DGX-2/2H System

User Guide

DU-09130-011_v15 | November 2020

Table of Contents

Chapter 1. Introduction to the NVIDIA DGX-2/2H System

The NVIDIA® DGX-2™ System is the world's first two-petaFLOPS system that engages 16 fully interconnected GPUs for accelerated deep learning performance. The DGX-2 System is powered by NVIDIA® DGX™ software stack and an architecture designed for Deep Learning, and High-Performance Computing and analytics.

A recent addition to the DGX family, the NVIDIA DGX-2H is a high-performance version of the DGX-2. See the Hardware Overview section for specification differences.

1.1 About this Document

This document is for users and administrators of the DGX-2 System. It is organized as follows:

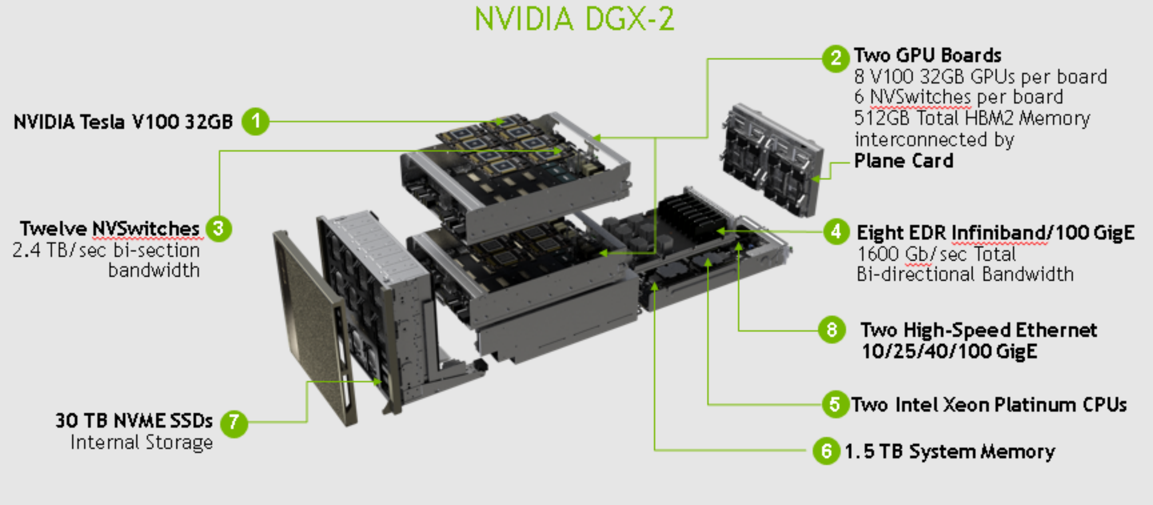
- **Chapters 1-4**: Overview of the DGX-2 System, including basic first-time setup and operation
- **Chapters 5-6**: Network and storage configuration instructions.
- **Chapter 7**: Special Features and Configuration
- **Chapters 8-10**: Software and firmware update instructions
- **Chapter 11**: How to use the BMC
- **Chapter 12**: How to configure and use the DGX-2 System as a Kernel Virtual Machine host

Unless otherwise indicated, references to the DGX-2 in this User Guide also apply to the DGX-2H.

1.2 Hardware Overview

1.2.1 Major Components

The following diagram shows the major components of the DGX-2 System.



ID	Component	Qty	Description
1	GPU	16	NVIDIA® Tesla V100 - 2 petaFLOPS (DGX-2) - 2.1 petaFLOPS (DGX-2H) - 512 GB total GPU memory - 81920 NVIDIA CUDA® Cores - 10240 NVIDIA Tensor Cores
2	GPU Boards	2	Each board consists of 8x NVIDIA® Tesla V100 6x NVSwitches 512 GB total HBM2 memory
3	NVSwitch	12	2.4 TB/s bi-section bandwidth
4	Network (cluster)	8	EDR InfiniBand or 100 GbE (1600 Gb/s total bi-directional bandwidth)
5	CPU	2	DGX-2: Dual Intel Xeon Platinum 8168, 2.7 GHz, 24-cores DGX-2H: Dual Intel Xeon Platinum 8174, 3.1 GHz, 24-cores
6	System Memory		1.5 TB
7	Storage (RAID 0) (Cache)	8	3.84 TB each (30TB total) NVMe SSDs
8	Network (storage)	2	High speed Ethernet 10/25/40/100 GbE Can be expanded with the purchase and installation of a second dual-port network adapter.

1.2.2 Other Components not in Exploded View

Component	Qty	Description
Power Supply	6	3000 W each
Storage (RAID 1) (OS)	2	960GB NVMe SSDs

1.2.3 Mechanical Specifications

Feature	Description
Form Factor	10U Rackmount
Height	17.32” (440 mm)
Width	19" (482.6 mm)
Depth	31.3" (795 mm)
Gross Weight	360 lbs (163.29 kg)

1.2.4 Power Specifications

Input		Specification for Each Power Supply	Comments
200-240 volts AC	DGX-2: 10 kW max. DGX-2H: 12 kW max.	3000 W @ 200-240 V, 16 A, 50-60 Hz	The DGX-2/2H System contains six load-balancing power supplies.

1.2.4.1 Support for Degraded Power

The DGX-2/2H includes six power supply units (PSU) configured for 5+1 redundancy. If one PSU fails, the system will continue to operate at full power with the remaining five PSUs.

The DGX-2/2H also supports operating in a degraded power mode when more than one PSU fails. If only 3 or 4 PSUs are operating, then performance is degraded slightly but the system is still operational.

Note: The DGX-2 will not operate with less than three PSUs.

1.2.4.2 DGX-2 Power Cord

The DGX-2 is shipped with a set of six (6) power cords that have been qualified for use with the DGX-2 to ensure regulatory compliance.

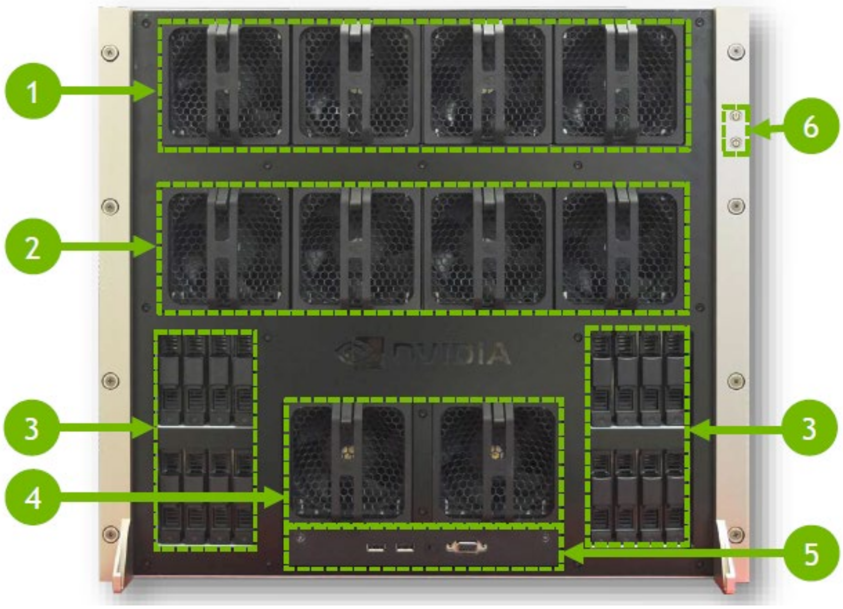
WARNING: To avoid electric shock or fire, do not connect other power cords to the DGX-2. For more details, see B.6. Electrical Precautions.

Power Cord Feature	Specification
Electrical	250VAC, 16A
Plug Standard	C19/C20
Dimension	1800mm length
Compliance	Cord: UL62, IEC60227 Connector/Plug: IEC60320-1

1.2.5 Environmental Specifications

Feature	Specification
Operating Temperature	DGX-2: 5 ° C to 35 ° C (41 ° F to 95 ° F) DGX-2H: 5 ° C to 25 ° C (41 ° F to 77 ° F)
Relative Humidity	20% to 80% noncondensing
Airflow	DGX-2: 1000 CFM @ 35 ° C DGX-2H: 1200 CFM @ 25 ° C
Heat Output	DGX-2: 34,122 BTU/hr DGX-2H: 40,945 BTU/hr

1.2.6 Front Panel Connections and Controls

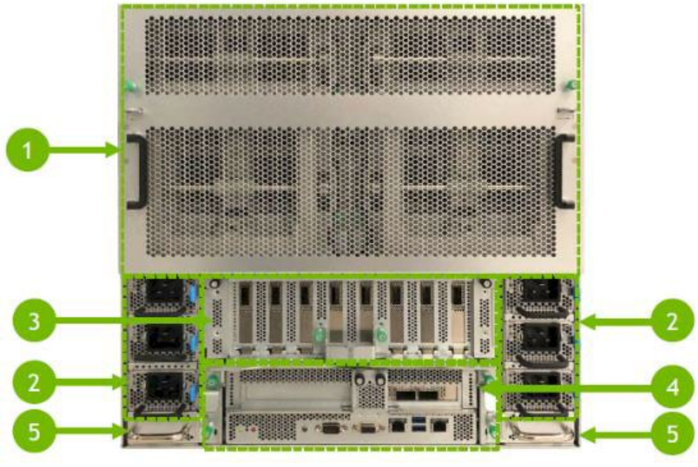


ID	Qty	Description
1	4	Upper GPU tray fans
2	4	Lower GPU tray fans
3	8 (default)	Solid State Drives. Additional SSDs available for purchase to expand to 16.
4	2	Motherboard tray fans
5	1	Front console board: USB 2.0 (2x) VGA (1x)
6	1	Power and ID buttons: Top: Power button and LED Press to turn the DGX-2 System on or off. Green steady: Power is On Amber steady: Power is Off Amber blinking: Status warning/error Bottom: ID button Press to cause an LED on the back of the unit to flash as an identifier during servicing.

! **IMPORTANT:** See the section Turning the DGX-2 On and Off for instructions on how to properly turn the system on or off.

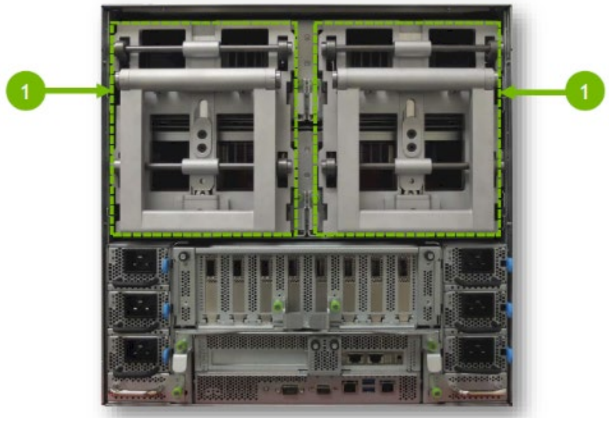
1.2.7 Rear Panel Connections and Controls

1.2.7.1 With EMI Shield Installed



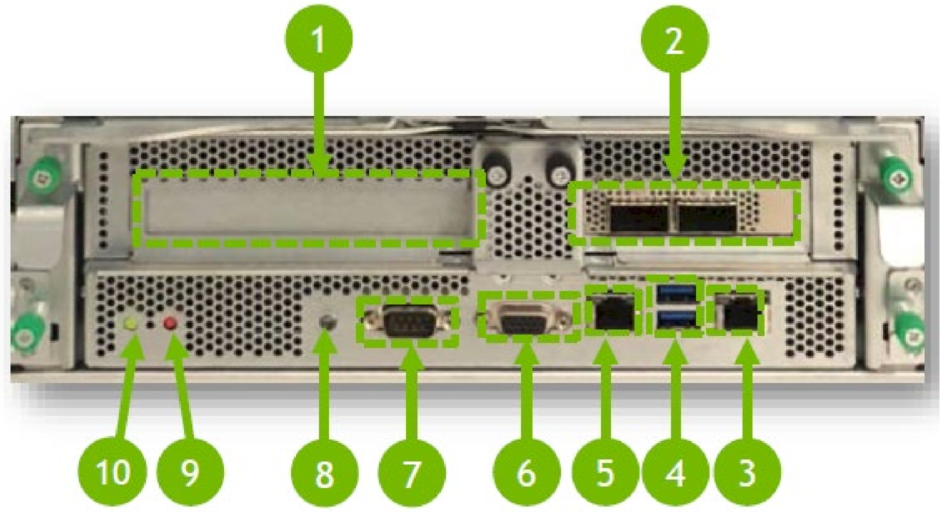
ID	Qty	Description
1	1	EMI shield
2	6	Power supplies and connectors
3	1	I/O tray
4	1	Motherboard tray
5	2	Handles to pull power supply carrier

1.2.7.2 With EMI Shield Removed



ID	Qty	Description
1	2	NVIDIA NVLink™ plane card

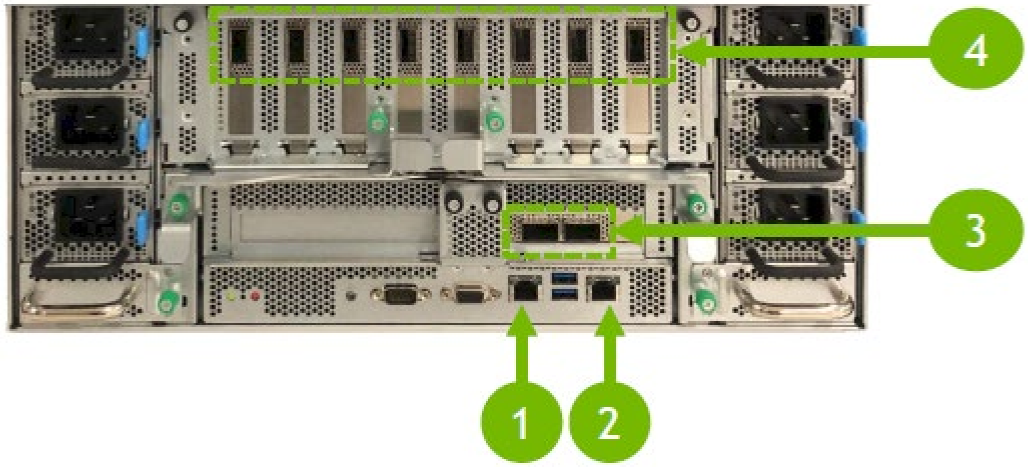
1.2.8 Motherboard Tray Ports and Controls



ID	Qty	Description
1	1	(Optional) High profile PCI card slot (for network storage)
2	2	(Default) QSFP28 network ports (for network storage) Left side port designation: `enp134s0f0` Right side port designation: `enp134s0f1`
3	1	RJ45 network port (for in-band management)
4	2	USB 3.0 ports
5	1	IPMI port (for out-of-band management (BMC))
6	1	VGA port
7	1	Serial port (DB-9)
8	1	System ID LED Blinks blue when ID button is pressed from the front of the unit as an aid in identifying the unit needing servicing
9	1	BMC reset button
10	1	Power and BMC heartbeat LED On/Off – BMC is not ready Blinking – BMC is ready

1.3 Network Ports

The following figure highlights the available network ports and their purpose.



ID	Connectivity	Uses	Number of Ports	Port Type	Cable Type
1	BMC (remote management and monitoring)	Out-of-band management	1	RJ45	100/1000 Ethernet Cat5E/6 Ethernet
2	Motherboard RJ45	In-band management, administration	1 (enp6s0)	RJ45	100/1000 Ethernet Cat5E/6 Ethernet
3	ConnectX-5 (LP) Ethernet mode	Storage (NFS) System communication	2 (Left): enp134s0f0 (Right): enp134s0f1	QSFP28	100 GbE (QSFP28) 1/10/25/40/100 GbE (QSFP28 to SFP28 or SFP+)
4	ConnectX-5 InfiniBand or Ethernet mode	Clustering Storage	8	QSFP28	InfiniBand EDR 100 Ethernet 100GbE

1.4 InfiniBand Cables

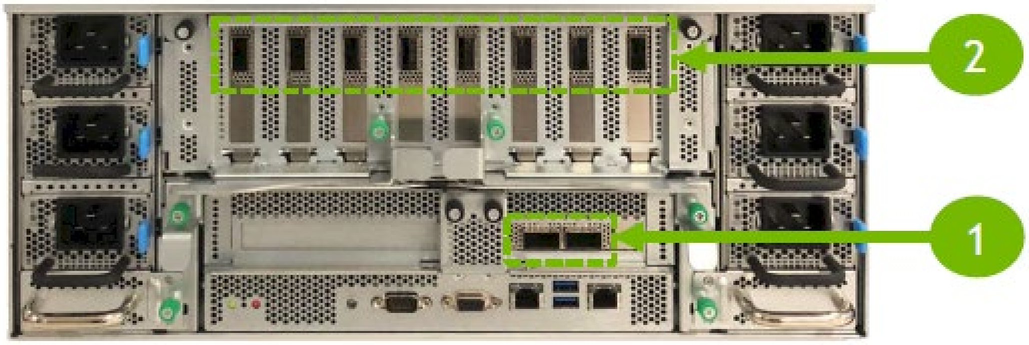
The DGX-2 System is not shipped with InfiniBand cables. For a list of cables compatible with the Mellanox ConnectX-5 VPI cards installed in the DGX-2 system, visit the Mellanox ConnectX-5 Firmware Download page, select the appropriate FW version, OPN (model), and PSID, and then select **Release Notes** from the Documentation column.

To connect the DGX-2 system to an existing 10 or 25 GbE network, you can purchase the following adaptors from NVIDIA.

Component	Mellanox MPN	Specification
Ethernet Cable Adapter	MAM1Q00A-QSA	40Gb/s to 10Gb/s, QSFP+ to SFP+
Passive Copper Hybrid Cable	MC2609130-003	Ethernet 40GbE to 4x10GbE, QSFP to 4xSFP+, 3m length
Passive Copper Hybrid Cable	MCP7F00-A003	Ethernet 100GbE to 4x25GbE, QSFP28 to 4xSFP28, 3m length, 28AWG

1.5 Recommended Ports to Use for External Storage

For clarity, the following figure reiterates the recommended ports to use for external storage. In most configurations, the storage ports (ID 1 below) should be used for connecting to high-speed NAS storage, while the cluster ports (ID 2 below) should be used for communication between nodes.



ID	Connectivity	Uses	Number of Ports	Port Type	Cable Type
1	ConnectX-5 (LP) (MCX-556A-ECAT) Ethernet mode	Storage (NFS)	2 (Left): `enp134s0f0` (Right): `enp134s0f1`	QSFP28	1/10/25/40/100 GbE
2	ConnectX-5 (MCX-555A-ECAT) InfiniBand or Ethernet mode	Cluster	8	QSFP28	EDR InfiniBand or 100 GbE

1.6 DGX OS Software

The DGX-2 System comes installed with a base OS incorporating

- An Ubuntu server distribution with supporting packages
- The NVIDIA driver
- Docker CE
- NVIDIA Container Runtime for Docker
- The following health monitoring software
  - NVIDIA System Management (NVSM)

    Provides active health monitoring and system alerts for NVIDIA DGX nodes in a data center. It also provides simple commands for checking the health of the DGX-2 SYSTEM 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.

1.7 Additional Documentation

Note: Some of the documentation listed below are not available at the time of publication. See https://docs.nvidia.com/dgx/ for the latest status.

- *DGX-2 System Service Manual*

  Instructions for servicing the DGX-2 System, including how to replace select components.
- *DGX OS Server Release Notes*

  Provides software component versions as well as a list of changes and known issues in the installed OS software.
- *NGC Container Registry for DGX*

  How to access the NGC container registry for using containerized deep learning GPU-accelerated applications on your DGX-2 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.

1.8 Customer Support

Contact NVIDIA Enterprise Support for assistance in reporting, troubleshooting, or diagnosing problems with your DGX-2 System. Also contact NVIDIA Enterprise Support for assistance in installing or moving the DGX-2 System. You can contact NVIDIA Enterprise Support in the following ways.

1.8.1 NVIDIA Enterprise Support Portal

The best way to file an incident is to log on to the NVIDIA Enterprise Support portal.

1.8.2 NVIDIA Enterprise Support Email

You can also send an email to enterprisesupport@nvidia.com.

1.8.3 NVIDIA Enterprise Support - Local Time Zone Phone Numbers

Visit NVIDIA Enterprise Support Phone Numbers.

Our support team can help collect appropriate information about your issue and involve internal resources as needed.

Chapter 2. Connecting to the DGX-2 Console

Connect to the DGX-2 console using either a direct connection, a remote connection through the BMC, or through an SSH connection.

CAUTION: Connect directly to the DGX-2 console if the DGX-2 System is connected to a 172.17.xx.xx subnet.

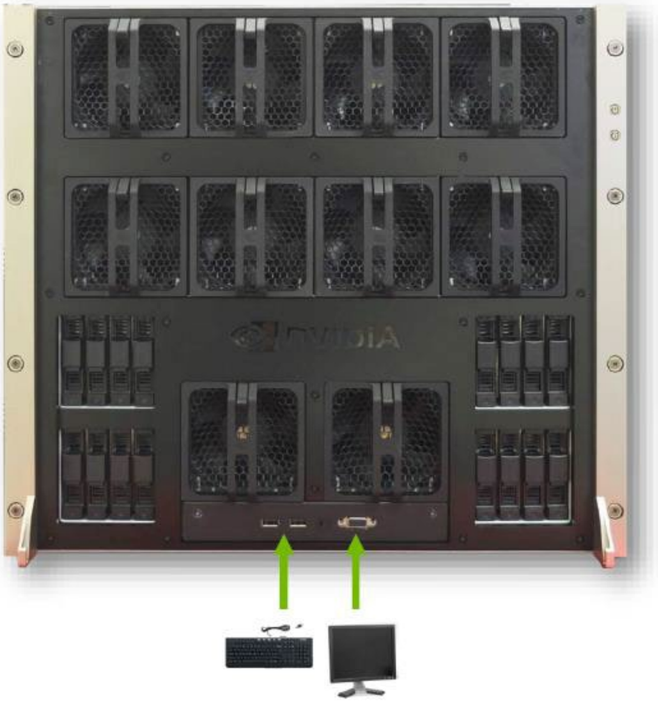
DGX OS Server software installs Docker CE which uses the 172.17.xx.xx subnet by default for Docker containers. If the DGX-2 System is on the same subnet, you will not be able to establish a network connection to the DGX-2 System.

Refer to the section Configuring Docker IP Addresses for instructions on how to change the default Docker network settings.

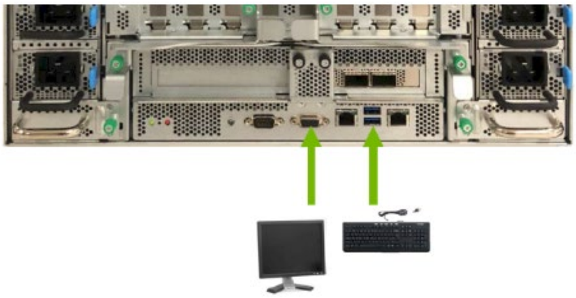
2.1 Direct Connection

At either the front or the back of the DGX-2 System, connect a display to the VGA connector, and a keyboard to any of the USB ports.

DGX-2 Server Front



DGX-2 Server Back



2.2 Remote Connection through the BMC

See the section Configuring Static IP Address for the BMC if you need to configure a static IP address for the BMC.

This method requires that you have the BMC login credentials. These credentials depend on the following conditions:

Prior to first time boot: The default credentials are

Username: admin

Password: admin

After first boot setup: The administrative user username that was set up during the initial boot is used for both the BMC username and BMC password.

Username: <administrator-username>

Password: <administrator-username>

After first boot setup with changed password: The BMC password can be changed from "<system-username>", in which case the credentials are

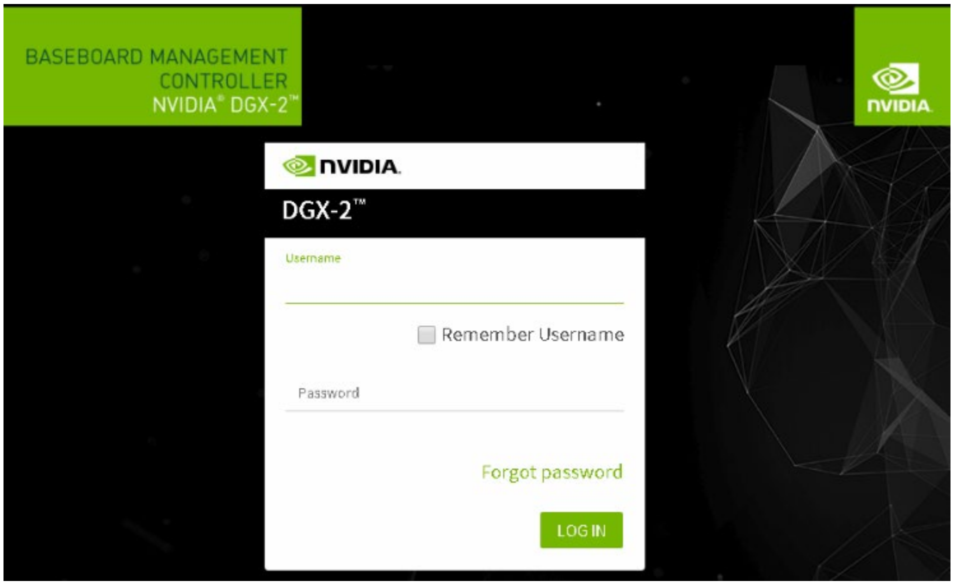
Username: <administrator-username>

Password: <new-bmc-password>

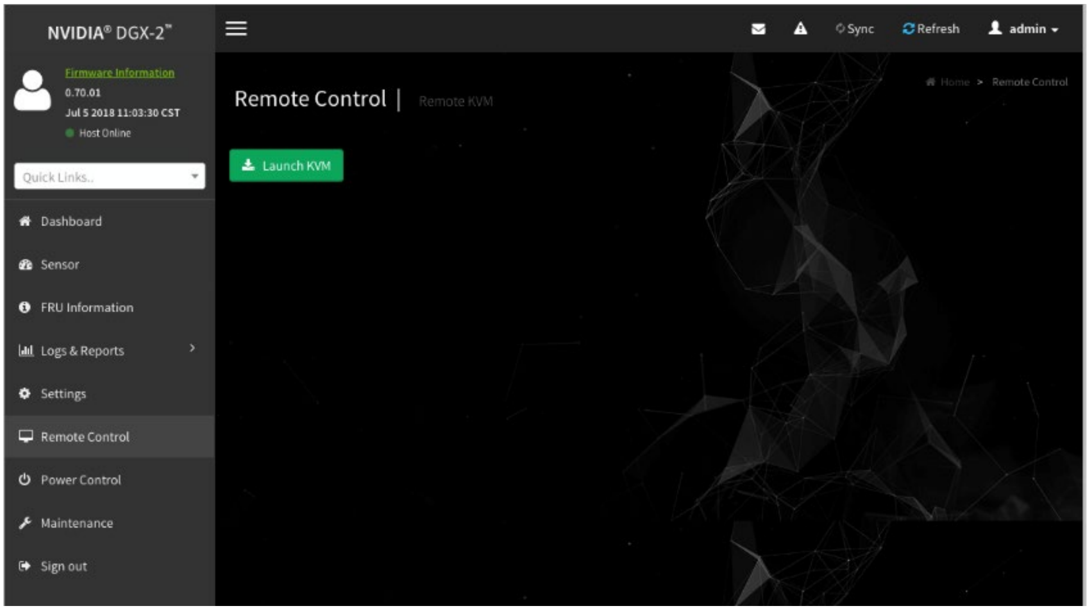
1. Make sure you have connected the BMC port on the DGX-2 System to your LAN.
2. Open a browser within your LAN and go to:

   ```
   https://<bmc-ip-address>/
   ```

   Make sure popups are allowed for the BMC address.
3. Log in.



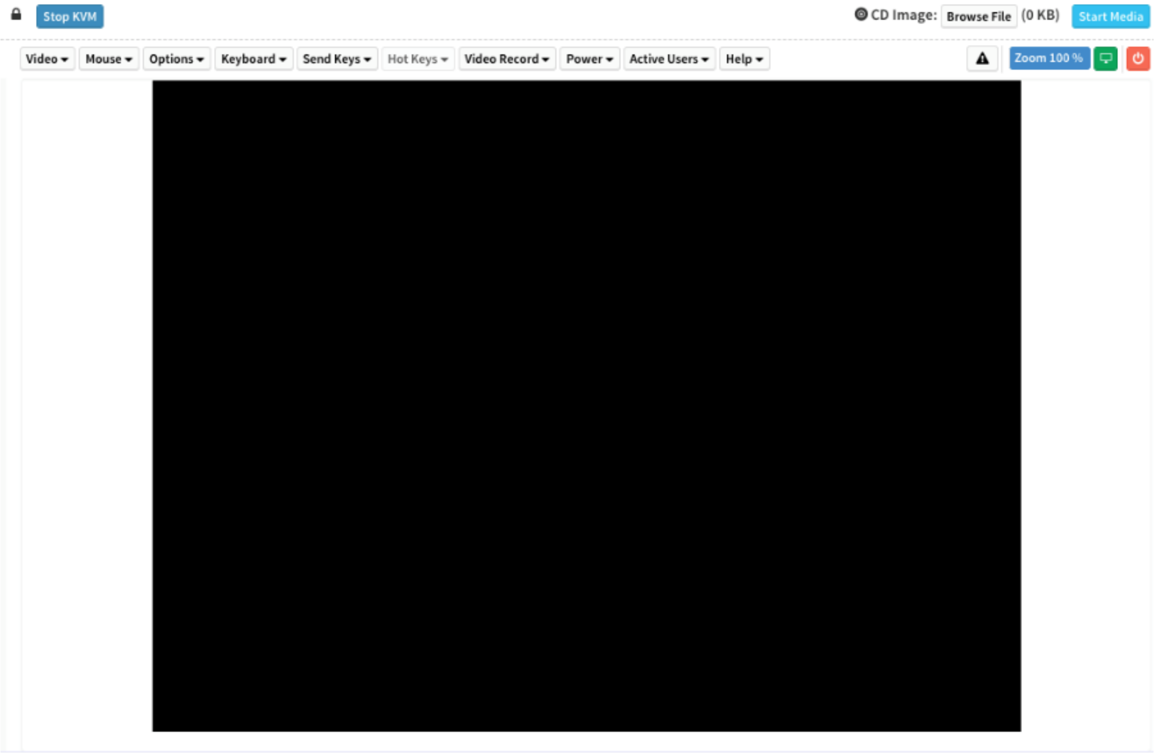
4. From the left-side navigation menu, click **Remote Control.**



The **Remote Control** page allows you to open a virtual Keyboard/Video/Mouse (KVM) on the DGX-2 System, as if you were using a physical monitor and keyboard connected to the front of the system.

5. Click Launch KVM.

   The DGX-2 console appears in your browser.



2.3 SSH Connection

You can also establish an SSH connection to the DGX-2 System through the network port. See the section Network Ports to identify the port to use, and the section Configuring Static IP Addresses for the Network Ports if you need to configure a static IP address.

Chapter 3. Setting Up the DGX-2 System

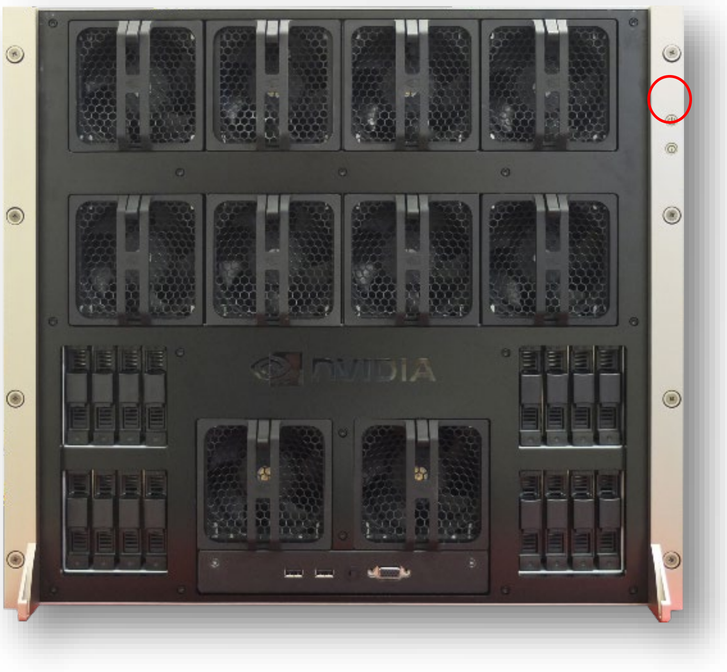
While NVIDIA service personnel will install the DGX-2 System at the site and perform the first boot setup, the first boot setup instructions are provided here for reference and to support any re-imaging of the server.

These instructions describe the setup process that occurs the first time the DGX-2 System is powered on after delivery or after the server is re-imaged.

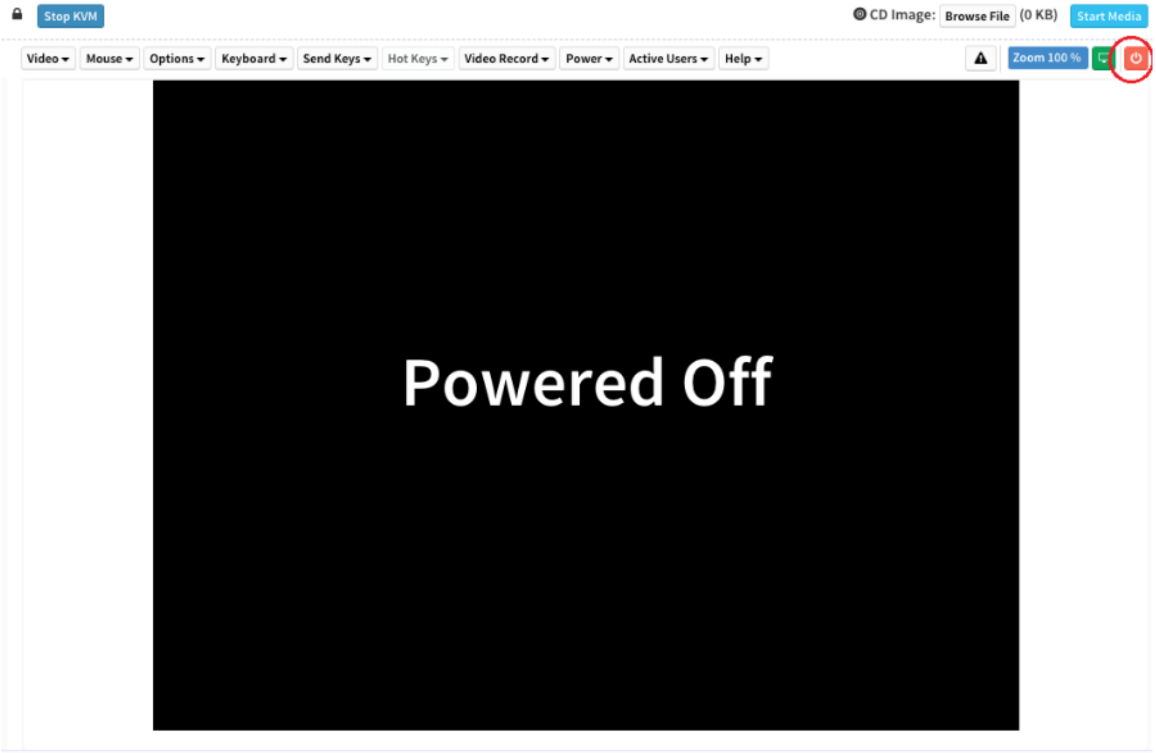
Be prepared to accept all End User License Agreements (EULAs) and to set up your username and password.

1. Connect to the DGX-2 console as explained in Connecting to the DGX-2 Console.
2. Power on the DGX-2 System.

   Using the physical power button



Using the Remote BMC



The system will take a few minutes to boot.

You are presented with end user license agreements (EULAs) for the NVIDIA software.

3. Accept all EULAs to proceed with the installation.

   The system boots and you are prompted to configure the DGX-2 software.

4. Perform the steps to configure the DGX-2 software.
   - Select your language and location.
   - Create a user account with your name, username, and password.

     You will need these credentials to log in to the DGX-2 System as well as to log in to the BMC remotely. When logging in to the BMC, enter your username for both the User ID as well as the password. Be sure to create a unique BMC password at the first opportunity.

> ! **CAUTION:** Once you create your login credentials, the default admin/admin login will no longer work.

> **Note:** The BMC software will not accept "sysadmin" for a user name. If you create this user name for the system log in, "sysadmin" will not be available for logging in to the BMC.

- Choose a primary network interface for the DGX-2 System; for example, enp6s0.

  This should typically be the interface that you will use for subsequent system configuration or in-band management.

> **Note:** After you select the primary network interface, the system attempts to configure the interface for DHCP and then asks you to enter a hostname for the system. If DHCP is not available, you will have the option to configure the network manually. If you need to configure a static IP address on a network interface connected to a DHCP network, select **Cancel** at the *Network configuration - Please enter the hostname for the system* screen. The system will then present a screen with the option to configure the network manually.

- Choose a host name for the DGX-2 System.

After completing the setup process, the DGX-2 System reboots automatically and then presents the login prompt.

5. Update the software to ensure you are running the latest version.

   Updating the software ensures your DGX-2 System contains important updates, including security updates. The Ubuntu Security Notice site (https://usn.ubuntu.com/) lists known Common Vulnerabilities and Exposures (CVEs), including those that can be resolved by updating the DGX OS software.

   a) Run the package manager.

   ```
   $ sudo apt update
   ```

   b) Upgrade to the latest version.

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

> **Note: RAID 1 Rebuild in Progress** - When the system is booted after restoring the image, software RAID begins the process of rebuilding the RAID 1 array - creating a mirror of (or resynchronizing) the drive containing the software. System performance may be affected during the RAID 1 rebuild process, which can take an hour to complete.
>
> During this time, the command “nvsm show health” will report a warning that the RAID volume is resyncing.
>
> You can check the status of the RAID 1 rebuild process using “sudo mdadm -D /dev/md0”.

Chapter 4. Quick Start Instructions

This chapter provides basic requirements and instructions for using the DGX-2 System, including how to perform a preliminary health check and how to prepare for running containers. Be sure to visit the DGX documentation website at https://docs.nvidia.com/dgx/ for additional product documentation.

4.1 Registration

To obtain support for your DGX-2 system, follow the instructions for registration in the Entitlement Certification email that was sent as part of the purchase.

Registration allows you access to the NVIDIA Enterprise Support Portal, technical support, software updates and access to set up an NVIDIA GPU Cloud for DGX account.

If you did not receive the information, open a case with our NVIDIA Enterprise Support Team at https://www.nvidia.com/en-us/support/enterprise/.

4.2 Installation and Configuration

> ! **IMPORTANT:** It is mandatory that your DGX-2 System be installed by NVIDIA service personnel or trained Advanced Technology Program (ATP) installation partner. If not performed by NVIDIA or an ATP partner, your DGX-2 hardware warranty will be voided.

Before installation, make sure you have completed the Site Survey and have given all relevant site information to your Installation Partner.

4.3 Obtaining an NVIDIA GPU Cloud Account

NVIDIA GPU Cloud (NGC) provides simple access to GPU-optimized software tools for deep learning and high-performance computing (HPC) that take full advantage of NVIDIA GPUs. An NGC account grants you access to these tools as well as the ability to set up a private registry to manage your customized tools.

Work with NVIDIA Enterprise Support to set up an NGC enterprise account if you are the organization administrator for your DGX-2 purchase. See the NGC Container Registry for DGX User Guide (https://docs.nvidia.com/dgx/ngc-registry-for-dgx-user-guide/) for detailed instructions on getting an NGC enterprise account.

4.4 Turning the DGX-2 On and OFF

The DGX-2 is a complex system, integrating a large number of cutting-edge components with specific startup and shutdown sequences. Observe the following startup and shutdown instructions.

4.4.1 Startup Considerations

In order to keep your DGX-2 running smoothly, allow up to a minute of idle time after reaching the login prompt. This ensures that all components are able to complete their initialization.

4.4.2 Shutdown Considerations

! **WARNING: Risk of Danger** - Removing power cables or using Power Distribution Units (PDUs) to shut off the system while the Operating System is running may cause damage to sensitive components in the DGX-2 server.

When shutting down the DGX-2, always initiate the shutdown from the operating system, momentary press of the power button, or by using Graceful Shutdown from the BMC, and wait until the system enters a powered-off state before performing any maintenance.

4.5 Verifying Basic Functionality

This section walks you through the steps of performing a health check on the DGX-2 System and verifying the Docker and NVIDIA driver installation.

1. Establish an SSH connection to the DGX-2 System.
2. Run a basic system check.

   ```
   $ sudo nvsm show health
   ```

   Verify that the output summary shows that all checks are Healthy and that the overall system status is Healthy.
3. Verify that Docker is installed by viewing the installed Docker version.

   ```
   $ sudo docker --version
   ```

   This should return the version as "`Docker version 18.03-ce`", where the actual version may differ depending on the specific release of the DGX OS Server software.
4. Verify connection to the NVIDIA repository and that the NVIDIA Driver is installed.

   ```
   $ sudo docker container run --runtime=nvidia --rm
   nvcr.io/nvidia/cuda:10.0-runtime nvidia-smi
   ```

   Docker pulls the nvidia/cuda container image layer by layer, then runs nvidia-smi.

   When completed, the output should show the NVIDIA Driver version and a description of each installed GPU.

See the NVIDIA Containers and Deep Learning Frameworks User Guide at https://docs.nvidia.com/deeplearning/frameworks/user-guide/index.html for further instructions, including an example of logging into the NGC container registry and launching a deep learning container.

4.6 Running NGC Containers with GPU Support

To obtain the best performance when running NGC containers on DGX-2 systems, two methods of providing GPU support for Docker containers have been developed:

- Native GPU support (included in Docker 19.03 and later)
- NVIDIA Container Runtime for Docker (`nvidia-docker2` package)

The method implemented in your system depends on the DGX OS version installed.

DGX OS Release	Method included
4.2 and later	Native GPU support NVIDIA Container Runtime for Docker
4.1	NVIDIA Container Runtime for Docker
4.0	NVIDIA Container Runtime for Docker

Each method is invoked by using specific Docker commands, described as follows.

4.6.1 Using Native GPU Support

Note: If Docker is updated to 19.03 on a system which already has the nvidia-docker2 package installed, then the instructions for using the NVIDIA Container Runtime for Docker can still be used.

- Use `docker run --gpus` to run GPU-enabled containers.
  - Example using all GPUs

    ```
    $ docker run --gpus all ...
    ```

  - Example using two GPUs

    ```
    $ docker run --gpus 2 ...
    ```

  - Examples using specific GPUs

    ```
    $ docker run --gpus "device=1,2" ...
    $ docker run --gpus "device=UUID-ABCDEF,1" ...
    ```

4.6.2 Using the NVIDIA Container Runtime for Docker

With the NVIDIA Container Runtime for Docker installed (`nvidia-docker2`), you can run GPU-accelerated containers in one of the following ways.

- Use docker run and specify runtime=nvidia.

```
$ docker run --runtime=nvidia ...
```

- Use nvidia-docker run.

```
$ nvidia-docker run ...
```

  The `nvidia-docker2` package provides backward compatibility with the previous `nvidia-docker` package, so you can run GPU-accelerated containers using this command and the new runtime will be used.

- Use `docker run` with `nvidia` as the default runtime.

  You can set `nvidia` as the default runtime, for example, by adding the following line to the `/etc/docker/daemon.json` configuration file as the first entry.

```
"default-runtime": "nvidia",
```

  The following is an example of how the added line appears in the JSON file. Do not remove any pre-existing content when making this change.

```
{
 "default-runtime": "nvidia",
  "runtimes": {
     "nvidia": {
         "path": "/usr/bin/nvidia-container-runtime",
         "runtimeArgs": []
     }
 },

}
```

  You can then use `docker run` to run GPU-accelerated containers.

```
$ docker run ...
```

> **CAUTION:** If you build Docker images while `nvidia` is set as the default runtime, make sure the build scripts executed by the Dockerfile specify the GPU architectures that the container will need. Failure to do so may result in the container being optimized only for the GPU architecture on which it was built.

Instructions for specifying the GPU architecture depend on the application and are beyond the scope of this document. Consult the specific application build process for guidance.

Chapter 5. Network Configuration

This chapter describes key network considerations and instructions for the DGX-2 System.

5.1 BMC Security

NVIDIA recommends that customers follow best security practices for BMC management (IPMI port). These include, but are not limited to, such measures as:

- Restricting the DGX-2 IPMI port to an isolated, dedicated, management network
- Using a separate, firewalled subnet
- Configuring a separate VLAN for BMC traffic if a dedicated network is not available

5.2 Configuring Network Proxies

If your network requires use of a proxy server, you will need to set up configuration files to ensure the DGX-2 System communicates through the proxy.

5.2.1 For the OS and Most Applications

Edit the file `/etc/environment` 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.

Example:

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

5.2.2 For apt

Edit (or create) a proxy config file `/etc/apt/apt.conf.d/myproxy` 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>/";
```

Where username and password are optional.

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/";
```

5.2.3 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, see https://docs.docker.com/engine/admin/systemd/#http-proxy.

5.3 Configuring Docker IP Addresses

To ensure that the DGX-2 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-2 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, then no changes are needed and you can skip this section.***

However, if your network uses the addresses within this range for the DGX-2 System, you should change the default Docker network addresses.

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

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

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

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

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

LimitMEMLOCK=infinity
LimitSTACK=67108864
```

   Save and close the `/etc/systemd/system/docker.service.d/docker-override.conf` file when done.

3. Reload the systemctl daemon.

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

4. Restart Docker.

```
$ sudo systemctl restart docker
```

5.4 Opening Ports

Make sure that the ports listed in the following table are open and available on your firewall to the DGX-2 System:

Port (Protocol)	Direction	Use
22 (TCP)	Inbound	SSH
53 (UDP)	Outbound	DNS
80 (TCP)	Outbound	HTTP, package updates
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
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

5.5 Connectivity Requirements

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://international.download.nvidia.com/dgx/repos/

  (To be accessed using apt-get, not through a browser.)
- https://apt.dockerproject.org/repo/
- https://download.docker.com/linux/ubuntu/
- 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
```

5.6 Configuring Static IP Address for the BMC

This section explains how to set a static IP address for the BMC. You will need to do this if your network does not support DHCP.

Use one of the methods described in the following sections:

- Configuring a BMC Static IP Address Using ipmitool
- Configuring a BMC Static IP Address Using the System BIOS
- Configuring a BMC Static IP Address Using the BMC Dashboard

5.6.1 Configuring a BMC Static IP Address Using ipmitool

This section describes how to set a static IP address for the BMC from the Ubuntu command line.

> **Note:** If you cannot access the DGX-2 System remotely, then connect a display (1440x900 or lower resolution) and keyboard directly to the DGX-2 System.

To view the current settings, enter the following command.

```
$ sudo ipmitool lan print 1
```

To set a static IP address for the BMC, do the following.

1. Set the IP address source to **static.**

   ```
   $ sudo ipmitool lan set 1 ipsrc static
   ```

2. Set the appropriate address information.
   - To set the IP address ("Station IP address" in the BIOS settings), enter the following and replace the italicized text with your information.

     ```
     $ sudo ipmitool lan set 1 ipaddr 10.31.241.190
     ```

   - To set the subnet mask, enter the following and replace the italicized text with your information.

     ```
     $ sudo ipmitool lan set 1 netmask 255.255.255.0
     ```

   - To set the default gateway IP ("Router IP address" in the BIOS settings), enter the following and replace the italicized text with your information.

     ```
     $ sudo ipmitool lan set 1 defgw ipaddr 10.31.241.1
     ```

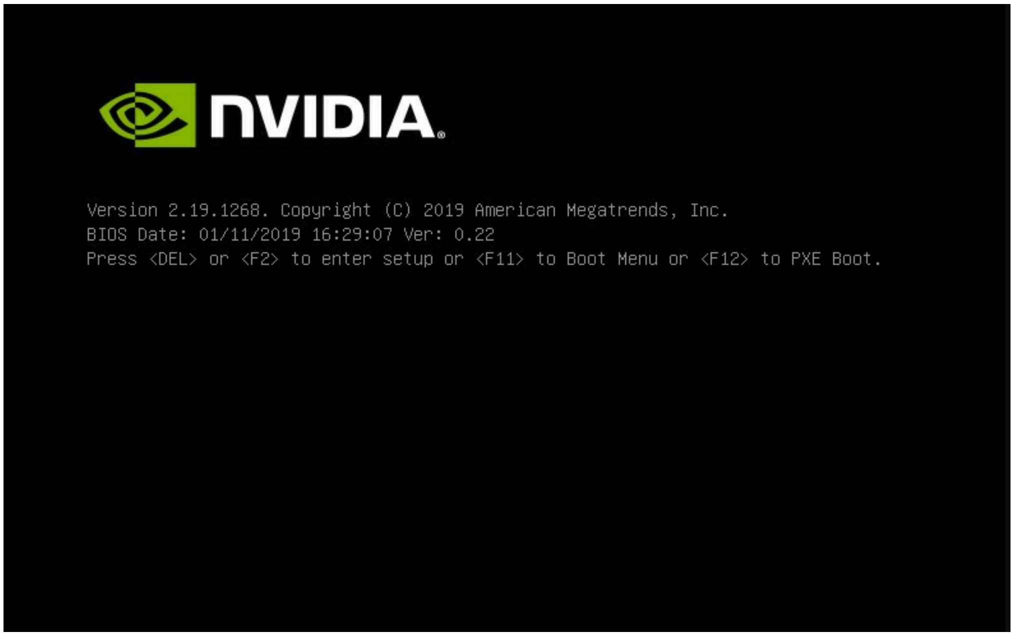
5.6.2 Configuring a BMC Static IP Address Using the System BIOS

This section describes how to set a static IP address for the BMC when you cannot access the DGX-2 System remotely. This process involves setting the BMC IP address during system boot.

1. Connect a keyboard and display (1440 x 900 maximum resolution) to the DGX-2 System, then turn on the DGX-2 System.

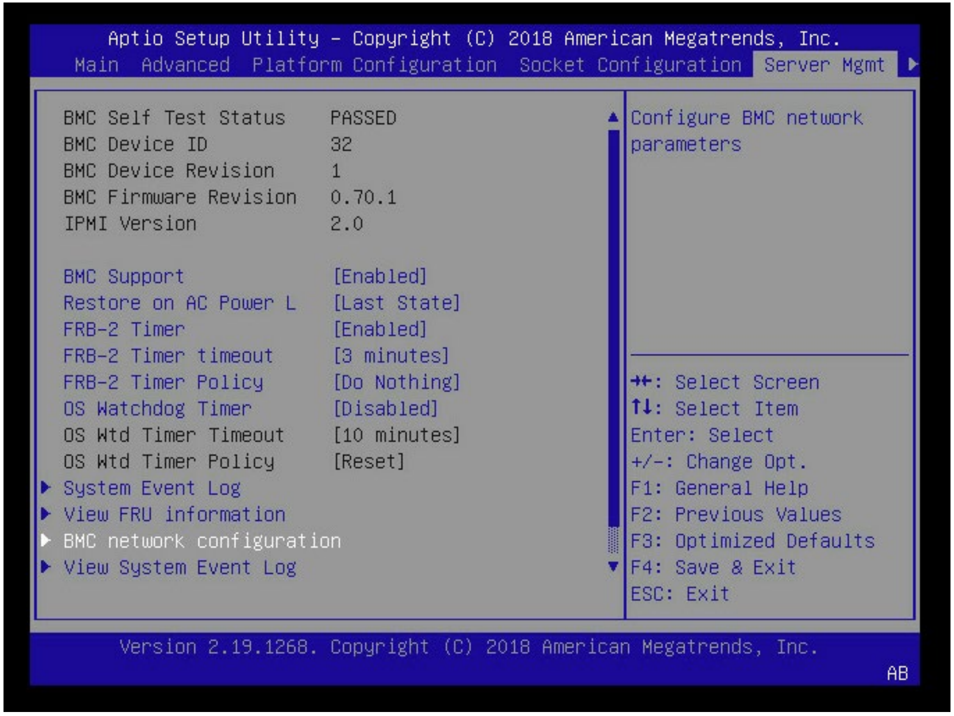
2. When you see the SBIOS version screen, press **Del** or **F2** to enter the BIOS Setup Utility screen.

   Example setup screen – details may vary depending on SBIOS version.



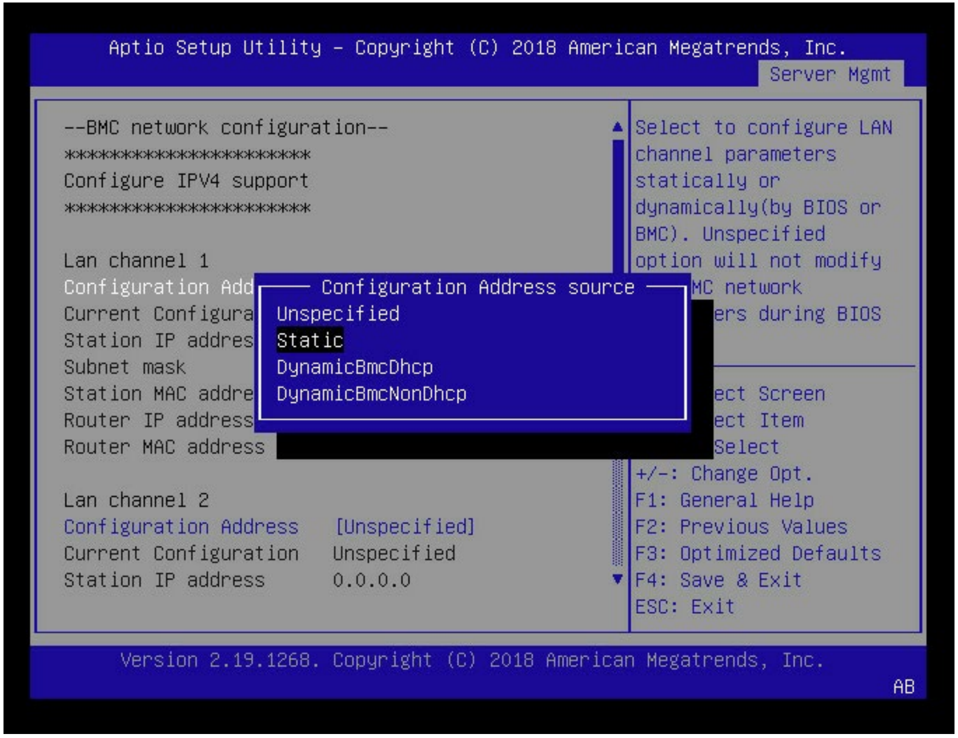
```
Version 2.19.1268. Copyright (C) 2019 American Megatrends, Inc.
BIOS Date: 01/11/2019 16:29:07 Ver: 0.22
Press <DEL> or <F2> to enter setup or <F11> to Boot Menu or <F12> to PXE Boot.
```

3. At the BIOS Setup Utility screen, navigate to the **Server Mgmt** tab on the top menu, then scroll to **BMC network configuration** and press **Enter**.



```
        Aptio Setup Utility – Copyright (C) 2018 American Megatrends, Inc.
   Main  Advanced  Platform Configuration  Socket Configuration  Server Mgmt ▶

   BMC Self Test Status     PASSED                   Configure BMC network
   BMC Device ID            32                       parameters
   BMC Device Revision      1
   BMC Firmware Revision    0.70.1
   IPMI Version             2.0

   BMC Support              [Enabled]
   Restore on AC Power L    [Last State]
   FRB-2 Timer              [Enabled]
   FRB-2 Timer timeout      [3 minutes]
   FRB-2 Timer Policy       [Do Nothing]             →←: Select Screen
   OS Watchdog Timer        [Disabled]               ↑↓: Select Item
   OS Wtd Timer Timeout     [10 minutes]             Enter: Select
   OS Wtd Timer Policy      [Reset]                  +/-: Change Opt.
 ▶ System Event Log                                  F1: General Help
 ▶ View FRU information                              F2: Previous Values
 ▶ BMC network configuration                         F3: Optimized Defaults
 ▶ View System Event Log                             F4: Save & Exit
                                                     ESC: Exit

        Version 2.19.1268. Copyright (C) 2018 American Megatrends, Inc.
                                                                        AB
```

4. Scroll to **Configuration Address Source** and press **Enter**, then at **the Configuration Address source** pop-up, select **Static** and then press **Enter**.

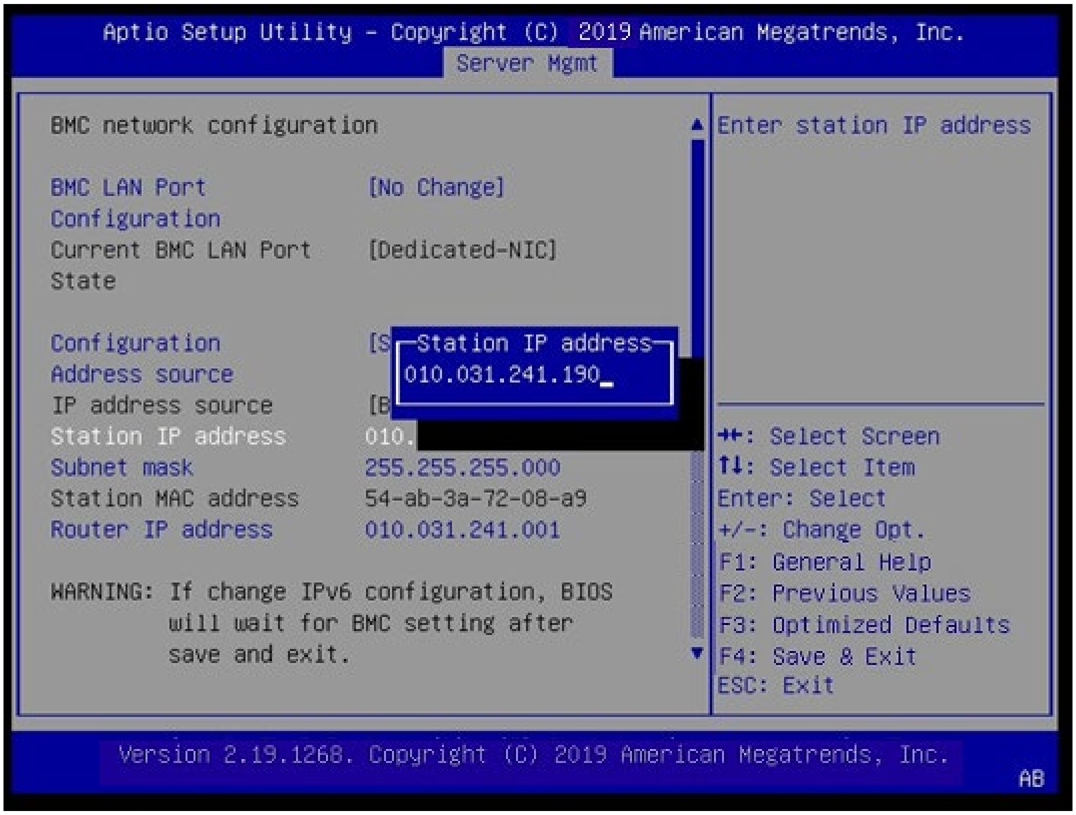


```
        Aptio Setup Utility – Copyright (C) 2018 American Megatrends, Inc.
                                                              Server Mgmt

  --BMC network configuration--                  ▲ Select to configure LAN
  **********************                           channel parameters
  Configure IPV4 support                           statically or
  **********************                           dynamically(by BIOS or
                                                   BMC). Unspecified
  Lan channel 1                                    option will not modify
  Configuration Add┌── Configuration Address source ──┐MC network
  Current Configura│ Unspecified                      │ers during BIOS
  Station IP addres│ Static                           │
  Subnet mask      │ DynamicBmcDhcp                   │
  Station MAC addre│ DynamicBmcNonDhcp                │ect Screen
  Router IP address└──────────────────────────────────┘ect Item
  Router MAC address                                    Select
                                                   +/-: Change Opt.
  Lan channel 2                                    F1: General Help
  Configuration Address    [Unspecified]           F2: Previous Values
  Current Configuration    Unspecified             F3: Optimized Defaults
  Station IP address       0.0.0.0               ▼ F4: Save & Exit
                                                   ESC: Exit

          Version 2.19.1268. Copyright (C) 2018 American Megatrends, Inc.
                                                                        AB
```

5. Set the addresses for the Station IP address, Subnet mask, and Router IP address as needed by performing the following for each:

   a). Scroll to the specific item and press **Enter**.

   b). Enter the appropriate information at the pop-up, then press **Enter**.



```
      Aptio Setup Utility - Copyright (C) 2019 American Megatrends, Inc.
                                  Server Mgmt
 BMC network configuration                          ▲ Enter station IP address

 BMC LAN Port               [No Change]
 Configuration
 Current BMC LAN Port       [Dedicated-NIC]
 State

 Configuration              [S┌Station IP address┐
 Address source               │010.031.241.190_  │
 IP address source          [B└──────────────────┘
 Station IP address         010.                      →←: Select Screen
 Subnet mask                255.255.255.000           ↑↓: Select Item
 Station MAC address        54-ab-3a-72-08-a9         Enter: Select
 Router IP address          010.031.241.001           +/-: Change Opt.
                                                      F1: General Help
 WARNING: If change IPv6 configuration, BIOS          F2: Previous Values
          will wait for BMC setting after             F3: Optimized Defaults
          save and exit.                            ▼ F4: Save & Exit
                                                      ESC: Exit
         Version 2.19.1268. Copyright (C) 2019 American Megatrends, Inc.
                                                                         AB
```

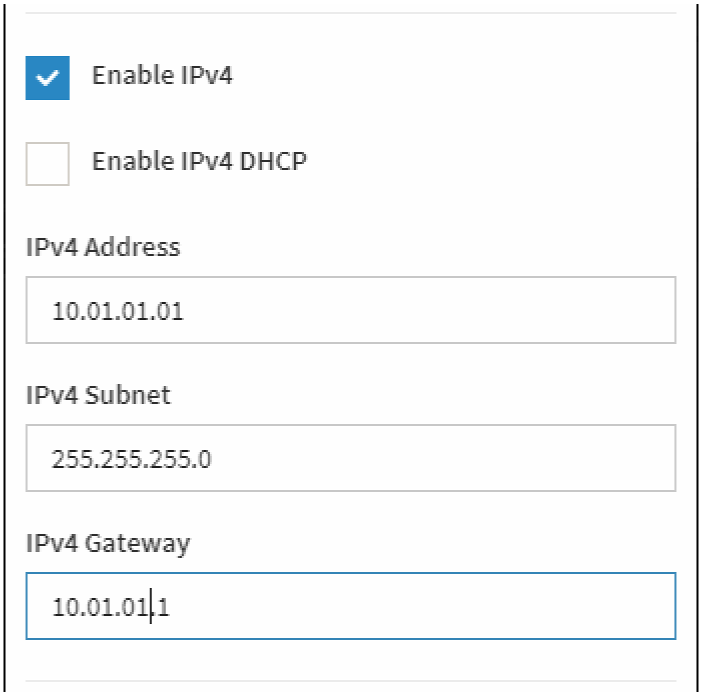
6. When finished making all your changes, press **F4** to save & exit

   You can now access the BMC over the network.

5.6.3 Configuring a BMC Static IP Address Using the BMC Dashboard

These instructions describe IPv4 addressing, but IPv6 addressing to the BMC can be configured if needed through the corresponding IPv6 fields.

1. Log into the BMC, then click **Settings**->**Network Settings**->**Network IP Settings**.
2. Clear the **Enable IPv4 DHCP** check box, then enter the appropriate values for the **IPv4 Address**, **IPv4 Subnet**, and **IPv4 Gateway** fields**.**



☑ Enable IPv4

□ Enable IPv4 DHCP

IPv4 Address

10.01.01.01

IPv4 Subnet

255.255.255.0

IPv4 Gateway

10.01.01.1

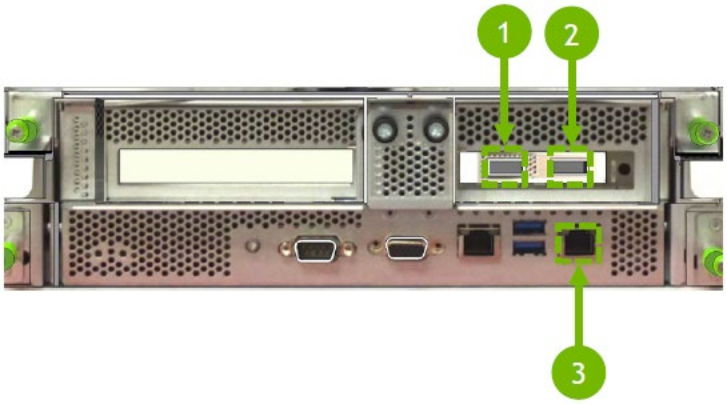
3. Click **Save** when done.

5.7 Configuring Static IP Addresses for the Network Ports

During the initial boot setup process for the DGX-2 System, you had an opportunity to configure static IP addresses for a single network interface. If you did not set this up at that time, you can configure the static IP addresses from the Ubuntu command line using the following instructions.

> **Note:** If you cannot access the DGX-2 System remotely, then connect a display (1440x900 or lower resolution) and keyboard directly to the DGX-2 System.

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



Ethernet Port Position	<port-designation>
1	enp134s0f0
2	enp134s0f1
3	enp6s0

2. Edit the network configuration yaml file.

```
$ 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. When finished with your edits, press **ESC** to switch to command mode, then save the file to the disk and exit the editor.
4. 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 https://help.ubuntu.com/lts/serverguide/network-configuration.html.en.

5.8 Switching Between InfiniBand and Ethernet

The NVIDIA DGX-2 System is equipped with eight QSFP28 network ports on the I/O board, typically used for cluster communications. By default these are configured as InfiniBand ports, but you have the option to convert these to Ethernet ports.

For these changes to work properly, the configured port must connect to a networking switch that matches the port configuration. In other words, if the port configuration is set to InfiniBand, then the external switch should be an InfiniBand switch with the corresponding InfiniBand cables. Likewise, if the port configuration is set to Ethernet, then the switch should also be Ethernet.

5.8.1 Starting the Mellanox Software Tools

1. Start the **mst** driver.

```
$ sudo mst start
```

2. To verify that the Mellanox Software Tools (MST) services are running, enter the following.

```
$ sudo mst status
```

- The following output indicates the services are ***not*** running.

```
MST modules:
------------
MST PCI module is not loaded
MST PCI configuration module is not loaded
```

- The following output indicates the services are running.

```
 MST modules:
------------
MST PCI module is not loaded
MST PCI configuration module loaded
MST devices:
------------
/dev/mst/mt4119_pciconf0    - PCI configuration cycles access.
                              domain:bus:dev.fn=0000:35:00.0 addr.reg=88 data.reg=92
                              Chip revision is: 00
/dev/mst/mt4119_pciconf1    - PCI configuration cycles access.
                              domain:bus:dev.fn=0000:3a:00.0 addr.reg=88 data.reg=92
                              Chip revision is: 00
/dev/mst/mt4119_pciconf2    - PCI configuration cycles access.
                              domain:bus:dev.fn=0000:58:00.0 addr.reg=88 data.reg=92
                              Chip revision is: 00
/dev/mst/mt4119_pciconf3    - PCI configuration cycles access.
                              domain:bus:dev.fn=0000:5d:00.0 addr.reg=88 data.reg=92
                              Chip revision is: 00
/dev/mst/mt4119_pciconf4    - PCI configuration cycles access.
                              domain:bus:dev.fn=0000:86:00.0 addr.reg=88 data.reg=92
                              Chip revision is: 00
/dev/mst/mt4119_pciconf5    - PCI configuration cycles access.
                              domain:bus:dev.fn=0000:b8:00.0 addr.reg=88 data.reg=92
                              Chip revision is: 00
/dev/mst/mt4119_pciconf6    - PCI configuration cycles access.
                              domain:bus:dev.fn=0000:bd:00.0 addr.reg=88 data.reg=92
                              Chip revision is: 00
/dev/mst/mt4119_pciconf7    - PCI configuration cycles access.
                              domain:bus:dev.fn=0000:e1:00.0 addr.reg=88 data.reg=92
                              Chip revision is: 00
/dev/mst/mt4119_pciconf8    - PCI configuration cycles access.
                              domain:bus:dev.fn=0000:e6:00.0 addr.reg=88 data.reg=92
                              Chip revision is: 00
$
```

5.8.2 Determining the Current Port Configuration

To determine the current port configuration, enter the following:

```
$ sudo mlxconfig query | egrep -e Device\|LINK_TYPE
Device #1:
Device type:    ConnectX5
Device:         0000:bd:00.0
        LINK_TYPE_P1                        IB(1)
Device #2:
Device type:    ConnectX5
Device:         0000:b8:00.0
        LINK_TYPE_P1                        IB(1)
Device #3:
Device type:    ConnectX5
Device:         0000:3a:00.0
        LINK_TYPE_P1                        IB(1)
Device #4:
Device type:    ConnectX5
Device:         0000:e1:00.0
        LINK_TYPE_P1                        IB(1)
Device #5:
Device type:    ConnectX5
Device:         0000:35:00.0
        LINK_TYPE_P1                        IB(1)
Device #6:
Device type:    ConnectX5
Device:         0000:5d:00.0
        LINK_TYPE_P1                        IB(1)
Device #7:
Device type:    ConnectX5
Device:         0000:e6:00.0
        LINK_TYPE_P1                        IB(1)
Device #8:
Device type:    ConnectX5
Device:         0000:58:00.0
        LINK_TYPE_P1                        IB(1)
Device #9:
Device type:    ConnectX5
Device:         0000:86:00.0
        LINK_TYPE_P1                        ETH(2)
        LINK_TYPE_P2                        ETH(2)
```

This output shows the first eight cards are configured for InfiniBand and correspond to the network cluster ports. The last card has two ports which correspond to the two network storage ports. These are configured for Ethernet should not be changed.

Map the Device bus numbers from your output to the device name from the `mst status` output on your system. For example, this example output shows that the device name for bus `bd` is `/dev/mst/mt4119_pciconf5.` You will need the device name when changing the configuration.

5.8.3 Switching the Port from InfiniBand to Ethernet

Make sure that you have started the Mellanox Software Tools (MST) services as explain in the section Starting the Mellanox Software Tools, and have identified the correct ports to change.

1. Change the configuration for the network cluster ports to Ethernet by setting LINK_TYPE_P1=2 for each port.

   The following example configures the 8 network cluster ports.

   ```
   ~$ sudo mlxconfig -y -d /dev/mst/mt4119_pciconf0 set LINK_TYPE_P1=2
   ~$ sudo mlxconfig -y -d /dev/mst/mt4119_pciconf1 set LINK_TYPE_P1=2
   ~$ sudo mlxconfig -y -d /dev/mst/mt4119_pciconf2 set LINK_TYPE_P1=2
   ~$ sudo mlxconfig -y -d /dev/mst/mt4119_pciconf3 set LINK_TYPE_P1=2
   ~$ sudo mlxconfig -y -d /dev/mst/mt4119_pciconf4 set LINK_TYPE_P1=2
   ~$ sudo mlxconfig -y -d /dev/mst/mt4119_pciconf5 set LINK_TYPE_P1=2
   ~$ sudo mlxconfig -y -d /dev/mst/mt4119_pciconf6 set LINK_TYPE_P1=2
   ~$ sudo mlxconfig -y -d /dev/mst/mt4119_pciconf7 set LINK_TYPE_P1=2
   ```

2. Reboot the server.
3. Verify the configuration changes have been applied.

   ```
   $ sudo mlxconfig query | egrep -e Device\|LINK_TYPE
   Device #1:
   Device type:    ConnectX5
   Device:         0000:bd:00.0
           LINK_TYPE_P1                        ETH (1)
   Device #2:
   Device type:    ConnectX5
   Device:         0000:b8:00.0
           LINK_TYPE_P1                        ETH (1)
   Device #3:
   Device type:    ConnectX5
   Device:         0000:3a:00.0
           LINK_TYPE_P1                        ETH (1)
   Device #4:
   Device type:    ConnectX5
   Device:         0000:e1:00.0
           LINK_TYPE_P1                        ETH (1)
   Device #5:
   Device type:    ConnectX5
   ```

```
Device:         0000:35:00.0
        LINK_TYPE_P1                         ETH (1)
Device #6:
Device type:    ConnectX5
Device:         0000:5d:00.0
        LINK_TYPE_P1                         ETH (1)
Device #7:
Device type:    ConnectX5
Device:         0000:e6:00.0
        LINK_TYPE_P1                         ETH (1)
Device #8:
Device type:    ConnectX5
Device:         0000:58:00.0
        LINK_TYPE_P1                         ETH (1)
Device #9:
Device type:    ConnectX5
Device:         0000:86:00.0
        LINK_TYPE_P1                         ETH(2)
        LINK_TYPE_P2                         ETH(2)
```

5.8.4 Switching the Port from Ethernet to InfiniBand

Make sure that you have started the Mellanox Software Tools (MST) as explain in the section Starting the Mellanox Software Tools, and have identified the correct ports to change.

1. Change the configuration for the network cluster ports to InfiniBand by setting LINK_TYPE_P1=1 for each port.

   The following example configures all 8 network cluster ports.

```
~$ sudo mlxconfig -y -d /dev/mst/mt4119_pciconf0 set LINK_TYPE_P1=1
~$ sudo mlxconfig -y -d /dev/mst/mt4119_pciconf1 set LINK_TYPE_P1=1
~$ sudo mlxconfig -y -d /dev/mst/mt4119_pciconf2 set LINK_TYPE_P1=1
~$ sudo mlxconfig -y -d /dev/mst/mt4119_pciconf3 set LINK_TYPE_P1=1
~$ sudo mlxconfig -y -d /dev/mst/mt4119_pciconf4 set LINK_TYPE_P1=1
~$ sudo mlxconfig -y -d /dev/mst/mt4119_pciconf5 set LINK_TYPE_P1=1
~$ sudo mlxconfig -y -d /dev/mst/mt4119_pciconf6 set LINK_TYPE_P1=1
~$ sudo mlxconfig -y -d /dev/mst/mt4119_pciconf7 set LINK_TYPE_P1=1
```

2. Verify the configuration changes have been applied.

```
$ sudo mlxconfig query | egrep -e Device\|LINK_TYPE
Device #1:
Device type:    ConnectX5
Device:         0000:bd:00.0
        LINK_TYPE_P1                         IB(1)
Device #2:
Device type:    ConnectX5
```

```
Device:         0000:b8:00.0
        LINK_TYPE_P1                        IB(1)
Device #3:
Device type:    ConnectX5
Device:         0000:3a:00.0
        LINK_TYPE_P1                        IB(1)
Device #4:
Device type:    ConnectX5
Device:         0000:e1:00.0
        LINK_TYPE_P1                        IB(1)
Device #5:
Device type:    ConnectX5
Device:         0000:35:00.0
        LINK_TYPE_P1                        IB(1)
Device #6:
Device type:    ConnectX5
Device:         0000:5d:00.0
        LINK_TYPE_P1                        IB(1)
Device #7:
Device type:    ConnectX5
Device:         0000:e6:00.0
        LINK_TYPE_P1                        IB(1)
Device #8:
Device type:    ConnectX5
Device:         0000:58:00.0
        LINK_TYPE_P1                        IB(1)
Device #9:
Device type:    ConnectX5
Device:         0000:86:00.0
        LINK_TYPE_P1                        ETH(2)
        LINK_TYPE_P2                        ETH(2)
```

Chapter 6. Configuring Storage – NFS Mount and Cache

By default, the DGX-2 System includes eight SSDs in a RAID 0 configuration. These SSDs are intended for application caching, so you must set up your own NFS storage for long term data storage.

Disabling cachefilesd

The DGX-2 system uses `cachefilesd` to manage caching of the NFS. If you do not want `cachefilesd` enabled, you can disable it as follows.

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

Using cachefilesd

The following instructions describe how to mount the NFS onto the DGX-2 System, and how to cache the NFS using the DGX-2 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-2 System, and that there is network access between the DGX-2 System and the NFS server.

1. **Configure an NFS mount for the DGX-2 System.**
   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.**

2. **Verify the NFS server is reachable.**

   ```
   ping <nfs_server>
   ```

   Use the server IP address or the server name provided by your network administrator.

3. **Mount the NFS export.**

   ```
   sudo mount /mnt
   ```

   `/mnt` is an example mount point.

4. **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-2 System automatically upon subsequent reboot cycles.

Chapter 7. Special Features and Configurations

This chapter describes specific features of the DGX-2 server to consider during setup and operation.

7.1 Setting MaxQ/MaxP

The maximum power consumption of the DGX-2 system is 10 kW. Beginning with DGX OS 4.0.5, you can reduce the power consumption of the GPUs in the DGX-2 system to accommodate server racks with a power budget of 18 kW. This allows you to install two DGX-2 systems in the rack, instead of being limited to one.

Use NVSM CLI to control the power mode of the GPUs.

Notes:

MaxQ is supported on DGX-2 systems with BMC firmware version 1.04.03 or later.

MaxQ is not supported on DGX-2H systems.

Commands to switch to MaxP or MaxQ, or to see the current power state, are not supported on DGX-2H systems.

Setting MaxP/MaxQ is not supported on DGX-2 systems configured to run kernel virtual machines (KVM mode).

7.1.1 MaxQ

- Maximum efficiency mode
- Allows two DGX-2 systems to be installed in racks that have a power budget of 18 kW.
- Switch to MaxQ mode as follows:

```
$ sudo nvsm set powermode=maxq
```

The settings are preserved across reboots.

7.1.2 MaxP

- Default mode that provides maximum performance
- GPUs operate unconstrained up to the thermal design power (TDP) level.

  In this setting, the maximum DGX-2 power consumption is 10 kW.
- Provides reduced but better performance than MaxQ when only 3 or 4 PSUs are working.
- If you switch to MaxQ mode, you can switch back to the default power mode (MaxP) as follows:

  ```
  $ sudo nvsm set powermode=maxp
  ```

  The settings are preserved across reboots.

7.1.3 Determining GPU Power Mode

Determine the GPU power mode as follows:

```
$ sudo nvsm show chassis/localhost
```

7.2 Managing the DGX Crash Dump Feature

Beginning with DGX OS Server 4.0.5, the Linux crash dump capability is enabled. The DGX OS includes a script to manage this feature.

7.2.1 Using the Script

- To enable dmesg crash dumps, enter

  ```
  /usr/sbin/dgx-kdump-config enable-dmesg-dump
  ```

  This option reserves memory for the crash kernel.
- To enable dmesg and vmcore crash dumps, enter

  ```
  /usr/sbin/dgx-kdump-config enable-vmcore-dump
  ```

  This option reserves memory for the crash kernel.
- To disable crash dumps, enter

  ```
  /usr/sbin/dgx-kdump-config disable
  ```

  This option disables the use of kdump and make sure no memory is reserved for the crash kernel.

7.2.2 Connecting to Serial Over LAN

While dumping vmcore, the BMC screen console goes blank approximately 11 minutes after the crash dump is started. To view the console output during the crash dump, connect to serial over LAN as follows:

```
$ ipmitool -I lanplus -H <bmc-ip-address> -U <BMC-USERNAME> -P <BMC-
PASSWORD> sol activate
```

7.3 Using PCIe Access Control Services

PCIe Access Control Services (ACS) is needed primarily if you are using the DGX-2 in KVM mode. When using the DGX-2 in bare-metal (non-KVM) mode, ACS affects GPUDirect performance and may cause InfiniBand failure.

NVIDIA enables/disables ACS according to whether the DGX-2 is in bare-metal or KVM mode as follows:

- Beginning with SBIOS version .18, the PCIe Access Control Services (ACS) is disabled by default.

  Since SBIOS updates do not over-write existing settings, the DGX-2 automatically disables ACS upon rebooting the system as part of the SBIOS update.
- If you are using the DGX-2 in KVM mode, ACS will be enabled automatically as part of the conversion from bare-metal to KVM host.
- When converting back to bare-metal mode from KVM mode and then rebooting, the DGX-2 automatically disables ACS.

7.4 Managing CPU Mitigations

The 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 installation of DGX systems 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 thereby increase performance. This capability is available starting with DGX OS 4.4 for Ubuntu, and EL7-20.02 for Red Hat Enterprise Linux.

This section provides instructions for Ubuntu. For instructions on managing CPU mitigations on Red Hat Enterprise Linux, refer to Managing CPU Mitigations for Red Hat Enterprise Linux.

7.4.1 Determining the CPU Mitigation State of the 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 if the output consists of multiple lines prefixed with `Mitigation:`.

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`.

Example:

```
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
```

7.4.2 Disabling 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 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 System for example output.

7.4.3 Re-enabling CPU Mitigations

1. Remove the `nv-mitigations-off` package.

```
$ sudo apt purge nv-mitigations-off
```

2. Reboot the system.
3. Verify CPU mitigations are enabled.

```
$ cat /sys/devices/system/cpu/vulnerabilities/*
```

The output should include several Mitigation lines. See Determining the CPU Mitigation State of the System for example output**.**

7.5 Using the DGX System in GPU Degraded Mode

If there is an issue with one or more GPUs, you can continue to use the system by excluding the faulty GPUs from use until there is an opportunity to service the system.

To exclude faulty GPUs and continue to use the DGX system in degraded mode, do the following.

1. If necessary, consult NVIDIA Enterprise Support to determine the failing GPU.
2. Query the GPU UUID using nvidia-smi.

```
$ nvidia-smi -q |egrep "GPU 00000000|GPU  UUID"
```

```
GPU 00000000:34:00.0 GPU
    UUID                    : GPU-8196613d-54af-3ef5-60e7-046d9a4783cf
GPU 00000000:36:00.0 GPU
    UUID                    : GPU-be8c9757-6874-2926-c5ac-366dc32147a4
.....
```

Example output

If you know the GPU index, you can also determine GPU UID by issuing the following.

```
$ nvidia-smi -i 3 -q | grep UUID
```

3. Mark a GPU for exclusion by adding the following line to **/etc/modprobe.d/ nvidia.conf**.

```
options nvidia NVreg_GpuBlacklist=<gpu-uuid>
```

4. Update initramfs.

   For Ubuntu

   ```
   $ sudo update-initramfs -u
   ```

   For Red Hat Enterprise Linux

   ```
   $ dracut --force /boot/initramfs-$(uname -r).img $(uname -r)
   ```

5. Reboot the system.

   ```
   $ sudo reboot
   ```

Chapter 8. Restoring the DGX-2 Software Image

If the **DGX-2** software image becomes corrupted (or both OS NVMe drives are replaced), restore the **DGX-2** software image to its original factory condition from a pristine copy of the image.

The process for restoring the **DGX-2** software image is as follows:

1. Obtain an ISO file that contains the image from NVIDIA Enterprise Support as explained in Obtaining the DGX-2 Software ISO Image and Checksum File.
2. Restore the **DGX-2** software image from this file either remotely through the BMC or locally from a bootable USB flash drive.
   - If you are restoring the image remotely, follow the instructions in Re-Imaging the System Remotely.
   - If you are restoring the image locally, prepare a bootable USB flash drive and restore the image from the USB flash drive as explained in the following topics:
     - Creating a Bootable Installation Medium
     - Re-Imaging the System From a USB Flash Drive

Note: The DGX OS Server software is restored on one of the two NMVe M.2 drives. When the system is booted after restoring the image, software RAID begins the process rebuilding the RAID 1 array - creating a mirror of (or resynchronizing) the drive containing the software. System performance may be affected during the RAID 1 rebuild process, which can take an hour to complete.

8.1 Obtaining the DGX-2 Software ISO Image and Checksum File

To ensure that you restore the latest available version of the **DGX-2** software image, obtain the current ISO image file from NVIDIA Enterprise Support. A checksum file is provided for the image to enable you to verify the bootable installation medium that you create from the image file.

1. Log on to the NVIDIA Enterprise Support site.
2. Click the **Announcements** tab to locate the download links for the DGX-2 software image.
3. Download the ISO image and its checksum file and save them to your local disk.

   The ISO image is also available in an archive file. If you download the archive file, be sure to extract the ISO image before proceeding.

8.2 Re-Imaging the System Remotely

These instructions describe how to re-image the system remotely through the BMC. For information about how to restore the system locally, see Re-Imaging the System from a USB Flash Drive.

Before re-imaging the system remotely, ensure that the correct **DGX-2** software image is saved to your local disk. For more information, see Obtaining the DGX-2 Software ISO Image and Checksum File.

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 the re-image ISO file and click **Open**.
   b). Click **Start Media**.
4. Reboot, install the image, and complete the DGX-2 system setup.
   a). From the top menu, click **Power** and then 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 and then come back online.
   c). At the boot selection screen, select **Install DGX Server**.

   If you are an advanced user who is not using the RAID disks as cache and want to keep data on the RAID disks, then select **Install DGX Server without formatting RAID.** See the section Retaining the RAID Partition While Installing the OS for more information.

d). Press **Enter**.

The DGX-2 System will reboot from ISO image and proceed to install the image. This can take approximately 15 minutes.

> **Note:** The Mellanox InfiniBand driver installation may take up to 10 minutes.

After the installation is completed, the system ejects the virtual CD and then reboots into the OS.

Refer to Setting Up the DGX-2 System for the steps to take when booting up the DGX-2 System for the first time after a fresh installation.

8.3 Creating a Bootable Installation Medium

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

> **Note:** If you are restoring the software image remotely through the BMC, you do not need a bootable installation medium and you can omit this task.

- If you are creating a bootable USB flash drive, follow the instructions for the platform that you are using:
  - On a text-only Linux distribution, 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, you can use any of the methods described in Burning the ISO on to a DVD on the Ubuntu Community Help Wiki.

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

On a Linux system, you can use the dd command to create a bootable USB flash drive that contains the **DGX-2** 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-2 software image is saved to your local disk. For more information, see Obtaining the DGX-2 Software ISO Image and Checksum File.
- The USB flash drive capacity is at least 4 GB.

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** command.

   ```
   lsblk
   ```

   You can identify the USB flash drive from its size.
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.

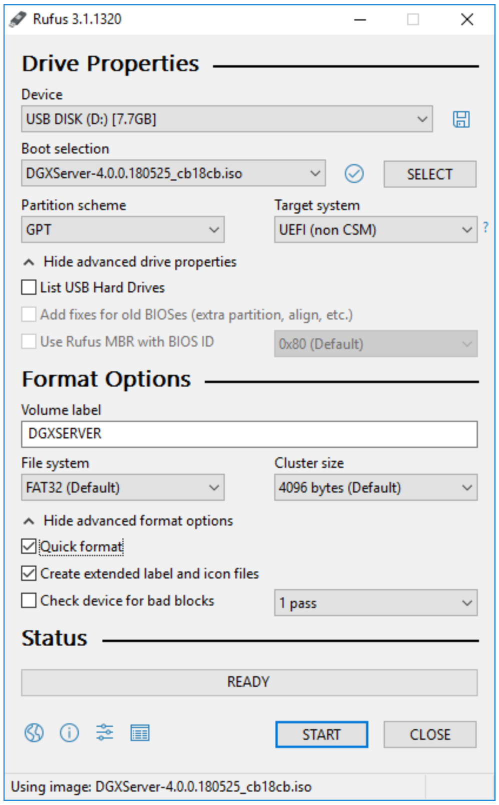
8.3.2 Creating a Bootable USB Flash Drive by Using Akeo Rufus

On a Windows system, you can use the Akeo Reliable USB Formatting Utility (Rufus) to create a bootable USB flash drive that contains the **DGX-2** software image.

Ensure that the following prerequisites are met:

- The correct **DGX-2** software image is saved to your local disk. For more information, see Obtaining the DGX-2 Software ISO Image and Checksum File.
- The USB flash drive has a capacity of at least 8 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).



Rufus 3.1.1320

Drive Properties

Device: USB DISK (D:) [7.7GB]

Boot selection: DGXServer-4.0.0.180525_cb18cb.iso

Partition scheme: GPT

Target system: UEFI (non CSM)

Hide advanced drive properties

☐ List USB Hard Drives

☐ Add fixes for old BIOSes (extra partition, align, etc.)

☐ Use Rufus MBR with BIOS ID: 0x80 (Default)

Format Options

Volume label: DGXSERVER

File system: FAT32 (Default)

Cluster size: 4096 bytes (Default)

Hide advanced format options

☑ Quick format

☑ Create extended label and icon files

☐ Check device for bad blocks: 1 pass

Status

READY

Using image: DGXServer-4.0.0.180525_cb18cb.iso

3. Under Device Properties, set the following options:

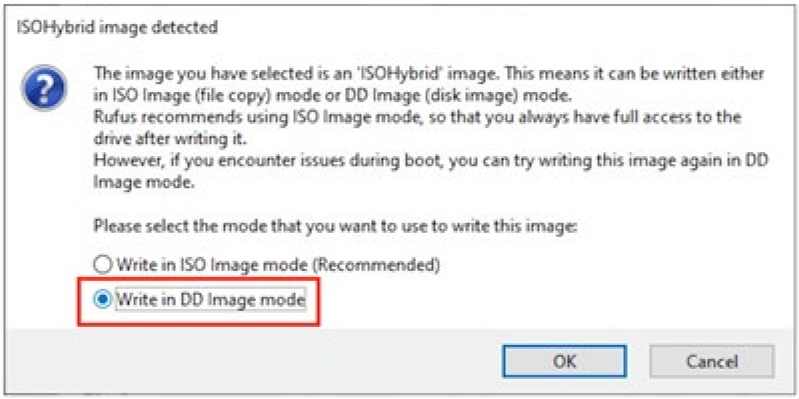
   e). In **Device**, select your USB flash drive.

   f). 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.** Because the image is a hybrid ISO file, you are prompted to select whether to write the image in ISO Image (file copy) mode or DD Image (disk image)

mode.



ISOHybrid image detected

The image you have selected is an 'ISOHybrid' image. This means it can be written either in ISO Image (file copy) mode or DD Image (disk image) mode.
Rufus recommends using ISO Image mode, so that you always have full access to the drive after writing it.
However, if you encounter issues during boot, you can try writing this image again in DD Image mode.

Please select the mode that you want to use to write this image:

○ Write in ISO Image mode (Recommended)

● Write in DD Image mode

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

8.4 Re-Imaging the System From a USB Flash Drive

These instructions describe how to re-image the system from a USB flash drive. For information about how to restore the system remotely, see Re-Imaging the System Remotely.

Before re-imaging the system from a USB flash drive, ensure that you have a bootable USB flash drive that contains the current **DGX-2** software image.

1. Plug the USB flash drive containing the OS image into the DGX-2 System.
2. Connect a monitor and keyboard directly to the DGX-2 System.
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. When the system boots up, select **Install DGX Server** on the startup screen.

   If you are an advanced user who is not using the RAID disks as cache and want to keep data on the RAID disks, then select **Install DGX Server without formatting RAID.** See the section Retaining the RAID Partition While Installing the OS for more information.
6. Press **Enter.**

   The DGX-2 System will reboot and proceed to install the image. This can take more than 15 minutes.

   Note: The Mellanox InfiniBand driver installation may take up to 10 minutes.

   After the installation is completed, the system then reboots into the OS.

Refer to Setting Up the DGX-2 System for the steps to take when booting up the DGX-2 System for the first time after a fresh installation.

8.5 Retaining the RAID Partition While Installing the OS

The re-imaging process creates a fresh installation of the DGX OS. During the OS installation or re-image process, you are presented with a boot menu when booting the installer image. The default selection is **Install DGX Software**. The installation process then repartitions all the SSDs, including the OS SSD as well as the RAID SSDs, and the RAID array is mounted as /raid. This overwrites any data or file systems that may exist on the OS disk as well as the RAID disks.

Since the RAID array on the DGX-2 System is intended to be used as a cache and not for long-term data storage, 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, then select the **Install DGX Server without formatting RAID** option at the boot menu during the boot installation. This option retains data on the RAID disks and performs the following:

- 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 configurations 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 at a later time by enabling `cachefilesd` and adding /`raid` to the file system table as follows:

1. Uncomment the #RUN=yes line in `/etc/default/cachefiled`.
2. Uncomment the /raid line in `etc/fstab`.
3. Run the following:
   a). Mount /raid.

   ```
   sudo mount /raid
   ```

   b). Start the cache daemon.

   ```
   systemctl start cachefilesd
   ```

   These changes are preserved across system reboots.

Chapter 9. Updating the DGX OS Software

You must register your DGX-2 System in order to receive email notification whenever a new software update is available.

These instructions explain how to update the DGX-2 software through an internet connection to the NVIDIA public repository. The process updates a DGX-2 System image to the latest QA'd versions of the entire DGX-2 software stack, including the drivers, for the latest update within a specific release; for example, to update to the latest Release 4.0 update from an earlier Release 4.0 version.

For instructions on ugrading from one Release to another (for example, from Release 3.1 to Release 4.1), consult the release notes for the target release.

9.1 Connectivity Requirements For Software Updates

Before attempting to perform the update, verify that the DGX-2 System network connection can access the public repositories and that the connection is not blocked by a firewall or proxy.

Enter the following on the DGX-2 System.

```
$ wget -O f1-changelogs http://changelogs.ubuntu.com/meta-release-lts
$ wget -O f2-archive
http://archive.ubuntu.com/ubuntu/dists/bionic/Release
$ wget -O f3-usarchive
http://us.archive.ubuntu.com/ubuntu/dists/bionic/Release
$ wget -O f4-security
http://security.ubuntu.com/ubuntu/dists/bionic/Release
$ wget -O f5-download
http://download.docker.com/linux/ubuntu/dists/bionic/Release
$ wget -O f6-international
http://international.download.nvidia.com/dgx/repos/bionic/dists/bionic/
Release
```

All the `wget` commands should be successful and there should be six files in the directory with non-zero content.

9.2 Update Instructions

> ! **CAUTION:** These instructions update 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 updated, 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.

Perform the updates using commands on the DGX-2 console.

1. Run the package manager.

   ```
   $ sudo apt update
   ```

2. Check to see which software will get updated.

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

   To prevent an application from being updated, instruct the Ubuntu package manager to keep the current version. See Introduction to Holding Packages.

3. Upgrade to the latest version.

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

   Answer any questions that appear.

   Most questions require a Yes or No response. If asked to select the grub configuration to use, select the current one on the system.

   Other questions will depend on what other packages were installed before the update and how those packages interact with the update. Typically, you can accept the default option when prompted.

4. Reboot the system.

Chapter 10. Updating Firmware

This section provides instructions for updating firmware for the NVIDIA® DGX server firmware using a Docker container.

The container supports

- NVIDIA DGX-2, starting with container `nvfw-dgx2_18.09.4`
- NVIDIA DGX-2H, starting with container `nvfw-dgx2:19.03.1.`

> **IMPORTANT:** DGX-2H is supported only with firmware container `nvfw-dgx2:19.03.1` or later. Do not update the DGX-2H firmware using an earlier container as this will result in version mismatch with the DGX-2H.

See the DGX-2 System Firmware Update Container Release Notes for information about each release.

For reference, the following naming scheme is used for the package, container image, and run file, depending on the FW update container version.

Component	Pre-19.03.1	19.03.1 and later
Tarball package	nvfw-dgx2_<version>.tar.gz	nvfw-dgx2_<version>.tar.gz
Container Image	nvfw-dgx2_<version>	nvfw-dgx2:<version>
Run file	N/A	nvfw-dgx2_<version>.run

10.1 General Firmware Update Guidelines

- Before updating the firmware, do the following to prevent corrupting the firmware due to a system crash or disruption to the update process.
  - Ensure the system is healthy
  - Stop system activities

> **Caution:** Stop all unnecessary system activities before attempting to update firmware, and do not add additional processing loads while an update is in

> progress. A high workload can disrupt the firmware update process and result in an unusable component.
>
> When initiating an update, the update software assists in determining the activity state of the DGX system and provides a warning if it detects that activity levels are above a predetermined threshold. If the warning is encountered, you are strongly advised to take action to reduce the workload before proceeding with the update.

- **Do not** terminate the firmware update console, close the browser, or shut down the system while updating the firmware.

  Component firmware corruption may occur if the update process is interrupted.
- Certain components, such as the system BIOS, require a system reboot for the new firmware to take effect.

  Reboot the system if prompted.
- When updating the BMC firmware, system management services are shut down first to allow the update to occur. Consequently, system management is not available during the BMC update.
- In the event of a firmware update failure, run **`nvsm dump health`** and then send the resulting archive containing the output to NVIDIA Enterprise Support (https://nvid.nvidia.com/dashboard/) for failure analysis.

  Do not attempt any further firmware updates until the issue is resolved or cleared by NVIDIA Enterprise Support.

10.2 Obtaining the Firmware Update Container

You can obtain the firmware update container in two ways.

- As a direct download from nvcr.io.

  This method requires that DGX OS 4.4 or later is installed.
- As a tarball obtained from NVIDIA Enterprise Support

To download from nvcr.io, issue the following.

```
$ sudo -E nvsm update firmware
```

Select the appropriate org and repo from the prompts, then enter **`Y`** to pull the container.

To use the tarball if DGX OS 4.3 or earlier is installed, perform the following.

1. Obtain the container tarball from the NVIDIA Enterprise Support portal and transfer it to the DGX-2 System.

The container is provided in the tarball <package-name>.tar.gz.

Beginning with container version 19.03.1, the container is also available from the NVIDIA Enterprise Support portal as a .run file <*run-file-name*>.run. See the section Using the .run File for instructions.

2. From the directory where you copied the tarball file, enter the following command.

```
$ sudo docker load -i <package-name>.tar.gz
```

3. To verify that the container image is loaded, enter the following.

```
$ sudo docker images
```

Example output after loading **nvfw-dgx2_18.09.3.tar.gz**.

```
REPOSITORY         TAG       IMAGE ID     CREATED       SIZE
nvfw-dgx2_18.09.3  latest    aa681a4ae600 1 hours ago   278MB
```

Starting with version 19.03.1, the container naming format has changed from nvfw_dgx2_version to nvfw_dgx2:tag, where tag indicates the version.

Example output after loading **nvfw-dgx2_19.03.1.tar.gz**.

```
REPOSITORY      TAG       IMAGE ID     CREATED       SIZE
nvfw-dgx2       19.03.1   fec80ce658ef 1 hours ago   532MB
```

10.3 Querying the Firmware Manifest

The manifest displays a listing of firmware components embedded in the containers that are qualified by NVIDIA.

To query the firmware manifest, enter the following:

```
# sudo docker run --rm --privileged -ti -v /:/hostfs <image-name>
show_fw_manifest
```

10.4 Querying the Currently Installed Firmware Versions

Display the onboard firmware version level of each component supported by the container. The output will show which component firmware is up to date, or whether it needs to be updated to the firmware level listed in the manifest.

To query the version information, enter the following.

```
# sudo docker run --privileged -v /:/hostfs <image-name> show_version
```

The output shows the onboard version, the version in the manifest, and whether the firmware is up to date.

10.5 Updating the Firmware

You can either update all the down-level firmware components at one time, or update just one or more components.

10.5.1 Command Syntax

```
sudo docker run --rm [-e auto=1] --privileged -ti -v /:/hostfs <image-
name> update_fw [-f] <target>
```

Where `<target>` specifies the hardware to update, and is either

all

to update all firmware components (SBIOS, BMC)

or one or more of the following:

SBIOS

to update the SBIOS

BMC

to update the BMC firmware

> **Note:** Other components may be supported beyond those listed here. Query the firmware manifest to see all the components supported by the container.

The command will scan the specified firmware components and update any that are down-level.

See the section Additional Options for an explanation of the `[-e auto=1]` and `[-f]` options.

10.5.2 Updating All Firmware components

The following instructions are an example of attempting to update all the firmware components using the container `nvfw-dgx2:19.03.1`. In this example, the SBIOS and BMC require an update.

1. Enter the following.

   ```
   $ sudo docker run --rm --privileged -ti -v /:/hostfs nvfw-
   dgx2:19.03.1 update_fw all
   ```

The container will scan the components and then prompt for confirmation before starting the update.

```
Following components will be updated with new firmware version:
 SBIOS
 BMC
IMPORTANT: Firmware update is disruptive and may require system
reboot.
Stop system activities before performing the update.
Ok to proceed with firmware update? <Y/N>
```

2. Press **Y** to proceed.

   The firmware update progress is displayed for each component.

   > **Note**: While the progress output shows the current and manifest firmware versions, the versions may be truncated due to space limitations. You can confirm the updated version after the update is completed using the show_version option.

   When the update completes successfully, the following message is displayed.

   ```
   Firmware update completed Component: SBIOS, update status: success,
   reboot required: yes
   Component: BMC, update status: success, new version: 3.20.30
   ```

3. If directed by the update message, reboot the system.

10.5.3 Updating Specific Firmware Components

Note: Be sure to consult the NVIDIA DGX-2 Firmware Update Container release notes for special instructions applicable to specific firmware versions.

The following is an example of updating the SBIOS firmware using the container `nvfw-dgx2_19.03.1`.

1. Enter the following.

```
$ sudo docker run --rm --privileged -ti -v /:/hostfs nvfw-
dgx2:19.03.1 update_fw SBIOS
```

   The container will scan the components and then prompt for confirmation before starting the update.

```
Following components will be updated with new firmware version:
IMPORTANT: Firmware update is disruptive and may require system
reboot.
Stop system activities before performing the update.
Ok to proceed with firmware update? <Y/N>
```

2. Press **`Y`** to proceed. When the update completes successfully, the following message is displayed.

```
Firmware update completed
Component: SBIOS, update status: success, reboot required: yes
```

You can also update a subset of all the components. For example, to update both the BMC firmware and the system BIOS, enter the following:

```
$ sudo docker run --rm --privileged -ti -v /:/hostfs nvfw-dgx2:19.03.1
update_fw BMC SBIOS
```

10.6 Additional Options

10.6.1 Forcing the Firmware Update

To update the firmware regardless of whether it is down-level, use the `-f` option as follows.

```
$ sudo docker run --rm --privileged -ti -v /:/hostfs <image-name>
update_fw -f <target>
```

The container will not check the onboard versions against the manifest.

10.6.2 Updating the Firmware Non-interactively

The standard way to run the container is interactively (`-ti` option). The container will prompt you to confirm before initiating the update.

To update the firmware without encountering the prompt, omit the **`-ti`** option and instead use the **`-e auto=1`** and **`-t`** options as follows.

```
$ sudo docker run -e auto=1 --rm --privileged -t -v /:/hostfs <image-
name> update_fw <target>
```

10.7 Command Summary

- Show the manifest.

  ```
  $ sudo docker run --rm --privileged -v /:/hostfs <image-name>
  show_fw_manifest
  ```

- Show version information.

  ```
  $ sudo docker run --rm --privileged -v /:/hostfs <image-name>
  show_version
  ```

- Check the onboard firmware against the manifest and update any down-level firmware.

  ```
  $ sudo docker run --rm --privileged -ti -v /:/hostfs <image-name>
  update_fw <target>
  ```

- Bypass the version check and update the firmware.

  ```
  $ sudo docker run --rm --privileged -ti -v /:/hostfs <image-name>
  update_fw -f <target>
  ```

- Update the firmware in non-interactive mode.

  ```
  $ sudo docker run --rm -e auto=1 --privileged -t -v /:/hostfs <image-
  name> update_fw <target>
  ```

10.8 Removing the Container

Remove the container and image from the DGX server when it is no longer needed. To remove the container and image, enter the following:

```
$ sudo docker rmi -f <image-name>
```

In this case, specify only the container repository and not the tag.

10.9 Using the .run File

Beginning with the firmware container version 19.03.1, a .run file is also available to run the firmware update container. The .run file is a self-extracting package embedding the firmware update container tarball. Using the .run file requires DGX OS Server 4.0.5 or later.

1. Before using, you need to make the file executable as follows:

   ```
   $ chmod +x /<run-file-name>.run
   ```

2. Run the file.

   ```
   $ sudo ./<run-file-name>.run
   ```

   This command is the same as running the container with the `update_fw all` option.

The .run file accepts the same options that are used when running the container.

Examples:

- Show the manifest.

  ```
  $ sudo ./<run-file-name>.run show_fw_manifest
  ```

- Show version information.

  ```
  $ sudo ./<run-file-name>.run show_version
  ```

- Check the onboard firmware against the manifest and update any down-level firmware.

  ```
  $ sudo ./<run-file-name>.run update_fw <target>
  ```

- Bypass the version check and update the firmware.

  ```
  $ sudo ./<run-file-name>.run update_fw -f <target>
  ```

10.10 Updating Secondary Firmware Images

Some firmware components provide a secondary image as a backup. This section describes the instructions for updating the secondary image for the system BIOS and the BMC. Consult the release notes for policies applicable to specific versions of the firmware update container.

10.10.1 Updating the Secondary BMC

To update the secondary BMC image, include the `--update-backup-bmc` option in the update command.

> **Note:** The `--update-backup-bmc` option is available only with firmware update container version 19.12.1 and later.

Example:

```
$ sudo docker run --rm --privileged -ti -v /:/hostfs nvfw-dgx2:19.12.1
update_fw BMC --update-backup-bmc
```

10.10.2 Updating the Secondary SBIOS

> **Note:** The ability to update the secondary SBIOS using the firmware update container is available only with firmware update container version 19.12.1 and later.

1. Update the active SBIOS using the firmware update container.
2. Designate booting from the secondary (inactive) SBIOS on the next boot.

   ```
   $ sudo ./<run-file-name>.run sbios_slot --switch-nextboot-slot
   ```

3. Reboot the DGX-2 to switch to the secondary SBIOS.

   ```
   $ telinit 1
   $ umount /raid
   $ sync
   $ ipmitool chassis power cycle
   ```

4. Update the secondary (now active) SBIOS.
5. Designate booting from the primary SBIOS on the next boot (to restore the primary SBIOS as the active SBIOS).

   ```
   $ sudo ./nvfw-dgx2_19.12.1_191204.run sbios_slot --switch-nextboot-
   slot
   ```

6. Reboot the DGX-2 to switch back to the primary SBIOS.

   ```
   $ telinit 1
   $ umount /raid
   $ sync
   $ ipmitool chassis power cycle
   ```

10.11 Troubleshooting

10.11.1 Redundant PSU fails to update

The system is still operational with only five of the six PSUs working, but the firmware update may fail.

Make sure all PSUs are fully inserted and that power cords to all PSUs are fully inserted and secured. If the firmware update still fails, then run **`nvsm dump health`** and send the resulting archive containing the output to NVIDIA Enterprise Support (https://nvid.nvidia.com/dashboard/) for failure analysis.

Do not attempt any further firmware updates until the issue is resolved or cleared by NVIDIA Enterprise Support.

Chapter 11. Using the BMC

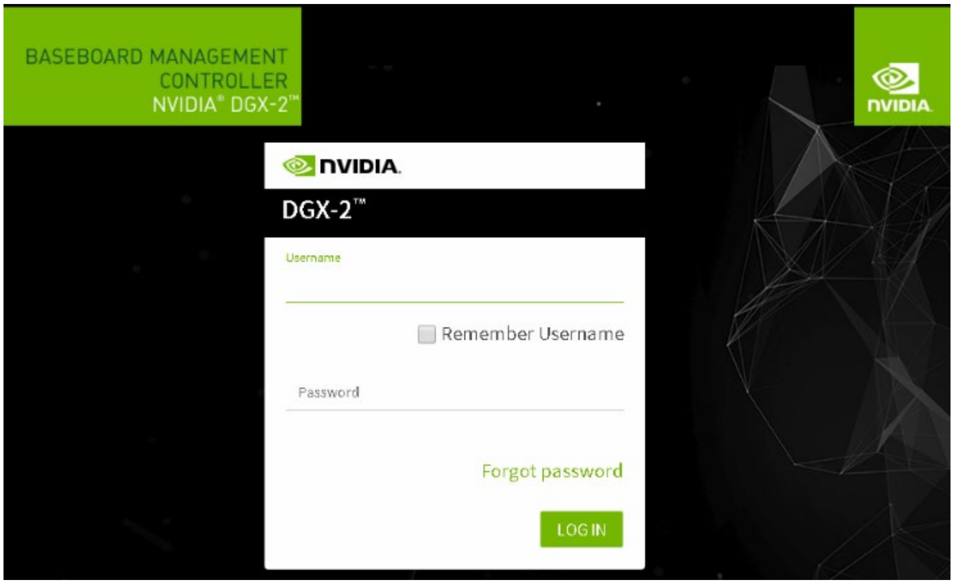
The NVIDIA DGX-2 System comes with a baseboard management controller (BMC) for monitoring and controlling various hardware devices on the system. It monitors system sensors and other parameters.

11.1 Connecting to the BMC

1. Make sure you have connected the BMC port on the DGX-2 System to your LAN.
2. Open a browser within your LAN and go to:

   ```
   https://<bmc-ip-address>/
   ```

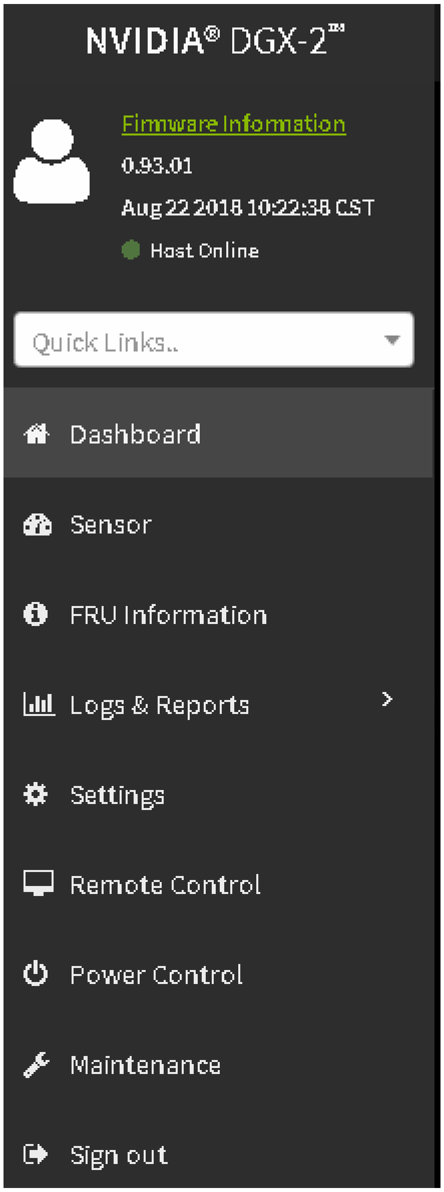
   Make sure popups are allowed for the BMC address.



3. Log in.

11.2 Overview of BMC Controls

The left-side navigation menu on the BMC dashboard contains the primary controls.



11.2.1 QuickLinks ...

Provides quick access to several tasks.

Note: Depending on the BMC firmware version, the following quick links may appear:

- Maintenance->Firmware Update
- Settings->NbMeManagement->NvMe P3700Vpd Info

Do not access these tasks using the Quick Links dropdown menu, as the resulting pages are not fully functional.

11.2.2 Sensor

Provides status and readings for system sensors, such as SSD, PSUs, voltages, CPU temperatures, DIMM temperatures, and fan speeds.

11.2.3 FRU Information

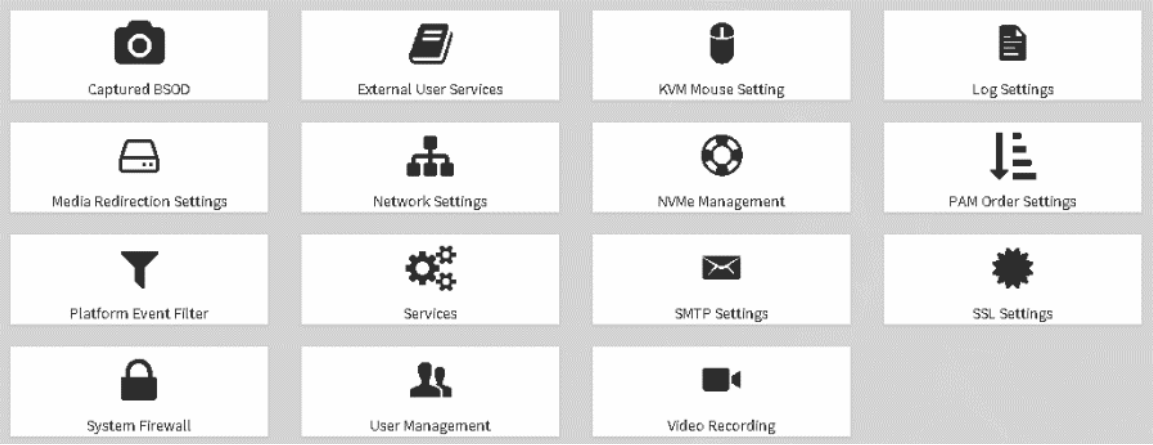
Provides, chassis, board, and product information for each FRU device.

11.2.4 Logs & Reports

View, and if applicable, download and erase, the IPMI event log, and system, audit, video and POST Code logs.

11.2.5 Settings

Configure the following settings

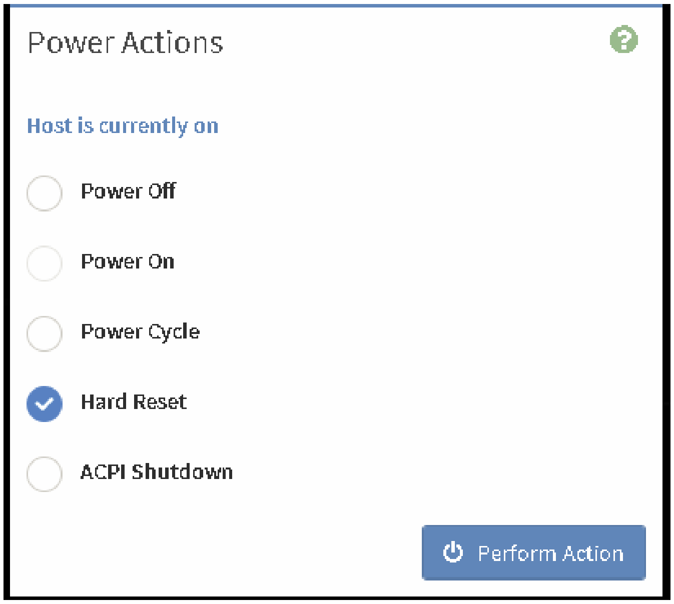


11.2.6 Remote Control

Opens the KVM Launch page for accessing the DGX-2 console remotely.

11.2.7 Power Control

Perform various power actions



Power Actions

Host is currently on

○ Power Off

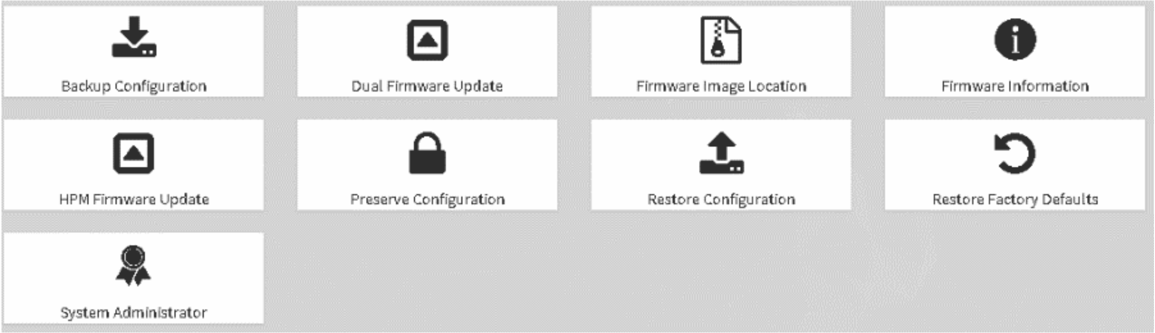
○ Power On

○ Power Cycle

☑ Hard Reset

○ ACPI Shutdown

11.2.8 Maintenance



IMPORTANT: While you can update the BMC firmware from this page, NVIDIA recommends using the NVIDIA Firmware Update Container instead (see section Updating Firmware for instructions).

Do not update from versions earlier than 01.04.02 using the BMC UI, as the sensor data record (SDR) is erroneously preserved which can result in the BMC UI reporting a critical 3V Battery sensor error. This is corrected in version 1.0.4.02 - updating from 1.04.02 does not preserve the SDR.

If you need to update from this page, click **Dual Firmware Update** and then select whichever is the **Current Active Image** to update.

11.3 Creating a Unique BMC Password

When you set up the DGX-2 upon powering it on for the first time, you set up a username and password for the system. These credentials are also used to log in to the BMC remotely, except that the BMC password is the username.

It is strongly recommended that you create a unique password as soon as possible.

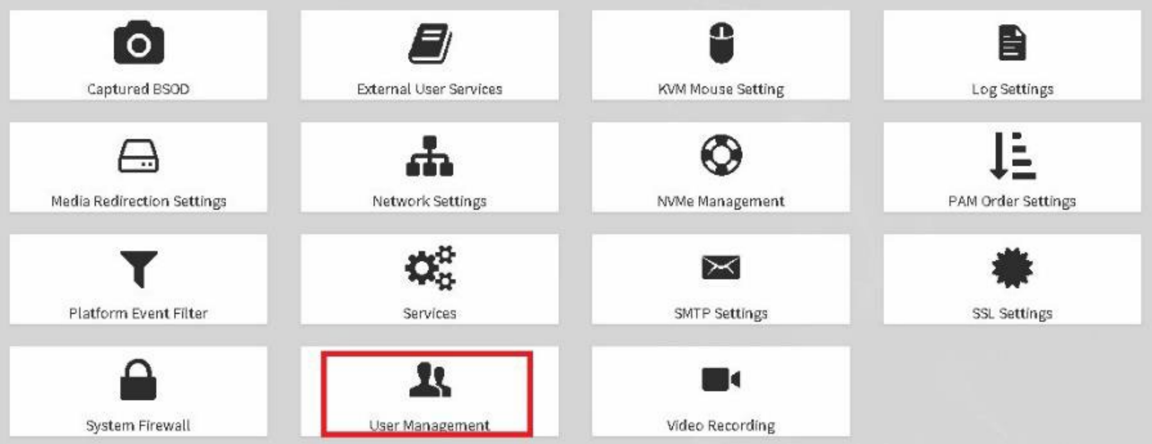
Create a unique BMC password as follows:

1. Log into the BMC.
   a). Open a browser within your LAN and go to `https://`*`<bmc-ip-address>`*`/`.
   b). Log in with the username that you created when you first set up the DGX-2.

   Enter your username for both the User ID as well as the password:

   User ID: <your username>

   Password: <your username>
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 create a strong password.
5. Log out and then log back in with the new password.

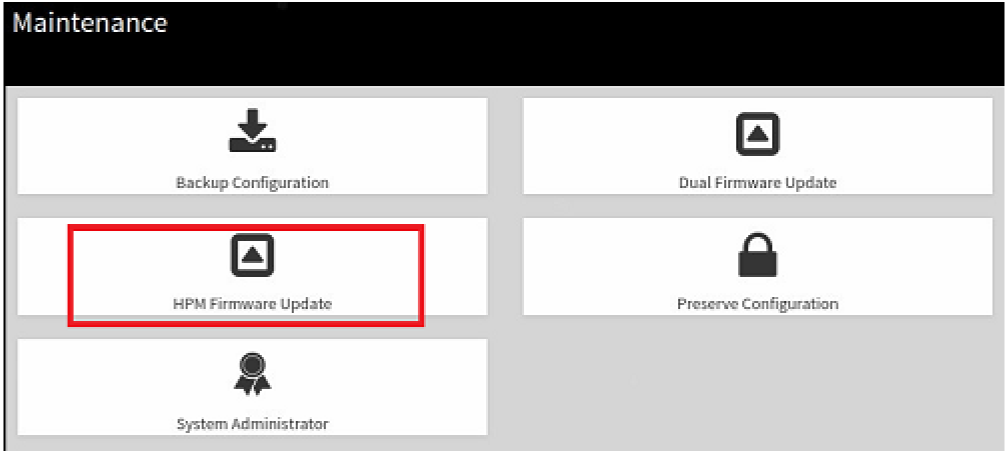
11.4 Updating the SBIOS

These instructions describe how to update the SBIOS from the BMC dashboard. *These instructions should be followed only under special circumstances*, such as

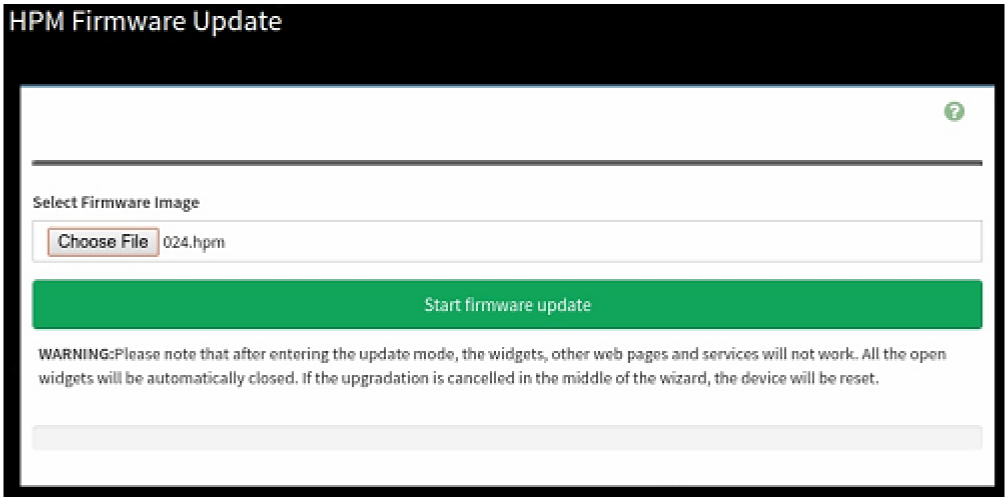
- When updating from SBIOS version 0.13 or 0.17
- When the SBIOS is corrupted and cannot be flashed using the firmware update container.

This process should take less than ten minutes and updates the inactive SBIOS.

1. Obtain the SBIOS `.hpm` file from the NVIDIA Enterprise Support announcement and copy it to your local machine.
2. Remove the DGX-2 from production to ensure against corrupting the BMC.
3. Log in to the BMC dashboard from your local machine and select **Maintenance** from the left-side navigation pane.
4. Select the **HPM Firmware Update** card from the list.



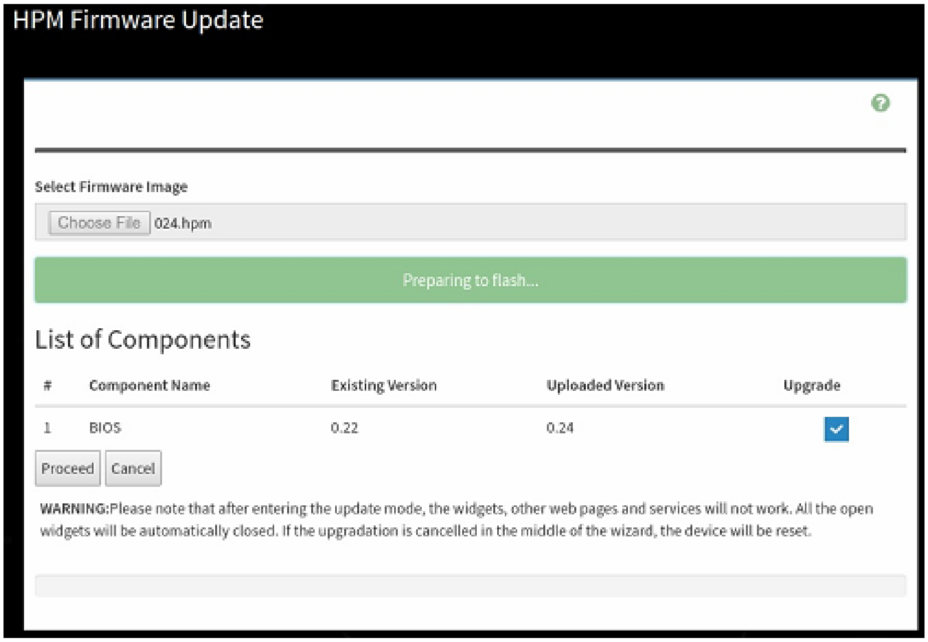
5. Click **Choose File**, and then locate and select the `.hpm` file corresponding to the update version.



WARNING:Please note that after entering the update mode, the widgets, other web pages and services will not work. All the open widgets will be automatically closed. If the upgradation is cancelled in the middle of the wizard, the device will be reset.

6. Click Start firmware update.

7. Under List of Components, confirm that the Uploaded Version is the intended version, then click **Proceed**.



024.hpm

List of Components

#	Component Name	Existing Version	Uploaded Version	Upgrade
1	BIOS	0.22	0.24	☑

WARNING:Please note that after entering the update mode, the widgets, other web pages and services will not work. All the open widgets will be automatically closed. If the upgradation is cancelled in the middle of the wizard, the device will be reset.

8. Click **OK** at the confirmation dialog. The progress bar shows the update progress.
9. Click **Cancel** at the Firmware update completed dialog box and perform a clean power cycle of the system as follows.
   a). Issue the following on the OS command line to perform a clean shutdown.

   ```
   $ telinit 1
   $ umount /raid
   $ sync
   $ ipmitool chassis power off
   ```

   b). After the shutdown, remove all AC cables from the DGX-2 and wait for ten minutes.

   c). Re-connect the AC cables, then push the power button to power on the DGX-2.

   The system reboots to the now updated secondary SBIOS (assuming it originally booted from the primary SBIOS).

10. Repeat the steps to update the primary SBIOS (assuming it originally booted from the primary SBIOS).
11. To verify the state of each SBIOS, log in to the BMC dashboard, select **Maintenance**, then select **Firmware Information** and view the information under the BIOS section.

Chapter 12. Using DGX-2 System in KVM Mode

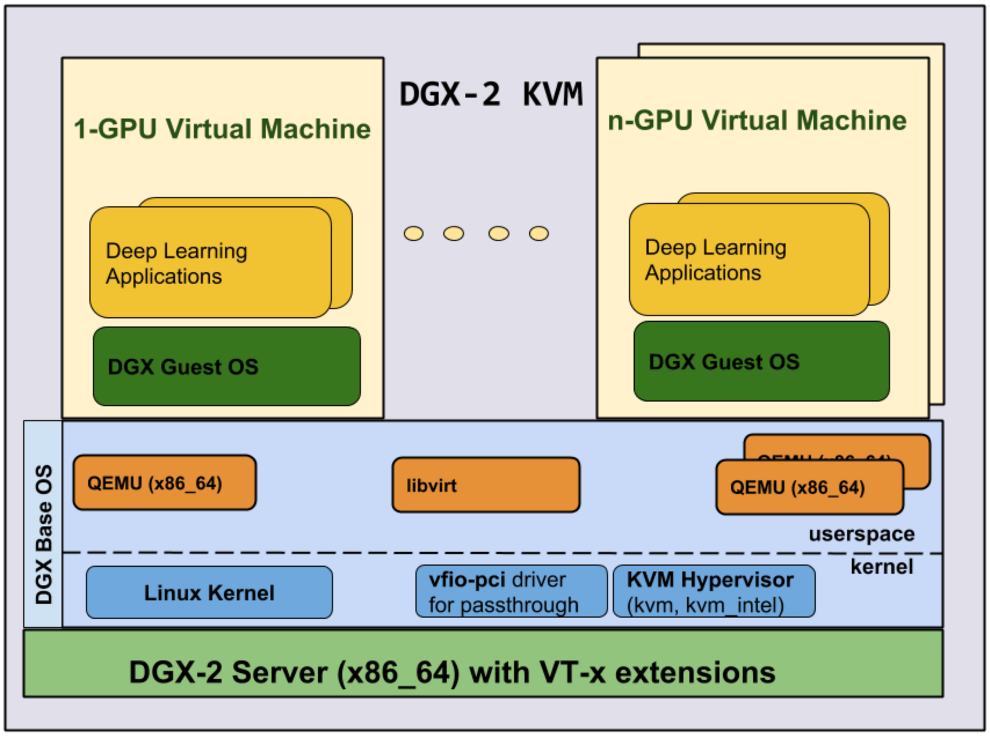
12.1 Overview

12.1.1 About NVIDIA KVM

The NVIDIA Kernel-based Virtual Machine (KVM) is a virtualization solution based on the Linux Kernel Virtual Machine (https://www.linux-kvm.org) and enhanced to enable GPU multi-tenancy. Since the KVM Hypervisor is part of the Linux kernel on the DGX-2 system, it contains the system-level components necessary to support multi-tenancy on the DGX-2 system, such as a memory manager, process scheduler, input/output (I/O) stack, device drivers, security manager, and a network stack.

Note: NVIDIA KVM is also supported on the NVIDIA DGX-2H. References to DGX-2 in this chapter also apply to DGX-2H.

The following diagram depicts an overview of the NVIDIA KVM architecture, showing the hardware layer, the DGX Server KVM OS, and the virtual machines.


DGX-2 KVM
1-GPU Virtual Machine
Deep Learning Applications
DGX Guest OS
n-GPU Virtual Machine
Deep Learning Applications
DGX Guest OS
DGX Base OS
QEMU (x86_64)
libvirt
QEMU (x86_64)
userspace
kernel
Linux Kernel
vfio-pci driver for passthrough
KVM Hypervisor (kvm, kvm_intel)
DGX-2 Server (x86_64) with VT-x extensions

Using NVIDIA KVM, the DGX-2 system can be converted to include a bare metal hypervisor to provide GPU multi-tenant virtualization. This is referred to as the *DGX-2 KVM host*. It allows different users to run concurrent deep learning jobs using multiple virtual machines (*guest GPU VMs*) within a single DGX-2 system. Just like the bare-metal DGX-2 system, each GPU-enabled VM contains a DGX OS software image which includes NVIDIA drivers, CUDA, the NVIDIA Container Runtime for Docker, and other software components for running deep learning containers.

Note: Unlike the-bare metal DGX-2 system or the KVM host OS, the guest VM OS is configured for English-only with no option to switch to languages such as Chinese. To set up a guest VM for a different language, install the appropriate language pack onto the guest VM.

Example of installing a Chinese language pack:

```
Guest-vm-2g4-5:~$ sudo apt install language-pack-zh-hant
language-pack-zh-hans  language-pack-zh-hans-base language-
pack-zh-hant-base
```

Running NVIDIA containers on the VM is just like running containers on a DGX-2 bare metal system with DGX OS software installed.

While NVIDIA KVM turns your DGX system into a hypervisor supporting multiple guest GPU VMs, it does not currently provide support for the following:

- oVirt, virt-manager

The DGX-2 OS incorporates Ubuntu server, which does not include a graphics manager required by oVirt and virt-manager.

- Orchestration/resource manager

  Created GPU VMs are static and cannot be altered once created.

- NVMe drives as pass-through devices

  To preserve the existing RAID configuration on the DGX-2 System and simplify the process of reusing this resource if the server were ever to be reverted from KVM, NVMe drives should not be up as pass-through devices. However, if you want to use NVMe as pass-through devices for performance reasons, refer to the KVM Performance Tuning section of the DGX Best Practices guide for instructions.

- The DGX-2 KVM host cannot be used to run containers.
- NVIDIA GPUDirect™ is not supported on multi-GPU guest VMs across InfiniBand.
- There is no guest UEFI BIOS support.

12.1.2 About the Guest GPU VM (Features and Limitations)

- Guest GPU VMs are based on an installed KVM image.
- Guest GPU VM size and resources are based on the number of GPUs assigned
- Once a GPU VM is created and resources assigned, reconfiguring the VM (adding or removing GPUs, modifying other resource allocations) is not supported.
- Access to the hardware is restricted from within the guest GPU VM such that
  - GPUs cannot be reset
  - GPU VBIOS cannot be updated
  - System firmware upgrade is not supported

12.1.3 About nvidia-vm

Guest GPU VMs can be managed using the **virsh** (see https://linux.die.net/man/1/virsh) program or using libvirt-based XML templates. For the NVIDIA KVM, NVIDIA has taken the most common virsh options and configuration steps and incorporated them into the tool **`nvidia-vm`**, provided with the DGX KVM package. **`nvidia-vm`** simplifies the process of creating guest GPU VMs and allocating resources. In addition, you can use **`nvidia-vm`** to modify default options to suit your needs for the VM and manage VM images installed on the system.

To view the top-level help, enter the following.

```
sudo nvidia-vm --help
```

You can view the man pages by entering the following from the DGX-2 KVM host.

```
man nvidia-vm
```

Details of basic commands are provided in the following sections.

> **Note:** Using nvidia-vm requires root or sudo privilege. This includes deleting VMs, running health-check, or other operations.

12.2 Converting the DGX-2 System to a DGX-2 KVM Host

To operate VMs from the DGX-2 System, you must first convert the DGX-2 System to a DGX-2 KVM host. Do this by installing the DGX KVM Software package and the DGX KVM image. Perform the following steps on the command line of the DGX-2 System.

1. Update the package list.

   ```
   sudo apt-get update
   ```

2. If you have not already installed the update packages, then perform the following to be sure to get the latest KVM packages.

   ```
   sudo apt install -y dgx-bionic-updates-repo
   ```

3. Get the latest updates.

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

4. Check available DGX KVM images.

   ```
   sudo apt-cache policy dgx-kvm-image*
   ```

   This returns a list of images in the repository.

5. Install the **dgx-kvm-sw** package as well as one of the images listed in the previous step.

   ```
   sudo apt-get install dgx-kvm-sw <dgx-kvm-image-x-y-z>
   ```

   This step updates the GRUB menu options so the Linux kernel is made KVM-ready, and binds the virtualization drivers to the NVIDIA devices. It also creates the GPU health database.

   Example of selecting image `dgx-kvm-image-4-1-1`:

   ```
   sudo apt-get install dgx-kvm-sw dgx-kvm-image-4-1-1
   ```

   See the section Installing Images for more information about installing KVM guest images, including how to view the image contents.

6. Reboot the system.

   Rebooting the system is needed to finalize the KVM preparation of the DGX-2 System.

   `sudo reboot`.

   Your DGX-2 system is now ready for you to create VMs.

12.2.1 Getting Updated KVM Packages

You can obtain the latest KVM packages, including any updates to KVM images by performing the following from the KVM host.

1. Update the package list.

   ```
   sudo apt-get update
   ```

2. If you have not already installed the update packages, then perform the following to be sure to get the latest KVM packages.

   ```
   sudo apt install -y dgx-bionic-updates-repo
   ```

3. Get the latest updates.

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

12.2.2 Restoring to Bare Metal

After setting up the DGX-2 System as a KVM host, you can restore the server to a bare metal system.

> ! **CAUTION:** Reverting the server back to a bare metal system destroys all guest GPU VMs that were created as well as any data. Be sure to save your data before removing the KVM software.

To restore the DGX-2 System to a bare metal system, do the following.

1. Remove the meta package and all its dependencies.

   ```
   sudo apt-get purge --auto-remove dgx-kvm-sw
   ```

2. Reboot the system.

   ```
   sudo reboot
   ```

12.3 Launching a Guest GPU VM Instance

To create and delete guest GPU VMs, use the NVIDIA utility `nvidia-vm` which simplifies the complex process of these tasks.

12.3.1 Determining the Guest GPU VMs on the DGX-2 System

GPUs cannot be assigned to more than one VM. Therefore, before you can create a VM that uses one or more GPUs, you must determine the number and position of the GPUs that are already allocated to VMs.

Run the following command.

```
sudo nvidia-vm list
```

The domain of each guest GPU VM is either based on the username of the VM creator appended with a timestamp, or is specified by the VM creator. The domain is then appended with a suffix to indicate the number of GPUs and their indices using the format

`<`*`number-of-gpus`*`>g<`*`starting-index`*`>-<`*`ending index`*`>`.

Examples:

`my-lab-vm1-8g0-7` : This VM is assigned 8 GPUs from index 0 through 7

`my-lab-vm2-1g0` : This VM is assigned 1 GPU from index 0

`my-lab-vm3-4g8-11` : This VM is assigned 4 GPUs from index 8 through 11

Inspect the list to determine the GPU indices that are available to you.

12.3.2 Creating a VM Using Available GPUs

Use **`nvidia-vm`** as explained in About nvidia-vm.

Syntax

```
sudo nvidia-vm create --gpucount N --gpu-index X [--image]
[options]
```

where

`--gpu-count`	The allowed number of GPUs to assign to the VM, depending on availability. Acceptable values: **1, 2, 4, 8, 16** A value of 0 is not supported, and can result in unexpected behavior.
`--gpu-index`	For the purposes of the KVM, GPUs on the DGX-2 System are distinguished by a zero-based, sequential index. `gpu_index` specifies the

<table>
<tr><td></td><td>starting index value for the group of sequentially indexed GPUs to be assigned to the VM.
Allowed values for <code>gpu_index</code> depend on the number of GPUs assigned to the VM, as shown in the following table.
<table>
<tr><th>Number of GPUs</th><th>Allowed values for gpu_index</th></tr>
<tr><td>1</td><td>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15</td></tr>
<tr><td>2</td><td>0,2,4,6,8,10,12,14</td></tr>
<tr><td>4</td><td>0,4,8,12</td></tr>
<tr><td>8</td><td>0,8</td></tr>
<tr><td>16</td><td>0</td></tr>
</table>
</td></tr>
<tr><td><code>--image</code></td><td>(Optional) Specifies the KVM image to use as the basis for the VM. If not specified, the latest version that is installed will be used. See the section Managing the Images for instructions on how to install images and also how to view which images are installed.</td></tr>
<tr><td><code>--user-data</code></td><td>(Optional) Starting with dgx-kvm-sw 4.1.1, you can use cloud-init by specifying the cloud-config file containing setup parameters for the VM. See section Using cloud-init to Initialize the Guest VM.</td></tr>
<tr><td><code>--meta-data</code></td><td>(Optional) Starting with dgx-kvm-sw 4.1.1, you can use cloud-init to specify the meta-data file containing the meta-data that you want to include in your VM. See section Using cloud-init to Initialize the Guest VM.</td></tr>
<tr><td><code>[options]</code></td><td>Optional parameters, including options to customize the default resource allocation (vCPUs, memory, OS disk size):
See the man pages or the Help for a detailed list of options.</td></tr>
</table>

Command Help:

```
[sudo] nvidia-vm create --help
```

This command does not require "`sudo`"; however, using `sudo` affects the VM name as described in this section.

Command Examples:

- Basic command

```
sudo nvidia-vm create --gpu-count 4 --gpu-index 12
```

This command creates a guest GPU VM with 4 GPUs, starting with index 12.

Since no domain was specified, the software generates a domain, the name of which depends on whether `sudo` is used.

- If using `sudo`, then the domain incorporates "root", day, hour, and minute.

  For example, **`rootTue1308-4g12-15`**
- If not using `sudo`, then the domain incorporates the username, day, hour, and minute.

  For example, **`jsmithTue1308-4g12-15`**

▶ Specifying a domain

```
sudo nvidia-vm create --gpu-count 2 --gpu-index 8 --domain
mydgx2vm
```

This command creates a VM with 2 GPUs, starting with index 8, named **`mydgx2vm-2g8-9`**.

> **IMPORTANT:** A value of **0x0** for the domain name is not supported.

▶ Specifying an image

```
sudo nvidia-vm create --gpu-count 2 --gpu-index 2 --image dgx-
kvm-image-4-1-1
```

This command creates a VM with 2 GPUs, starting with index 2, named **`rootTue1308-2g2-3`**, and based on the image `dgx-kvm-image-4-1-1`.

> **Note:** If you encounter the following message when creating a VM,
>
> ```
> Error setting up logfile: No write access to directory
> /home/$USER/.cache/virt-manager
> ```
>
> remove the `/home/$USER/.cache/virt-manager` directory and then create the VM again.

12.3.2.1 Using cloud-init to Initialize the Guest VM

The NVIDIA DGX KVM software installs cloud-init in guest VMs. Cloud-init is a tool for automating the initial setup of VMs such as configuring the host name, network interfaces, and authority keys.

For more information about using cloud-init,
see https://cloudinit.readthedocs.io/en/latest/index.html.

Using cloud-init involves creating two configuration files - cloud-config and instance-data.json - and then calling them when creating the guest VM using the following options:

- `--user-data <cloud-config>`
- `--meta-data <meta-data file>`

Example:

```
$ nvidia-vm create --gpu-count <#> --verbose --user-data
/home/lab/cloud-config --meta-data /home/lab/instance-data.json
```

See Using Cloud-init for details on setting up the files.

12.4 Stopping, Restarting, and Deleting a Guest GPU VM

Once a guest GPU is created, it can be stopped if you want to temporarily free resources while keeping your data. You can then restart the stopped guest GPU VM. You can also permanently delete a guest GPU VM, which frees resources and deletes associated data.

12.4.1 Shutting Down a VM

You can perform a graceful shutdown of a VM, which does the following:

- Releases the CPUs, memory, GPUs, and NVLink
- Retains allocation of the OS and data disks

Note: Since allocation of the OS and data disks are retained, the creation of other VMs is still impacted by the shut-down VM.

To shut down a VM, enter the following.

```
sudo nvidia-vm shutdown <vm-domain>
```

In the event that **`nvidia-vm shutdown`** fails to shut down the VM, for example, if the VM OS is unresponsive, then you'll need to delete the VM as explained in the section Deleting a VM

12.4.2 Starting an Inactive VM

To restart a VM that has been shut down (not deleted), run the following.

```
sudo nvidia-vm start <vm-domain>
```

You can also connect to the console automatically upon restarting the VM using the following command.

```
sudo nvidia-vm start --console <vm-domain>
```

12.4.3 Deleting a VM

Like the process of creating a guest GPU VM, deleting a VM involves several `virsh` commands. For this reason, NVIDIA provides a simple way to delete a VM using `nvidia-vm`. Deleting a VM using `nvidia-vm` does the following:

- Stops the VM if it is running
- Erases data on disks that the VM is using and releases the disks
- Deletes any temporary support files

You should delete your VM instead of merely stopping it in order to release all resources and to remove unused files.

> ! **CAUTION:** VMs that are deleted cannot be recovered. Be sure to save any data before deleting any VMs.

Use **`nvidia-vm`** as explained in About nvidia-vm.

Syntax

```
sudo nvidia-vm delete --domain <vm-domain>
```

Command Help

```
sudo nvidia-vm delete --help
```

Command Examples

- Deleting an individual VM

  ```
  sudo nvidia-vm delete --domain dgx2vm-labTue1308-4g12-15
  ```

- Deleting all the VMs on the system

  ```
  sudo nvidia-vm delete --domain ALL
  ```

12.4.4 Stopping a VM

You can stop (or destroy) a VM, which forcefully stops the VM but leaves its resources intact.

```
sudo nvidia-vm destroy --domain <vm-domain>
```

To terminate gracefully, use the `--graceful` option:

```
sudo nvidia-vm destroy --domain <vm-domain> --graceful
```

12.4.5 Rebooting a VM

```
sudo nvidia-vm reboot --domain <vm-domain> --mode <shutdown-mode>
```

Where the shutdown mode string is one of the following: `acpi`, `agent`, `initctl`, `signal`, `paravirt`.

12.5 Connecting to Your Guest GPU VM

12.5.1 Determining IP Addresses

You can determine the IP address of your VM by entering the following.

```
virsh domifaddr <vm-domain> --source agent
```

This command returns IP addresses for the default network configuration (macvtap) as well as private networks. Refer to the section Network Configuration for a description of each network type.

Example:

```
$ virsh domifaddr 1gpu-vm-1g1 --source agent

Name        MAC address          Protocol     Address
-------------------------------------------------------------------------------
lo          00:00:00:00:00:00    ipv4         127.0.0.1/8
-           -                    ipv6         ::1/128
enp1s0      52:54:00:1e:23:2b    ipv4         10.120.28.219/24
-           -                    ipv6         fe80::5054:ff:fe1e:232b/64
enp2s0      52:54:00:1b:3b:1c    ipv4         192.168.254.150/24
-           -                    ipv6         fe80::5054:ff:fe1b:3b1c/64
docker0     02:42:d9:4e:00:b6    ipv4         172.17.0.1/16
```

In this example, 10.120.28.219/24 refers to the macvtap network, and 192.168.254.150/24 refers to the private network.

12.5.2 Connecting to the Guest GPU VM

You can connect to your VM in the following ways.

- Option 1 (connecting to the VM from the Host OS)

  ```
  virsh console <vm-domain>
  ```

- Option 2 (connecting to the VM using SSH)

```
ssh <username>@<IP ADDRESS>
```

The default credentials for logging into the VM are -

Login: `nvidia`

Password: `nvidia`

These can be changed. See the section Changing Login Credentials for instructions.

12.6 Making Your VM More Secure

There are a couple of things you can do to make your VM more secure.

- Change the Login Credentials
- Add SSH Keys

12.6.1 Changing Login Credentials

When the guest GPU VM is created, the default login credentials are nvidia/nvidia. As a security practice, use the standard Ubuntu methods to create a new user account and then delete the nvidia user account from the GPU VM. The basic commands are provided below for convenience. Consult the Ubuntu/Linux documentation for additional options.

- Creating a new user account

```
sudo useradd -m <new-username> -p <new-password>
```

- Deleting the `nvidia` user account

```
deluser -r nvidia
```

To run virsh commands, the new user must then be added to the libvirt and libvirt-qemu groups.

```
sudo usermod -a -G libvirt <new-username>
sudo usermod -a -G libvirt-qemu <new-username>
```

Using cloud-init

You can also use **cloud-init** to establish a unique username and password when you create the VM. Specify the username and password in the cloud config file. See the section Using cloud-init to Initialize the Guest VM for more information about using cloud-init.

12.6.2 Adding SSH Keys

You can incorporate SSH keys to increase security over password authentication.

Refer to the following websites for instructions.

- How to set up SSH so you aren't asked for a password
- How to disable password authentication

Using cloud-init

You can also use cloud-init to establish the SSH keys on a per-user basis when you create the VM. Specify the SSH authorized keys in the cloud config file. See the section Using cloud-init to Initialize the Guest VM for more information about using cloud-init.

12.7 Managing Images

Guest GPU VMs are based on an installed KVM image using thin provisioning for resource efficiency.

> ! **IMPORTANT:** A KVM guest VM runs a thin-provisioned copy of the source image. If the source image is ever uninstalled, the guest VM may not work properly. To keep guest VMs running uninterrupted, save the KVM source image to another location before uninstalling it.

You can manage these images as explained in this section.

Use **nvidia-vm** as explained in About nvidia-vm.

Syntax

```
sudo nvidia-vm image [options]
```

This section describes common command options.

Command Help

```
sudo nvidia-vm image --help
```

12.7.1 Installing Images

The KVM image is typically installed at the time the KVM package is installed. Since updated KVM images may be available from the repository, you can install any of these images for use in creating a guest GPU VM.

- To check available DGX KVM images, enter the following.

```
apt-cache policy dgx-kvm-image*
```

This returns a list of images in the repository.

- You can get detailed information about a specific KVM image using the `apt show` command as follows.

Syntax

```
apt show <kvm-image>
```

Example

```
apt show dgx-kvm-image-4-1-1

<snip>

Description: NVIDIA DGX bionic KVM hard disk image
   DGX BaseOS image for KVM
   OS Version: Ubuntu 18.04
   Kernel Version: 4.15.0-47.50
   Nvidia Driver Version: 418.67
   Nvidia Docker Version: 2.0.3+docker18.09.4-1
   Nvidia Container Runtime Version: 2.0.0+docker18.09.4-1
   Libnvidia Container Version: 1.0.2-1
```

- To install a KVM image from the list, use the `nvidia-vm image install` command.

Syntax

```
sudo nvidia-vm image install <kvm-image>
```

Example

```
sudo nvidia-vm image install dgx-kvm-image-4-1-1
```

12.7.2 Viewing a List of Installed Images

To view a list of all the VM images that are installed in the guest OS image directory, enter the following.

```
sudo nvidia-vm image show
```

12.7.3 Viewing Image Usage

To view a list of created VMs and the images they are using, enter the following.

```
sudo nvidia-vm image vmshow
```

Note: This command applies only to VMs that are not running. Currently, the command returns “Unknown” for any guest VMs that are running.

12.7.4 Uninstalling Images

If you convert the DGX-2 System from a KVM OS back to the bare metal system, you need to uninstall all the dgx-kvm images that were installed.

Perform the following for each installed image.

```
sudo nvidia-vm image uninstall dgx-kvm-image-x-y-z
```

```
Ok to remove image package "dgx-kvm-image-x-y-z"?  (y/N) :
```

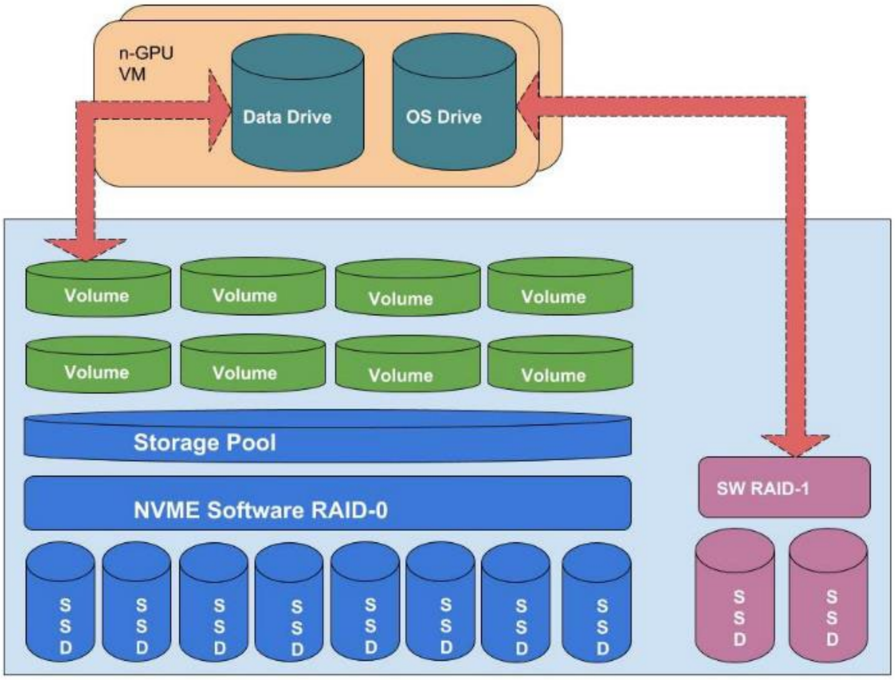
where *x-y-z* is the version for each installed image.

! **IMPORTANT:** If you uninstall KVM images without converting the system back to bare metal - or example, to recover space on the Hypervisor or to upgrade to a newer image - then you should make a copy of the image first.

A KVM guest VM runs a thin-provisioned copy of the source image. If the source image is ever uninstalled, the guest VM may not work properly. To keep guest VMs running uninterrupted, save the KVM source image to another location before uninstalling it.

12.8 Using the Guest OS Drives and Data Drives

The figure below depicts how NVIDIA KVM generates the Guest OS Drive and Data Drive from the physical drives on the DGX-2 System.



12.8.1 Guest OS Drive

DGX-2 KVM Host software uses the existing RAID-1 volume as the OS drive of each Guest (`/dev/vda1`) which by default is 50 GB. Since the OS drive resides on the RAID-1 array of the KVM Host, its data shall always be persistent.

Using the *nvidia-vm* tool, a system administrator can change the default OS drive size.

12.8.2 Data Drives

The DGX-2 KVM host software assigns a virtual disk to each guest GPU VM, referred to here as the Data Drive. It is based on filesystem directory-based volumes and can be used either as scratch space or as a cache drive.

DGX-2 software sets up a storage pool on top of the existing RAID-0 volume on the KVM Host for Data Drives on the Guests. The Data drive is automatically carved, by *nvidia-vm* tool, out of the Storage Pool and allocated to each GPU VM as a Data Drive (`/dev/vdb1`) which is automatically mounted on `/raid`. The Data Drive size is pre-configured according to the size of the GPU VM. For example, a 16-GPU VM gets a very large Data Drive (See the Resource Allocation section for size details).

Since the Data Drive is created on the Host RAID-0 array, data is not intended to be persistent. Therefore, when the GPU VM is destroyed, the Data Drive is automatically deleted and data is not preserved.

Using the *nvidia-vm* tool, a system administrator can change the default Data Drive size.

12.8.3 Storage Pool Examples

This section shows how to view the storage pool, and how disk space is assigned to a VM from the storage pool.

Show storage pool

Enter the following to verify the storage pool is active.

```
$ virsh pool-list
 Name                 State      Autostart
-------------------------------------------
 dgx-kvm-pool         active     yes
```

Create a VM:

```
$ sudo nvidia-vm create --gpu-count 1 --gpu-index 0
dgx2vm-rootTue1616-1g0: create define start mac: 52:54:00:16:b9:ff  ip:
10.120.28.219/24
```

Viewing the Volume from the DGX-2 KVM Host

To see the volumes that are created for each VM, enter the following.

```
$ virsh vol-list dgx-kvm-pool --details

 Name                       Path                                    Type Capacity Allocation
-----------------------------------------------------------------------------------------------
 vol-dgx2vm-rootTue1616-1g0  /raid/dgx-kvm/vol-dgx2vm-labTue1616-1g0  file  1.74 TiB  3.71 GiB
```

Viewing the Data Volume from the Guest VM

1. Connect to the guest GPU VM.

```
$ virsh console dgx2vm-labTue1616-1g0
Connected to domain dgx2vm-rootTue1616-1g0
```

2. List the virtual storage on the guest GPU VM.

```
nvidia@dgx2vm-rootTue1616-1g0:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda    252:0    0   50G  0 disk
└─vda1 252:1    0   50G  0 part /
vdb    252:16   0 54.9G  0 disk
└─vdb1 252:17   0 54.9G  0 part /raid
nvidia@dgx2vm-rootTue1616-1g0:~$
```

12.9 Network Configuration

Networks can be configured in several way. The following table shows the available options and describes their application.

Configuration	**Host<->VM**	**VM<->VM**	**VM<->External**
macvtap (bridged mode – default)	No	Yes	Yes
macvtap (VEPA mode)	Yes	Yes	Yes
Private	**Yes**	No	No

Macvtap (bridged mode)

This is the default mode. Each guest VM has a virtual network interface based on the *macvtap-net* network.

Macvtap (VEPA mode)

Set macvtap in Virtual Ethernet Port Aggregator (VEPA) mode to support communication between the VM and an outside network. This mode requires that the network switch supported "hairpin" mode. Refer to the "KVM Networking Best Practices Guide" for more details.

Private Network

Specify **--privateIP** while creating the VM so that the second virtual network interface will be added based on *private-net* network for Host-to-VM connectivity.

Example:

```
sudo nvidia-vm create --gpu-count 4 --gpu-index 12 --privateIP
```

12.10 Updating the Software

12.10.1 Updating the Host OS

You can update the DGX OS software for the host using standard Ubuntu `apt` process with an internet connection.

Since the reboot step will stop any running guest VMs, they should be stopped first to avoid an uncontrolled or unexpected interruption which can lead to corruption of the VM.

> **!** **IMPORTANT:** Updating the DGX OS software may result in an over-write of the associated KVM image. Guest VMs created from this older image will no longer be available. To keep guest VMs, save the older KVM image to another location and then and then restore the image after updating the DGX OS.

Perform the following from the host OS.

1. Shut down all running VMs.

   Failure to shut down the VMs may result in corruption of one or more VMs after the final reboot step.
2. Update the list of available packages and their versions.

   ```
   $ sudo apt update
   ```
3. Review the packages that will be updated.

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

   To prevent an application from being updated, instruct the Ubuntu package manager to keep the current version. See Introduction to Holding Packages.
4. Upgrade to the latest version.

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

   - Answer any questions that appear.
     - Most questions require a Yes or No response. When asked to select the grub configuration to use, select the current one on the system.
     - 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 indicating that nvidia-docker.service failed to start, you can disregard it and continue with the next step. The service will start normally at that time.
5. Reboot the system.

   ```
   $ sudo reboot
   ```

12.10.2 Updating the Guest VM OS

You can update the DGX OS software for the guest VM using standard Ubuntu apt process with an internet connection. This is the same process that is used when updating the DGX OS software on the bare metal system.

> **!** **IMPORTANT:** A KVM guest VM runs a thin-provisioned copy of the source image. If the source image is ever uninstalled, the guest VM may not work properly. To keep guest VMs running uninterrupted, save the KVM source image to another location before uninstalling it.

Perform the following from the guest VM.

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

   ```
   $ sudo apt update
   ```

2. Review the packages that will be updated.

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

   To prevent an application from being updated, instruct the Ubuntu package manager to keep the current version. See Introduction to Holding Packages.

3. Upgrade to the latest version.

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

   - Answer any questions that appear.
     - Most questions require a Yes or No response. When asked to select the grub configuration to use, select the current one on the system.
     - 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 indicating that nvidia-docker.service failed to start, you can disregard it and continue with the next step. The service will start normally at that time.

4. Reboot the guest VM.

   ```
   $ sudo reboot
   ```

12.11 Supplemental Information

12.11.1 Resource Allocations

By default, the KVM software assigns the following resources in approximate proportion to the number of assigned GPUs:

GPU	1	2	4	8	16
vCPU/HT	5	10	22	46	92
Memory (GB)	90	180	361	723	1446
InfiniBand	N/A	1	2	4	8
OS Drive (GB)	50	50	50	50	50
Data Drive (TB)	1.92	3.84	7.68	15.36	31.72
Ethernet	macvtap	macvtap	macvtap	macvtap	macvtap
NVLink	N/A	1	3	6	6

- Data drive values indicate the maximum space that will be used. The actual space is allocated as needed.
- You can use command options to customize memory allocation, OS disk size, and number of vCPUs to assign.
- `virtio-net` and `virtio-blk` devices are configured with multiple queues to increase performance.

12.11.2 Resource Management

NVIDIA KVM optimizes resources to maximize the performance of the VM.

- vCPU

  vCPUs are pinned to each VM to be NUMA-aware and to provide better VM performance.
- InfiniBand

  InfiniBand (IB) devices are set up as passthrough devices to maximize performance.
- GPU

  GPUs are set up as passthrough devices to maximize performance.
- Data Drive

Data drives are intended to be used as scratch space cache.

- NVSwitch

  NVSwitch assignments are optimized for NVLink peer-to-peer performance.

- NVLink

  An NVLink connection is the connection between each GPU and the NVSwitch fabric. Each NVLink connection allows up to 25 GB/s uni-directional performance.

12.11.3 NVIDIA KVM Security Considerations

Consult the security policies of your organization to determine firewall needs and settings.

12.11.4 Launching VMs in Degraded Mode

On DGX-2 KVM systems, degraded mode is a mechanism that allows one or more GPUs to fail without affecting the operation or creation of other VMs on the server. This allows the DGX-2 System to run GPU VMs with fewer than 16 GPUs present. System administrators can then keep a subset of GPU VMs available for use while waiting to replace GPUs that may have failed.

12.11.4.1 When the DGX-2 is Put in Degraded Mode

The following are the type of GPU errors that will put the system in degraded mode:

- GPU double-bit ECC errors
- GPU failure to enumerate on the PCIe bus
- GPU side NVLink training error
- GPU side unexpected XID error

To identify failed GPUs, the KVM host automatically polls the state of any GPUs to be used upon launching a VM. When a failed GPU is identified by the software, the DGX-2 System is marked as 'degraded' and operates in degraded mode until all bad GPUs are replaced.

12.11.4.2 Performing a GPU Health Check

You can create the initial GPU health database after installing the KVM software but before rebooting the system.

The following command tests all the GPUs in the system.

```
$ sudo nvidia-vm health-check [options]
```

Where [options] are

`--force`	Forces health-check to run, and rebuilds the database
`--help`	Prints this help text
`--fulltest`	Runs an extensive test, approximately 2 minutes per GPU
`--timelimit`	Approximate time to limit running of the test

Examples:

To run a quick test if health info does not exist.

```
$ sudo nvidia-vm health-check
```

To run an extended test.

```
$ sudo nvidia-vm health-check --force --fulltest
```

To see the GPU status as recorded in the database:

```
$ sudo nvidia-vm health-check show
```

12.11.4.3 Getting GPU Health Information from Within the VM

Enable fetching the GPU health from the KVM host by enabling monitoring of the guest VM.

1. Shut down the guest VM.

   ```
   $ virsh shutdown <vm-name>
   ```

2. Edit the NVIDIA fabric manager service.

   ```
   $ sudo virt-edit <vm-name> /lib/systemd/system/nvidia-fabricmanager.service
   ```

   Change the following line from

   ```
   ExecStart=/usr/bin/nv-hostengine -l -g --log-level 4 --log-rotate --log-filename /var/log/fabricmanager.log
   ```

   to

   ```
   ExecStart=/usr/bin/nv-hostengine -l -g --log-level 4 -b ALL --log-rotate --log-filename /var/log/fabricmanager.log
   ```

3. Restart the guest VM.

   ```
   $ virsh start <vm-name>
   ```

4. Get the guest VM's IP address

   ```
   $ virsh domifaddr <vm-name> --source agent
   ```

5. Get the guest VM's GPU information

   ```
   $ dcgmi discovery --host <vm-ip-address> -l
   ```

6. Enable the guest VM health monitoring.

   ```
   $ dcgmi health --host <vm-ip-address> -s a
   ```

7. Get the guest VM health status.

   ```
   $ dcgmi health --host <vm-ip-address> --check
   ```

   Example output

   ```
   +---------------------------+---------------------------------------+
   | Health Monitor Report                                             |
   +===========================+=======================================+
   | Overall Health            | Healthy                               |
   +---------------------------+---------------------------------------+
   ```

12.11.4.4 Creating VMs with the DGX-2 System in Degraded Mode

You can still create guest GPU VMs on a DGX-2 System in degraded mode as long as you do not try to assign a failed GPU. If you attempt to create a VM with a failed GPU after its state has been marked as 'bad' by the system, the VM will fail to start and an appropriate error message is returned. Restarting an existing VM after a GPU fails will result in the same failure and error message.

The following is an example of launching a VM when GPU 12 and 13 have been marked as degraded or in a failed state.

```
sudo nvidia-vm create --gpu-count 8 --gpu-index 8
```

```
ERROR: GPU 12 is in unexpected state "missing", can't use it -
BDF:e0:00.0 SXMID:13 UUID:GPU-b7187786-d894-2266-d11d-21124dc61dd3
```

```
ERROR: GPU 13 is in unexpected state "missing", can't use it -
BDF:e2:00.0 SXMID:16 UUID:GPU-9a6a6a52-c6b6-79c3-086b-fcf2d5b1c87e
```

```
ERROR: 2 GPU's are unavailable, unable to start this VM "dgx2vm-
labMon1559-8g8-15"
```

> **Note:** If you attempt to launch a VM with a failed GPU before the system has identified its failed state, the VM will fail to launch but without an error message. If this happens, keep trying to launch the VM until the message appears.

12.11.4.5 Restarting a VM After the System or VM Crashes

Some GPU errors may cause the VM or the system to crash.

- If the system crashes, you can attempt to restart the VM.
- If the VM crashes (but not the system), you can attempt to restart the VM.

Your VM should restart successfully if none of the associated GPUs failed. However, if one or more of the GPUs associated with your VM failed, then the response depends on whether the system has had a chance to identify the GPU as unavailable.

- Failed GPU identified as unavailable

  The system will return an error indicating that the GPU is missing or unavailable and that the VM is unable to start.

- Failed GPU not yet identified as unavailable

  The VM crashes upon being restarted.

12.11.4.6 Restoring a System from Degraded Mode

All GPUs need to be replaced to restore the DGX-2 from degraded mode.

The server must be powered off when performing the replacement. After GPU replacement and upon powering on the server, the KVM software runs a health scan to add any new GPUs to the health database.

12.12 Troubleshooting Tools

This section discusses tools to assist in gathering data to help NVIDIA Enterprise Services troubleshoot GPU VM issues. Most of the tools are used at the KVM host level. If you are using guest VMs and do not have access to the KVM host, then request the help of your system administrator.

12.12.1 Reporting Issues and Collecting Information

Log on to the NVIDIA Enterprise Services site for assistance with troubleshooting, diagnosing, or reporting DGX-2 KVM issues.

Run `nvsysinfo` from within the guest VM as well as on the KVM host, collect the output and provide to NVIDIA Enterprise Services.

The following sections focus on detecting

- Guest VM launch and shutdown issues
- Guest VM connection issues
- Guest VM GPU issues
- Host KVM and Guest VM Storage issues

12.12.2 How to Detect Guest Launch/Shutdown Issues

Obtain the following log files and information:

12.12.2.1.1virt-install.log

```
$ grep -i 'error|fail' $HOME/.cache/virt-manager/virt-install.log
```

12.12.2.1.2Guest VM log file

```
$ sudo egrep -i 'error|fail' /var/log/libvirt/qemu/<vm-name>.log
```

From the KVM host, connect to the VM console to verify guest VM operation.

```
$ virsh console <vm-name>
```

12.12.3 How to Detect Guest Networking Issues

There are several steps to take to verify suspected guest connection issues.

12.12.3.1.1 Verify Enabled Networks

Check whether macvtap and private networks are still active.

```
$ virsh net-list
```

Example output

```
 Name                     State      Autostart     Persistent
----------------------------------------------------------
 macvtap-net              active     yes           yes
 private-net              active     yes           yes
```

12.12.3.1.2 Verify Guest VM IP Address

Check the guest IP address to see if there is an IP address and whether it is as expected.

```
$ virsh domifaddr <vm-name> --source agent
```

Example output:

```
$ virsh domifaddr 1gpu-vm-1g2 --source agent

 Name       MAC address          Protocol   Address
-----------------------------------------------------------------
 lo         00:00:00:00:00:00    ipv4       127.0.0.1/8
 -          -                    ipv6       ::1/128
 enp1s0     52:54:00:3c:07:62    ipv4       10.120.28.227/24
 -          -                    ipv6       fe80::5054:ff:fe3c:762/64
 docker0    02:42:9f:5c:39:da    ipv4       172.17.0.1/16
```

12.12.3.1.3 Configuring VMs for Host-to-VM Network Connectivity

By default, each VM is created with a virtual network interface based on the macvtap-net network for VM-to-External and VM-to-VM network connectivity. The default macvtapnet network is configured in "bridge" mode and it will not provide Host-to-VM network connectivity. Read more on macvtap and its limitation.

You can configure the VM for Host-to-VM network connectivity by using privateIP. See How to Configure the Guest VM with privateIP for instructions.

12.12.4 How to Detect GPU Issues in a Guest VM

See Getting GPU Health Information from Within the VM.

12.12.5 How to Detect Storage Issues in a Guest VM or on the KVM Host

NVIDIA KVM reuses SSDs available to hypervisor to create Guest OS and Data Drives. This section covers how to check the health of SSDs from Host as well as inside a GPU Guest.

Instructions for fixing degraded RAID-0 and its impact to GPU VMs is currently beyond the scope of this document.

12.12.5.1 How to Check Storage Health in the KVM Host

The DGX-2 Host has two OS drives in RAID-1 array (/dev/nvme0n1, /dev/nvme1n1), and the remaining NVMe drives are in aRAID-0 array as Data Drives. Check the health of SSDs and RAID separately.

12.12.5.1.1To check health of SSDs:

1. Run the following command to list the SSDs.

   ```
   :~$ sudo nvme list
   ```

   Example output

   ```
   Node         SN              Model                      Namespace  Usage                Format      FW Rev

   ------------ -------------- -------------------------- -- -------------------- ----------  --------

   /dev/nvme0n1 S2X6NX0K501953 SAMSUNG MZ1LW960HMJP-00003  1 61.79 GB / 960.20 GB 512 B + 0 B CXV8601Q


   <snip> ...


   /dev/nvme9n1 18141C246847   Micron_9200_MTFDHAL3T8TCT   1 3.84 TB / 3.84 TB    512 B + 0 B 101008R0
   ```

2. Check whether the number of drives is less than two.

   If less than two, then there is an issue.

3. View the health of each NVMe drive by running the following command:

   ```
   :-$ sudo nvme smart-log /dev/nvme9n1
   ```

Example output

```
Smart Log for NVME device:nvme9n1 namespace-id:fffffff
critical_warning                   : 0
<snip> ...
```

If smart-log shows *critical_warning* as a non-zero value, your NVMe SSD is faulted.

12.12.5.1.2To check health of the RAID Array:

Run the following command on the raid devices (/dev/md0, /dev/md1).

```
$ sudo mdadm -S -D  /dev/md0
```

Example output

```
/dev/md0:
           Version : 1.2
     Creation Time : Tue Aug 13 08:23:52 2019
        Raid Level : raid1
        Array Size : 937034752 (893.63 GiB 959.52 GB)
     Used Dev Size : 937034752 (893.63 GiB 959.52 GB)
      Raid Devices : 2
     Total Devices : 2
       Persistence : Superblock is persistent

     Intent Bitmap : Internal

       Update Time : Thu Aug 15 16:31:29 2019
             State : clean
    Active Devices : 2
   Working Devices : 2
    Failed Devices : 0
     Spare Devices : 0

Consistency Policy : bitmap

              Name : dgx-18-04:0
              UUID : 98c0057f:11b0d131:2b689147:c780f126
            Events : 2014

    Number   Major   Minor   RaidDevice State
       0     259        2        0      active sync   /dev/nvme0n1p2
       1     259        5        1      active sync   /dev/nvme1n1p2
```

If highlighted areas show error, your RAID is faulted.

12.12.5.1.3To check health of the KVM Storage Pools:

The DGX-2 KVM software creates Data Drives for each GPU VM on the RAID-0 array as a volume. OS drives are also created as a Volume. You could check the health of KVM pools as follows.

```
$ virsh pool-list --details
```

Example Output

```
 Name           State     Autostart  Persistent    Capacity  Allocation    Available
------------------------------------------------------------------------------------
 dgx-kvm-pool   running   yes        yes          27.83 TiB  171.71 GiB    27.66 TiB
 images         running   yes        yes         878.57 GiB   19.62 GiB   858.95 GiB
```

If highlighted areas show error, your Storage Pool is faulted.

12.12.5.2 How to Check Storage Health in KVM Guests

The DGX-2 KVM Host software uses the existing RAID-1 volume as the OS drive of each Guest (/dev/vda1), which by default is 50 GB. Each GPU VM also gets a virtual disk called the Data Drive. It is based on filesystem directory-based volumes and can be used either as scratch space or as a cache drive.

- From within the guest VM, run the following command.

  ```
  :~# lsblk
  ```

  Example output

  ```
  NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
  vda     252:0    0   50G  0 disk
  └─vda1  252:1    0   50G  0 part /
  vdb     252:16   0 13.9T  0 disk
  └─vdb1  252:17   0 13.9T  0 part /raid
  ```

  If you do not see both drives, report this as an error.

- From within the Hypervisor, perform the following command and verify that the output shows “No errors found”.

  ```
  $ virsh domblkerror <vm-name>
  No errors found
  ```

12.12.6 Using libguestfs-tools to Gather Log Files and Perform Console Scraping

The NVIDIA KVM software installs the libguestfs-tool utility which provides useful tools for viewing log files, console scraping, etc. These tools shouldbe run from the KVM host.

The following are some usage examples.

12.12.6.1.1 Partial List of tools for checking log files

- virt-cat
- virt-log
- virt-tail
- virt-edit

12.12.6.1.2 Partial List of tools for checking block devices and filesystems

- virt-df
- virt-filesystems
- virt-ls
- virt-copy-in
- virt-copy-out
- virt-list-filesystems
- virt-list-partitions

12.12.6.1.3 Examples

- Check console/syslog

```
$ sudo virt-log -d <vm-name>
```

- Read/Write specific log files

```
$ sudo virt-cat -d <vm-name> /var/log/syslog

$ sudo virt-edit -d <vm-name> /var/log/syslog
```

- Display free space on guest VM filesystems.

```
$ sudo virt-df -d <vm-name>
```

Example

```
$ sudo virt-df -d 1gpu-vm-1g0

Filesystem                  1K-blocks   Used      Available  Use%
1gpu-vm-1g0:/dev/sda1       51341792    3313160   45390912    7%
```

- List filesystems, partitions, block devices, LVM on the guest VM's disk image.

```
$ sudo virt-filesystems -d <vm-name>
```

12.12.7 Known Issues

For a list of known issues with using GPU VMs on DGX-2 systems, refer to DGX-2 Server Software Release Notes.

12.12.8 Reference Resources

The following are some useful resources for debugging and troubleshooting KVM issues.

- Linux KVM: Guest OS debugging

Chapter 13. Replacing Components

Be sure to familiarize yourself with the NVIDIA Terms & Conditions documents before attempting to perform any modification or repair to the DGX-1. These Terms & Conditions for the DGX-1 can be found through the NVIDIA DGX Systems Support page (http://www.nvidia.com/object/dgxsystems-support.html).

Contact NVIDIA Enterprise Support to obtain an RMA number for any system or component that needs to be returned for repair or replacement. When replacing a component, use only the replacement supplied to you by NVIDIA.

The following components are customer-replaceable:

Bezel	DIMMs	Motherboard Tray Battery
Boot Drives Riser Assembly	EMI Shield	PCIe Riser Assembly
Boot (M.2) NVMe Drives	Front Fan Modules	Power Supplies
Cache (U.2) NVMe Drives	Front Console Board	Power Supply Carrier
ConnectX-5 Network Adapter Card	I/O Expander Tray	Power Supply Carrier Fan

Return failed high-value components to NVIDIA. Low-cost items such as batteries, power supplies, and fans do not need to be returned. See the NVIDIA DGX-2 Service Manual for instructions on replacing these components.

Chapter 14. Security

The NVIDIA DGX-2 Server is a specialized server designed to be deployed in a data center. It must be configured to protect the hardware from unauthorized access and unapproved use. The DGX-2 Server is designed with a dedicated BMC Management Port and multiple Ethernet network ports.

When installing the DGX-2 Server in the data center, follow best practices as established by your organization to protect against unauthorized access.

Updating BMC Username and Password

If you have not already updated the default BMC username and passwords, NVIDIA recommends the BMC username and passwords be updated immediately to protect your system or nodes from unauthorized access. Follow the instructions in the section Creating a Unique BMC Password to update the default password.

Securing SNMP

Administrators must also confirm that the SNMP Read and Read/Write community strings are changed to a secure phrase or that you have disabled SNMP access. These settings can be found in the BMC dashboard under Configuration > SNMP.

Securing the BMC Port

NVIDIA recommends that the BMC port of the DGX-2 Server be connected to a dedicated management network with firewall protection. If remote access to the BMC is required (such as for a system hosted at a co-location provider), it should be accessed through a secure method that provides isolation from the internet, such as through a VPN server.

Chapter 15. Secure Data Deletion of SSDs

This section explains how to securely delete data from the NVIDIA DGX-2 system SSDs to permanently destroy all the data that was stored there. This performs a more secure SSD data deletion than merely deleting files or reformatting the SSDs.

Prerequisite

Prepare a bootable installation medium that contains the current DGX OS Server ISO image.

See:

- Obtaining the DGX-2 Software ISO Image and Checksum File
- Creating a Bootable Installation Medium

Instructions

1. Boot the system from the ISO image, either remotely or from a bootable USB key.

   The ISO image must contain the `nvmecli` package.

2. At the GRUB menu, choose ‘**Rescue a broken system**’, then configure the locale and network information.
3. When asked to choose a root file system, choose

   ‘**`Do not use a root file system`**’

   and then

   ‘**`Execute a shell in the installer environment`**’

4. Log in.
5. Run the following command to identify the devices available in the system:

   ```
   $ sudo nvme list
   ```

   Example output showing eight cache and two boot devices.

```
Node           SN         Model           Namespace Usage                  Format
FM Rev
```

```
------------  --------  --------------  --------- --------------------   ------------
-----------
/dev/nvme0n1  Sxxxxxxx  Samsung MZxxxx   1        88.99 GB / 960.20 GB   512  B + 0 B
CXV8501Q
/dev/nvme1n1  Sxxxxxxx  Samsung MZxxxx   1        90.11 GB / 960.20 GB   512  B + 0 B
CXV8501Q
/dev/nvme2n1  18xxxxxx  Micron_9200_xx   1        3.84 TB  / 3.84 TB     512  B + 0 B
101008R0
/dev/nvme3n1  18xxxxxx  Micron_9200_xx   1        3.84 TB  / 3.84 TB     512  B + 0 B
101008R0
/dev/nvme4n1  18xxxxxx  Micron_9200_xx   1        3.84 TB  / 3.84 TB     512  B + 0 B
101008R0
/dev/nvme5n1  18xxxxxx  Micron_9200_xx   1        3.84 TB  / 3.84 TB     512  B + 0 B
101008R0
/dev/nvme6n1  18xxxxxx  Micron_9200_xx   1        3.84 TB  / 3.84 TB     512  B + 0 B
101008R0
/dev/nvme7n1  18xxxxxx  Micron_9200_xx   1        3.84 TB  / 3.84 TB     512  B + 0 B
101008R0
/dev/nvme8n1  18xxxxxx  Micron_9200_xx   1        3.84 TB  / 3.84 TB     512  B + 0 B
101008R0
/dev/nvme9n1  18xxxxxx  Micron_9200_xx   1        3.84 TB  / 3.84 TB     512  B + 0 B
101008R0
```

If `nvmecli` is not installed, then install the CLI as follows and then run `nvme list`.

```
$ udpkg -i /cdrom/extras/pool/main/n/nvme-cli/nvme-cli_1.5-
1ubuntu1_amd64.deb
```

6. Run `nvme format -s1` on all storage devices listed.

   The following example shows 8 cache and 2 boot devices.

```
$ sudo nvme format -s1 /dev/nvme0n1
$ sudo nvme format -s1 /dev/nvme1n1
$ sudo nvme format -s1 /dev/nvme2n1
$ sudo nvme format -s1 /dev/nvme3n1
$ sudo nvme format -s1 /dev/nvme4n1
$ sudo nvme format -s1 /dev/nvme5n1
$ sudo nvme format -s1 /dev/nvme6n1
$ sudo nvme format -s1 /dev/nvme7n1
$ sudo nvme format -s1 /dev/nvme8n1
$ sudo nvme format -s1 /dev/nvme9n1
```

Appendix A. Installing Software on Air-gapped DGX-2 Systems

For security purposes, some installations require that systems be isolated from the internet or outside networks. Since most DGX-2 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 Docker containers as well.

A.1 Installing NVIDIA DGX-2 Software

One method for updating DGX-2 software on an air-gapped DGX-2 System is to download the ISO image, copy it to removable media and then re-image the DGX-2 System from the media. This method is available only for software versions that are available as ISO images for download.

Alternately, you can update the DGX-2 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.2 Re-Imaging the System

! **CAUTION:** This process destroys all data and software customizations that you have made on the DGX-2 System. Be sure to back up any data that you want to preserve and push any Docker images that you want to keep to a trusted registry.

1. Obtain the ISO image from the Enterprise Support site.
   a) Log on to the NVIDIA Enterprise Support site and click the Announcements tab to locate the DGX OS Server image ISO file.
   b) Download the image ISO file.

2. Refer to the instructions in the Restoring the DGX-2 Software Image section for additional instructions.

A.3 Creating a Local Mirror of the NVIDIA and Canonical Repositories

The procedure below describes how to download all the necessary packages to create a mirror of the repositories that are needed to update Nvidia DGX systems. The steps are specific to versions 4.0.X and 4.1.X, but they can be edited to work with other versions. For more information on DGX OS versions and the release notes available, please visit https://docs.nvidia.com/dgx/dgx-os-server-release-notes/index.html#dgx-os-release-number-scheme.

For more information on how to upgrade from versions 4.0.X to 4.1.X, review the release notes: https://docs.nvidia.com/dgx/pdf/DGX-OS-server-4.1-relnotes-update-guide.pdf.

> **Note:** These procedures apply only to upgrades within the same major release, such as 4.x → 4.y. It does not support upgrades across major releases, such as 3.x → 4.x..

A.3.1 Create the Mirror in a DGX OS 4 System

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

Prerequisites

- A system installed with Ubuntu OS is needed to create the mirror because there are several Ubuntu tools that need to be used.
- The system must contain enough storage space to replicate the repositories to a filesystem; the space requirement could be as high as 250GB.
- An efficient way to move large amount of data; 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 the portable drive is formatted using ext4 or FAT32.

1. Make sure the storage device is attached to the system with network access and identify the mount point.

   Example mount point: `/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 as well as 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 bionic-security main
deb http://security.ubuntu.com/ubuntu bionic-security universe
deb http://security.ubuntu.com/ubuntu bionic-security multiverse
deb http://archive.ubuntu.com/ubuntu/ bionic main multiverse universe
deb http://archive.ubuntu.com/ubuntu/ bionic-updates main multiverse universe
#
deb-i386 http://security.ubuntu.com/ubuntu bionic-security main
deb-i386 http://security.ubuntu.com/ubuntu bionic-security universe
deb-i386 http://security.ubuntu.com/ubuntu bionic-security multiverse
deb-i386 http://archive.ubuntu.com/ubuntu/ bionic main multiverse universe
deb-i386 http://archive.ubuntu.com/ubuntu/ bionic-updates main multiverse universe
#
# DGX specific repositories:
deb http://international.download.nvidia.com/dgx/repos/bionic bionic main restricted universe multiverse
```

```
deb http://international.download.nvidia.com/dgx/repos/bionic bionic-updates main restricted universe multiverse
deb http://international.download.nvidia.com/dgx/repos/bionic bionic-r418+cuda10.1 main multiverse restricted universe
#
deb-i386 http://international.download.nvidia.com/dgx/repos/bionic bionic main restricted universe multiverse
deb-i386 http://international.download.nvidia.com/dgx/repos/bionic bionic-updates main restricted universe multiverse
# Only for DGX OS 4.1.0
deb-i386 http://international.download.nvidia.com/dgx/repos/bionic bionic-r418+cuda10.1 main multiverse restricted universe
# 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.3.2 Configure the Target System

The instructions in this section are to be performed on the target system.

Prerequisites

- The target DGX system is installed, has gone through the first boot process, and is ready to be updated with the latest packages.
- A USB storage device is attached to the target DGX server.

  There are other ways to transfer the data that are not covered in this document as they will depend on the data center policies for the air-gapped environment.

1. Mount the storage device on the air-gapped system to `/media/usb/repository` for consistency.
2. Configure the `apt` command to use the filesystem 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 bionic-security main
   deb file:///media/usb/repository/mirror/security.ubuntu.com/ubuntu bionic-security universe
   ```

```
deb file:///media/usb/repository/mirror/security.ubuntu.com/ubuntu
bionic-security multiverse
deb file:///media/usb/repository/mirror/archive.ubuntu.com/ubuntu/
bionic main multiverse universe
deb file:///media/usb/repository/mirror/archive.ubuntu.com/ubuntu/
bionic-updates main multiverse universe
```

3. Configure apt to use the NVIDIA DGX OS packages in the file `/etc/apt/sources.list.d/dgx.list`.

   ```
   deb
   file:///media/usb/repository/mirror/international.download.nvidia.com/dgx/repos/bionic
   bionic main multiverse restricted universe
   ```

4. If present, remove the file `/etc/apt/sources.list.d/docker.list` as it is no longer needed and it will eliminate error messages during the update process.
5. **(For DGX OS Release 4.1 and later only)** Configure apt to use the NVIDIA DGX OS packages in the file `/etc/apt/sources.list.d/dgx-bionic-r418-cuda11-0-repo.list`

   ```
   deb
   file:///media/usb/repository/mirror/international.download.nvidia.com/dgx/repos/bionic
   / bionic-r418+cuda10.1 main multiverse restricted universe
   ```

6. **(Optional for DGX OS Release 4.5 and later only, for using the Release 450 driver package and CUDA Toolkit 11.0)** Configure apt to use the NVIDIA DGX OS packages in the file `/etc/apt/sources.list.d/dgx-bionic-r450-cuda10-1-repo.list`

   ```
   deb
   file:///media/usb/repository/mirror/international.download.nvidia.com/dgx/repos/bionic
   / bionic-r450+cuda11.0 main multiverse restricted universe
   ```

   Note: If you want to continue using earlier releases, for example the R418 NVIDIA graphic driver and CUDA Toolkit 10.1, omit this step.

7. Edit the file `/etc/apt/preferences.d/nvidia` to update the Pin parameter as follows.

   ```
   Package: *
   #Pin: origin international.download.nvidia.com
   Pin: release o=DGX Server
   Pin-Priority: 600
   ```

8. Update the apt repository and confirm there are no errors.

   ```
   $ sudo apt update
   ```

   Output from this command is similar to the following example.

```
Get:1 file:/media/usb/repository/mirror/security.ubuntu.com/ubuntu
bionic-security InRelease [88.7 kB]
Get:1 file:/media/usb/repository/mirror/security.ubuntu.com/ubuntu
bionic-security InRelease [88.7 kB]
```

```
Get:2 file:/media/usb/repository/mirror/archive.ubuntu.com/ubuntu
bionic InRelease [242 kB]
Get:2 file:/media/usb/repository/mirror/archive.ubuntu.com/ubuntu
bionic InRelease [242 kB]
Get:3 file:/media/usb/repository/mirror/archive.ubuntu.com/ubuntu
bionic-updates InRelease [88.7 kB]
Get:4
file:/media/usb/repository/mirror/international.download.nvidia.com/dgx
/repos/bionic bionic-r418+cuda10.1 InRelease [13.0 kB]
Get:5
file:/media/usb/repository/mirror/international.download.nvidia.com/dgx
/repos/bionic bionic InRelease [13.1 kB]
Get:3 file:/media/usb/repository/mirror/archive.ubuntu.com/ubuntu
bionic-updates InRelease [88.7 kB]
Get:4
file:/media/usb/repository/mirror/international.download.nvidia.com/dgx
/repos/bionic bionic-r418+cuda10.1 InRelease [13.0 kB]
Get:5
file:/media/usb/repository/mirror/international.download.nvidia.com/dgx
/repos/bionic bionic InRelease [13.1 kB]
Hit:6 https://download.docker.com/linux/ubuntu bionic InRelease
Get:7
file:/media/usb/repository/mirror/international.download.nvidia.com/dgx
/repos/bionic bionic-r418+cuda10.1/multiverse amd64 Packages [10.1 kB]
Get:8
file:/media/usb/repository/mirror/international.download.nvidia.com/dgx
/repos/bionic bionic-r418+cuda10.1/restricted amd64 Packages [10.3 kB]
Get:9
file:/media/usb/repository/mirror/international.download.nvidia.com/dgx
/repos/bionic bionic-r418+cuda10.1/restricted i386 Packages [516 B]
Get:10
file:/media/usb/repository/mirror/international.download.nvidia.com/dgx
/repos/bionic bionic/multiverse amd64 Packages [44.5 kB]
Get:11
file:/media/usb/repository/mirror/international.download.nvidia.com/dgx
/repos/bionic bionic/multiverse i386 Packages [8,575 B]
Get:12
file:/media/usb/repository/mirror/international.download.nvidia.com/dgx
/repos/bionic bionic/restricted i386 Packages [745 B]
Get:13
file:/media/usb/repository/mirror/international.download.nvidia.com/dgx
/repos/bionic bionic/restricted amd64 Packages [8,379 B]
Get:14
file:/media/usb/repository/mirror/international.download.nvidia.com/dgx
/repos/bionic bionic/universe amd64 Packages [2,946 B]
Get:15
file:/media/usb/repository/mirror/international.download.nvidia.com/dgx
/repos/bionic bionic/universe i386 Packages [496 B]
Reading package lists... Done
Building dependency tree
```

```
Reading state information... Done
249 packages can be upgraded. Run 'apt list --upgradable' to see them.
$
```

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

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

   If you configured `apt` to use the NVIDIA DGX OS packages in the file `/etc/apt/sources.list.d/dgx-bionic-r450-cuda11-0-repo.list`, the NVIDIA graphics driver is upgraded to the R450 driver and the package sources are updated to obtain future updates from the R450 driver repositories.
10. Optional: (For DGX OS Release 4.5 and later only) If you configured apt to use the NVIDIA DGX OS packages in the file `/etc/apt/sources.list.d/dgx-bionic-r450-cuda11-0-repo.list` and want to use CUDA Toolkit 11.0, install it.

    ```
    $ sudo apt install cuda-toolkit-11-0
    ```

> **Note:** If you did not configure apt to use the NVIDIA DGX OS packages in the file /etc/apt/sources.list.d/dgx-bionic-r450-cuda11-0-repo.list, omit this step. If you try to install CUDA Toolkit 11.0, the attempt fails...

A.4 Installing Docker Containers

This method applies to Docker containers hosted on the NVIDIA NGC Container Registry, and requires that you have an active NGC account.

1. On a system with internet access, log in to the NGC Container Registry by entering the following command and credentials.

   ```
   $ docker login nvcr.io
   Username: $oauthtoken
   Password: apikey
   ```

   Type "$oauthtoken" exactly as shown for the Username. This is a special username that enables API key authentication. In place of apikey, paste in the API Key text that you obtained from the NGC website.
2. Enter the docker pull command, specifying the image registry, image repository, and tag:

   ```
   $ docker pull nvcr.io/nvidia/repository:tag
   ```

3. Verify the image is on your system using docker images.

   ```
   $ docker images
   ```

4. Save the Docker image as an archive. .

   ```
   $ docker save nvcr.io/nvidia/repository:tag > framework.tar
   ```

5. Transfer the image to the air-gapped system using removable media such as a USB flash drive.

6. Load the NVIDIA Docker image.

   ```
    $ docker load –i framework.tar
   ```

7. Verify the image is on your system.

   ```
   $ docker images
   ```

Appendix B. Supplemental KVM Information

B.1 Using Cloud-init

Setting Up the Cloud-Config File

Set up a cloud-config file that your guest VM will use when it is created. Refer to https://cloudinit.readthedocs.io/en/latest/topics/examples.html for a description of the file format and options.

Following are the minimum set of options to appear in the cloud-config file.

name

The user's login name. The default file contains a dummy value which must be replaced with your own.

primary_group

Define the primary group. Defaults to a new group created named after the user. The default file contains a dummy value which must be replaced with your own.

groups

Optional. Additional groups to add the user to. Defaults to none.

shell

Shell for the created user

lock_passwd

Defaults to true. Lock the password to disable password login.

passwd

The hash - not the password itself - of the password you want to use for this user.

ssh_authorized_keys

Optional. [list] Add keys to user's authorized keys file.

Setting Up Instance-Data

Set up an `instance-data.json` file with the metadata that you want to include in your VM. Refer to https://cloudinit.readthedocs.io/en/latest/topics/instancedata.html#format-of-instance-data-json for a list of metadata attributes and file format.

Appendix C. Safety

C.1 Safety Information

To reduce the risk of bodily injury, electrical shock, fire, and equipment damage, read this document and observe all warnings and precautions in this guide before installing or maintaining your server product.

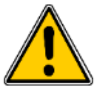
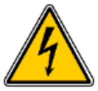
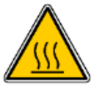
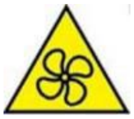
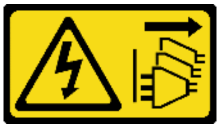
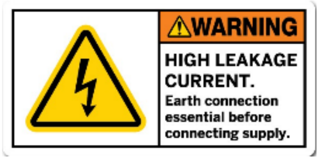
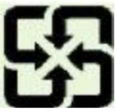
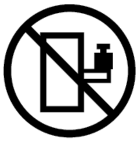
In the event of a conflict between the information in this document and information provided with the product or on the website for a particular product, the product documentation takes precedence.

Your server should be integrated and serviced only by technically qualified persons.

You must adhere to the guidelines in this guide and the assembly instructions in your server manuals to ensure and maintain compliance with existing product certifications and approvals. Use only the described, regulated components specified in this guide. Use of other products I components will void the UL Listing and other regulatory approvals of the product, and may result in noncompliance with product regulations in the region(s) in which the product is sold.

C.2 Safety Warnings and Cautions

To avoid personal injury or property damage, before you begin installing the product, read, observe, and adhere to all of the following safety instructions and information. The following safety symbols may be used throughout the documentation and may be marked on the product and/or the product packaging.

Symbol	Meaning
CAUTION	Indicates the presence of a hazard that may cause minor personal injury or property damage if the CAUTION is ignored.
WARNING	Indicates the presence of a hazard that may result in serious personal injury if the WARNING is ignored.
	Indicates potential hazard if indicated information is ignored.
	Indicates shock hazards that result in serious injury or death if safety instructions are not followed
	Indicates hot components or surfaces.
	Indicates do not touch fan blades, may result in injury.
	Shock hazard – Product might be equipped with multiple power cords. To remove all hazardous voltages, disconnect all power cords.
 WARNING HIGH LEAKAGE CURRENT. Earth connection essential before connecting supply.	High leakage current ground(earth) connection to the Power Supply is essential before connecting the supply.
	Recycle the battery.
	The rail racks are designed to carry only the weight of the server system. Do not use rail-mounted equipment as a workspace. Do not place additional load onto any rail-mounted equipment.

C.3 Intended Application Uses

This product was evaluated as Information Technology Equipment (ITE), which may be installed in offices, schools, computer rooms, and similar commercial type locations. The suitability of this product for other product categories and environments (such as medical, industrial, residential, alarm systems, and test equipment), other than an ITE application, may require further evaluation.

C.4 Site Selection

Choose a site that is:

- Clean, dry, and free of airborne particles (other than normal room dust).
- Well-ventilated and away from sources of heat including direct sunlight and radiators.
- Away from sources of vibration or physical shock.
- In regions that are susceptible to electrical storms, we recommend you plug your system into a surge suppressor and disconnect telecommunication lines to your modem during an electrical storm.
- Provided with a properly grounded wall outlet.
- Provided with sufficient space to access the power supply cord(s), because they serve as the product's main power disconnect.

C.5 Equipment Handling Practices

Reduce the risk of personal injury or equipment damage:

- Conform to local occupational health and safety requirements when moving and lifting equipment.
- Use mechanical assistance or other suitable assistance when moving and lifting equipment.

C.6 Electrical Precautions

C.6.1 Power and Electrical Warnings

Caution: The power button, indicated by the stand-by power marking, DOES NOT completely turn off the system AC power; standby power is active whenever the system is plugged in. To remove power from system, you must unplug the AC power cord from the wall outlet. Make sure all AC power cords are unplugged before you open the chassis, or add or remove any non hot-plug components.

Do not attempt to modify or use an AC power cord if it is not the exact type required. A separate AC cord is required for each system power supply.

Some power supplies in servers use Neutral Pole Fusing. To avoid risk of shock use caution when working with power supplies that use Neutral Pole Fusing.

The power supply in this product contains no user-serviceable parts. Do not open the power supply. Hazardous voltage, current and energy levels are present inside the power supply. Return to manufacturer for servicing.

When replacing a hot-plug power supply, unplug the power cord to the power supply being replaced before removing it from the server.

To avoid risk of electric shock, tum off the server and disconnect the power cords, telecommunications systems, networks, and modems attached to the server before opening it.

C.6.2 Power Cord Warnings

Use certified AC power cords to connect to the server system installed in your rack.

Caution: To avoid electrical shock or fire, check the power cord(s) that will be used with the product as follows:

- Do not attempt to modify or use the AC power cord(s) if they are not the exact type required to fit into the grounded electrical outlets.
- The power cord(s) must meet the following criteria:

The power cord must have an electrical rating that is greater than that of the electrical current rating marked on the product.

The power cord must have safety ground pin or contact that is suitable for the electrical outlet.

The power supply cord(s) is/are the main disconnect device to AC power. The socket outlet(s) must be near the equipment and readily accessible for disconnection.

The power supply cord(s) must be plugged into socket-outlet(s) that is/are provided with a suitable earth ground.

C.7 System Access Warnings

Caution: To avoid personal injury or property damage, the following safety instructions apply whenever accessing the inside of the product:

- Turn off all peripheral devices connected to this product.
- Turn off the system by pressing the power button to off.
- Disconnect the AC power by unplugging all AC power cords from the system or wall outlet.
- Disconnect all cables and telecommunication lines that are connected to the system.
- Retain all screws or other fasteners when removing access cover(s). Upon completion of accessing inside the product, refasten access cover with original screws or fasteners.
- Do not access the inside of the power supply. There are no serviceable parts in the power supply.
- Return to manufacturer for servicing.
- Power down the server and disconnect all power cords before adding or replacing any non hot-plug component.
- When replacing a hot-plug power supply, unplug the power cord to the power supply being replaced before removing the power supply from the server.

Caution: If the server has been running, any installed processor(s) and heat sink(s) may be hot.

Unless you are adding or removing a hot-plug component, allow the system to cool before opening the covers. To avoid the possibility of coming into contact with hot component(s) during a hot-plug installation, be careful when removing or installing the hot-plug component(s).

Caution: To avoid injury do not contact moving fan blades. Your system is supplied with a guard over the fan, do not operate the system without the fan guard in place.

.

C.8 Rack Mount Warnings

Note: The following installation guidelines are required by UL for maintaining safety compliance when installing your system into a rack.

The equipment rack must be anchored to an unmovable support to prevent it from tipping when a server or piece of equipment is extended from it. The equipment rack must be installed according to the rack manufacturer's instructions.

Install equipment in the rack from the bottom up with the heaviest equipment at the bottom of the rack.

Extend only one piece of equipment from the rack at a time.

You are responsible for installing a main power disconnect for the entire rack unit. This main disconnect must be readily accessible, and it must be labeled as controlling power to the entire unit, not just to the server(s).

To avoid risk of potential electric shock, a proper safety ground must be implemented for the rack and each piece of equipment installed in it.

Elevated Operating Ambient- If installed in a closed or multi-unit rack assembly, the operating ambient temperature of the rack environment may be greater than room ambient. Therefore, consideration should be given to installing the equipment in an environment compatible with the maximum ambient temperature (T_{ma}) specified by the manufacturer.

Reduced Air Flow -Installation of the equipment in a rack should be such that the amount of air flow required for safe operation of the equipment is not compromised.

Mechanical Loading- Mounting of the equipment in the rack should be such that a hazardous condition is not achieved due to uneven mechanical loading.

Circuit Overloading- Consideration should be given to the connection of the equipment to the supply circuit and the effect that overloading of the circuits might have on overcurrent protection and supply wiring. Appropriate consideration of equipment nameplate ratings should be used when addressing this concern.

Reliable Earthing- Reliable earthing of rack-mounted equipment should be maintained.

Particular attention should be given to supply connections other than direct connections to the branch circuit (e.g. use of power strips).

C.9 Electrostatic Discharge (ESD)

Caution: ESD can damage drives, boards, and other parts. We recommend that you perform all procedures at an ESD workstation. If one is not available, provide some ESD protection by wearing an antistatic wrist strap attached to chassis ground -- any unpainted metal surface -- on your server when handling parts.

Always handle boards carefully. They can be extremely sensitive to ESO. Hold boards only by their edges. After removing a board from its protective wrapper or from the server, place the board component side up on a grounded, static free surface. Use a conductive foam pad if available but not the board wrapper. Do not slide board over any surface.

C.10 Other Hazards

C.10.1 CALIFORNIA DEPARTMENT OF TOXIC SUBSTANCES CONTROL:

Perchlorate Material – special handling may apply. See www.dtsc.ca.gov/perchlorate.

Perchlorate Material: Lithium battery (CR2032) contains perchlorate. Please follow instructions for disposal.

C.10.2 NICKEL

Ni
NICKEL

NVIDIA Bezel. The bezel's decorative metal foam contains some nickel. The metal foam is not intended for direct and prolonged skin contact. Please use the handles to remove, attach or carry the bezel. While nickel exposure is unlikely to be a problem, you should be aware of the possibility in case you're susceptible to nickel-related reactions.

C.10.3 Battery Replacement

Caution: There is the danger of explosion if the battery is incorrectly replaced. When replacing the battery, use only the battery recommended by the equipment manufacturer.

Dispose of batteries according to local ordinances and regulations. Do not attempt to recharge a battery.

Do not attempt to disassemble, puncture, or otherwise damage a battery.

C.10.4 Cooling and Airflow

Caution: Carefully route cables as directed to minimize airflow blockage and cooling problems. For proper cooling and airflow, operate the system only with the chassis covers installed. Operating the system without the covers in place can damage system parts. To install the covers:

- Check first to make sure you have not left loose tools or parts inside the system.
- Check that cables, add-in cards, and other components are properly installed.
- Attach the covers to the chassis according to the product instructions.

.

Appendix D. Compliance

The NVIDIA DGX-2 is compliant with the regulations listed in this section.

D.1 United States

Product: DGX-2, DGX-2H

Federal Communications Commission (FCC)

FCC Marking (Class A)

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including any interference that may cause undesired operation of the device.

NOTE: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

California Department of Toxic Substances Control: Perchlorate Material - special handling may apply. See www.dtsc.ca.gov/hazardouswaste/perchlorate.

D.2 United States / Canada

Product: DGX-2

cULus Listing Mark

c UL US
LISTED
I.T.E. E145483

D.3 Canada

Product: DGX-2

Innovation, Science and Economic Development Canada (ISED)

CAN ICES-3(A)/NMB-3(A)

The Class A digital apparatus meets all requirements of the Canadian Interference-Causing Equipment Regulation.

Cet appareil numerique de la class A respecte toutes les exigences du Reglement sur le materiel brouilleur du Canada.

D.4 CE

Product: DGX-2

European Conformity; Conformité Européenne (CE)

This is a Class A product. In a domestic environment this product may cause radio frequency interference in which case the user may be required to take adequate measures.

This device bears the CE mark in accordance with Directive 2014/53/EU.

This device complies with the following Directives:

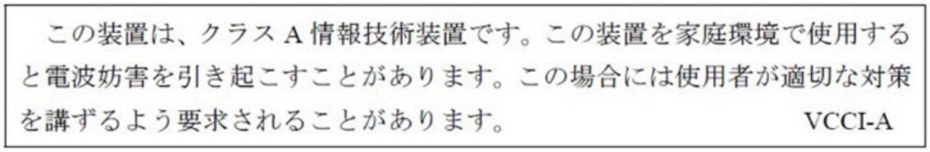
- EMC Directive A, I.T.E Equipment.

- Low Voltage Directive for electrical safety.
- RoHS Directive for hazardous substances.
- Energy-related Products Directive (ErP).

D.5 Japan

Product: DGX-2, DGX-2H

Voluntary Control Council for Interference (VCCI)



この装置は、クラス A 情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。 VCCI-A

This is a Class A product.

In a domestic environment this product may cause radio interference, in which case the user may be required to take corrective actions. VCCI-A

2008年、日本における製品含有表示方法、JISC0950が公示されました。製造事業者は、2006年7月 1 日

以降に販売される電気・電子機器の特定

化学物質の含有に付きまして情報提供を義務付けられました。製品の部材表示に付きましては、

A Japanese regulatory requirement, defined by specification JIS C 0950, 2008, mandates that manufacturers provide Material Content Declarations for certain categories of electronic products offered for sale after July 1, 2006.
To view the JIS C 0950 material declaration for this product, visit www.nvidia.com

Japan RoHS Material Content Declaration

日本工業規格JIS C
0950:2008により、2006年7月1日以降に販売される特定分野の電気および電子機器について、製造者による含有物質の表示が義務付けられます。

機器名称：DGX-2

主な分類	特定化学物質記号					
	Pb	Hg	Cd	Cr(VI)	PBB	PBDE
筐体	除外項目	0	0	0	0	0
プリント基板	除外項目	0	0	0	0	0
プロセッサー	除外項目	0	0	0	0	0
マザーボード	除外項目	0	0	0	0	0
電源	除外項目	0	0	0	0	0
システムメモリ	除外項目	0	0	0	0	0
ハードディスクドライブ	除外項目	0	0	0	0	0
機械部品 (ファン、ヒートシンク、ベゼル..)	除外項目	0	0	0	0	0
ケーブル/コネクター	除外項目	0	0	0	0	0
はんだ付け材料	0	0	0	0	0	0
フラックス、クリームはんだ、ラベル、その他消耗品	0	0	0	0	0	0

注：

1.「0」は、特定化学物質の含有率が日本工業規格JIS C 0950:2008に記載されている含有率基準値より低いことを示します。

2.「除外項目」は、特定化学物質が含有マークの除外項目に該当するため、特定化学物質について、日本工業規格JIS C
0950:2008に基づく含有マークの表示が不要であることを示します。

3.「0.1wt%超」または「0.01wt%超」は、特定化学物質の含有率が日本工業規格JIS C 0950:2008 に記載されている含有率基準値を超えていることを示します。

A Japanese regulatory requirement, defined by specification JIS C 0950: 2008, mandates that manufacturers provide Material Content Declarations for certain categories of electronic products offered for sale after July 1, 2006.

Product Model Number: DGX-2

Major Classification	Symbols of Specified Chemical Substance					
	Pb	Hg	Cd	Cr(VI)	PBB	PBDE
Chassis	Exempt	0	0	0	0	0
PCA	Exempt	0	0	0	0	0
Processor	Exempt	0	0	0	0	0
Motherboard	Exempt	0	0	0	0	0
Power supply	Exempt	0	0	0	0	0
System memory	Exempt	0	0	0	0	0
Hard drive	Exempt	0	0	0	0	0
Mechanical parts (fan, heat sink, bezel…)	Exempt	0	0	0	0	0
Cables/Connectors	Exempt	0	0	0	0	0
Soldering material	0	0	0	0	0	0
Flux, Solder Paste, label and other consumable materials	0	0	0	0	0	0

Notes:

1. “0” indicates that the level of the specified chemical substance is less than the threshold level specified in the standard, JIS C 0950: 2008.

2. “Exempt” indicates that the specified chemical substance is exempt from marking and it is not required to display the marking for that specified chemical substance per the standard, JIS C 0950: 2008.

3. “Exceeding 0.1wt%” or “Exceeding 0.01wt%” is entered in the table if the level of the specified chemical substance exceeds the threshold level specified in the standard, JIS C 0950: 2008.

D.6 Australia and New Zealand

Product: DGX-2

Australian Communications and Media Authority

This product meets the applicable EMC requirements for Class A, I.T.E equipment

D.7 China

Product: DGX-2

China RoHS Material Content Declaration

产品中有害物质的名称及含量
The Table of Hazardous Substances and their Content
根据中国《电器电子产品有害物质限制使用管理办法》
as required by China's Management Methods for Restricted of Hazardous Substances Used in Electrical and Electronic Products

部件名称 Parts	有害物质 Hazardous Substances					
	铅 (Pb)	汞 (Hg)	镉 (Cd)	六价铬 (Cr(VI))	多溴联苯 (PBB)	多溴联苯醚 (PBDE)
机箱 Chassis	X	O	O	O	O	O
印刷电路部件 PCA	X	O	O	O	O	O
处理器 Processor	X	O	O	O	O	O
主板 Motherboard	X	O	O	O	O	O
电源设备 Power supply	X	O	O	O	O	O
存储设备 System memory	X	O	O	O	O	O
硬盘驱动器 Hard drive	X	O	O	O	O	O
机械部件（风扇、散热器、面板等） Mechanical parts (fan, heat sink, bezel…)	X	O	O	O	O	O
线材/连接器 Cables/Connectors	X	O	O	O	O	O
焊接金属 Soldering material	O	O	O	O	O	O
助焊剂，锡膏，标签及其他耗材 Flux, Solder Paste, label and other consumable materials	O	O	O	O	O	O

本表格依据SJ/T 11364-2014 的规定编制
The table according to SJ/T 11364-2014

O：表示该有害物质在该部件所有均质材料中的含量均在GB/T 26572-2011 标准规定的限量要求以下。
O: Indicates that this hazardous substance contained in all of the homogeneous materials for this part is below the limit requirement in GB/T 26572-2011.
X：表示该有害物质至少在该部件的某一均质材料中的含量超出GB/T 26572-2011 标准规定的限量要求。
X: Indicates that this hazardous substance contained in at least one of the homogeneous materials used for this part is above the limit requirement in GB/T 26572-2011.

此表中所有名称中含 "X" 的部件均符合欧盟 RoHS 立法。
All parts named in this table with an "X" are in compliance with the European Union's RoHS Legislation.

注：环保使用期限的参考标识取决于产品正常工作的温度和湿度等条件
Note: The referenced Environmental Protection Use Period Marking was determined according to normal operating use conditions of the product such as temperature and humidity.

Notice

THE INFORMATION IN THIS GUIDE AND ALL OTHER INFORMATION CONTAINED IN NVIDIA DOCUMENTATION REFERENCED IN THIS GUIDE IS PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE INFORMATION FOR THE PRODUCT, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the product described in this guide shall be limited in accordance with the NVIDIA terms and conditions of sale for the product.

THE NVIDIA PRODUCT DESCRIBED IN THIS GUIDE IS NOT FAULT TOLERANT AND IS NOT DESIGNED, MANUFACTURED OR INTENDED FOR USE IN CONNECTION WITH THE DESIGN, CONSTRUCTION, MAINTENANCE, AND/OR OPERATION OF ANY SYSTEM WHERE THE USE OR A FAILURE OF SUCH SYSTEM COULD RESULT IN A SITUATION THAT THREATENS THE SAFETY OF HUMAN LIFE OR SEVERE PHYSICAL HARM OR PROPERTY DAMAGE (INCLUDING, FOR EXAMPLE, USE IN CONNECTION WITH ANY NUCLEAR, AVIONICS, LIFE SUPPORT OR OTHER LIFE CRITICAL APPLICATION). NVIDIA EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR SUCH HIGH RISK USES. NVIDIA SHALL NOT BE LIABLE TO CUSTOMER OR ANY THIRD PARTY, IN WHOLE OR IN PART, FOR ANY CLAIMS OR DAMAGES ARISING FROM SUCH HIGH RISK USES.

NVIDIA makes no representation or warranty that the product described in this guide will be suitable for any specified use without further testing or modification. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to ensure the product is suitable and fit for the application planned by customer and to do the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this guide. NVIDIA does not accept any liability related to any default, damage, costs or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this guide, or (ii) customer product designs.

Other than the right for customer to use the information in this guide with the product, no other license, either expressed or implied, is hereby granted by NVIDIA under this guide. Reproduction of information in this guide is permissible only if reproduction is approved by NVIDIA in writing, is reproduced without alteration, and is accompanied by all associated conditions, limitations, and notices.

Trademarks

NVIDIA, the NVIDIA logo, and DGX-2 are trademarks and/or registered trademarks of NVIDIA Corporation in the Unites States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

DU-09130-011_v15

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