NVIDIA DOCA

Installation Guide for Linux

MLNX-15-060470 _v2.2.1 | October 2023

Table of Contents

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

DOCA is supported on the devices listed in the following subsections.

> Note: This is a special GA release for improved stability that supports BlueField-3 DPUs only.

1.1.1. Supported BlueField DPUs

The following NVIDIA® BlueField® DPUs are supported with DOCA:

NVIDIA SKU	PSID	Description
900-9D3B6-00CN-AB0	MT_0000000883	BlueField-3 B3240 P-Series Dual-slot FHHL DPU; 400GbE / NDR IB (default mode); Dual-port QSFP112; PCIe Gen5.0 x16 with x16 PCIe extension option; 16 Arm cores; 32GB on-board DDR; integrated BMC; Crypto Enabled
900-9D3B6-00CV-AA0	MT_0000000884	BlueField-3 B3220 P-Series FHHL DPU; 200GbE (default mode) / NDR200 IB; Dual-port QSFP112; PCIe Gen5.0 x16 with x16 PCIe extension option; 16 Arm cores; 32GB on-board DDR; integrated BMC; Crypto Enabled
900-9D3B6-00SN-AB0	MT_0000000964	BlueField-3 B3240 P-Series Dual-slot FHHL DPU; 400GbE / NDR IB (default mode); Dual-

NVIDIA SKU	PSID	Description
		port QSFP112; PCIe Gen5.0 x16 with x16 PCIe extension option; 16 Arm cores; 32GB on-board DDR; integrated BMC; Crypto Disabled
900-9D3B6-00SV-AA0	MT_0000000965	BlueField-3 B3220 P-Series FHHL DPU; 200GbE (default mode) / NDR200 IB; Dual-port QSFP112; PCIe Gen5.0 x16 with x16 PCIe extension option; 16 Arm cores; 32GB on-board DDR; integrated BMC; Crypto Disabled
900-9D3B4-00EN-EA0	MT_0000001010	BlueField-3 B3140L E-Series FHHL DPU; 400GbE / NDR IB (default mode); Single-port QSFP112; PCIe Gen5.0 x16; 8 Arm cores; 16GB on-board DDR; integrated BMC; Crypto Enabled
900-9D3B4-00PN-EA0	MT_0000001011	BlueField-3 B3140L E-Series FHHL DPU; 400GbE / NDR IB (default mode); Single-port QSFP112; PCIe Gen5.0 x16; 8 Arm cores; 16GB on-board DDR; integrated BMC; Crypto Disabled
900-9D3B6-00CC-AA0	MT_0000001024	BlueField-3 B3210 P-Series FHHL DPU; 100GbE (default mode) / HDR100 IB; Dual-port QSFP112; PCIe Gen5.0 x16 with x16 PCIe extension option; 16 Arm cores; 32GB on-board DDR; integrated BMC; Crypto Enabled
900-9D3B6-00SC-AA0	MT_0000001025	BlueField-3 B3210 P-Series FHHL DPU; 100GbE (default mode) / HDR100 IB; Dual-port QSFP112; PCIe Gen5.0 x16 with x16 PCIe extension option; 16 Arm cores; 32GB on-board DDR; integrated BMC; Crypto Disabled
900-9D3D4-00EN-HA0	MT_0000001069	BlueField-3 B3140H E-series HHHL DPU; 400GbE(default mode)/NDR IB; Single-port QSFP112; PCIe Gen5.0 x16; 8 Arm cores; 16GB on board DDR; integrated BMC; Crypto Enabled

NVIDIA SKU	PSID	Description
900-9D3D4-00NN-HA0	MT_0000001070	BlueField-3 B3140H E-series HHHL DPU; 400GbE(default mode)/NDR IB; Single-port QSFP112; PCIe Gen5.0 x16; 8 Arm cores; 16GB on board DDR; integrated BMC; Crypto Disabled

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	2.2.1	Software development kit package for developing host software
	DOCA Runtime	2.2.1	Runtime libraries required to run DOCA-based software applications on host
	DOCA Tools	2.2.1	Tools for developers and administrators on host
	DOCA Extra	2.2.1	Contains helper scripts (doca-info, doca-kernel-support)
	DOCA OFED	2.2.1	Software stack which operates across all NVIDIA network adapter solutions
	Arm emulated (QEMU) development container	4.2.2	Linux-based BlueField Arm emulated container for developers
Target BlueField DPU (Arm)	BlueField BSP	4.2.2	BlueField image and firmware
	DOCA SDK	2.2.1	Software development kit packages for developing Arm software

Device	Component	Version	Description
	DOCA Runtime	2.2.1	Runtime libraries requied to run DOCA-based software applications on Arm
	DOCA Tools	2.2.1	Tools for developers and administrators for Arm target

1.4. Supported Operating System Distributions

The default operating system of the BlueField DPU (Arm) is Ubuntu 22.04.

The supported operating systems on the host machine are the following:

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

OS	Kernel	x86	aarch64
Alinux 3.2	5.10	#	
Debian 10.13	5.10.135 5.4.210	#	
Debian 10.8	4.19.0	#	
Oracle Linux 8.7	5.10/5.15	#	
RHEL/CentOS 7.6	3.10	#	
RHEL/CentOS 8.2	4.18	#	
Ubuntu 18.04	4.15	#	
Ubuntu 20.04	5.4	#	
Ubuntu 22.04	5.15	#	
Windows	DOCA support for Windows is on DOCA's roadmap. For Windows driver support, refer to WinOF-2 Release Notes.		

Chapter 2. SDK Manager

NVIDIA SDK Manager (SDKM) supports DOCA installation, including software packages on the host and the BlueField target. The SDKM automates the process of DOCA installation and other related configuration of the system.

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

Note: If installing DOCA using SDKM, please skip the remaining sections and follow the wizard instead.

Note: SDKM installation requires Internet connection through out-of-band (OOB) port.

The following example uses the SDKM CLI to install DOCA without any additional SDKs:

```
host# sdkmanager --cli install --logintype devzone --product DOCA --version 2.2.1 --
targetos Linux --host --target BLUEFIELD2_DPU_TARGETS --flash all
```

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	OS and Arch	Link
Host	These files contain the following components suitable for their respective OS version. ▶ DOCA SDK v2.2.1 ▶ DOCA Runtime v2.2.1 ▶ DOCA Tools v2.2.1 ▶ DOCA Extra v2.2.1 ▶ DOCA OFED v2.2.1	Alinux 3.2 on x86	doca-host-repo-alinux32-2.2.1-0.0.3.2.2.1009.1.al8.23.0
		Debian 10.13 on x86	doca-host-repo-debian1013_2.2.1-0.0.3.2.2.1009.1.23.0
		Debian 10.8 on x86	doca-host-repo-debian108_2.2.1-0.0.3.2.2.1009.1.23.07
		Oracle Linux 8.7 on x86	doca-host-repo-ol87-2.2.1-0.0.3.2.2.1009.1.el8.23.07.0.5
		RHEL/CentOS 7.6 on x86	doca-host-repo-rhel76-2.2.1-0.0.3.2.2.1009.1.el7.23.07.0
		RHEL/CentOS 8.2 on x86	doca-host-repo-rhel82-2.2.1-0.0.3.2.2.1009.1.el8.23.07.0
		Ubuntu 18.04 on x86	doca-host-repo-ubuntu1804_2.2.1-0.0.3.2.2.1009.1.23.0
		Ubuntu 20.04 on x86	doca-host-repo-ubuntu2004_2.2.1-0.0.3.2.2.1009.1.23.0
		Ubuntu 22.04 on x86	doca-host-repo-ubuntu2204_2.2.1-0.0.3.2.2.1009.1.23.0
	Arm Emulated Development Container	Arm container v4.2.2 on aarch64	doca_devel_ubuntu_20.04-inbox-5.5.tar
Target BlueField DPU (Arm)	BlueField Software v4.2.2	Ubuntu 22.04 on aarch64	doca_2.2.1_bsp_4.2.2_ubuntu_22.04-13
	DOCA SDK v2.2.1	Ubuntu 22.04 on aarch64	doca-dpu-repo-ubuntu2204-local_2.2.1009-1.23.07.0.5.0.0.bf.4.2.2.1
	DOCA Runtime v2.2.1		

Device	Component	OS and Arch	Link
	DOCA Tools v2.2.1		

3.2. Uninstalling Software from Host

If an older DOCA software version is installed on your host, make sure to uninstall it before proceeding with the installation of the new version:

<table>
<tr><th>OS</th><th>Commands</th></tr>
<tr><td>Ubuntu/ Debian</td><td>$ for f in $(dpkg --list | grep doca | awk '{print $2}'); do echo $f ; apt remove --purge $f -y ; done
$ sudo apt-get autoremove</td></tr>
<tr><td>CentOS/ RHEL</td><td>host# for f in $(rpm -qa |grep -i doca) ; do yum -y remove $f; done
host# yum autoremove
host# yum makecache</td></tr>
</table>

3.3. Installing Prerequisites on Host for Target DPU

Install `doca-tools` to manage and flash the BlueField DPU.

OS	Instructions

For Ubuntu/ Debian	1. Download the DOCA host repo package from the Installation Files section.
	2. Unpack the deb repo. Run:
	`host# sudo dpkg -i doca-host-repo-ubuntu<version>_amd64.deb`
	3. Perform apt update. Run:
	`host# sudo apt-get update`
	4. Run `apt install` for DOCA Tools:
	`host# sudo apt install doca-tools`

For CentOS/RHEL 8.x/
Rocky 8.6
	1. Download the DOCA host repo package from the Installation Files section.
	2. Unpack the RPM repo. Run:
	`host# sudo rpm -Uvh doca-host-repo-rhel<version>.x86_64.rpm`
	3. Enable new dnf repos. Run:
	`host# sudo dnf makecache`
	4. Run `yum install` to install DOCA Tools:
	`host# sudo yum install doca-tools`

For CentOS/RHEL 7.x	1. Download the DOCA host repo package from the Installation Files section.
	2. Unpack the RPM repo. Run:
	`host# sudo rpm -Uvh doca-host-repo-rhel<version>.x86_64.rpm`
	3. Enable new yum repos. Run:
	`host# sudo yum makecache`

OS	Instructions
	4. Run `dnf install` to install DOCA Tools: `host# sudo dnf install doca-tools`

3.4. Determining DPU Device ID

It is is important to learn your DPU's `device-id` for performing some of the software installations or upgrades in this guide.

To determine the device ID of the DPUs on your setup, run:

```
host# mst start
host# mst status -v
```

Example output:

```
MST modules:
------------
    MST PCI module is not loaded
    MST PCI configuration module loaded
PCI devices:
------------
DEVICE_TYPE               MST                           PCI       RDMA            NET
                        NUMA
BlueField2(rev:1)         /dev/mst/mt41686_pciconf0.1   3b:00.1   mlx5_1          net-
ens1f1                  0

BlueField2(rev:1)         /dev/mst/mt41686_pciconf0     3b:00.0   mlx5_0          net-
ens1f0                  0

BlueField3(rev:1)         /dev/mst/mt41692_pciconf0.1   e2:00.1   mlx5_1          net-
ens7f1np1               4

BlueField3(rev:1)         /dev/mst/mt41692_pciconf0     e2:00.0   mlx5_0          net-
ens7f0np0               4
```

The device IDs for the BlueField-2 and BlueField-3 DPUs in this example are `/dev/mst/mt41686_pciconf0` and `/dev/mst/mt41692_pciconf0` respectively.

3.5. Installing Software on Host

Important: Skip this section if you intend to update only the BlueField software (`*.bfb`).

1. Follow the instructions under Installing Prerequisites on Host for Target DPU.
2. Install DOCA local repo package for host.

The following table provides instructions for installing the DOCA host repo on your device depending on your OS and desired profile.

OS	Profi	Devic	Instructions
Ubuntu/ Debian	doca-all	BlueField	a). Download the DOCA host repo package from Installation Files for the host.

OS	Profi	Devic	Instructions

			b). Unpack the deb repo. Run:
			`host# dpkg -i <repo_file>`
			c). Perform apt update. Run:
			`host# apt-get update`
			d). If the kernel version on your host is not supported (not shown under Supported Operating System Distributions), refer to DOCA Extra Package.
			e). Run apt install for DOCA SDK, DOCA runtime, DOCA tools:
			`host# sudo apt install -y doca-runtime doca-sdk doca-tools`

	doca-cx	ConnectX	a). Download the DOCA host repo package from Installation Files for the host.
			b). Unpack the deb repo. Run:
			`host# dpkg -i <repo_file>`
			c). Perform apt update. Run:
			`host# apt-get update`
			d). If the kernel version on your host is not supported (not shown under Supported Operating System Distributions), refer to DOCA Extra Package.
			e). Run apt install for DOCA SDK, DOCA runtime, DOCA tools:
			`host# sudo apt install -y doca-cx-runtime doca-cx-sdk doca-cx-tools`

	doca-ofed	All	a). Download the DOCA host repo package from Installation Files 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). If the kernel version on your host is not supported (not shown under Supported Operating System Distributions), refer to DOCA Extra Package.
			e). Install `doca-ofed`. Run:
			`host# sudo apt install -y doca-ofed`

CentOS/RHEL/Alinux/
Rocky/Oracle Linux
	doca-all	BlueField	Note: RHEL users need a valid subscription to install packages.
			a). Install the following software dependencies:
			▶ For RedHat 8.x, run:
			`host# yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm`
			▶ For Alinux 3.x, run:
			`host# yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm`
			▶ For Rocky 8.x, run:
			`host# sudo dnf makecache`
			`host# sudo dnf install epel-release`
			`host# sudo dnf install -y yum-utils`

OS	Profi	Devic	Instructions

			`host# sudo dnf config-manager --enable PowerTools`
			`host# sudo dnf clean dbcache`
			For CentOS 8.x, run:
			`host# sudo dnf install epel-release`
			`host# sudo dnf config-manager --set-enabled PowerTools`
			For CentOS 7.x, run:
			`host# sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm`
			`host# sudo yum-config-manager --add-repo http://mirror.centos.org/centos/7/os/x86_64`
			`host# sudo rpm --import http://mirror.centos.org/centos/7/os/x86_64/RPM-GPG-KEY-CentOS-7`
			`host# sudo yum-config-manager --add-repo http://mirror.centos.org/centos/7/extras/x86_64`
			`host# sudo yum-config-manager --save --setopt=mirror.centos.org_centos_7_os_x86_64.exclude='pciutils* libnl3*'`
			`host# yum makecache`
			b). Download the DOCA host repo package from Installation Files for the host.
			c). Unpack the rpm repo. Run:
			`host# rpm -Uvh <repo_file>.rpm`
			d). Perform yum update. Run:
			`host# sudo yum makecache`
			e). If the kernel version on your host is not supported (not shown under Supported Operating System Distributions), refer to DOCA Extra Package.
			f). Run yum install for DOCA SDK, DOCA runtime, DOCA tools:
			`host# sudo yum install -y doca-runtime doca-sdk doca-tools`

	doca-cx	ConnectX	RHEL users need a valid subscription to install packages.
			a). Install the following software dependencies. Run:
			For RedHat 8.x, run:
			`host# yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm`
			For Alinux 3.x, run:
			`host# yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm`
			For Rocky 8.x, run:
			`host# sudo dnf makecache`
			`host# sudo dnf install epel-release`
			`host# sudo dnf install -y yum-utils`
			`host# sudo dnf config-manager --enable PowerTools`
			`host# sudo dnf clean dbcache`
			For CentOS 8.x, run:
			`host# sudo dnf install epel-release`
			`host# sudo dnf config-manager --set-enabled PowerTools`
			For CentOS 7.x, run:
			`host# sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm`
			`host# sudo yum-config-manager --add-repo http://mirror.centos.org/centos/7/os/x86_64`
			`host# sudo rpm --import http://mirror.centos.org/centos/7/os/x86_64/RPM-GPG-KEY-CentOS-7`

OS	Profil	Devic	Instructions

			`host# sudo yum-config-manager --add-repo http://mirror.centos.org/centos/7/extras/x86_64`
			`host# sudo yum-config-manager --save --setopt=mirror.centos.org_centos_7_os_x86_64.exclude='pciutils* libnl3*'`
			`host# yum makecache`
			▸ For Oracle Linux 8.7, run:
			`host# yum config-manager --set-enabled ol8_codeready_builder`
			b). Download the DOCA host repo package from Installation Files for the host.
			c). Unpack the rpm repo. Run:
			`host# rpm -Uvh <repo_file>.rpm`
			d). Perform yum update. Run:
			`host# sudo yum makecache`
			e). If the kernel version on your host is not supported (not shown under Supported Operating System Distributions), refer to DOCA Extra Package.
			f). Run yum install for DOCA SDK, DOCA runtime, DOCA tools:
			`host# sudo yum install -y doca-cx-runtime doca-cx-sdk doca-cx-tools`

	doca-ofed	All	a). Download the DOCA host repo package from Installation Files for the host.
			b). Unpack the RPM repo. Run:
			`host# sudo rpm -Uvh doca-host-repo-rhel<version>.x86_64.rpm`
			c). Install doca-extra. Run:
			`host# sudo yum/dnf install -y doca-extra`
			d). If the kernel version on your host is not supported (not shown under Supported Operating System Distributions), refer to DOCA Extra Package.
			e). Install doca-ofed. Run:
			`host# sudo yum install -y doca-ofed`

3. Initialize MST. Run:

```
host# sudo mst start
```

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

```
host# sudo mlxconfig -d <device-id> -y reset

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

> Note: To learn the DPU's device ID, refer to Determining DPU Device ID.

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 <device-id> -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 <device-id> s LINK_TYPE_P1=2 LINK_TYPE_P2=2
```

6. 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
```

7. 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
```

3.5.1. DOCA Extra Package

If the kernel version on on your host is not supported (not shown under "Supported Operating System Distributions"), two options are available:

- Switch to a compatible kernel
- Install `doca-extra` package

  1. Run:

     ```
     host# sudo apt install -y doca-extra
     ```

  2. Execute the `doca-kernel-support` script which downloads the appropriate missing packages from your kernel to support DOCA:

     ```
     host# sudo /opt/mellanox/doca/tools/doca-kernel-support
     ```

     > Note: `doca-kernel-support` does not support customized or unofficial kernels.

3.6. 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.

3.6.1. Installing Full DOCA Image on DPU

Important: This step overwrites the entire boot partition.

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.

3.6.1.1. 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
```

3.6.1.2. Option 2 - Set Pre-defined Password

Ubuntu users can provide a unique password that will be applied at the end of the BlueField software 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...
```

3.6.2. 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# ./multi-bfb-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/RHEL 7.6 – the script creates a configuration file `/etc/sysconfig/network-scripts/ifcfg-br_tmfifo`
- For CentOS/RHEL 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.

3.6.3. Installing DOCA Local Repo Package on DPU

> Note: If you have already installed BlueField image, be aware that the DOCA SDK, Runtime, and Tools are already contained in the BFB, and this installation is not mandatory. If you have not installed the BlueField image and wish to update DOCA Local Repo package, proceed with the following procedure.

> Note: Before installing DOCA on the target DPU, make sure the out-of-band interface (mgmt) is connected to the Internet.

1. Download the DOCA SDK, DOCA Runtime, and DOCA Tools package from section Installation Files.
2. Copy deb repo package into BlueField. Run:
   ```
   host# sudo scp -r doca-repo-aarch64-ubuntu2004-local_<version>_arm64.deb
    ubuntu@192.168.100.2:/tmp/
   ```
3. Unpack the deb repo. Run:
   ```
   dpu# sudo dpkg -i doca-dpu-repo-ubuntu2004-local_<version>_arm64.deb
   ```
4. Run apt update:
   ```
   dpu# sudo apt-get update
   ```
5. Check for any DOCA package content upgrade. Run:
   ```
   dpu# sudo apt install doca-runtime
   dpu# sudo apt install doca-tools
   dpu# sudo apt install doca-sdk
   ```

3.7. 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 tmfifo 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. Power cycle the host server for the firmware upgrade to take effect.

> Note: The entire DPU will experience reset.

3.8. 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 the bfb-build project in 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. Additional SDKs for DOCA

6.1. 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: The CUDA version tested to work with DOCA SDK is 12.1.

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.

6.1.1. 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 <device-id> q PCI_DOWNSTREAM_PORT_OWNER[4]
```

Note: To learn the DPU's device ID, refer to Determining DPU Device ID.

- Standard mode output:

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

- BlueField-X mode output:

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

To configure BlueField-X mode, run:

```
host# mlxconfig -d <device-id> s PCI_DOWNSTREAM_PORT_OWNER[4]=0xF
```

To configure standard mode, run:

```
host# mlxconfig -d <device-id> s PCI_DOWNSTREAM_PORT_OWNER[4]=0x0
```

Note: To learn the DPU's device ID, refer to Determining DPU Device ID.

Power cycle is required for configuration to take effect. To power cycle the host run:

```
host# ipmitool power cycle
```

6.1.2. 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 Downloads webpage.

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

   Note: This section shows the native compilation option either on x86 or aarch64 hosts.

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

```
dpu# nvidia-smi

Tue Apr  5 13:37:59 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 510.47.03    Driver Version: 510.47.03    CUDA Version: 11.8     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  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). Download CUDA samples repo. Run:

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

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

   ```
   dpu# cd cuda-samples/Samples/0_Introduction/vectorAdd
   dpu# make
   dpu# ./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:

```
dpu# sudo setpci -v -d ::0302 800.L=201 # CPL_VC0 = 32
```

6.1.3. GPUDirect RDMA

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 the `gdrcopy` library. Run:

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

2. Install dependencies:

   - For RHEL/CentOS:

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

   - For Ubuntu/Debian:

     ```
     dpu# sudo apt install check libsubunit0 libsubunit-dev
     ```

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

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

4. Setup the path to `gdrcopy`. Run:

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

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

6.2. Installing Rivermax on DPU

NVIDIA Rivermax offers a unique IP-based solution for any media and data streaming use case.

This section provides the steps to install Rivermax assuming that a BFB image has already been installed on your environment.

6.2.1. Downloading Rivermax Driver

1. Navigate to the NVIDIA Rivermax SDK product page.
2. Register to be able to download the driver package using the JOIN button at the top of the page.
3. Download the appropriate driver package according to your BFB under the "Linux" subsection. For example, for Ubuntu 22.04 BFB, download `rivermax_ubuntu2204_<version>.tar.gz`.

6.2.2. Installing Rivermax Driver

1. Copy the `.tgz` file to the DPU:

   ```
   host# sudo scp -r rivermax_ubuntu2204_<version>.tar.gz ubuntu@192.168.100.2:/tmp/
   ```

2. Extract the Rivermax file:

   ```
   dpu# sudo tar xzf rivermax_ubuntu2204_<version>.tar.gz
   ```

3. Install the Rivermax driver package:

   ```
   dpu# cd <rivermax-version>/Ubuntu.22.04/deb-dist/aarch64/
   dpu# sudo dpkg -i rivermax_<version>.deb
   ```

6.2.3. Installing Rivermax Libraries from DOCA

Rivermax libraries are compatibles with DOCA components and can be found inside the doca-dpu-repo.

1. Unpack the doca-dpu-repo:

   ```
   dpu# sudo dpkg -i doca-dpu-repo-ubuntu2204-local_<version>_arm64.deb
   ```

2. Run apt update:

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

3. Install the Rivermax libraries:

   ```
   dpu# sudo apt install doca-rmax-libs
   dpu# sudo apt install libdoca-rmax-libs-dev
   ```

For additional details and guidelines, please visit the NVIDIA Rivermax SDK product page.

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 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.

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

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