



# NVIDIA DOCA

## Installation Guide

# Table of Contents

<b>Chapter 1. Introduction.....</b>	<b>1</b>
1.1. Supported Platforms.....	1
1.2. Hardware Prerequisites.....	2
1.3. DOCA Packages.....	2
1.4. Supported Operating System.....	3
1.5. Supported Kernel Versions.....	3
<b>Chapter 2. SDK Manager.....</b>	<b>5</b>
<b>Chapter 3. Manual BlueField Image Installation.....</b>	<b>6</b>
3.1. Installation Files.....	6
3.2. Installing Software on Host.....	7
3.3. Installing Software on DPU.....	9
3.3.1. Installing Full DOCA Image on DPU.....	9
3.3.1.1. Option 1 - No Pre-defined Password.....	10
3.3.1.2. Option 2 - Set Pre-defined Password.....	10
3.3.2. Installing Full DOCA Image on Multiple DPUs.....	11
3.3.3. Installing DOCA Local Repo Package on DPU.....	11
3.4. Upgrading Firmware.....	11
3.5. Post-installation Procedure.....	13
<b>Chapter 4. Building Your Own BFB Installation Image.....</b>	<b>14</b>
<b>Chapter 5. Setting Up Build Environment for Developers.....</b>	<b>15</b>
<b>Chapter 6. Installing CUDA on NVIDIA Converged Accelerator.....</b>	<b>16</b>
6.1. Configuring Operation Mode.....	16
6.2. Downloading and Installing CUDA Toolkit and Driver.....	17
6.3. GPUDirect RDMA.....	18
6.4. DPDK GPUDEV.....	18

---

# Chapter 1. Introduction

There are two ways to install the NVIDIA BlueField DPU software:

- ▶ Using the SDK Manager which provides a GUI/CLI for full DPU software installation
- ▶ Manual installation with a step-by-step procedure

## 1.1. Supported Platforms

Model Number	Description
MBF2H322A-AEEOT	NVIDIA® BlueField®-2 P-Series DPU 25GbE Dual-Port SFP56, PCIe Gen4 x8, Crypto Enabled, 8GB on-board DDR, 1GbE OOB management, Tall Bracket, HHHH
MBF2H322A-AENOT	NVIDIA BlueField-2 P-Series DPU 25GbE Dual-Port SFP56, PCIe Gen4 x8, Crypto Disabled, 8GB on-board DDR, 1GbE OOB management, Tall Bracket, HHHH
MBF2H332A-AEEOT	NVIDIA BlueField-2 P-Series DPU 25GbE Dual-Port SFP56, PCIe Gen3/4 x8, Crypto Enabled, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, HHHH
MBF2H332A-AENOT	NVIDIA BlueField-2 P-Series DPU 25GbE Dual-Port SFP56, PCIe Gen3/4 x8, Crypto Disabled, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, HHHH
MBF2H516A-CEEOT	NVIDIA BlueField-2 P-Series DPU 100GbE Dual-Port QSFP56, PCIe Gen4 x16, Crypto Enabled, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, FHHL
MBF2H516A-CENOT	NVIDIA BlueField-2 P-Series DPU 100GbE Dual-Port QSFP56, PCIe Gen4 x16, Crypto Disabled, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, FHHL
MBF2H516A-EEEOT	NVIDIA BlueField-2 P-Series DPU 100GbE/EDR/HDR100 VPI Dual-Port QSFP56, PCIe Gen4 x16, Crypto Enabled, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, FHHL
MBF2H516A-EENOT	NVIDIA BlueField-2 P-Series DPU 100GbE/EDR VPI Dual-Port QSFP56; PCIe Gen4 x16; Crypto Disabled; 16GB on-board DDR; 1GbE OOB management; FHHL
MBF2H516B-CENOT	NVIDIA BlueField-2 P-Series BF2500 DPU Controller, 100GbE Dual-Port QSFP56, PCIe Gen4 x16, Crypto Disabled, 16GB on-board DDR, 1GbE OOB Management, Tall Bracket, FHHL

Model Number	Description
MBF2H516B-EENOT	NVIDIA BlueField-2 P-Series BF2500 DPU Controller, 100GbE/EDR/HDR100 VPI Dual-Port QSFP56, PCIe Gen4 x16, Crypto Disabled, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, FHHL
MBF2M322A-AEEOT	NVIDIA BlueField-2 E-Series DPU 25GbE Dual-Port SFP56, PCIe Gen3/4 x8, Crypto, 8GB on-board DDR, 1GbE OOB management, Tall Bracket, HHHH
MBF2M322A-AENOT	NVIDIA BlueField-2 E-Series DPU 25GbE Dual-Port SFP56, PCIe Gen3/4 x8, Crypto Disabled, 8GB on-board DDR, 1GbE OOB management, Tall Bracket, HHHH
MBF2M332A-AEEOT	NVIDIA BlueField-2 E-Series DPU 25GbE Dual-Port SFP56, PCIe Gen4 x8, Crypto, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, HHHH
MBF2M332A-AENOT	NVIDIA BlueField-2 E-Series DPU 25GbE Dual-Port SFP56, PCIe Gen4 x8, Crypto Disabled, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, HHHH
MBF2M516A-CEEOT	NVIDIA BlueField-2 E-Series DPU 100GbE Dual-Port QSFP56; PCIe Gen4 x16; Crypto Enabled; 16GB on-board DDR; 1GbE OOB management; FHHL
MBF2M516A-CENOT	NVIDIA BlueField-2 E-Series DPU 100GbE Dual-Port QSFP56, PCIe Gen4 x16, Crypto Disabled, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, FHHL
MBF2M516A-EEEEOT	NVIDIA BlueField-2 E-Series DPU 100GbE/EDR/HDR100 VPI Dual-Port QSFP56, PCIe Gen4 x16, Crypto Enabled, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, FHHL
MBF2M516A-EENOT	NVIDIA BlueField-2 E-Series DPU 100GbE/EDR/HDR100 VPI Dual-Port QSFP56; PCIe Gen4 x16; Crypto Disabled; 16GB on-board DDR; 1GbE OOB management; FHHL
900-21004-0030-000	NVIDIA BlueField-2 A30X, P1004 SKU 205, Generic, GA100, 24GB HBM2e, PCIe Passive Dual Slot 230W Gen 4.0, DPU Crypto ON W/ Bkt, 1 Dongle, Black, HF, VCPD
900-21004-0010-000	NVIDIA BlueField-2 A100X, P1004 SKU 230, Generic, GA100, 80GB HBM2e, PCIe Passive Dual Slot 300W Gen 4.0, DPU Crypto ON W/ Bkt, 1 Dongle, Black, HF, VCPD

## 1.2. Hardware Prerequisites

This quick start guide assumes that an NVIDIA® BlueField® DPU has been installed in a server according to the instructions detailed in your [DPU's hardware user guide](#).

## 1.3. DOCA Packages

Device	Component	Version	Description
Host	DOCA SDK	0.5.0	Software development kit package for developing host software

Device	Component	Version	Description
Target BlueField-2 DPU (Arm)	DOCA Runtime	1.4.0	Runtime libraries required to run DOCA-based software applications on host
	DOCA Tools	1.4.0	Tools for developers and administrators on host
	Arm emulated (Qemu) development container	3.9.2	Linux-based BlueField Arm emulated container for developers
	BlueField OS	3.9.2	BlueField OS image and firmware
	DOCA SDK	0.5.0	Software development kit packages for developing Arm software
	DOCA Runtime	1.4.0	Runtime libraries required to run DOCA-based software applications on Arm
	DOCA Tools	1.4.0	Tools for developers and administrators for Arm target

## 1.4. Supported Operating System

The operating system supported on the BlueField DPU is Ubuntu 20.04.

The following operating systems are supported on the host machine:

- ▶ Ubuntu 18.04/20.04
- ▶ CentOS/RHEL 7.6/8.0/8.2
- ▶ Debian 10.8

## 1.5. Supported Kernel Versions



**Note:** Only the following generic kernel versions are supported for DOCA local repo package for host installation (whether by SDKM or manually).

Host Operation System	Kernel Support	Arch Support
CentOS 7.6	4.14.0-115.el7a.aarch64	aarch64
	3.10.0-957.el7.x86_64	x86

Host Operation System	Kernel Support	Arch Support
CentOS 8.0	4.18.0-80.el8.x86_64	
CentOS 8.2	4.18.0-193.el8.x86_64	
RHEL 7.6	3.10.0-957.el7.x86_64	
RHEL 8.0	4.18.0-80.el8.x86_64	
RHEL 8.2	4.18.0-193.el8.x86_64	
Ubuntu 18.04	4.15.0-20-generic	
Ubuntu 20.04	5.4.0-26-generic	
Debian 10.8	4.19.0-14-amd64	

---

# Chapter 2. SDK Manager

[NVIDIA SDK Manager](#) supports DOCA installation, including software packages on the host and the BlueField-2 target.

- ▶ To use the SDK Manager GUI, please refer to [NVIDIA SDK Manager GUI installation guide for DOCA](#) for detailed instructions.
- ▶ To use the SDK Manager CLI, please refer to [NVIDIA SDK Manager CLI installation guide for DOCA](#) for detailed instructions.



**Note:** SDK manager installation requires internet connection through out-of-band (OOB) port.

# Chapter 3. Manual BlueField Image Installation

This guide provides the minimal first-step instructions for setting up DOCA on a standard system.

## 3.1. Installation Files

Device	Component	Arch and OS	Link
Host	These files contain the following components suitable for their respective OS version. <ul style="list-style-type: none"> <li>▶ DOCA SDK v0.5.0</li> <li>▶ DOCA Runtime v1.4.0</li> <li>▶ DOCA Tools v1.4.0</li> </ul>	CentOS 7.6 on aarch64	<a href="#">doca-host-repo-rhel76-1.4.0-0.1.9.1.4.0079.1.el7a.5.7.1.0.2</a>
		CentOS/RHEL 7.6 on x86	<a href="#">doca-host-repo-rhel76-1.4.0-0.1.9.1.4.0079.1.el7.5.7.1.0.2</a>
		CentOS/RHEL 8.0 on x86	<a href="#">doca-host-repo-rhel80-1.4.0-0.1.9.1.4.0079.1.el8.5.7.1.0.2</a>
		CentOS/RHEL 8.2 on x86	<a href="#">doca-host-repo-rhel82-1.4.0-0.1.9.1.4.0079.1.el8.5.7.1.0.2</a>
		Ubuntu 18.04 on x86	<a href="#">doca-host-repo-ubuntu1804_1.4.0-0.1.9.1.4.0079.1.5.7.1.0.</a>
		Ubuntu 20.04 on x86	<a href="#">doca-host-repo-ubuntu2004_1.4.0-0.1.9.1.4.0079.1.5.7.1.0.</a>
		Debian 10.8 on x86	<a href="#">doca-host-repo-debian108_1.4.0-0.1.9.1.4.0079.1.5.7.1.0.2</a>
	Arm Emulated Development Container	Arm container v3.9.2 on aarch64	<a href="#">doca_devel_ubuntu_20.04-inbox-5.5.tar</a>
Target BlueField-2 DPU (Arm)	BlueField OS image v3.9.2	Ubuntu 20.04 on aarch64	<a href="#">doca_1.4.0_bsp_3.9.2_ubuntu_20.04-4.sig</a>
	DOCA SDK v0.5.0		<a href="#">doca-repo-aarch64-ubuntu2004-local_1.4.0079-1.5.7.1.0.2.0.bf.3.9.2.12271</a>
	DOCA Runtime v1.4.0		
	DOCA Tools v1.4.0		



## 3.2. Installing Software on Host

1. Installation of MFT and RShim for managing and flashing the BlueField DPU.

### For Ubuntu/Debian

- a). Download the DOCA Tools package from [Installation Files](#) section for the host.

- b). Unpack the deb repo. Run:

```
host# sudo dpkg -i doca-host-repo-ubuntu<version>_amd64.deb
```

- c). Perform apt update. Run:

```
host# sudo apt-get update
```

- d). Run `apt install` for DOCA Tools.

- ▶ For DPU

```
host# sudo apt install doca-tools
```

- ▶ For ConnectX on Ubuntu 20.04

```
host# sudo apt install doca-cx-tools
```

### For CentOS/RHEL


- a). Download the DOCA Tools package from [Installation Files](#) section for the x86 host.

- b). Unpack the RPM repo. Run:

```
host# sudo rpm -Uvh doca-host-repo-rhel<version>.x86_64.rpm
```

- c). Run `yum install` to install DOCA Tools.

```
host# sudo yum install doca-tools
```

 **Note:** Skip the following step to proceed without the DOCA local repo package for host.

2. Alternatively, to continue with the DOCA local repo package for host installation:

### Installing DOCA Local Repo Package on Ubuntu Host

- a). Download the DOCA SDK and DOCA Runtime packages from [Installation Files](#) section for the host.

- b). Unpack the deb repo. Run:

```
host# sudo dpkg -i doca-host-repo-ubuntu<version>_amd64.deb
```

- c). Perform apt update. Run:

```
host# sudo apt-get update
```

- d). Run `apt install` for DOCA runtime, tools, and SDK.

- ▶ For DPU

```
host# sudo apt install -y doca-runtime doca-sdk
```

- ▶ For ConnectX on Ubuntu 20.04

```
host# sudo apt install -y doca-cx-runtime doca-cx-sdk
```

### Installing DOCA Local Repo Package on CentOS Host

- a). Download the DOCA SDK and DOCA Runtime packages from [Installation Files](#) section for the x86 host.

- b). Install the following software dependencies. Run:

```
host# sudo yum install -y epel-release
```

- c). For CentOS 8.2 only, also run:

```
host# yum config-manager --set-enabled PowerTools
```

- d). Unpack the RPM repo. Run:

```
host# sudo rpm -Uvh doca-host-repo-rhel<version>.x86_64.rpm
```

- e). Run `yum install` for DOCA runtime, tools, and SDK.

```
host# sudo yum install -y doca-runtime doca-sdk
```

## Installing DOCA Local Repo Package on RHEL Host



**Note:** For RHEL 7.6, only perform step d. from the following procedure.

- a). Open a RedHat account.

- i. Log into RedHat website via the [developers tab](#).
- ii. [Create a developer user](#).

- b). Run:

```
host# subscription-manager register --username=<username> --password=PASSWORD
```

To extract pool ID:

```
host# subscription-manager list --available --all
```

```
...
Subscription Name:   Red Hat Developer Subscription for Individuals
Provides:            Red Hat Developer Tools (for RHEL Server for ARM)
                    ...
                    Red Hat CodeReady Linux Builder for x86_64
...
Pool ID:             <pool-id>
...
```

And use the pool ID for the Subscription Name and Provides that include Red Hat CodeReady Linux Builder for x86\_64.

- c). Run:

```
host# subscription-manager attach --pool=<pool-id>
host# subscription-manager repos --enable codeready-builder-for-rhel-8-x86_64-rpms
host# yum makecache
```

- d). Install the DOCA local repo package for host. Run:

```
host# rpm -Uvh doca-host-repo-rhel<version>.x86_64.rpm
host# sudo yum install -y doca-runtime doca-sdk
```

- e). Sign out from your RHEL account. Run:

```
host# subscription-manager remove --all
host# subscription-manager unregister
```

3. Initialize MST. Run:

```
host# sudo mst start
```

4. Reset the `nvconfig` params to their default values:

```
host# sudo mlxconfig -d /dev/mst/mt41686_pciconf0 -y reset
```

```
Reset configuration for device /dev/mst/mt41686_pciconf0? (y/n) [n] : y
Applying... Done!
-I- Please reboot machine to load new configurations.
```

5. Skip this step if your BlueField DPU is Ethernet only. Please refer to [Supported Platforms](#) to learn your DPU type.

If you have a VPI DPU, the default link type of the ports will be configured to IB. To verify your link type, run:

```
host# sudo mst start
host# sudo mlxconfig -d /dev/mst/mt41686_pciconf0 -e q | grep -i link_type
Configurations:          Default          Current          Next
Boot
*      LINK_TYPE_P1      IB(1)           ETH(2)
  IB(1)
*      LINK_TYPE_P2      IB(1)           ETH(2)
  IB(1)
```



**Note:** If your DPU is Ethernet capable only, then the `sudo mlxconfig -d <device>` command will not provide an output.

If the current link type is set to IB, run the following command to change it to Ethernet:

```
host# sudo mlxconfig -d /dev/mst/mt41686_pciconf0 s LINK_TYPE_P1=2 LINK_TYPE_P2=2
```

- Assign a dynamic IP to `tmfifo_net0` interface (RShim host interface).



**Note:** Skip this step if you are [installing the DOCA image on multiple DPUs](#).

```
host# ifconfig tmfifo_net0 192.168.100.1 netmask 255.255.255.252 up
```

- Verify that RShim is active.

```
host# sudo systemctl status rshim
```

This command is expected to display `active (running)`. If RShim service does not launch automatically, run:

```
host# sudo systemctl enable rshim
host# sudo systemctl start rshim
```

## 3.3. Installing Software on DPU

Users have two options for installing DOCA on the DPU:

- ▶ Upgrading the full DOCA image on the DPU (recommended) - this option overwrites the entire boot partition.
- ▶ Upgrading DOCA local repo package on the DPU - this option upgrades DOCA components without overwriting the boot partition. Use this option to preserve configurations or files on the DPU itself.

### 3.3.1. Installing Full DOCA Image on DPU



**Note:** This installation sets up the OVS bridge.



**Note:** If you are installing DOCA on multiple DPUs, skip to section [Installing Full DOCA Image on Multiple DPUs](#).



**Note:** This step overwrites the entire boot partition.

### 3.3.1.1. Option 1 - No Pre-defined Password



**Note:** To set the password in advance, proceed to [Option 2](#).

BFB installation is executed as follows:

```
host# sudo bfb-install --rshim <rshimN> --bfb <image_path.bfb>
```

Where `rshimN` is `rshim0` if you only have one DPU. You may run the following command to verify:

```
host# ls -la /dev/ | grep rshim
```

### 3.3.1.2. Option 2 - Set Pre-defined Password

Ubuntu users can provide a unique password that will be applied at the end of the BlueField OS image installation. This password needs to be defined in a `bf.cfg` configuration file.

To set the password for the "ubuntu" user:

1. Create password hash. Run:

```
host# openssl passwd -1
Password:
Verifying - Password:
$1$3B0RIrfX$TlHry93NFUJzg3Nya00rE1
```

2. Add the password hash in quotes to the `bf.cfg` file:

```
host# sudo vim bf.cfg
ubuntu_PASSWORD='$1$3B0RIrfX$TlHry93NFUJzg3Nya00rE1'
```

When running the installation command, use the `--config` flag to provide the file containing the password:

```
host# sudo bfb-install --rshim <rshimN> --bfb <image_path.bfb> --config bf.cfg
```



**Note:** If `--config` is not used, then upon first login to the BlueField device, users will be asked to update their password.

The following is an example of Ubuntu installation assuming the "pv" Linux tool has been installed (to view the installation progress).

```
host# sudo bfb-install --rshim rshim0 --bfb DOCA_<version>-aarch64.bfb --config
bf.cfg
Pushing bfb
1.08GiB 0:00:57 [19.5MiB/s] [      <=>      ]
Collecting BlueField booting status. Press Ctrl+C to stop...
INFO[BL2]: start
INFO[BL2]: DDR POST passed
INFO[BL2]: UEFI loaded
INFO[BL31]: start
INFO[BL31]: runtime
INFO[UEFI]: eMMC init
INFO[UEFI]: eMMC probed
INFO[UEFI]: PCIe enum start
INFO[UEFI]: PCIe enum end
INFO[MISC]: Ubuntu installation started
INFO[MISC]: Installation finished
INFO[MISC]: Rebooting...
```

### 3.3.2. Installing Full DOCA Image on Multiple DPUs

On a host with multiple DPUs, the BFB image can be installed on all of them using the [multi-bfb-install script](#).

```
host# ./bfb-multi-install --bfb <bfb-file> --password <password>
```

This script detects the number of RShim devices and configures them statically.

- ▶ For Ubuntu – the script creates a configuration file `/etc/netplan/20-tmfifo.yaml`
- ▶ For CentOS/RH 7.6 – the script creates a configuration file `/etc/sysconfig/network-scripts/ifcfg-br_tmfifo`
- ▶ For CentOS/RH 8.0 and 8.2 – the script installs `bridge-utils` package to use the command `brctl`, creates bridge `tm-br` and connects all RShim interfaces to it

After the installation is complete, the configuration of the bridge and each RShim interface can be observed using `ifconfig`. The expected result is to see the IP on the bridge `tm-br` configured to `192.168.100.1` with subnet `255.255.255.0`.



**Note:** To log into BlueField with `rshim0`, run:

```
ssh ubuntu@192.168.100.2
```

For each RShim after that, add 1 to the fourth octet of the IP address (e.g., `ubuntu@192.168.100.3` for `rshim1`, `ubuntu@192.168.100.4` for `rshim2`, etc).

The script burns a new MAC address to each DPU and configures a new IP, `192.168.100.x`, as described earlier.

### 3.3.3. Installing DOCA Local Repo Package on DPU



**Note:** DOCA repo files are already installed by default as part of the BlueField OS image.

To upgrade DOCA packages on the DPU, use the Debian package manager to upgrade the new packages:

1. Download the DOCA repo file `doca-repo-aarch64-<version>.deb` for the target DPU from section [Installation Files](#).
2. Copy the DOCA repo file into the BlueField DPU.
3. Unpack and install the DOCA repo file. Run:
 

```
dpu# sudo dpkg -i doca-repo-aarch64-<version>.deb
```
4. Check for any DOCA package content upgrade. Run:
 

```
dpu# sudo apt-get update
dpu# sudo apt-get upgrade
```


## 3.4. Upgrading Firmware



**Note:** If multiple DPUs are installed, the following steps must be performed on all of them after [BFB installation](#).

To upgrade firmware:

1. SSH to your BlueField device via 192.168.100.2 (preconfigured).

 **Note:** If multiple DPUs are installed, the tmfif0 IP interface does not have to be 192.168.100.2. The last octate changes and depends on the RShim number.

The default credentials for Ubuntu are as follows:

- ▶ Username: ubuntu
- ▶ Password: ubuntu or a unique password that you set in `bf.cfg`

For example:

```
host# ssh ubuntu@192.168.100.2 Password: <configured-password>
```

2. Upgrade firmware in BlueField DPU. Run:

```
dpu# sudo /opt/mellanox/mlnx-fw-updater/mlnx_fw_updater.pl --force-fw-update
```

Example output:

```
Device #1:
-----
Device Type:      BlueField-2
[...]
Versions:         Current          Available
FW                <Old_FW>          <New_FW>
```

3. For the firmware upgrade to take effect:

- a). Run the following command on the BlueField DPU and host:

```
dpu# sudo mst start
```

- b). Query the available reset flows:

```
dpu# sudo mlxfwreset -d /dev/mst/mt41686_pciconf0 q
```


Example output:

```
Reset-levels:
...

Reset-types (relevant only for reset-levels 3,4):
...

Reset-sync (relevant only for reset-level 3):
0: Tool is the owner                -Supported (default)
1: Driver is the owner              -Supported
```

- c). If `reset-sync 1` is not supported, perform host power cycle. Otherwise, trigger reset by running the following:

 **Note:** The following operation **can only be used after** upgrading to DOCA 1.4.0. It is not supported from previous versions and may cause unexpected behavior if used.

```
dpu# sudo mlxfwreset -d /dev/mst/mt41686_pciconf0 --sync 1 -y reset
```

 **Note:** The entire DPU will experience reset.

## 3.5. Post-installation Procedure

1. Restart the driver. Run:

```
host# sudo /etc/init.d/openibd restart
Unloading HCA driver:                [ OK ]
Loading HCA driver and Access Layer: [ OK ]
```

2. Configure the physical function (PF) interfaces.

```
host# sudo ifconfig <interface-1> <network-1/mask> up
host# sudo ifconfig <interface-2> <network-2/mask> up
```

For example:

```
host# sudo ifconfig p2p1 192.168.200.32/24 up
host# sudo ifconfig p2p2 192.168.201.32/24 up
```

Pings between the source and destination should now be operational.

---

# Chapter 4. Building Your Own BFB Installation Image

Users wishing to build their own customized BlueField OS image can use the BFB build environment. Please refer to [this GitHub webpage](#) for more information.



**Note:** For a customized BlueField OS image to boot on the UEFI secure-boot-enabled DPU (default DPU secure boot setting), the OS must be either signed with an existing key in the UEFI DB (e.g., the Microsoft key), or UEFI secure boot must be disabled. Please refer to the [Secure Boot](#) section and its subpages of the *NVIDIA BlueField DPU Platform Operating System Documentation* for more details.



---

## Chapter 5. Setting Up Build Environment for Developers

For full instructions about setting up a development environment, refer to the [NVIDIA DOCA Developer Guide](#).

---

# Chapter 6. Installing CUDA on NVIDIA Converged Accelerator

NVIDIA® CUDA® is a parallel computing platform and programming model developed by NVIDIA for general computing GPUs.

This section details the necessary steps to set up CUDA on your environment. This section assumes that a BFB image has already been installed on your environment.

To install CUDA on your converged accelerator:

1. Download and install the latest NVIDIA Data Center GPU driver.
2. Download and install CUDA.



**Note:** Downloading CUDA includes the latest NVIDIA Data Center GPU driver and CUDA toolkit. For more information about CUDA and driver compatibility please refer to [NVIDIA CUDA Toolkit Release Notes](#).

## 6.1. Configuring Operation Mode

There are two modes that the NVIDIA Converged Accelerator may operate in:

- ▶ Standard mode (default) – the BlueField DPU and the GPU operate separately
- ▶ BlueField-X mode – the GPU is exposed to the DPU and is no longer visible on the host

To verify which mode the system is operating in, run:

```
host# sudo mst start
host# sudo mlxconfig -d /dev/mst/mt41686_pciconf0 q PCI_DOWNSTREAM_PORT_OWNER[4]
```

Standard mode output:

```
Device #1:
[...]
Configurations:
    PCI_DOWNSTREAM_PORT_OWNER[4]
Next Boot
    DEVICE_DEFAULT(0)
```

BlueField-X mode output:

```
Device #1:
[...]
Configurations:
    PCI_DOWNSTREAM_PORT_OWNER[4]
Next Boot
    EMBEDDED_CPU(15)
```

To configure BlueField-X mode, run:

```
host# mlxconfig -d /dev/mst/mt41686_pciconf0 s PCI_DOWNSTREAM_PORT_OWNER[4]=0xF
```

To configure standard mode, run:

```
host# mlxconfig -d /dev/mst/mt41686_pciconf0 s PCI_DOWNSTREAM_PORT_OWNER[4]=0x0
```



**Note:** Firmware reset or power cycle is required for configuration to take effect.

- ▶ For firmware reset, run the following command on the BlueField DPU and **immediately** afterwards on the host. **Do not** wait for the command to complete on the BlueField DPU before issuing the command on the host.

```
dpu+host# sudo mlxfwreset -d /dev/mst/mt41686_pciconf0 -l 3 -y reset
```

- ▶ For power cycle the host run:

```
host# ipmitool power cycle
```

## 6.2. Downloading and Installing CUDA Toolkit and Driver

This section details the necessary steps to set up CUDA on your environment. It assumes that a BFB image has already been installed on your environment.

1. Install CUDA by visiting the [CUDA Toolkit 11.6.2 Downloads](#) webpage.



**Note:** Select the Linux distribution and version relevant for your environment.

2. Test that the driver installation completed successfully. Run:

```
nvidia-smi
```

```
Tue Apr  5 13:37:59 2022
```

```
+-----+
| NVIDIA-SMI 510.47.03      Driver Version: 510.47.03   CUDA Version: 11.6     |
+-----+-----+-----+
| GPU  Name          Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|====+=====+=====+=====+=====+=====+=====+
|   0  NVIDIA BF A10      Off      | 00000000:06:00:0 Off |             0         |
|  0%   43C    P0      N/A / 225W |  0MiB / 23028MiB |           0%      Default |
|                                     |                      | N/A             |
+-----+-----+-----+

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

3. Verify that the installation completed successfully.

- a). Install Download CUDA samples repo. Run:

```
git clone https://github.com/NVIDIA/cuda-samples.git
```

- b). Build and run vectorAdd CUDA sample. Run:

```
cd cuda-samples/Samples/0_Introduction/vectorAdd
```

```
make
./vectorAdd
```

**Note:** If the `vectorAdd` sample works as expected, it should output "Test Passed".

**Note:** If it seems that the GPU is slow or stuck, stop execution and run:

```
sudo setpci -v -d ::0302 800.L=201 # CPL_VCO = 32
```

## 6.3. GPUDirect RDMA

To enable GPUDirect RDMA with a network card on NVIDIA Converged Accelerator, you need an additional kernel module. Run:

```
sudo modprobe nvidia-peermem
```

## 6.4. DPDK GPUDEV

To enable CPU map GPU memory feature in DPDK's `gpudev` library, you need the `GDRCopy` library and driver to be installed on your system.

1. Install `GDRCopy` library. Run:

```
git clone https://github.com/NVIDIA/gdrcopy.git
```

2. Install dependencies.

- ▶ For RHEL:

```
# DKMs can be installed from epel-release. See https://fedoraproject.org/wiki/EPEL.
$ sudo yum install dkms check check-devel subunit subunit-devel
```

- ▶ For Debian:

```
$ sudo apt install check libsubunit0 libsubunit-dev
```

3. Build the library and install the driver. Run:

```
cd gdrcopy
make
# Launch gdrdrv kernel module on the system
./insmod.sh
```

4. Setup `GDRCopy` path. Run:

```
export GDRCOPY_PATH_L=/path/to/libgdrapi
```

**Note:** In general, the path to `libgdrapi` is `/path/to/gdrcopy/src/`.

## 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 nor any of its direct or indirect subsidiaries and affiliates (collectively: "NVIDIA") make no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assume 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 NON-INFRINGEMENT, 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.

## Trademarks

NVIDIA, the NVIDIA logo, and Mellanox are trademarks and/or registered trademarks of Mellanox Technologies Ltd. and/or NVIDIA Corporation in the U.S. and in other countries. The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis. Other company and product names may be trademarks of the respective companies with which they are associated.

## Copyright

© 2022 NVIDIA Corporation & affiliates. All rights reserved.