat customer portal knowledge base article How to register and subscribe a system offline to the Red Hat Customer Portal.

A.2. Creating a Local Mirror of the NVIDIA Repository

This section presents both a general procedure for copying repositories, and also an example of copying the NVIDIA provided software repository for DGX-specific software.

The Red Hat customer portal provides a knowledge base article for creating a local mirror. Pay particular attention to the instructions under *Create a local repo with Red Hat Enterprise Linux 8.* The reposync command can now download repository metadata as well, so there is no longer a need to use createrepo after.

The repositories that will need to be mirrored are:

- rhel-8-for-x86_64-appstream-rpms
- rhel-8-for-x86_64-baseos-rpms

- codeready-builder-for-rhel-8-x86_64-rpms
- nvidia-dgx-8
- CUDA

Once mirrored, be sure to configure the target system to use your local repository. This can be accomplished by creating file under `/etc/yum.repos.d/my_mirror.repo` with the following contents:

```
[nvidia-dgx-8]
name=NVIDIA DGX EL8
baseurl=file://path/to/your/nvidia-dgx-8-repo-mirror/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-dgx-cosmos-support

[CUDA]
name=NVIDIA CUDA for EL8
baseurl=file://path/to/your/CUDA-repo-mirror/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-cuda
```

Note: The instructions assume that you have the repositories enabled on the local machine. See Enabling the DGX Software Repository for instructions on enabling the NVIDIA DGX EL8 repository.

A.3. Installing Docker Containers

This method applies to Docker containers hosted on the NGC Container Registry. Most container images are freely available, but some are locked and require that you have an NGC account to access. See the NGC Private Registry User Guide for instructions on accessing locked container images.

1. Enter the docker pull command, specifying the image registry, image repository, and tag.
   ```
   docker pull nvcr.io/nvidia/repository:tag
   ```
2. Verify the image is on your system using docker images.
   ```
   docker images
   ```
3. Save the Docker image as an archive.
   ```
   docker save nvcr.io/nvidia/repository:tag > framework.tar
   ```
4. Transfer the image to the air-gapped system using removable media such as a USB flash drive.
5. Load the NVIDIA Docker image.
   ```
   docker load -i framework.tar
   ```
6. Verify the image is on your system.
   ```
   docker images
   ```

Appendix B. Installing Software Using Local Repositories

As an alternative to the reposync method described in Appendix A, software can also be installed using local repos. These repos provided by NVIDIA will install all of the available packages locally in the filesystem, allowing installation of packages without network access.

B.1. Installing the Base Red Hat Enterprise Linux System

The Red Hat components of the software must be installed either using the standard network based installation, or as described in Appendix A. This includes registering and subscribing to the Red Hat Customer Portal.

If the system is connected to the network, follow the steps of the standard Red Hat installation, then enable the additional Red Hat repositories. Finally, execute:

```
sudo dnf update -y
sudo dnf install -y kernel-devel kernel-headers
sudo reboot
```

If the system is not connected to the network, follow the instructions in Appendix A to mirror the Red Hat repositories using reposync.

B.2. Downloading the Local Repositories

Download the DGX Software local repository from https://repo.download.nvidia.com/baseos/el/el-files/8/dgx-local-repo-22.08-4.el8.x86_64.rpm

Download the NVIDIA CUDA driver local repository from

https://us.download.nvidia.com/tesla/470.161.03/nvidia-driver-local-repo-rhel8-470.161.03-1.0-1.x86_64.rpm

Download the CUDA 11.4 Toolkit local repository from https://developer.download.nvidia.com/compute/cuda/11.4.4/local_installers/cuda-repo-rhel8-11-4-local-11.4.4_470.82.01-1.x86_64.rpm

B.3. Installing the Local Repositories

Install the local repositories

```
sudo dnf install -y ./dgx-local-repo-22.08-4.el8.x86_64.rpm
sudo dnf install -y ./nvidia-driver-local-repo-rhel8-470.161.03-1.0-1.x86_64.rpm
sudo dnf install -y ./cuda-repo-rhel8-11-4-local-11.4.4_470.82.01-1.x86_64.rpm
```

With the local repositories installed, proceed with the standard Installing the DGX Software instructions, with the exception of the CUDA driver installation. To install the CUDA driver from the local repository, follow the steps in the next section Installing the NVIDIA CUDA Driver from the Local Repository.

B.4. Installing the NVIDIA CUDA Driver from the Local Repository

- For non-NVSwitch systems such as DGX-1, DGX Station, and DGX Station A100, install the driver using the default profile:

  ```
  sudo dnf module install -y nvidia-driver:470-dkms
  sudo dnf install -y nv-persistence-mode libnvidia-nscq-470
  ```

- For NVSwitch systems such as DGX-2 and DGX A100, install the driver using the fabricmanager (fm) profile:

  ```
  sudo dnf module install -y nvidia-driver:470-dkms/fm
  sudo dnf install -y nv-persistence-mode nvidia-fm-enable
  ```

Appendix C. Changing the BMC Login

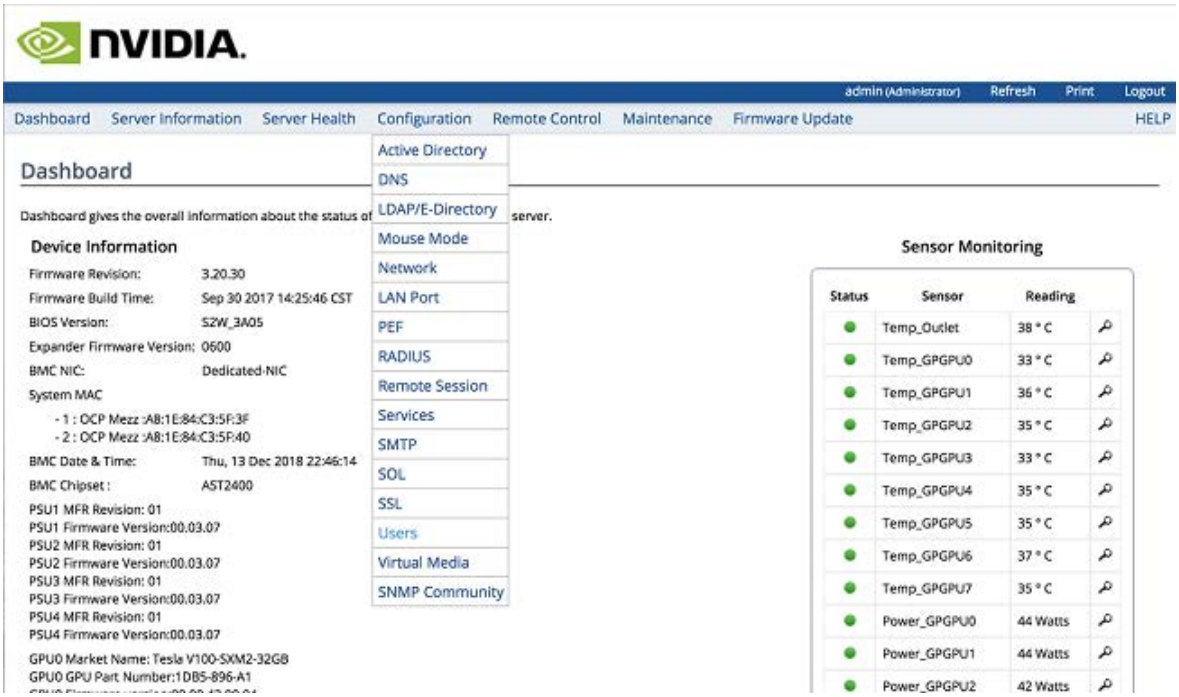
The NVIDIA DGX server includes a base management controller (BMC) for out-of-band management of the DGX system. NVIDIA recommends disabling the default username and creating a unique username and password as soon as possible.

C.1. Changing the BMC Login on the DGX-1

1. Log into the BMC.
   a). Open a browser within your LAN and go to `http://<BMC-ip-address>/`.

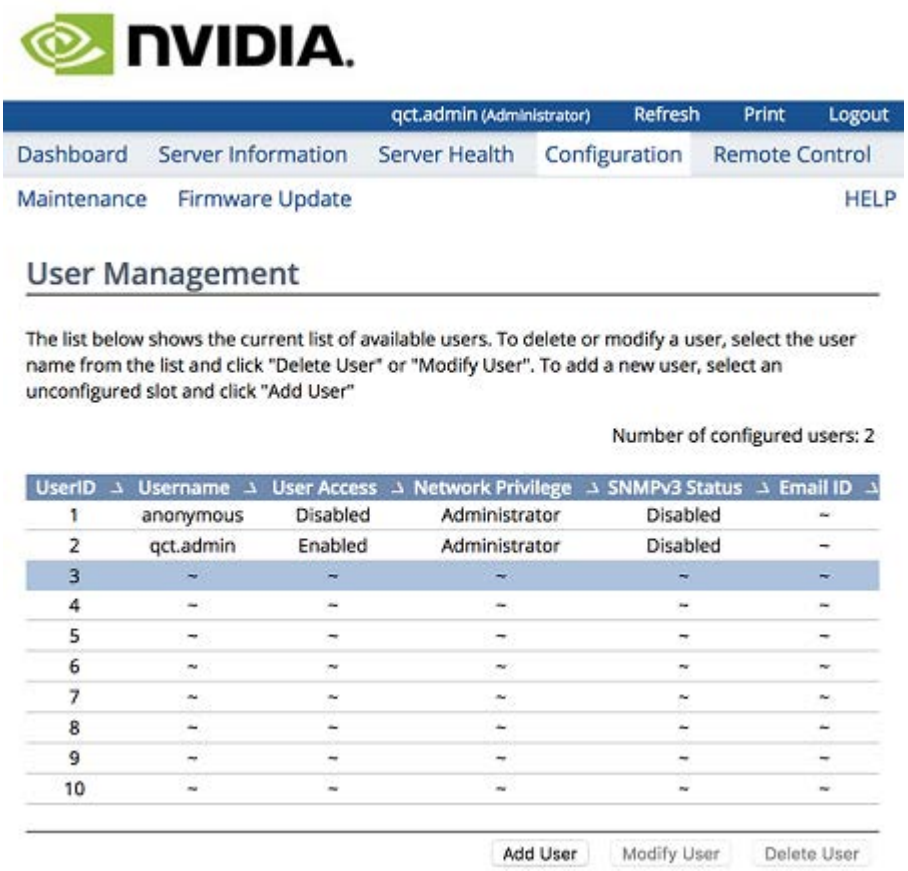
   Use Firefox or Internet Explorer. Google Chrome is not officially supported by the DGX-1 BMC.

   b). Log in, using qct.admin/qct.admin for the User ID/Password.
2. Select Configuration → Users.

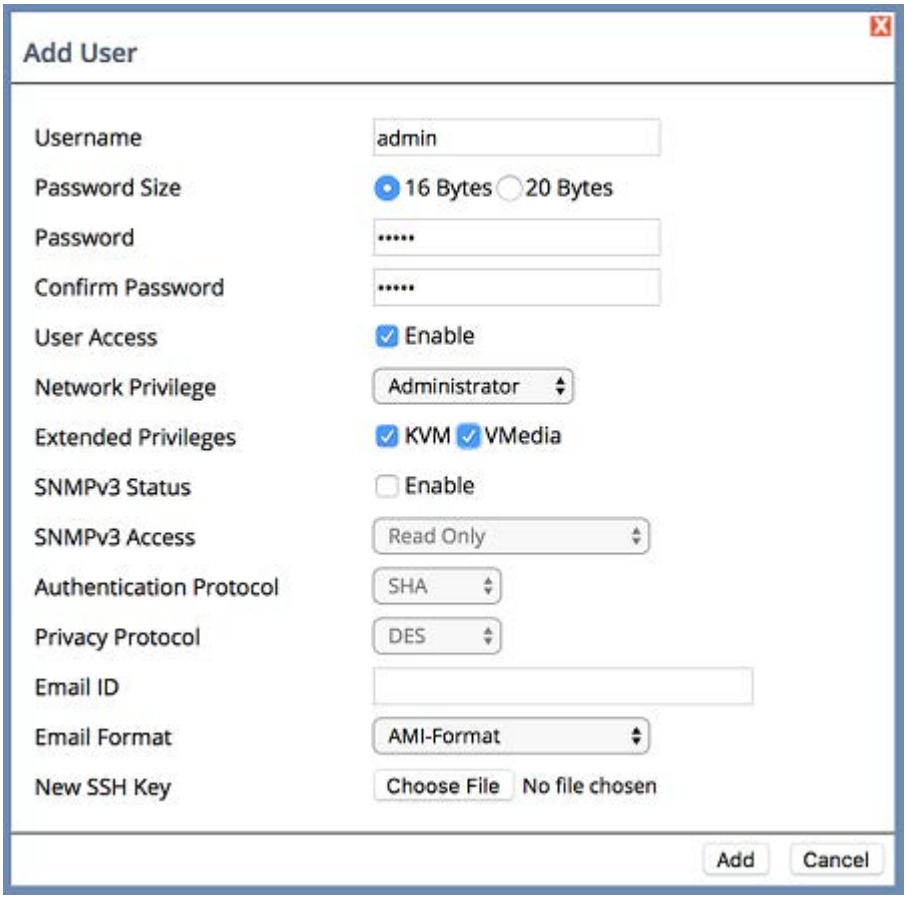


3. Add a new user.

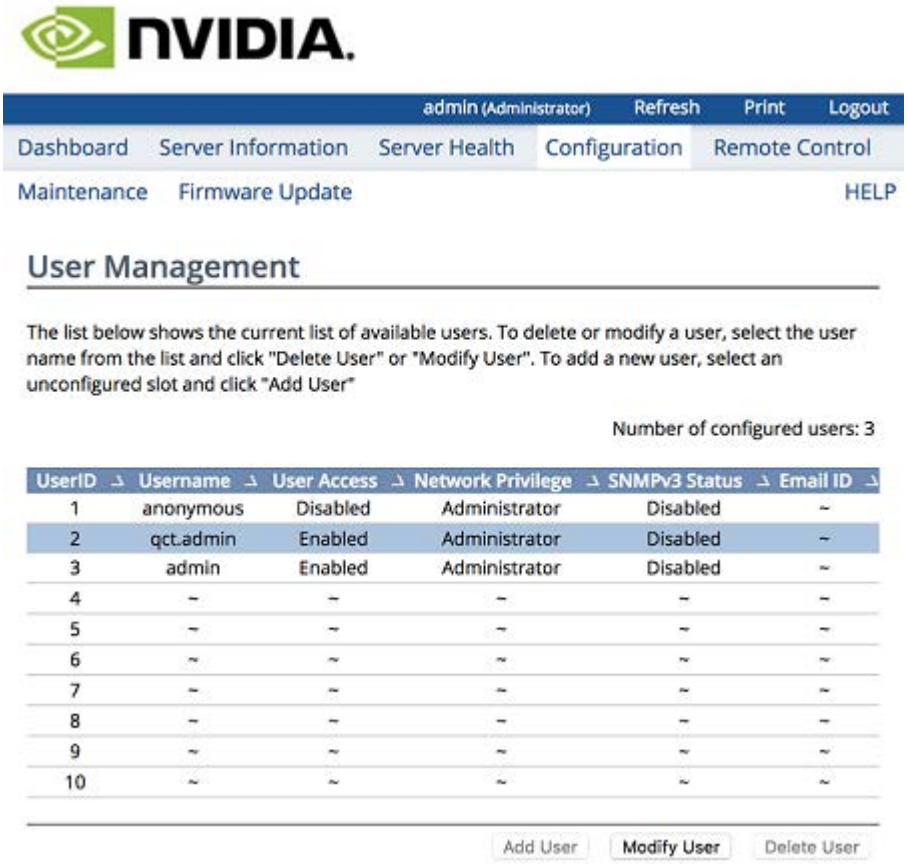
   a). Select an empty field and click Add User.



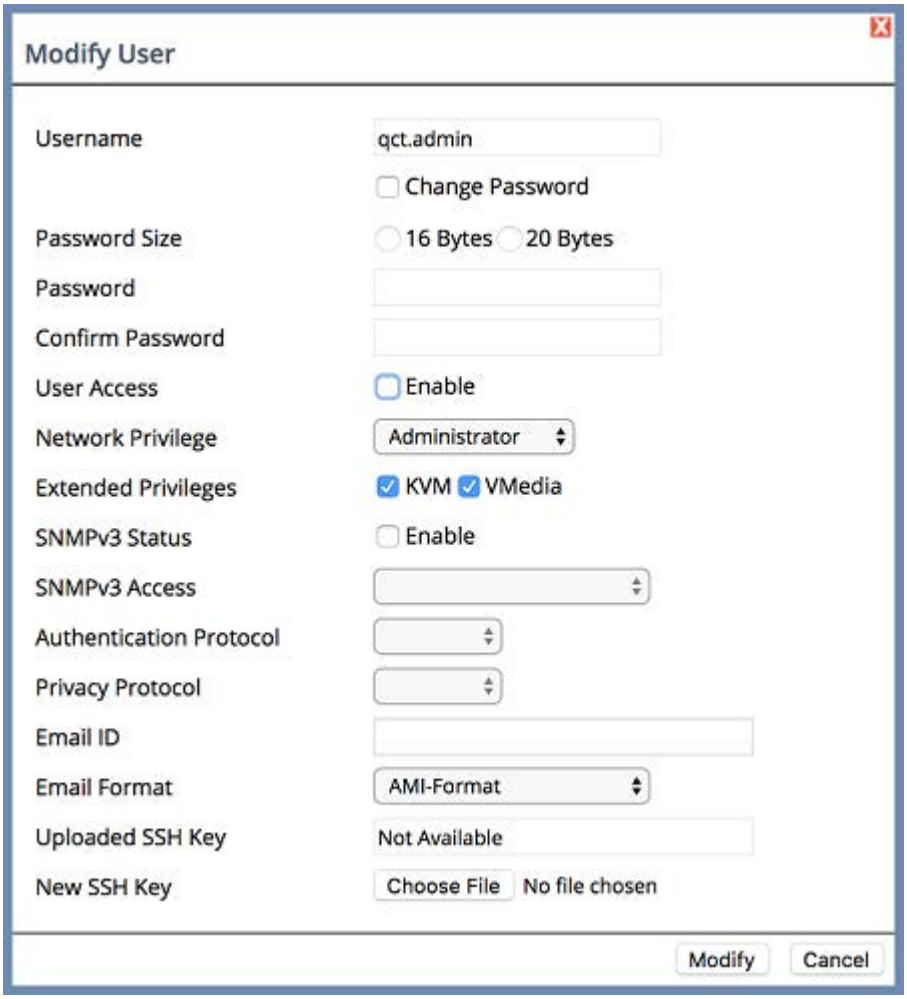
   b). Enter new user information and click Add.



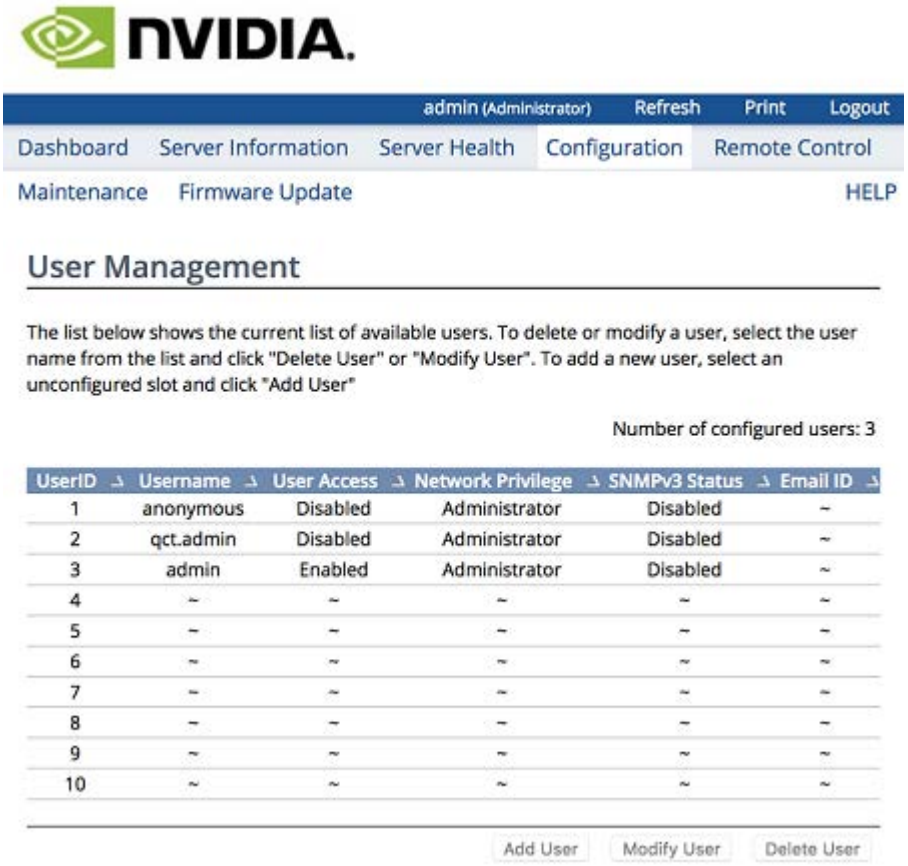
4. Log out and then log back in as the new user.
5. Select Configuration → Users.
6. Disable User Access for the user qct.admin.
   a). Select the user qct.admin user and select Modify User



b). Deselect Enable in User Access and click Modify.



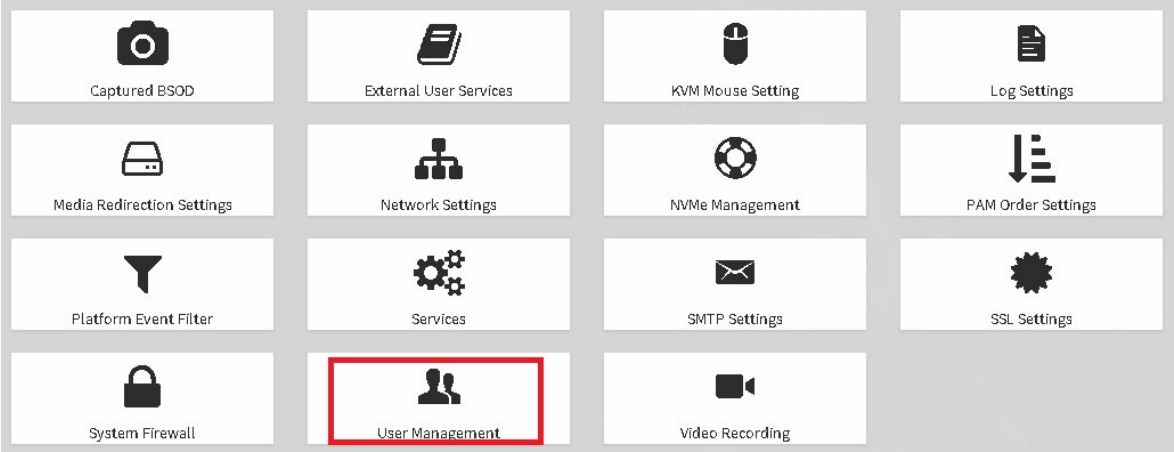
c). Ensure User Access is disabled for the user qct.admin.



7. Log out.

C.2. Changing the BMC Login on the DGX-2 or DGX A100

1. Log into the BMC.
   a). Open a browser within your LAN and go to `https://<BMC-ip-address>/`.
   b). Log in, using admin/admin for the User ID/Password.
2. Select **Settings** from the left-side navigation menu.
3. Select the **User Management** card.



4. Click the green Help icon (?) for information about configuring users, then add a new user with unique username and strong password.
5. Log out and then log back in as the new user.
6. Select **Settings** → **User Management**.
7. Disable the **admin** and **anonymous** users.

Appendix D. Installing the Mellanox ConnectX-6 Firmware

MOFED version 5.4-3.5.8.0 contains adapter card firmware version xx.31.2354, while the latest GA version of the ConnectX-6 firmware is 20.34.1002.

NVIDIA has tested Red Hat Enterprise Linux 8 and Rocky Linux with the ConnectX-6 firmware v20.31.2354. The ConnectX-6 adapter card versions used in DGX A100 are:

- OPN MCX653105A-HDA_Ax, PSID MT_0000000223
- OPN MCX653106A-HDA_Ax, PSID MT_0000000225

Appendix E. Changing NVIDIA Driver Branches

To switch driver branches, you must first remove the existing branch before installing the new branch. Currently, removing the nvidia-driver module clears the GRUB_CMDLINE_LINUX setting and this can result in the server failing to boot. Be sure to note the current GRUB_CMDLINE_LINUX setting and restore it after switching driver branches as explained in this section.

1. Note the existing GRUB_CMDLINE_LINUX setting in the file etc/default/grub.

   Example:

   ```
   GRUB_CMDLINE_LINUX=”crashkernel=auto
     rd.md.uuid=09a9380c:87edd4b6:8f5d9bbc:45e834c7 rhgb quiet
     rd.driver.blacklist=nouveau”
   ```

   Note: The "`rd.driver.blacklist=nouveau`" parameter was added when installing the driver and should not be included in the restoration.

2. Remove and clear the existing stream.

   ```
   $ sudo dnf module remove --all nvidia-driver
   $ sudo dnf module reset nvidia-driver
   ```

3. Follow the instructions in step 5 of the section Installing Required Components to install the new driver branch.
4. If the `nvidia-peer-memory-dkms` driver is installed, it must be reinstalled to match the new driver branch.

   ```
   $ sudo dnf reinstall -y nvidia-peer-memory-dkms
   ```

5. In the file /etc/default/grub, remove any extra instances of GRUB_CMDLINE_LINUX and manually edit the file to restore the original setting (except for the blacklist parameter).

   Example:

   ```
   GRUB_CMDLINE_LINUX="crashkernel=auto
    rd.md.uuid=09a9380c:87edd4b6:8f5d9bbc:45e834c7 rhgb quiet"
   ```

6. Reboot the system.

   ```
   sudo reboot
   ```