



DGX Spark User Guide

NVIDIA Corporation

Nov 12, 2025

Release Notes

1	Overview	3
2	Getting Started	5
2.1	DGX Spark Release Notes	5
2.1.1	Current Software Versions	5
2.1.2	Known Issues	5
2.2	Known Issues	5
2.2.1	Use the supplied power adapter for optimal performance	5
2.2.2	nvidia-smi reports “Memory-Usage: Not Supported”	6
2.2.3	Memory reporting differences with unified memory architecture	6
2.3	Hardware Overview	7
2.3.1	System Overview	7
2.3.1.1	Component Descriptions	8
2.3.2	Physical Specifications	8
2.3.2.1	Form Factor	8
2.3.2.2	Environmental Requirements	8
2.3.3	Connectivity and I/O	8
2.3.3.1	Rear Panel	8
2.3.4	Performance Specifications	9
2.3.4.1	Compute Performance	9
2.3.4.2	AI/ML Capabilities	9
2.3.5	Power and Thermal Management	9
2.3.5.1	Power Requirements	9
2.3.5.2	Thermal Management	10
2.4	Initial Setup - First Boot	10
2.4.1	What You’ll Do	10
2.4.2	Choose how to access your system during initial setup	10
2.4.3	Get Ready	11
2.4.4	Run the First-Time Setup	11
2.4.4.1	Getting Started	12
2.4.4.2	What to Expect During Setup	12
2.4.5	Next Steps	14
2.5	System Configuration and Operation	15
2.5.1	System Overview	15
2.5.1.1	Flexible Access and Usage	15
2.5.1.2	Key Capabilities	15
2.5.1.3	System Architecture	15
2.5.1.4	Software	16
2.5.1.5	Getting Started	16
2.5.2	UEFI Settings	16
2.5.2.1	Accessing UEFI	16
2.5.2.2	UEFI Documentation	17
2.5.2.3	General UEFI Configuration	17

2.5.2.4	Troubleshooting	17
2.5.3	Spark Stacking	17
2.5.3.1	Connecting DGX Spark Systems into a Virtual Cluster	17
2.6	Software	20
2.6.1	DGX OS	20
2.6.1.1	Overview	20
2.6.1.2	Release Cadence	20
2.6.2	NVIDIA Sync	20
2.6.2.1	Overview	20
2.6.2.2	Installation	21
2.6.2.3	Connecting to Your Spark	21
2.6.2.4	Configuring Applications	21
2.6.3	DGX Dashboard	22
2.6.3.1	Overview	22
2.6.3.2	Integrated JupyterLab	23
2.6.3.3	Accessing the Dashboard	23
2.6.4	NVIDIA Container Runtime for Docker	23
2.6.4.1	Overview	23
2.6.4.2	Installation	23
2.6.4.3	Usage	24
2.6.4.4	Validation	24
2.6.4.5	Troubleshooting	25
2.6.5	NGC	26
2.6.5.1	Overview	26
2.6.5.2	Getting Started	26
2.6.5.3	Basic Usage	27
2.6.5.4	Common Workflows	28
2.6.5.5	Best Practices	28
2.6.5.6	Troubleshooting	28
2.7	Common Use Cases	29
2.8	OS and Component Update Guide	29
2.8.1	Update Methods	30
2.8.2	Using DGX Dashboard for Updates	30
2.8.3	Manual System Updates	30
2.8.4	Update Best Practices	31
2.9	System Recovery	31
2.9.1	System Recovery Overview	31
2.9.1.1	Recovery Requirements	31
2.9.1.2	Recovery Process Steps	31
2.10	Third-Party License Notices	35
2.10.1	LIBICONV Library	36
2.11	Notices	49
2.11.1	Notice	49
2.11.2	Trademarks	50

This guide is also available for download as a [PDF](#).

Chapter 1. Overview

The DGX Spark is NVIDIA's compact AI computer designed for developers, data scientists, and AI researchers who need powerful computing capabilities for AI development and deployment.

Chapter 2. Getting Started

2.1. DGX Spark Release Notes

This section provides release notes for the DGX Spark, including information about new features, known issues, and software version updates.

2.1.1. Current Software Versions

The following table shows the current version information for the DGX Spark software stack.

Component	Version
Operating System	NVIDIA DGX OS 7.2.3
NVIDIA GPU Driver	580.95.05
NVIDIA CUDA Toolkit	13.0.2

2.1.2. Known Issues

For an updated list of known issues, see [Known Issues](#).

2.2. Known Issues

This topic provides a summary of known issues with DGX Spark systems.

2.2.1. Use the supplied power adapter for optimal performance

For optimal performance, use the supplied power adapter with the DGX Spark system. Using a different adapter may reduce performance, prevent boot, or cause unexpected shutdowns.

2.2.2. nvidia-smi reports “Memory-Usage: Not Supported”

On iGPU platforms, `nvidia-smi` will display “Memory-Usage: Not Supported” even though per-process GPU memory is listed. This is expected because iGPUs do not have dedicated framebuffer memory.

2.2.3. Memory reporting differences with unified memory architecture

DGX Spark systems use a unified memory architecture (UMA), where the GPU shares system memory (DRAM) with the CPU and other compute engines. This design reduces latency and allows larger amounts of memory to be used for GPU workloads. On UMA systems, the CPU can dynamically manage DRAM contents, including freeing up memory by swapping pages between DRAM and the system’s SWAP area. However, the `cudaMemGetInfo` API does not account for memory that could potentially be reclaimed from SWAP. As a result, the memory size reported by `cudaMemGetInfo` may be smaller than the actual allocatable memory, since the CPU may be able to release additional DRAM pages by moving them to SWAP.

To more accurately estimate the amount of allocatable device memory on DGX Spark platforms, CUDA application developers should consider the possibility of DRAM reclamation via SWAP and not rely solely on the values returned by `cudaMemGetInfo`. The following provides an example implementation using C standard libraries:

```
#include <stdio.h>

int getAvailableMemory(long *availableMemoryKb, long *freeSwapKb) {
    FILE *meminfoFile = NULL;
    char lineBuffer[256];
    long hugeTlbTotalPages = -1;
    long hugeTlbFreePages = -1;
    long hugeTlbPageSize = -1;

    if (availableMemoryKb == NULL || freeSwapKb == NULL) {
        return 1;
    }

    meminfoFile = fopen("/proc/meminfo", "r");
    if (meminfoFile == NULL) {
        return 1;
    }

    *availableMemoryKb = -1;
    *freeSwapKb = -1;

    while (fgets(lineBuffer, sizeof(lineBuffer), meminfoFile)) {
        long value;

        if (sscanf(lineBuffer, "MemAvailable: %ld kB", &value) == 1) {
            *availableMemoryKb = value;
        } else if (sscanf(lineBuffer, "SwapFree: %ld kB", &value) == 1) {
            *freeSwapKb = value;
        } else if (sscanf(lineBuffer, "HugePages_Total: %ld", &value) == 1) {
```

(continues on next page)

(continued from previous page)

```

        hugeTlbTotalPages = value;
    } else if (sscanf(lineBuffer, "HugePages_Free: %ld", &value) == 1) {
        hugeTlbFreePages = value;
    } else if (sscanf(lineBuffer, "Hugepagesize: %ld kB", &value) == 1) {
        hugeTlbPageSize = value;
    }

    if (*availableMemoryKb != -1 &&
        *freeSwapKb != -1 &&
        hugeTlbTotalPages != -1 &&
        hugeTlbFreePages != -1 &&
        hugeTlbPageSize != -1) {
        break;
    }
}

fclose(meminfoFile);

if (hugeTlbTotalPages != 0 && hugeTlbTotalPages != -1) {
    *availableMemoryKb = hugeTlbFreePages * hugeTlbPageSize;

    // Hugetlbfs pages are not swappable.
    *freeSwapKb = 0;
}

return 0;
}

```

As a workaround for debugging purposes, you can flush the buffer cache manually with the following command:

```
sudo sh -c 'sync; echo 3 > /proc/sys/vm/drop_caches'
```

After flushing the cache, restart your application.

2.3. Hardware Overview

Powered by the NVIDIA Grace Blackwell architecture, DGX Spark enables developers, researchers, and data scientists to prototype, deploy, and fine-tune large AI models on their desktop. This section provides information about the hardware components and specifications.

2.3.1. System Overview

The DGX Spark features:

- ▶ NVIDIA Grace Blackwell architecture with integrated GPU and CPU
- ▶ 20-core Arm processor with high-performance cores
- ▶ 128 GB unified system memory
- ▶ Compact desktop form factor

- ▶ Advanced connectivity including Wi-Fi 7, 10 GbE, and ConnectX-7
- ▶ Support for AI models up to 200 billion parameters (or 405B for dual-Spark configuration)

2.3.1.1 Component Descriptions

The DGX Spark includes the following components:

Table 1: Component Specifications

Component	Specification
GPU	NVIDIA Blackwell Architecture with 5th Generation Tensor Cores, 4th Generation RT Cores
CPU	20-core Arm processor (10 Cortex-X925 + 10 Cortex-A725)
Memory	128 GB LPDDR5x unified system memory, 256-bit interface, 4266 MHz, 273 GB/s bandwidth
Storage	1 TB or 4 TB NVMe M.2 with self-encryption
Network	1x RJ-45 (10 GbE), ConnectX-7 Smart NIC, Wi-Fi 7, Bluetooth 5.4
Connectivity	4x USB Type-C, 1x HDMI 2.1a, HDMI multichannel audio
Video Processing	1x NVENC, 1x NVDEC

2.3.2. Physical Specifications

2.3.2.1 Form Factor

- ▶ **Chassis Type:** Small form factor (SFF)
- ▶ **Dimensions:** 150 mm (L) x 150 mm (W) x 50.5 mm (H)
- ▶ **Weight:** 1.2 kg (2.6 lbs)

2.3.2.2 Environmental Requirements

Table 2: Environmental Specifications

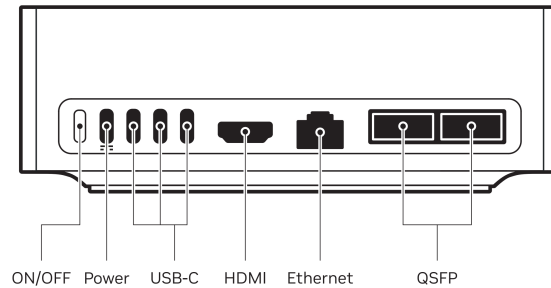
Specification	Value
Ideal Operating Temperature	0°C to 35°C (32°F to 95°F)
Operating Humidity	10% to 90% (non-condensing)
Operating Altitude	Up to 3,000 meters (9,843 feet)

2.3.3. Connectivity and I/O

2.3.3.1 Rear Panel

- ▶ Power button

- ▶ 4x USB Type-C (one for power delivery)
- ▶ 1x HDMI 2.1a display connector
- ▶ 1x RJ-45 Ethernet connector (10 GbE)
- ▶ 2x QSFP Network connectors (ConnectX-7)



2.3.4. Performance Specifications

2.3.4.1 Compute Performance

- ▶ **AI Compute:** Up to 1,000 TOPS (trillion operations per second) inference and up to 1 PFLOP (petaFLOP) at FP4 precision with sparsity
- ▶ **CUDA Cores:** 6,144
- ▶ **Copy Engines:** 2 (enables simultaneous data transfers to and from GPU memory, improving throughput for AI workloads)
- ▶ **CPU Performance:** 20 cores (10 Cortex-X925 + 10 Cortex-A725)
- ▶ **Memory Bandwidth:** 273 GB/s
- ▶ **Memory Channels:** 16 channels (256 bit) LPDDR5X 8533

2.3.4.2 AI/ML Capabilities

- ▶ **Model Support:** AI models up to 200 billion parameters
- ▶ **Tensor Performance:** 5th Generation Tensor Cores with FP4 support
- ▶ **Framework Support:** PyTorch, TRT-LLM, and other AI frameworks
- ▶ **Use Cases:** Inference, deployment, and fine-tuning of large language models

2.3.5. Power and Thermal Management

2.3.5.1 Power Requirements

- ▶ **Power Supply:** 240W external power supply (included)
 - ▶ GB10 SOC Thermal Design Power (TDP) is 140W
 - ▶ 100W is available for other system components (ConnectX-7, Wi-Fi, SSD, USB-C, etc.)

- ▶ **Usage Requirement:** Use of the provided 240W power supply is required for optimal performance. Using a different or lower-rated power supply may result in reduced system performance, failure to boot, or unexpected shutdowns.
- ▶ **Input Voltage:** Standard AC power input

2.3.5.2 Thermal Management

- ▶ **Cooling Solution:** Integrated thermal management system
- ▶ **Form Factor:** Compact design optimized for desktop placement
- ▶ **Operating Temperature:** 0°C to 35°C (32°F to 95°F) *Preliminary specifications*

2.4. Initial Setup - First Boot

This guide walks you through setting up your DGX Spark for the first time. You'll choose how to access your system during the initial setup, and run the first-time setup utility to configure everything. The access method you choose is only for completing the initial setup - after setup is complete, you can access your DGX Spark any way you like: locally with a monitor and keyboard, over the local network from another computer, or a mix of both.

2.4.1. What You'll Do

This setup process includes:

- ▶ Choosing how to access the system during initial setup (with a display, or as a network appliance)
- ▶ Preparing your system and connections
- ▶ Running the first-time setup utility to configure your system

2.4.2. Choose how to access your system during initial setup

To complete the initial setup, you'll need to access your DGX Spark. You can do this in one of two ways:

With a Display (Local Setup)

- ▶ Connect keyboard and mouse via USB or Bluetooth
- ▶ Connect a display to work directly on the system
- ▶ Follow the setup wizard on screen

Over the Network (as a Network Appliance)

- ▶ Access the system over your local network from another computer
- ▶ Use another computer to complete setup via web browser
- ▶ No Spark display or keyboard required for the setup process

Note

This choice is only about how you'll complete the initial setup process. After setup is finished, you can access your DGX Spark however you prefer. You're not locked into your original choice.

2.4.3. Get Ready

Important

The DGX Spark device starts up immediately when power is applied. Please attach all peripherals (display, keyboard, mouse, network, etc.) before connecting the power supply.

Before starting, ensure you have:

- ▶ A fast, reliable internet connection (Wi-Fi or Ethernet) to download required updates during the initial setup process. Connections using captive portals (such as hotel or airport Wi-Fi) or those prone to disconnections (like phone hotspots) are not recommended. If you do not have access to a stable connection, consider downloading the system recovery media and using [System Recovery](#) to install the latest software for your DGX Spark.
- ▶ For Local Setup: A display, keyboard, and mouse connected (or available using Bluetooth).
- ▶ For Network Setup: A computer on the same network to access the setup interface.
- ▶ Power connected to the system (the system will start automatically when power is applied).

Note

Display Troubleshooting: Some displays may have trouble with Spark out of the box. If you are connecting over USB-C/DisplayPort and there is no display, try using HDMI instead.

Note

If you plan to use a wired network connection, plug in the network cable before starting the installation. This helps avoid connection issues later in the process.

2.4.4. Run the First-Time Setup

The first-time setup utility will guide you through:

- ▶ Powering on and initializing the system
- ▶ Selecting your preferred setup mode
- ▶ Downloading and installing critical updates
- ▶ Completing your initial configuration

Warning

Critical: Do not shut down or reboot the system during the update process. The installation cannot be interrupted once the download begins, and powering down during updates can cause system damage.

2.4.4.1 Getting Started

The way you start the installation depends on your chosen access method:

With a Display (Local Setup)

1. Power on the system
2. The first-time setup utility will start automatically on the connected display
3. Use your wired keyboard and mouse (already connected) to navigate
4. If a keyboard or mouse is not detected, you will be prompted to put your Bluetooth devices in pairing mode:

USB devices can be plugged in at any time and should start working, even if detected improperly. Bluetooth devices can be put into pairing mode and will generally still pair while on the “Get Started” screen (exception - keyboards that require a passcode to type in won’t work on this screen). Once you click on “Get Started,” Bluetooth pairing stops, so you will have to power cycle to try again.

5. Follow the on-screen prompts to complete the setup process

Over the Network (as a Network Appliance)

1. Power on the system. This creates a Wi-Fi hotspot that you will use to connect to the system and continue the setup process. The SSID and password for the Wi-Fi hotspot are printed on a sticker attached to the Quick Start Guide included with your DGX Spark’s packaging
2. From another computer, connect to the Spark’s Wi-Fi hotspot using the SSID and password provided on the Quick Start Guide. A captive portal page will open in the default web browser on your computer. If it does not open automatically, use your browser to navigate to the Spark’s system setup page listed on the Quick Start Guide.
3. Follow the on-screen prompts to continue the setup process.

When the DGX Spark joins your home network, its Wi-Fi hotspot will turn off and your computer will reconnect to the device through your Wi-Fi network to resume the setup process. If you are not able to connect to the DGX Spark after it joins your home network, you must connect a display/keyboard/mouse to continue.

Note

It may take as long as 10 minutes for the DGX Spark to install all of its updates and join your home network.

2.4.4.2 What to Expect During Setup

The first-time setup utility will guide you through several configuration steps. Simply follow the on-screen prompts to complete each step.

Setup Process Steps:

1. Language and Time Zone Selection

Choose your preferred language and time zone settings for the system. Note that the input fields will filter as you type.

2. Keyboard Layout Selection (Local Setup only)

Select your keyboard layout (e.g. US keyboard vs. Russian keyboard). This screen only appears when using a display during setup.

3. Terms and Conditions

Review and accept the terms and conditions to continue with the installation.

4. User Account Creation

Create your username and password for system access.

5. Information Sharing Settings (Optional)

Configure analytics and crash reporting preferences. You can skip this step if desired.

6. Wi-Fi Network Selection

Select your Wi-Fi network. This step is automatically skipped if an Ethernet cable that is providing internet access is connected.

7. Wi-Fi Password

Enter the password for your selected Wi-Fi network.

8. Joining Wi-Fi Network

The system connects to your Wi-Fi network and tears down the access point. Your computer will automatically reconnect to your default network.

Note

Network Connection Issues:

- ▶ If your computer automatically reconnects to the same network as the Spark, the installation should continue seamlessly
- ▶ If not, you'll need to connect your computer to the same network as the Spark while the setup app is waiting for the network setup process to complete
- ▶ If the setup fails, you must connect a display/keyboard/mouse to continue
- ▶ The modal instructs you to try to reconnect to the Spark's hotspot and try again. This will work if the Spark actually failed to join the network (e.g. wrong password) versus your laptop can't communicate with the Spark
- ▶ If you DO NOT see the hotspot available when this error modal appears, that means the Spark did join the network but your laptop can't communicate with it. This could be because:
 - ▶ Device isolation
 - ▶ You failed to join the same network as your Spark
 - ▶ mDNS does not work on your network due to its configuration (e.g. a complex corporate network)

9. Software Download and Installation

After connecting to the network, the system will automatically download and install the full software image. This process can take some time, and the system may reboot more than once before it's complete. If you are setting up over the network from another computer, you will not be able to access the device during this time. The process may continue for up to 10 minutes after the interface shows that the device is rebooting to finish the setup

⚠ Warning

Do not shut down or reboot the system during this process. The installation cannot be interrupted once the download begins.

10. Installation Complete

The device will reboot automatically when installation is complete, and you can then use it normally.

2.4.5. Next Steps

Congratulations! Your DGX Spark is now ready to use. Here are the recommended ways to access and start working with your system:

Access Your System

- ▶ **Locally:** Use your DGX Spark like any desktop computer with a monitor, keyboard, and mouse
- ▶ **Over the Network:** Connect to your DGX Spark from another computer on the same network using the NVIDIA Sync tool (see [NVIDIA Sync](#)), SSH, or remote desktop
- ▶ **Mix Both:** Use whatever access method fits your current workflow - you can switch freely between local and network access
- ▶ **Dashboard Access:** Use the built-in DGX Dashboard for system monitoring, updates, and JupyterLab access. See [DGX Dashboard](#) for more information

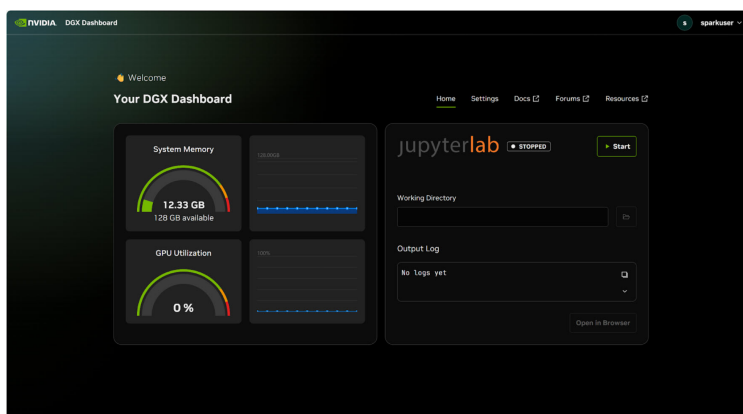


Fig. 1: The DGX Dashboard provides an intuitive interface for system monitoring and development

Additional Resources

- ▶ Visit the [NVIDIA Spark Developer Portal](https://build.nvidia.com/spark) at <https://build.nvidia.com/spark> for the latest guides, tutorials, and updates

- ▶ Refer to *DGX Spark Release Notes* for the latest software updates and features
- ▶ See *Known Issues* for troubleshooting common problems

Your DGX Spark is now ready to power your AI development and deployment workflows!

2.5. System Configuration and Operation

Configuring and operating your DGX Spark effectively is key to delivering consistent results across AI/ML workflows. This section provides an overview of the platform, recommended UEFI settings, clustering procedures, and foundational security guidance to help you deploy, manage, and scale with confidence.

2.5.1. System Overview

Powered by the NVIDIA Grace Blackwell architecture, DGX Spark enables developers, researchers, and data scientists to prototype, deploy, and fine-tune large AI models on their desktop.

2.5.1.1 Flexible Access and Usage

The DGX Spark is designed for maximum flexibility in how you access and use it. You can seamlessly switch between different access methods based on your needs:

- ▶ **Local Access:** Connect a keyboard, mouse, and monitor to work directly on the system
- ▶ **Network Access:** Access your system from another computer on the same network using SSH, NVIDIA Sync, or remote desktop tools
- ▶ **Hybrid Usage:** Mix and match access methods - work locally one day and over the network the next, or even simultaneously

All access methods are fully supported and equally capable. Your DGX Spark adapts to your workflow, whether you're working at your desk with a monitor or accessing it remotely as a network appliance on the same network.

2.5.1.2 Key Capabilities

Your DGX Spark enables you to:

- ▶ **Run Inference:** Deploy models for real-time AI applications
- ▶ **Develop AI Models:** Train and fine-tune models with up to 200 billion parameters
- ▶ **Process Data:** Handle large datasets with high-performance computing
- ▶ **Experiment Freely:** Test new ideas without cloud computing costs
- ▶ **Scale Workloads:** Connect multiple systems for larger projects

2.5.1.3 System Architecture

The DGX Spark is built on NVIDIA's Grace Blackwell architecture, providing:

- ▶ **Unified Memory:** 128 GB of high-bandwidth memory for large models
- ▶ **High-Performance Computing:** 20-core ARM64-based processor with integrated GPU
- ▶ **Advanced Connectivity:** Wi-Fi 7, 10 GbE, CX7 NIC, and multiple I/O options

- **Compact Form Factor:** 150mm x 150mm x 50.5mm desktop design

For detailed hardware specifications, see [Hardware Overview](#).

2.5.1.4 Software

Your system comes pre-configured with:

- **NVIDIA DGX OS:** Optimized operating system for AI workloads
- **Development Tools:** CUDA, cuDNN, and NVIDIA's development ecosystem
- **Container Support:** Docker and NVIDIA Container Runtime for easy deployment
- **NGC Integration:** Access to NVIDIA's container registry

For detailed software information, see [Software](#).

2.5.1.5 Getting Started

To begin using your DGX Spark:

1. **Initial Setup:** Follow the [Initial Setup - First Boot](#) to configure your system
2. **Explore Examples:** Try sample workloads to understand capabilities
3. **Configure Development Environment:** Set up your preferred tools and frameworks
4. **Start Building:** Begin your AI development projects

Note

For the most up-to-date tutorials and examples, visit <https://build.nvidia.com/spark>. This site is regularly updated with new content and serves as the primary resource for practical guides and use cases.

2.5.2. UEFI Settings

This topic provides guidance on accessing and configuring the UEFI settings for the DGX Spark system. While there are no Spark-specific features that require UEFI configuration, you may need to access the UEFI for general system configuration or troubleshooting purposes.

2.5.2.1 Accessing UEFI

Important

To access the UEFI setup menu, you must be using a keyboard connected directly to the DGX Spark device.

To access the UEFI setup menu:

1. Power on or restart the system
2. Immediately press **Esc** or **Del** during the boot process and keep holding it until the UEFI setup menu appears
3. The UEFI setup menu will appear

Note

The timing for pressing Esc or Del is critical. Press the key as soon as the system starts booting, before the operating system begins loading.

2.5.2.2 UEFI Documentation

The DGX Spark system uses an AMI UEFI. For detailed information about UEFI settings and configuration options, refer to the AMI UEFI Manual.

2.5.2.3 General UEFI Configuration

While there are no Spark-specific UEFI requirements, you may need to configure general system settings such as:

- ▶ Basic settings (date and time, system language)
- ▶ Power-on behavior
- ▶ Device boot order
- ▶ Security settings (TPM, secure Boot, passwords)
- ▶ Network configuration
- ▶ RAM disk configuration
- ▶

Important

Always save your UEFI configuration changes before exiting from the **Save and Exit** menu to restart the system with your new settings applied.

2.5.2.4 Troubleshooting

If you experience issues after making UEFI changes:

- ▶ Revert to UEFI defaults and restart
- ▶ Apply changes incrementally to identify problematic settings
- ▶ Update to the latest UEFI version if available
- ▶ Consult the AMI UEFI manual for specific setting descriptions
- ▶ For additional troubleshooting guidance and support options, see [spark-maintenance-troubleshooting](#)

2.5.3. Spark Stacking

2.5.3.1 Connecting DGX Spark Systems into a Virtual Cluster

2.5.3.1.1 Overview

This guide explains how to connect two DGX Spark systems into a virtual compute cluster using simplified networking configuration and a QSFP/CX7 cable for high-performance interconnect.

The goal is to enable distributed workloads across Grace Blackwell GPUs using MPI (for inter-process CPU communication) and NCCL v2.28.3 (for GPU-accelerated collective operations).

Additional Information can be found in the [Connect Two Sparks](#) playbook.

2.5.3.1.2 System Requirements

Before you begin, ensure the following:

- ▶ Both DGX Spark systems have Grace Blackwell GPUs, are connected to each other using a QSFP/CX7 cable, and are running Ubuntu 24.04 (or later) with NVIDIA drivers installed
- ▶ The systems have internet access for initial software setup
- ▶ You have sudo/root access on both systems

2.5.3.1.3 Setup Networking Between Spark Systems

2.5.3.1.3.1 Option 1: Automatic IP assignment (recommended)

Follow these steps on **both DGX Spark nodes** to configure network interfaces using netplan. The following commands should be run in a terminal session (either local or remote).

1. Download the netplan configuration file

```
sudo wget -O /etc/netplan/40-cx7.yaml https://github.com/NVIDIA/dgx-spark-  
→playbooks/raw/main/nvidia/connect-two-sparks/assets/cx7-netplan.yaml
```

2. Set appropriate permissions on the configuration file

```
sudo chmod 600 /etc/netplan/40-cx7.yaml
```

3. Apply the netplan configuration

```
sudo netplan apply
```

2.5.3.1.3.2 Option 2: Manual IP assignment (advanced)

Follow these steps to manually assign IP addresses for dedicated cluster networking.

1. On Node 1, assign a static IP address and bring up the interface

```
sudo ip addr add 192.168.100.10/24 dev enP2p1s0f1np1  
sudo ip link set enP2p1s0f1np1 up
```

2. On Node 2, assign a static IP address and bring up the interface

```
sudo ip addr add 192.168.100.11/24 dev enP2p1s0f1np1  
sudo ip link set enP2p1s0f1np1 up
```

3. From Node 1, verify connectivity by testing connection to Node 2

```
ping -c 3 192.168.100.11
```

4. From Node 2, verify connectivity by testing connection to Node 1

```
ping -c 3 192.168.100.10
```

2.5.3.1.4 Run the DGX Spark Discovery Script

This step will automatically identify interconnected DGX Spark systems and set up SSH authentication without requiring a password.

The following commands should be run in a terminal session (either local or remote) on both nodes.

On both nodes:

1. **Download the discovery script**

```
wget https://github.com/NVIDIA/dgx-spark-playbooks/raw/refs/heads/main/
→nvidia/connect-two-sparks/assets/discover-sparks
```

2. **Make the script executable**

```
chmod +x discover-sparks
```

3. **Run the discovery script**

```
./discover-sparks
```

Example output:

```
Found: 192.168.100.10 (spark-1b3b.local)
Found: 192.168.100.11 (spark-1d84.local)

Copying your SSH public key to all discovered nodes using ssh-copy-id.
You may be prompted for your password on each node.
Copying SSH key to 192.168.100.10 ...
Copying SSH key to 192.168.100.11 ...
nvidia@192.168.100.11's password:

SSH key copy process complete. These two sparks can now talk to each
→other.
```

2.5.3.1.5 Install Required Software and Verify the Configuration

With the networking configured and the systems able to communicate with each other, the next step is to install the required software for distributed workloads and run test workloads to verify that GPU-to-GPU communication is working correctly and to measure performance across the stacked systems.

For complete instructions on building NCCL, running the NCCL test suite, and interpreting the results, see the [NCCL Stacked Sparks](#) playbook.

2.5.3.1.6 Troubleshooting

- ▶ Ensure the QSFP/CX7 interface is active and used for IP assignment
- ▶ Verify connectivity between nodes via `ping`
- ▶ Check your interface bindings with `ip a` and `ethtool`
- ▶ If the discovery script fails, manually verify SSH connectivity between nodes
- ▶ For additional troubleshooting guidance and support options, see `spark-maintenance-troubleshooting`

2.5.3.1.7 Next Steps

Once tested, this configuration can be scaled to support:

- ▶ Job orchestration with Slurm or Kubernetes
- ▶ Containerized execution with Singularity or Docker

2.6. Software

The DGX Spark comes with a comprehensive software stack optimized for AI development, machine learning, and data science workflows. This section provides detailed information about the included software components and their configuration.

2.6.1. DGX OS

2.6.1.1 Overview

NVIDIA DGX OS is a customized Linux distribution that provides a stable, tested, and supported operating system foundation for running AI, machine learning, and analytics applications on DGX systems. It includes platform-specific optimizations, drivers, and diagnostic tools tailored for NVIDIA hardware.

DGX OS serves as the underlying operating system for your DGX Spark, providing:

- ▶ A robust Linux foundation optimized for AI workloads
- ▶ Pre-configured drivers and system settings for NVIDIA hardware
- ▶ Security updates and system maintenance capabilities
- ▶ Compatibility with the broader NVIDIA software ecosystem

Note

For more information about DGX OS, see the official documentation at: <https://docs.nvidia.com/dgx/dgx-os-7-user-guide/introduction.html>

2.6.1.2 Release Cadence

DGX OS follows a regular release schedule with updates typically provided twice per year, around February and August, for the first two years after initial release. Additional updates and security patches are provided between major releases and throughout the support lifecycle.

2.6.2. NVIDIA Sync

2.6.2.1 Overview

NVIDIA Sync is a system tray utility that provides a simple way to access your DGX Spark system remotely from another machine. Whether your DGX Spark is connected to a monitor or not, NVIDIA Sync makes it easy to connect your favorite development tools and access the system from anywhere on your network.

2.6.2.2 Installation

1. Download the latest version of NVIDIA Sync from <https://build.nvidia.com/spark/connect-to-your-spark/sync>. Installers are available for Windows, macOS, and Linux.
2. Run the installer.

2.6.2.3 Connecting to Your Spark

The first time you launch NVIDIA Sync, you'll be prompted to connect to your DGX Spark.

Provide your DGX Spark's *machine name* and your *login credentials*.

About the machine name

The machine name is your DGX Spark hostname with `.local` appended, such as `spark-xxxx.local`. You can find the default hostname on the Quick Start Guide that came in the box.

The `.local` address uses mDNS (multicast DNS) to automatically locate your DGX Spark on the network without needing to know its IP address. This is particularly useful if your router periodically reassigns IP addresses.

For Windows users: mDNS requires Bonjour Print Services from Apple. If you have iTunes or other Apple software installed, you likely already have this. Otherwise, you can download it from Apple's website. Alternatively, you can try using just the hostname without `.local` (such as `spark-xxxx`), though this method is less reliable on modern networks.

Why .local might not work: `.local` hostnames may not work in enterprise networks with strict security policies, networks that block multicast traffic, or other restricted network environments.

Using an IP address instead: If `.local` hostnames do not work, you will need to use the IP address. To find the IP address, physically log in to your DGX Spark and click the network icon in the top right corner of the Ubuntu desktop. Select **Settings** from the dropdown menu, then navigate to the **Network** section. The IP address will be displayed under your active connection. Alternatively, you can log in to your router's administration console to view connected devices and their IP addresses.

2.6.2.4 Configuring Applications

After connecting to your DGX Spark, NVIDIA Sync will automatically scan your local machine for compatible applications that can connect remotely to the DGX Spark. Select the applications you want to configure.

The selected applications will be configured to use your DGX Spark as a remote compute resource. This means you run the application interface locally on your workstation, but all compute workloads (such as code execution, model training, and data processing) run on the DGX Spark. Your files, code, and development environment are synchronized to the DGX Spark, giving you the full power of the DGX Spark's GPUs while using familiar local tools.

All configured applications are accessible from the NVIDIA Sync system tray icon. You can change your application selections at any time from the NVIDIA Sync settings window. If an application is marked as "unavailable" in the settings, it means the application is not installed on your local machine.

2.6.2.4.1 Available Applications

The following applications can be detected and configured by NVIDIA Sync if they are installed on your local machine:

- ▶ **NVIDIA AI Workbench:** Download from <https://www.nvidia.com/workbench>
- ▶ **Cursor:** Download from <https://cursor.com/downloads>
- ▶ **VS Code:** Download from <https://code.visualstudio.com/download>

The following connection methods are always available and do not require any local application installation:

- ▶ **DGX Dashboard** - Web-based system monitoring and management interface
- ▶ **SSH Terminal** - Command-line access with SSH keys automatically managed by NVIDIA Sync

2.6.3. DGX Dashboard

2.6.3.1 Overview

The DGX Spark comes with a built-in dashboard that provides an overview of the system's current operational metrics, the ability to apply updates, change some system settings, and access local Jupyter Notebooks.

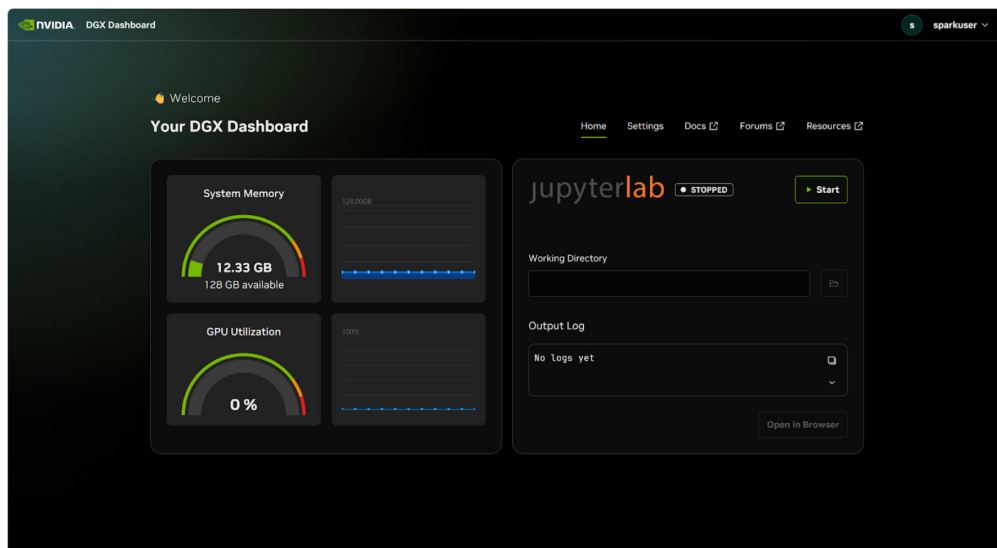


Fig. 2: The DGX Dashboard provides real-time system monitoring and integrated JupyterLab access

Note

To run updates and change the device name, you must have *sudo* access. The account created during initial setup will have access.

2.6.3.2 Integrated JupyterLab

The dashboard includes an integrated JupyterLab instance that provides a convenient development environment:

- ▶ When started, JupyterLab creates a virtual environment in the specified working directory and automatically installs a set of recommended packages
- ▶ If you enter a new working directory and start JupyterLab, a new environment will be created
- ▶ Each user account on the device is assigned a port located in `/opt/nvidia/dgx-dashboard-service/jupyterlab_ports.yaml`
- ▶ To access JupyterLab remotely, you must tunnel it just like the dashboard itself. The port to tunnel is in the ports file. Using NVIDIA Sync, this tunnel is managed for you automatically and just works

2.6.3.3 Accessing the Dashboard

The dashboard can be accessed locally by clicking on the “Show Apps” button in the bottom left corner of the Ubuntu desktop. Then, in the app grid, select the “DGX Dashboard” shortcut to open the dashboard in your default web browser.

Remotely, the dashboard can be accessed using NVIDIA Sync or via a manually created SSH tunnel. If using NVIDIA Sync, after connecting, simply click on the “DGX Dashboard” button and the dashboard will open in your default web browser at `http://localhost:11000`.

To manually access over SSH, first open a tunnel, e.g., `ssh -L 11000:localhost:11000 <username>@<IP or spark-abcd.local>`. Then, open the dashboard in your web browser at `http://<spark-host-ip>:11000`.

2.6.4. NVIDIA Container Runtime for Docker

2.6.4.1 Overview

The NVIDIA Container Runtime enables Docker containers to access GPU resources on DGX Spark systems. This runtime acts as a bridge between Docker and the NVIDIA drivers, allowing containers to utilize GPU acceleration for AI/ML workloads, CUDA applications, and other GPU-accelerated software.

Key benefits: - Seamless GPU access within containers - Automatic driver and library management - Support for multi-GPU configurations - Compatibility with popular container orchestration platforms

The runtime works in conjunction with the NVIDIA Container Toolkit, which provides the necessary components to expose GPU devices and CUDA libraries to containerized applications.

2.6.4.2 Installation

The NVIDIA Container Toolkit is preinstalled and configured on DGX Spark systems. This includes:

- ▶ NVIDIA Container Runtime
- ▶ Docker integration
- ▶ GPU device access configuration
- ▶ CUDA library management

The runtime is ready to use out of the box for running GPU-accelerated containers.

2.6.4.2.1 Optional: Add User to Docker Group

By default, Docker requires *sudo* privileges to run commands. Adding your user to the docker group allows you to run Docker commands without *sudo*, which provides:

- ▶ **Convenience:** No need to type *sudo* before every Docker command
- ▶ **Better workflow:** Seamless integration with development tools and scripts
- ▶ **Reduced friction:** Faster iteration when working with containers

To add your user to the docker group:

```
sudo usermod -aG docker $USER
newgrp docker
```

Note: This step is optional. You can continue using Docker with *sudo* if you prefer not to modify group memberships.

2.6.4.3 Usage

2.6.4.3.1 Basic GPU Access

Run a container with GPU access using the *-gpus* flag:

```
docker run -it --runtime=nvidia --gpus=all nvcr.io/nvidia/cuda:13.0.1-devel-
↳ubuntu24.04 nvidia-smi
```

This command: - Runs an interactive container (*-it*) - Enables access to all GPUs and uses the NVIDIA Container Runtime for Docker (*--runtime=nvidia* *--gpus=all*) - Uses the NVIDIA CUDA development image - Executes *nvidia-smi* to display GPU information

2.6.4.3.2 Set GPU Capabilities

Control which GPU capabilities are available to the container:

```
docker run -it --runtime=nvidia --gpus '"capabilities=compute,utility"' nvcr.
↳io/nvidia/cuda:13.0.1-devel-ubuntu24.04 nvidia-smi
```

2.6.4.4 Validation

2.6.4.4.1 Test GPU Access

1. Run the test command to verify GPU access:

```
docker run -it --runtime=nvidia --gpus=all nvcr.io/nvidia/cuda:13.0.1-
↳devel-ubuntu24.04 nvidia-smi
```

Expected output should show: - GPU device information - Driver version - CUDA version - Memory usage and temperature

2. Check runtime configuration:

```
docker info | grep -A 10 "Runtimes"
```

3. Verify NVIDIA runtime is available:

```
docker run --rm --runtime=nvidia nvcr.io/nvidia/cuda:13.0.1-devel-
↳ubuntu24.04 nvidia-smi
```

2.6.4.4.2 Inspect Container GPU Access

Check what GPU resources are available inside a running container:

```
docker run -it --runtime=nvidia --gpus=all nvcr.io/nvidia/cuda:13.0.1-devel-
↳ubuntu24.04 bash
# Inside the container:
nvidia-smi
ls /dev/nvidia*
```

2.6.4.5 Troubleshooting

2.6.4.5.1 Runtime Not Found

If you encounter “runtime not found” errors:

1. Verify NVIDIA Container Toolkit is installed:

```
nvidia-ctk --version
```

2. Check Docker daemon configuration:

```
cat /etc/docker/daemon.json
```

3. Restart Docker service:

```
sudo systemctl restart docker
```

2.6.4.5.2 Driver/Container CUDA Mismatch

If you see CUDA version mismatches:

1. Check host CUDA driver version:

```
nvidia-smi
```

2. Use a container image with compatible CUDA version:

```
docker run -it --runtime=nvidia --gpus=all nvcr.io/nvidia/cuda:13.0.1-
↳devel-ubuntu24.04 nvidia-smi
```

2.6.4.5.3 Permission Issues

If you encounter permission errors:

1. Ensure your user is in the docker group (if not using sudo):

```
groups $USER
```

2. Check device permissions:

```
ls -la /dev/nvidia*
```

3. Verify Docker daemon has access to GPU devices:

```
sudo docker run -it --runtime=nvidia --gpus=all nvcr.io/nvidia/cuda:13.0.1-
→1-devel-ubuntu24.04 nvidia-smi
```

2.6.4.5.4 Container Startup Issues

If containers fail to start:

1. Check Docker logs:

```
docker logs <container_id>
```

2. Verify GPU devices are available on host:

```
ls /dev/nvidia*
```

3. Test with a minimal container:

```
docker run --rm --runtime=nvidia --gpus=all nvcr.io/nvidia/cuda:13.0.1-
→devel-ubuntu24.04 echo "GPU test successful"
```

2.6.5. NGC

2.6.5.1 Overview

NVIDIA GPU Cloud (NGC) is a comprehensive registry of GPU-optimized containers, pre-trained models, and AI/ML software that enables rapid development and deployment of AI applications. For DGX Spark users, NGC provides access to the latest frameworks, tools, and optimized environments specifically designed for the Grace Blackwell architecture.

Key benefits for DGX Spark users:

- **Optimized Containers:** Pre-configured environments with the latest AI/ML frameworks, CUDA, and libraries optimized for Grace Blackwell GPUs
- **Pre-trained Models:** Access to state-of-the-art models and model collections for various AI tasks
- **Rapid Development:** Skip complex environment setup and focus on your AI/ML projects
- **Cutting-edge Software:** Access to the latest NVIDIA software stack and experimental features

NGC is particularly valuable for DGX Spark users because it provides the most current and optimized software stack for this new platform, ensuring you have access to the latest performance optimizations and features.

2.6.5.2 Getting Started

2.6.5.2.1 Create an NGC Account

1. Visit the [NGC website](#)
2. Click **Sign Up** and create a free account
3. Verify your email address

4. Complete your profile information

2.6.5.2.2 Generate an API Key

1. Log in to your NGC account
2. Navigate to **Setup** -> **API Key**
3. Click **Generate API Key**
4. Copy and securely store your API key

Note

Your API key is required for pulling containers and accessing NGC resources. Keep it secure and never share it publicly.

2.6.5.2.3 Install NGC CLI (Optional)

The NGC CLI provides convenient command-line access to NGC resources. Your DGX Spark system requires the ARM64 version of the NGC CLI which can be found on the **ARM64 Linux** tab at <https://org.nvidia.com/setup/installers/cli>.

For more information on installing and using the NGC CLI, see [NGC CLI Documentation](#).

2.6.5.2.4 Authenticate with Docker

Configure Docker to access NGC registries:

```
# Login to NGC with Docker
docker login nvcr.io
# Username: $oauthtoken
# Password: <your-api-key>
```

2.6.5.3 Basic Usage

2.6.5.3.1 Pull and Run a Container

Start with a popular AI/ML framework container:

```
# Pull a PyTorch container optimized for Grace Blackwell
docker pull nvcr.io/nvidia/pytorch:24.08-py3

# Run the container with GPU access
docker run -it --gpus=all nvcr.io/nvidia/pytorch:24.08-py3
```

2.6.5.3.2 Explore Available Resources

Browse NGC resources through the web interface:

- **Containers:** AI/ML frameworks, development environments, and specialized tools
- **Models:** Pre-trained models for computer vision, natural language processing, and more
- **Helm Charts:** Kubernetes deployment configurations

- **Jupyter Notebooks:** Interactive tutorials and examples

2.6.5.4 Common Workflows

2.6.5.4.1 Development Environment

Use NGC containers as your development environment:

```
# Run a development container with persistent storage
docker run -it --gpus=all \
  -v /path/to/your/project:/workspace \
  nvcr.io/nvidia/pytorch:24.08-py3
```

2.6.5.4.2 Model Inference and Training

Access pre-trained models and training scripts:

```
# Pull a model from NGC
ngc registry model download-version nvidia/bert-base-uncased:1

# Or use models directly in containers
docker run -it --gpus=all --runtime=nvidia \
  nvcr.io/nvidia/pytorch:24.08-py3
```

2.6.5.5 Best Practices

2.6.5.5.1 Container Management

- **Pin Versions:** Use specific container tags for reproducible environments
- **Regular Updates:** Periodically update to newer container versions for latest optimizations
- **Resource Limits:** Set appropriate memory and CPU limits for your workloads

2.6.5.5.2 Data Persistence

- **Volume Mounts:** Mount your data directories into containers for persistence
- **Model Storage:** Store trained models and checkpoints outside containers
- **Configuration:** Keep configuration files in version control

2.6.5.5.3 Security

- **API Key Security:** Store your NGC API key securely and rotate it regularly
- **Container Scanning:** Scan containers for vulnerabilities before use
- **Network Security:** Use appropriate network configurations for your environment

2.6.5.6 Troubleshooting

2.6.5.6.1 Common Issues

Authentication Failures

```
# Verify your API key is correct
docker login nvcr.io
# Check if your account has access to the requested resource
```

Container Pull Issues

```
# Check network connectivity to NGC registry
curl -I https://nvcr.io

# View container catalog on NGC website
# Visit https://catalog.ngc.nvidia.com/containers

# Or use NGC CLI to list available containers
ngc registry image list nvidia/*

# Try pulling with verbose output to see specific error
docker pull nvcr.io/nvidia/pytorch:24.08-py3
```

GPU Access Problems

```
# Verify NVIDIA Container Runtime is installed
docker run --rm --runtime=nvidia --gpus=all nvcr.io/nvidia/cuda:13.0.1-devel-
↳ubuntu24.04 nvidia-smi
```

2.6.5.6.2 Getting Help

- ▶ **NGC Documentation:** Visit the [NGC documentation](#)
- ▶ **Community Forums:** Join the [NVIDIA Developer Forums](#)
- ▶ **Additional Support:** For troubleshooting guidance and support options, see [spark-maintenance-troubleshooting](#)

2.7. Common Use Cases

The DGX Spark is designed to support a wide range of AI, machine learning, and data science workflows. Visit <https://build.nvidia.com/spark> for practical guides to help you get started. That site is regularly updated with new content and information and will be the single source of truth for your Spark device.

2.8. OS and Component Update Guide

This section provides guidance for updating the operating system, software components, and firmware on your DGX Spark. The DGX Spark runs on NVIDIA DGX OS, which is an Ubuntu-based Linux distribution optimized for AI workloads.

Note

Founders Edition Only: The update information in this guide applies only to the DGX Spark Founders Edition. Devices from other manufacturers may have different update procedures.

2.8.1. Update Methods

We strongly recommend using the DGX Dashboard for all system updates on your [spark]. The dashboard ensures your system stays up to date with the latest NVIDIA-optimized components and configurations. While standard Ubuntu package management methods will work, the dashboard provides the most reliable and tested update path for your DGX system.

2.8.2. Using DGX Dashboard for Updates

The DGX Dashboard is the **primary and recommended** way to perform system updates on your DGX Spark. It provides a centralized interface for:

- ▶ Viewing available system updates
- ▶ Installing security patches and system updates
- ▶ Managing NVIDIA driver updates
- ▶ Managing firmware updates
- ▶ Monitoring update status and progress

For detailed information about using the DGX Dashboard, including how to access it and perform updates, see [DGX Dashboard](#).

Warning

Before performing any system or firmware updates:

- ▶ Ensure your system is connected to a stable power source
- ▶ Close all running applications and save your work
- ▶ Have a recovery plan in place
- ▶ Schedule updates during maintenance windows when possible

2.8.3. Manual System Updates

Note: While manual updates using standard Ubuntu package management commands will work, we strongly recommend using the DGX Dashboard for all updates to ensure optimal system performance and compatibility.

For advanced users or when the DGX Dashboard is not available, you can perform system updates manually using the following steps:

1. Open a remote or local terminal on the DGX Spark device.
2. Run the following commands:

```
sudo apt update
sudo apt dist-upgrade
sudo fwupdmgr refresh
sudo fwupdmgr upgrade
sudo reboot
```

These commands will:

- ▶ Update the package lists and upgrade all installed packages (including OS components and drivers)
- ▶ Refresh the firmware metadata and upgrade all firmware components
- ▶ Reboot the system to apply updates

2.8.4. Update Best Practices

- ▶ **Use the DGX Dashboard:** Always prefer the DGX Dashboard for system updates to ensure compatibility and optimal performance
- ▶ **Regular updates:** Check for updates regularly, especially security patches
- ▶ **Backup before major updates:** Always backup critical data before major system changes
- ▶ **Stable power:** Ensure your system has a stable power supply during updates
- ▶ **Maintenance windows:** Schedule updates during planned maintenance windows when possible

2.9. System Recovery

This section provides information about system recovery procedures for your DGX Spark.

Note

Founders Edition Only: This recovery information applies only to the DGX Spark Founders Edition. Devices from other manufacturers may have different recovery procedures.

2.9.1. System Recovery Overview

The system recovery process for the DGX Spark allows you to restore the operating system and firmware to their original factory state. This process is useful when dealing with system corruption, fatal configuration errors, or other issues that prevent normal system operation.

2.9.1.1 Recovery Requirements

Before beginning the recovery process, ensure you have the following:

- ▶ A USB flash drive with 16GB or larger capacity
- ▶ A keyboard and display connected to your DGX Spark
- ▶ Access to download the recovery media from NVIDIA support

2.9.1.2 Recovery Process Steps

The recovery process involves downloading recovery media, creating a bootable USB drive, and following a series of UEFI configuration steps. Follow these steps carefully to ensure successful system recovery.

1. **Download Recovery Media:**

- ▶ Download the recovery media archive file (tar.gz format) from <https://developer.nvidia.com/downloads/dgx-spark/dgx-spark-recovery-image-1.91.51-1.tar.gz>
- ▶ Save the file to your local system

2. Create Recovery USB Drive:

- ▶ Download and unzip the recovery media archive file
- ▶ Insert the USB drive into your system
- ▶ Within the extracted files, use one of the following commands to create the recovery drive:

Note

Creating the recovery drive requires administrator privileges. On Windows systems, this is typically done by running the command as an administrator (i.e. from an elevated PowerShell or Command Prompt). If supported by the host system, you will be prompted by the system to run the command with elevated privileges.

Windows:

```
CreateUSBKey.cmd
```

Linux:

```
CreateUSBKey.sh
```

MacOS:

```
CreateUSBKeyMacOS.sh
```

Warning

This process will erase all data on your USB drive. Ensure you have backed up any important data before proceeding.

3. Boot from Recovery USB Drive:

- ▶ Disconnect any external storage devices from the DGX Spark
- ▶ Connect the USB drive to one of the USB ports on your DGX Spark
- ▶ If your DGX Spark is powered off, boot the device into UEFI settings by holding down the **Del** key immediately after powering on the device

4. Restore UEFI Defaults:

- ▶ Tap the **Right Arrow** key to select the “Save & Exit” page within the UEFI settings
- ▶ Tap **Down Arrow** to select “Restore Defaults” and select “Yes” in response to “Load Optimized Defaults”
- ▶ Select “Save Changes and Reset” - the device will reboot
- ▶ While the device is rebooting, hold down the **Del** key to re-enter UEFI settings a second time

5. Enable Secure Boot:

- ▶ Use the **Right Arrow** key to select “Security”

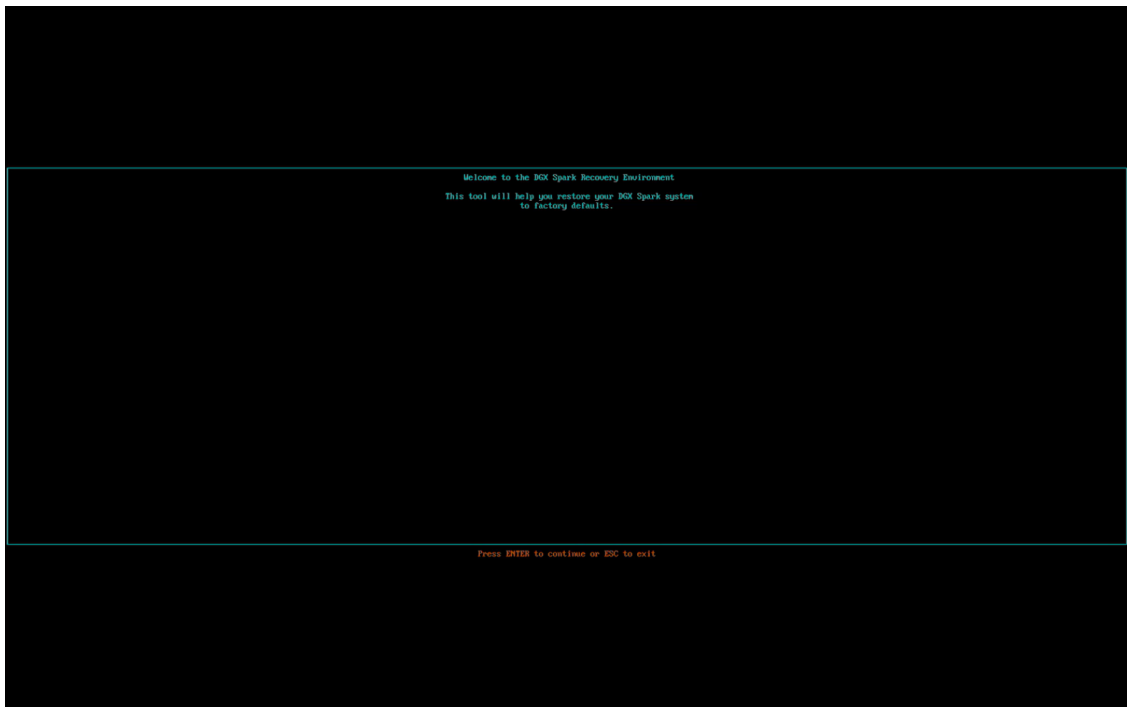
- ▶ Confirm that “Secure Boot” is set to “Enabled”
- ▶ Select “Restore Factory Keys”
- ▶ Use the arrow keys to select the “Save and Exit” BIOS screen, and select “Save Changes and Reset”

6. Boot from Recovery Media:

- ▶ While the device is rebooting, hold down the Del key to re-enter UEFI settings a third time
- ▶ Tap the Right Arrow key to select the “Save & Exit” page in UEFI settings
- ▶ Tap Down Arrow to move to the “Boot Override” section
- ▶ Select the USB drive and tap Enter
- ▶ The device will reboot using the USB drive - follow the on-screen steps to update your firmware and re-install the OS on the device’s SSD

7. Restore System from Recovery Environment

- ▶ On the welcome screen, press Enter to continue. To cancel the process, press Esc.



- ▶ On the warning screen, select [EXIT] to restart without proceeding, or select [START RECOVERY] to begin reflashing the SSD. Be aware that this will completely erase the internal SSD on the DGX Spark.



- Monitor the Recovery Progress screen for detailed output from the recovery process and the progress of the SSD reflash.



- When the process is complete, review the progress screen and follow the prompt to continue. Scroll or page through the recovery output if needed, for example during customer support.

```

Recovery Process - Complete
DGX Spark System Recovery Completed

Recovery Output--
0x698d1e02 bytes (25 GB, 23 GiB) copied, 35 s, 782 MB/s
0x76f98b3776 bytes (26 GB, 24 GiB) copied, 41 s, 626 MB/s
0x81c5dc5699 bytes (27 GB, 25 GiB) copied, 43 s, 627 MB/s
0x172102624 bytes (28 GB, 26 GiB) copied, 45 s, 624 MB/s
0x991ba923d8 bytes (29 GB, 27 GiB) copied, 47 s, 618 MB/s
0xc6af4f7d72 bytes (30 GB, 28 GiB) copied, 50 s, 620 MB/s
0x1138512896 bytes (31 GB, 29 GiB) copied, 54 s, 573 MB/s
0x2212254759 bytes (32 GB, 30 GiB) copied, 58 s, 566 MB/s
0x20259e2544 bytes (33 GB, 31 GiB) copied, 59 s, 557 MB/s
0x1537f73660 bytes (34 GB, 32 GiB) copied, 59 s, 589 MB/s
0x253ab98192 bytes (35 GB, 33 GiB) copied, 61 s, 581 MB/s
0x603627960 bytes (36 GB, 34 GiB) copied, 63 s, 579 MB/s
351 records in
351 records out
36403627960 bytes (36 GB, 34 GiB) copied, 65.4722 s, 556 MB/s
[Task 1] 47.0 G-Feb-2023
Pass 1: Checking inode, blocks, and sizes
inode 286 extent tree (at least 1) could be narrower. Optimized yes
inode 292166 extent tree (at least 2) could be narrower. Optimized yes
inode 288372 extent tree (at least 2) could be narrower. Optimized yes
Pass 2: Optimize extent trees
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
root ===== FILE SYSTEM INFO =====
root: 279385/2228224 files (1.5% non-configurable), 8376811/8887450 blocks
non-root: 147.0 G-Feb-2023
Resizing the filesystem on /dev/nvme0n1p2 to 1000129145 (4k) blocks.
The filesystem on /dev/nvme0n1p2 is now 1000129145 (4k) blocks long.
Set UID for each partition.
+ Set FFI UID
FFI UID=====4247
+ Get Beta IID
Beta IID=====3-a208-4a28-e7fe-379da26d3fa
Adjust zeta/zetaib
+ Adjust FFIs/beta-grub.cfg
Mount UUID: 345d166-f7ae-6513-bc79-fdb0bcMeta9 Replacing with: cca5df3a-2e28-4a28-e7fe-379da26d3fa
Updated grub.cfg with new UUID
Updating boot loader entries
bootCurrent: 0000
Fileset: 1 records
bootDriver: 0000,0003,BIOS
Boot0002=UEFI; Pxe IPv4 Bootable PCle 10 GHz Family Controller PCIeBoot(0x72)/Pci(0x0,dm0)/Pci(0x0,dm0)/MBC(cchbf00000903)/PIV#(0,0,0,0,0,0,0,0)
Boot0003=UEFI; CD/DVD IDE Drive, Partition 1          UbuntuGAS(Dm0)-977e-622e-c2eb-caef82621fa,00003ff1600000000/cGB(0,0,0,cGB(0,0,0,0,0,0,0,0))/HUB,hdcdScsiTar,hddbm,bcbcm00000
Boot0006=Ubuntu(FR0-F70)(1,CPT,addFc2fc8-894d-8428-8a28-98277c7ad88e,e8080,e9590000)/File(EFT)/ubuntu@v61.efi)
Successfully added Ushin entry and set it as first boot option
Installing MZ Capsize Update
GRUB mapping: m88247a-3875-frcf-hlf-d7965e2d45c:33589235 d41a238a-5fb8-66d4-9981-ab9da95a596:1 3413769-f6ad-46ad-f5ac-921f9968f65:33571331
Installing MZ CapSize Update
Copying oc.fw.cap: cap version 33571331 < cart version 33571331
Copying oc.mzcap: cap version 33589235 < cart version 33589235
Copying tpm.cap: cap version 11572269 < cart v relion
Copying ukpl.cap: cap version hdd > cart version 1
Copying jnlm.oc.mzcap: cap version 33571331 < cart version 4.e(f1)
Setting Delimitation Flag to 0x00000000 for Capsize Update
Recovery command completed

Press ENTER to continue...

```

- On the final screen, confirm that the DGX Spark has been reset to factory state and press Enter to restart the device.

The image shows a black screen with white text. At the top left, the text "Recovery Complete" is displayed. In the center, the text "Recovery Completed" is shown. Below this, a message states: "Your BOE Spark has been restored to factory defaults." followed by "Press ENTER to restart your BOE Spark". At the bottom center, there is a red rectangular button with the text "(RESTART)" in white.

2.10. Third-Party License Notices

This NVIDIA product contains third party software that is being made available to you under their respective open source software licenses. Some of those licenses also require specific legal information

to be included in the product. This section provides such information.

2.10.1. LIBICONV Library

NVTelemetry is using the libiconv library, which is licensed under the GNU Lesser General Public License (LGPL). libiconv source code can be obtained from the following link: <http://repo.download.nvidia.com/baseos/sources/libiconv-1.15.tar.gz>.

The libiconv library is provided under the following terms:

GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

(continues on next page)

(continued from previous page)

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

1. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying,

(continues on next page)

(continued from previous page)

distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

2. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users

(continues on next page)

(continued from previous page)

can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

3. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

4. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

5. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice;

(continues on next page)

(continued from previous page)

keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

6. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 1. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

7. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

(continues on next page)

(continued from previous page)

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status

(continues on next page)

(continued from previous page)

of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

8. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of

(continues on next page)

(continued from previous page)

it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

9. Termination.

(continues on next page)

(continued from previous page)

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

10. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

11. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the

(continues on next page)

(continued from previous page)

Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

12. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

(continues on next page)

(continued from previous page)

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

13. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

14. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

15. Revised Versions of this License.

(continues on next page)

(continued from previous page)

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

16. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

17. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

18. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms,

(continues on next page)

(continued from previous page)

reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

(continues on next page)

(continued from previous page)

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>

2.11. Notices

2.11.1. Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation (“NVIDIA”) makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

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

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

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

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

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or

services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

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

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

2.11.2. Trademarks

NVIDIA, the NVIDIA logo, DGX, DGX-1, DGX-2, DGX A100, DGX H100/H200, DGX B200, DGX Station, DGX Station A100, and DGX Spark are trademarks and/or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

©2022-2025, NVIDIA Corporation