Installing and Upgrading Aerial cuBB

Table of contents

If you are using AX800 converged accelerator, follow Installing Tools on Dell R750 then go directly to Installing New cuBB Container.

This page describes how to install or upgrade Aerial cuBB and the dependent CUDA driver, NIC firmware on the host system per release. You must update the dependent software components to the specific version listed in the Release Manifest.

> **Note**
>
> To prevent dependency errors, you must perform all steps in this section during the initial installation and each time you upgrade to a new Aerial release.

Update RSHIM and Mellanox Firmware Tools on the Host

> **Note**
>
> 1. Aerial has been using the Mellanox inbox driver instead of MOFED since the 23-4 release. MOFED must be removed if it is installed on the system.
>
> 2. RSHIM package is shared via a PID account after the official product release.

If the host system has previous MOFED and nv-peer-mem containers running, stop and remove the MOFED and nv-peer-mem driver containers first.

```
$ sudo docker stop OFED $ sudo docker rm OFED $ sudo docker stop nv_peer_mem
$ sudo docker rm nv_peer_mem
```

To check if there is an existing MOFED installed on the host system:

```
$ ofed_info -s MLNX_OFED_LINUX-23.07-0.5.0.0:
```

To uninstall MOFED, if it is present:

```
$ sudo /usr/sbin/ofed_uninstall.sh
```

If you are upgrading a Grace Hopper MGX system, follow Installing Tools on Grace Hopper to install DOCA OFED and MFT. The rshim is already included in the DOCA OFED. The following step to install rshim is for the x86 platform only.

Download the rshim package here and copy it to the local file system on the server.

Enter the following commands to install the rshim driver.

```
# Install rshim $ sudo apt-get install libfuse2 $ sudo dpkg -i
rshim_2.0.17.g0caa378_amd64.deb
```

Enter the following commands to install the Mellanox firmware tools.

```
# Install Mellanox Firmware Tools $ export MFT_VERSION=4.26.1-3 $ wget
https://www.mellanox.com/downloads/MFT/mft-$MFT_VERSION-x86_64-deb.tgz $
tar xvf mft-$MFT_VERSION-x86_64-deb.tgz $ sudo mft-$MFT_VERSION-x86_64-
deb/install.sh # Verify the install Mellanox firmware tool version $ sudo mst version
mst, mft 4.26.1-3, built on Nov 27 2023, 15:24:39. Git SHA Hash: N/A $ sudo mst
start # check NIC PCIe bus addresses and network interface names $ sudo mst status -v
MST modules: ------------ MST PCI module is not loaded MST PCI configuration
module loaded PCI devices: ------------ DEVICE_TYPE MST PCI RDMA NET NUMA
ConnectX6DX(rev:0) /dev/mst/mt4125_pciconf0.1 b5:00.1 mlx5_1 net-ens6f1 0
ConnectX6DX(rev:0) /dev/mst/mt4125_pciconf0 b5:00.0 mlx5_0 net-ens6f0 0
```

Update CX6-DX NIC Firmware

This section describes how to update the Mellanox NIC firmware.

To download the NIC firmware, see Mellanox firmware download page. For example, to update the CX6-DX NIC firmware, download it from the ConnectX-6 Dx Ethernet Firmware Download Center page.

In the download menu, there are multiple versions of the firmware specific to the NIC hardware, as identified by its OPN and PSID. To look up the OPN and PSID, use this command:

```
$ sudo mlxfwmanager -d /dev/mst/mt4125_pciconf0 Querying Mellanox devices
firmware ... Device #1: ---------- Device Type: ConnectX6DX Part Number:
MCX623106AE-CDA_Ax Description: ConnectX-6 Dx EN adapter card; 100GbE; Dual-
port QSFP56; PCIe 4.0 x16; Crypto; No Secure Boot PSID: MT_0000000528 PCI Device
Name: b5:00.0 Base GUID: b8cef6030033fdee Base MAC: b8cef633fdee Versions:
Current Available FW 22.34.1002 N/A PXE 3.6.0700 N/A UEFI 14.27.0014 N/A
```

The OPN and PSID are the “Part Number” and the “PSID” shown in the output. For example, depending on the hardware, the CX6-DX OPN and PSID could be:

```
OPN = MCX623106AC-CDA_Ax, PSID = MT_0000000436 OPN = MCX623106AE-
CDA_Ax, PSID = MT_0000000528
```

Download the firmware bin file that matches the hardware OPN. If the file has a `.zip` extension, unzip it with the unzip command to get the `.bin` file.

```
# For MCX623106AC-CDA_Ax, PSID = MT_0000000436 $ wget
https://www.mellanox.com/downloads/firmware/fw-ConnectX6Dx-rel-22_39_2048-
MCX623106AC-CDA_Ax-UEFI-14.32.17-FlexBoot-3.7.300.signed.bin.zip $ unzip fw-
ConnectX6Dx-rel-22_39_2048-MCX623106AC-CDA_Ax-UEFI-14.32.17-FlexBoot-
3.7.300.signed.bin.zip # For MCX623106AE-CDA_Ax, PSID = MT_0000000528 $ wget
https://www.mellanox.com/downloads/firmware/fw-ConnectX6Dx-rel-22_39_2048-
MCX623106AE-CDA_Ax-UEFI-14.32.17-FlexBoot-3.7.300.bin.zip $ unzip fw-
ConnectX6Dx-rel-22_39_2048-MCX623106AE-CDA_Ax-UEFI-14.32.17-FlexBoot-
3.7.300.bin.zip
```

To flash the firmware image onto the CX6-DX NIC, enter the following commands:

```
# For MCX623106AC-CDA_Ax, PSID = MT_0000000436 $ sudo flint -d
/dev/mst/mt4125_pciconf0 --no -i fw-ConnectX6Dx-rel-22_39_2048-MCX623106AC-
CDA_Ax-UEFI-14.32.17-FlexBoot-3.7.300.signed.bin b # For MCX623106AE-CDA_Ax,
PSID = MT_0000000528 $ sudo flint -d /dev/mst/mt4125_pciconf0 --no -i fw-
ConnectX6Dx-rel-22_39_2048-MCX623106AE-CDA_Ax-UEFI-14.32.17-FlexBoot-
3.7.300.bin b Current FW version on flash: 22.35.1012 New FW version: 22.39.2048
FSMST_INITIALIZE - OK Writing Boot image component - OK # Reset the NIC $ sudo
mlxfwreset -d /dev/mst/mt4125_pciconf0 --yes --level 3 r
```

Perform the steps below to enable the NIC firmware features required for Aerial CUDA-Accelerated RAN:

```
# eCPRI flow steering enable $ sudo mlxconfig -d /dev/mst/mt4125_pciconf0 --yes set
FLEX_PARSER_PROFILE_ENABLE=4 $ sudo mlxconfig -d /dev/mst/mt4125_pciconf0 --
yes set PROG_PARSE_GRAPH=1 # Accurate TX scheduling enable $ sudo mlxconfig -d
/dev/mst/mt4125_pciconf0 --yes set REAL_TIME_CLOCK_ENABLE=1 $ sudo mlxconfig
-d /dev/mst/mt4125_pciconf0 --yes set ACCURATE_TX_SCHEDULER=1 # Maximum
level of CQE compression $ sudo mlxconfig -d /dev/mst/mt4125_pciconf0 --yes set
CQE_COMPRESSION=1 # Reset NIC $ sudo mlxfwreset -d /dev/mst/mt4125_pciconf0
--yes --level 3 r
```

To verify that the above NIC features are enabled:

```
$ sudo mlxconfig -d /dev/mst/mt4125_pciconf0 q | grep
"CQE_COMPRESSION\|PROG_PARSE_GRAPH\|FLEX_PARSER_PROFILE_ENABLE\|REAL_
FLEX_PARSER_PROFILE_ENABLE 4 PROG_PARSE_GRAPH True(1)
ACCURATE_TX_SCHEDULER True(1) CQE_COMPRESSION AGGRESSIVE(1)
REAL_TIME_CLOCK_ENABLE True(1)
```

Update A100X BFB Image and NIC Firmware

NOTE: The following instructions are specifically for A100X boards. Validate and ensure that RSHIM and MFT are installed on the system.

```
# Enable MST $ sudo mst start $ sudo mst status MST modules: ------------ MST PCI
module is not loaded MST PCI configuration module loaded MST devices: ------------
/dev/mst/mt41686_pciconf0 - PCI configuration cycles access.
domain:bus:dev.fn=0000:cc:00.0 addr.reg=88 data.reg=92 cr_bar.gw_offset=-1 Chip
revision is: 01 # Update BFB image $ wget
https://content.mellanox.com/BlueField/BFBs/Ubuntu22.04/DOCA_2.5.0_BSP_4.5.0_Ub
1.23-10.prod.bfb $ sudo bfb-install -r rshim0 -b
DOCA_2.5.0_BSP_4.5.0_Ubuntu_22.04-1.23-10.prod.bfb Pushing bfb 920MiB 0:01:51
[8.22MiB/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]: lifecycle Secured (development) INFO[BL31]: runtime INFO[UEFI]:
eMMC init INFO[UEFI]: UPVS valid INFO[UEFI]: eMMC probed INFO[UEFI]: PMI:
updates started INFO[UEFI]: PMI: boot image update INFO[UEFI]: PMI: updates
completed, status 0 INFO[UEFI]: PCIe enum start INFO[UEFI]: PCIe enum end
INFO[UEFI]: exit Boot Service INFO[MISC]: Ubuntu installation started INFO[MISC]:
Installing OS image INFO[MISC]: Installation finished # Update NIC firmware $ wget
https://www.mellanox.com/downloads/firmware/fw-BlueField-2-rel-24_39_2048-
699210040230_Ax-NVME-20.4.1-UEFI-21.4.13-UEFI-22.4.12-UEFI-14.32.17-FlexBoot-
3.7.300.signed.bin.zip $ unzip fw-BlueField-2-rel-24_39_2048-699210040230_Ax-
NVME-20.4.1-UEFI-21.4.13-UEFI-22.4.12-UEFI-14.32.17-FlexBoot-
3.7.300.signed.bin.zip $ sudo flint -d /dev/mst/mt41686_pciconf0 -i fw-BlueField-2-
rel-24_39_2048-699210040230_Ax-NVME-20.4.1-UEFI-21.4.13-UEFI-22.4.12-UEFI-
14.32.17-FlexBoot-3.7.300.signed.bin -y b Current FW version on flash: 24.35.1012
New FW version: 24.39.2048 FSMST_INITIALIZE - OK Writing Boot image component
- OK Restoring signature - OK # NOTE: need full Power cycle from host with cold boot #
Verify NIC FW version after reboot $ sudo mst start $ sudo flint -d
/dev/mst/mt41686_pciconf0 q Image type: FS4 FW Version: 24.39.2048 FW Release
Date: 29.11.2023 Product Version: 24.39.2048 Rom Info: type=UEFI Virtio net
version=21.4.13 cpu=AMD64,AARCH64 type=UEFI Virtio blk version=22.4.12
cpu=AMD64,AARCH64 type=UEFI version=14.32.17 cpu=AMD64,AARCH64 type=PXE
version=3.7.300 cpu=AMD64 Description: UID GuidsNumber Base GUID:
48b02d03005f770c 16 Base MAC: 48b02d5f770c 16 Image VSD: N/A Device VSD: N/A
PSID: NVD0000000015 Security Attributes: secure-fw
```

Run the following code to switch the A100x to the **BF2-as-CX** mode:

```
# Setting BF2 port to Ethernet mode (not Infiniband) $ sudo mlxconfig -d
/dev/mst/mt41686_pciconf0 --yes set LINK_TYPE_P1=2 $ sudo mlxconfig -d
/dev/mst/mt41686_pciconf0 --yes set LINK_TYPE_P2=2 # Setting BF2 Embedded CPU
mode $ sudo mlxconfig -d /dev/mst/mt41686_pciconf0 --yes set
INTERNAL_CPU_MODEL=1 $ sudo mlxconfig -d /dev/mst/mt41686_pciconf0 --yes set
INTERNAL_CPU_PAGE_SUPPLIER=EXT_HOST_PF $ sudo mlxconfig -d
/dev/mst/mt41686_pciconf0 --yes set
INTERNAL_CPU_ESWITCH_MANAGER=EXT_HOST_PF $ sudo mlxconfig -d
/dev/mst/mt41686_pciconf0 --yes set INTERNAL_CPU_IB_VPORT0=EXT_HOST_PF $
sudo mlxconfig -d /dev/mst/mt41686_pciconf0 --yes set
INTERNAL_CPU_OFFLOAD_ENGINE=DISABLED # Accurate scheduling related settings $
sudo mlxconfig -d /dev/mst/mt41686_pciconf0 --yes set CQE_COMPRESSION=1 $
sudo mlxconfig -d /dev/mst/mt41686_pciconf0 --yes set PROG_PARSE_GRAPH=1 $
sudo mlxconfig -d /dev/mst/mt41686_pciconf0 --yes set
ACCURATE_TX_SCHEDULER=1 $ sudo mlxconfig -d /dev/mst/mt41686_pciconf0 --yes
set FLEX_PARSER_PROFILE_ENABLE=4 $ sudo mlxconfig -d
/dev/mst/mt41686_pciconf0 --yes set REAL_TIME_CLOCK_ENABLE=1 # NOTE: You
must power cycle the host for those settings to take effect # Verify that the NIC FW
changes have been applied $ sudo mlxconfig -d /dev/mst/mt41686_pciconf0 q | grep
"CQE_COMPRESSION\|PROG_PARSE_GRAPH\|ACCURATE_TX_SCHEDULER\
\|FLEX_PARSER_PROFILE_ENABLE\|REAL_TIME_CLOCK_ENABLE\|INTERNAL_CPU_MOD
\|INTERNAL_CPU_PAGE_SUPPLIER\|INTERNAL_CPU_ESWITCH_MANAGER\|INTERNAL_
        INTERNAL_CPU_MODEL                EMBEDDED_CPU(1)
        INTERNAL_CPU_PAGE_SUPPLIER        EXT_HOST_PF(1)
        INTERNAL_CPU_ESWITCH_MANAGER      EXT_HOST_PF(1)
        INTERNAL_CPU_IB_VPORT0            EXT_HOST_PF(1)
        INTERNAL_CPU_OFFLOAD_ENGINE       DISABLED(1)
        FLEX_PARSER_PROFILE_ENABLE        4
        PROG_PARSE_GRAPH                  True(1)
        ACCURATE_TX_SCHEDULER             True(1)
        CQE_COMPRESSION                   AGGRESSIVE(1)
        REAL_TIME_CLOCK_ENABLE            True(1)
        LINK_TYPE_P1                      ETH(2)        LINK_TYPE_P2                      ETH(2)
```

Update the NVIDIA CUDA Driver

If the installed CUDA driver is older than the version specified in the release manifest, follow these instructions to remove the old CUDA driver and install the driver version that matches the version specified in the release manifest.

Removing an Old CUDA Driver

If this is the first time you are installing a CUDA driver on the system, you can skip directly to “Installing CUDA Driver”. If this is an existing system and you don’t know how the CUDA driver was previously installed, see How to remove old CUDA toolkit and driver.

Run the following commands to remove the old CUDA driver.

```
# Unload dependent kernel modules $ sudo service nvidia-persistenced stop $ sudo
service nvidia-docker stop $ for m in $(lsmod | awk "/^[^[:space:]]*
(nvidia|nv_|gdrdrv)/ {print \$1}"); do echo Unload $m...; sudo rmmod $m; done #
Check the installed CUDA driver version $ apt list --installed | grep cuda-drivers #
Remove the driver if you have the older version installed. # For example, cuda-drivers-
520 was installed on the system. $ sudo apt purge cuda-drivers-520 $ sudo apt
autoremove # Remove the driver if it was installed by runfile installer before. $ sudo
/usr/bin/nvidia-uninstall
```

Installing CUDA Driver

Note

Aerial has been using the open-source GPU kernel driver (OpenRM) since the 23-4 release.

Run the following commands to install the **NVIDIA open-source GPU kernel driver** (OpenRM).

```
# Install CUDA driver $ wget https://download.nvidia.com/XFree86/Linux-
x86_64/535.54.03/NVIDIA-Linux-x86_64-535.54.03.run $ sudo sh NVIDIA-Linux-
x86_64-535.54.03.run --silent -m kernel-open # Verify if the driver is loaded
```

```
successfully $ nvidia-smi +-----------------------------------------------------------------------------------
---+ | NVIDIA-SMI 535.54.03 Driver Version: 535.54.03 CUDA Version: 12.2 | |-----------
------------------------------+----------------------+----------------------+ | 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 A100-PCIE-40GB Off | 00000000:B6:00.0 Off | 0 | | N/A 26C P0 38W /
250W | 4MiB / 40960MiB | 44% Default | | | | Disabled | +-----------------------------------
-----+---------------------+---------------------+ +-------------------------------------------------------------
----------------------+ | Processes: | | GPU GI CI PID Type Process name GPU Memory |
| ID ID Usage |
|==================================================================
| No running processes found | +----------------------------------------------------------------------
---------------+
```

Removing the Old Aerial cuBB Container

This step is optional. To remove the old cuBB container, enter the following commands:

```
$ sudo docker stop <cuBB container name> $ sudo docker rm <cuBB container
name>
```

Installing the New Aerial cuBB Container

The cuBB container is available on the NVIDIA GPU Cloud (NGC). Follow the instructions on that page to pull the container and to run the container.

Note

If you receive the cuBB container image via nvonline, run “docker load < cuBB container image file” to load the image. Then use the same docker run command detailed on the NGC page to launch it.