First-time Installation Procedure

Table of contents

The installation procedure depends on the active mode of operation on the NVIDIA® BlueField® networking platform:

- Installation for DPU Mode (default mode for BlueField DPUs)

- Installation for NIC Mode (default mode for BlueField SuperNICs)

Installation for DPU Mode

Contents:

> ⓘ **Note**
>
> DPU mode is the default mode for BlueField DPUs, while BlueField SuperNICs are shipped with NIC mode as their default. To switch between the modes, see NVIDIA BlueField Modes of Operation. To check which mode your BlueField is currently running, refer to section "Identifying Which Mode BlueField is Currently Operating In" on the same page.
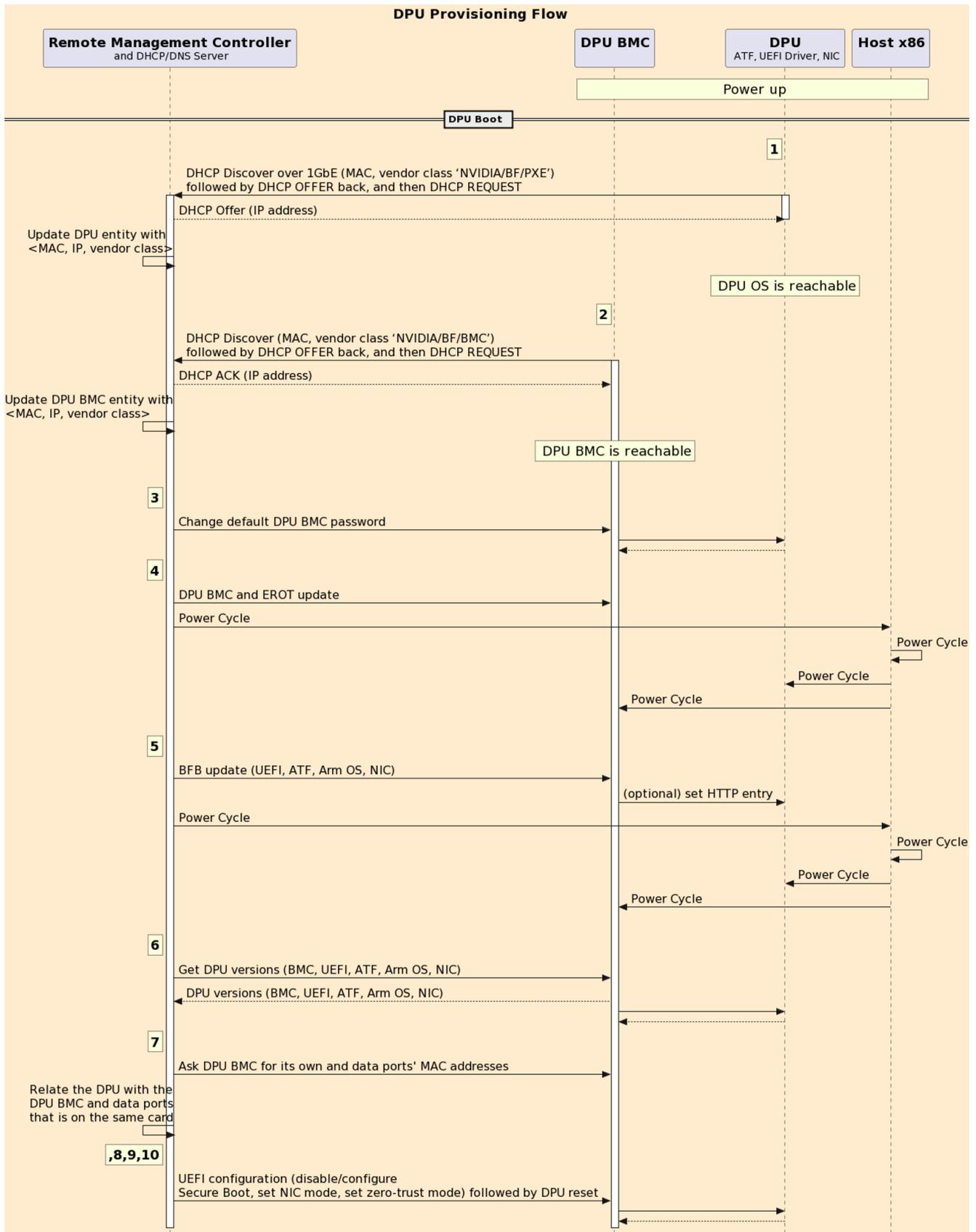
> ⓘ **Note**
>
> In the out-of-box state of the BlueField the host is assumed to be trusted. Later in this procedure, after performing BFB Bundle update, a step is provided to disable the host RShim which the user must perform to protect the BlueField from potential security threats from the host.

The following diagram illustrates the sequence of events and actions from first time power-up of the NVIDIA® BlueField® networking platform (DPU or SuperNIC) in the data center environment through provisioning and maintenance.

> ⓘ **Info**

The numbers indicated in the sequence diagram correspond to the steps that follow it.

DPU Provisioning Flow

Remote Management Controller
and DHCP/DNS Server

DPU BMC

DPU
ATF, UEFI Driver, NIC

Host x86

Power up

DPU Boot

1

DHCP Discover over 1GbE (MAC, vendor class 'NVIDIA/BF/PXE')
followed by DHCP OFFER back, and then DHCP REQUEST

DHCP Offer (IP address)

Update DPU entity with
<MAC, IP, vendor class>

DPU OS is reachable

2

DHCP Discover (MAC, vendor class 'NVIDIA/BF/BMC')
followed by DHCP OFFER back, and then DHCP REQUEST

DHCP ACK (IP address)

Update DPU BMC entity with
<MAC, IP, vendor class>

DPU BMC is reachable

3

Change default DPU BMC password

4

DPU BMC and EROT update

Power Cycle

Power Cycle

Power Cycle

Power Cycle

5

BFB update (UEFI, ATF, Arm OS, NIC)

(optional) set HTTP entry

Power Cycle

Power Cycle

Power Cycle

Power Cycle

6

Get DPU versions (BMC, UEFI, ATF, Arm OS, NIC)

DPU versions (BMC, UEFI, ATF, Arm OS, NIC)

7

Ask DPU BMC for its own and data ports' MAC addresses

Relate the DPU with the
DPU BMC and data ports
that is on the same card

,8,9,10

UEFI configuration (disable/configure
Secure Boot, set NIC mode, set zero-trust mode) followed by DPU reset

At the end of this procedure, the BlueField should be configured with an IP address, all required settings, has up-to-date software component versions, and is ready to use.

Step 1 – BlueField SoC Boots

The BlueField SoC boots to the UEFI BIOS and DHCP DISCOVER is sent

1. BlueField SoC runs UEFI/PXE which sends a DHCP DISCOVER over the 1GbE OOB interface, including vendor class (`"NVIDIA/BF/PXE"`) for BlueField SoC (to allow customer's server to differentiate between BlueField SoC and BlueField BMC), and MAC for identification and discovery. See Appendix B for more information.

2. A customer's DHCP server inspects the MAC address and the vendor class, allocates IP, and continues the standard DHCP.

3. DHCP server updates RMC of the new BlueField discovered with detailed information (e.g., MAC, IP address, vendor class).

Step 2 – BlueField BMC Boots

BlueField BMC issues DHCP DISCOVER over the 1GbE OOB interface, including vendor class (`"NVIDIA/BF/BMC"`) for BlueField-BMC, and MAC for identification and discovery. Example of BlueField BMC DHCP DISCOVER packet structure (note `"NVIDIA/BF/BMC"` in line 13):

```
root@bf-bmc:~# 18:18:10.563269 IP (tos 0xc0, ttl 64, id 0, offset 0,
flags [none], proto UDP (17), length 320)
0.0.0.0.bootpc > 255.255.255.255.bootps: [udp sum ok] BOOTP/DHCP,
Request from b8:3f:d2:ca:4b:26 (oui Unknown), length 292, xid
0xfc2acdec, secs 1, Flags [none] (0x0000)
Client-Ethernet-Address b8:3f:d2:ab:cd:ef (oui Unknown)
Vendor-rfc1048 Extensions
Magic Cookie 0x63825363
DHCP-Message (53), length 1: Discover
Client-ID (61), length 7: ether b8:3f:d2:ab:cd:ef
Parameter-Request (55), length 9:
Subnet-Mask (1), Default-Gateway (3), Domain-Name-Server (6),
Hostname (12)
```

```
Domain-Name (15), Static-Route (33), NTP (42), Unknown (120)
Classless-Static-Route (121)
MSZ (57), length 2: 576
Hostname (12), length 7: "bf-bmc" Vendor-Class (60), length 13:
"NVIDIA/BF/BMC" END (255), length 0
18:18:10.565261 IP (tos 0x0, ttl 63, id 0, offset 0, flags [DF], proto
UDP (17), length 353)
(example) dhcp01.XX.YY > ldev-platform-13-043-bmc.bootpc: [no
cksum] BOOTP/DHCP, Reply, length 325, hops 1, xid 0xfc2acdec, secs 1,
Flags [none] (0x0000)
(example) Your-IP ldev-platform-13-043-bmc.XX.YY
(example) Server-IP l-pxe02.XX.YY
Gateway-IP 10.237.0.255
Client-Ethernet-Address b8:3f:d2:ab:cd:ef (oui Unknown)
file "pxelinux.0" Vendor-rfc1048 Extensions
Magic Cookie 0x63825363
DHCP-Message (53), length 1: Offer
Server-ID (54), length 4: (example) dhcp01.XX.YY
Lease-Time (51), length 4: 43200
Subnet-Mask (1), length 4: 255.255.0.0
Default-Gateway (3), length 4
(example) GW.XX.YY
Hostname (12), length 24: "ldev-platform-13-043-bmc" Domain-Name (15),
length 13: "<local domain name>" NTP (42), length 4: (example) NTP.XX.YY
END (255), length 0
18:18:10.565261 IP (tos 0x0, ttl 62, id 0, offset 0, flags [DF], proto
UDP (17), length 353)
dhcp01.XX.YY > ldev-platform-13-043-bmc.<local domain name>: [no
cksum] BOOTP/DH
```

1. DHCP server inspects the MAC address and the vendor class, allocates IP and continues the standard DHCP flow.

2. DHCP server updates RMC of the new BlueField BMC discovered with detailed information: MAC, IP address, vendor classes, etc.

Step 3 – Change Default Password

To communicate with the BlueField BMC, change the default password (`0penBmc`) by sending the following Redfish schema to the BlueField BMC:

```
curl -k -u root:0penBmc -H "Content-Type: application/json" -X
PATCH https://<BF-BMC-IP>/redfish/v1/AccountService/Accounts/root
-d '{"Password" : "<user-password>"}'
```

Where `<BF-BMC-IP>` is the IP address for the BlueField BMC (e.g., 10.10.1.2), and `<user-password>` is the chosen password to log into the BlueField BMC with root privileges.

The BMC password must comply with the following policy parameters:

- Using ASCII and Unicode characters is permitted

- Minimum length: 12

- Maximum length: 20

- Maximum number of consecutive character pairs: 4

> **ⓘ Info**
>
> Two characters are consecutive if `|hex(char_1)-hex(char_2)|=1`.
>
> Examples of passwords with 5 consecutive character pairs (invalid): `DcB` `a123456AbCd!`; `ab1XbcYcdZdeGef!`; `Testing_123abcgh!`.

The following is a valid example password:

- `HelloNvidia3D!`

For example:

```
[redfish_scripts] $ curl -k -u root:0penBmc -H "Content-Type:
application/json" -X PATCH https://<BF-BMC-
IP>/redfish/v1/AccountService/Accounts/root -d '{"Password" :
"HelloNvidia3D!"}'
Response:
{
  "@Message.ExtendedInfo": [
    {
      "@odata.type": "#Message.v1_1_1.Message",
      "Message": "The request completed successfully.",
      "MessageArgs": [],
      "MessageId": "Base.1.15.0.Success",
      "MessageSeverity": "OK",
      "Resolution": "None"
    }
  ]
}
```

Step 4 – Upgrade BlueField BMC Firmware

Upgrade BlueField BMC firmware via the Redfish "update service schema" through the 1GbE OOB.

- If a BlueField-2 is in your possession and it is the first time you are upgrading BlueField BMC, follow Appendix A.

- If a BlueField-3 is in your possession, follow the instructions in the following subsections

> **ⓘ Info**
>
> Make sure to download the latest BlueField BMC image available from the [BlueField Runtime and Driver Downloader](#).

Update BMC Firmware

1. Run the following Redfish command over the 1GbE out-of-band interface on the BlueField BMC to trigger a secure BlueField BMC firmware update:

```
curl -k -u root:'<password>' -H "Content-Type:
application/octet-stream" -X POST -T <package_path>
https://<BF-BMC-IP>/redfish/v1/UpdateService/update
```

Where:

- `<password>` – BlueField BMC password

- `<package_path>` – BMC firmware update package path pointing to BMC `*.fwpkg` binary (e.g., `bf3-bmc-23.09-6_opn.fwpkg`)

- `<BF-BMC-IP>` – BMC IP address

  After pushing the image to the BlueField BMC, a new task is created. Example:

```
{
  "@odata.id": "/redfish/v1/TaskService/Tasks/0",
  "@odata.type": "#Task.v1_4_3.Task",
  "Id": "0",
  "TaskState": "Running"
```

```
}
```

> **ⓘ Info**
>
> BMC firmware update takes ~12 minutes.

2. To track the progress of the update, use the task `Id` received in the response above (i.e., 0) in your query and monitor the value of the task's `PercentComplete` field:

```
curl -k -u root:'<password>' -X GET https://<BF-BMC-
IP>/redfish/v1/TaskService/Tasks/<task_id> | jq -r '
.PercentComplete'
```

Where:

- `<password>` – BlueField BMC password

- `<BF-BMC-IP>` – BMC IP address

- `<task_id>` – task ID of the update process as received in the response under the `Id` value

  Example output:

```
  % Total    % Received % Xferd  Average Speed   Time
Time     Time  Current Dload  Upload   Total   Spent
Left  Speed
100  2123  100  2123    0     0  38600      0 --:--:-- -
-:--:-- --:--:-- 37910
20
```

See `PercentComplete` is at 20 percent.

3. Proceed to the next step when the process reaches 100%.

Update eROT Firmware

1. Trigger a secure firmware update:
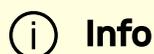
```
curl -k -u root:'<password>' -H "Content-Type:
application/octet-stream" -X POST -T <package_path>
https://<BF-BMC-IP>/redfish/v1/UpdateService/update
```

Where:

- `<password>` – BlueField BMC password

- `<package_path>` – eROT firmware update package path pointing to eROT `*.fwpkg` binary (e.g. `cec1736-ecfw-00.02.0127.0000-n02-rel-prod.fwpkg`)

- `<BF-BMC-IP>` – BMC IP address

  After initiating the eROT secure update, a new task is created. Example:

  ```
  {
    "@odata.id": "/redfish/v1/TaskService/Tasks/0",
    "@odata.type": "#Task.v1_4_3.Task",
    "Id": "0",
    "TaskState": "Running"
  }
  ```

  ⓘ **Info**

eROT firmware update takes ~20 seconds.

2. To track the progress of the update, use the task `Id` received in the response above (i.e., 0) in your query and monitor the value of the task's `PercentComplete` field:

```
curl -k -u root:'<password>' -X GET https://<BF-BMC-
IP>/redfish/v1/TaskService/Tasks/<task_id> | jq -r '
.PercentComplete'
```

> ⓘ **Note**
>
> Run this command several times until `PercentComplete` shows 100 before proceeding to other operations.

Where:

- `<password>` – BlueField BMC password

- `<BF-BMC-IP>` – BMC IP address

- `<task_id>` – task ID of the update process as received in the response under the `Id` value

> ⓘ **Note**
>
> For the firmware of the BMC and CEC to apply and to allow new Redfish APIs which are required for the following steps, a power cycle of the BlueField is required. The BlueField-3 is installed in the host's

PCIe slot. To initiate the power cycle sequence for the BlueField, the entire server on which it is installed must be power cycled.

Possible Error Codes During BMC/eROT Upgrade

Fault	Diagnosis and Possible Solution
Connection to BMC breaks during firmware package transfer	• Redfish task URI is not returned by the Redfish server • The Redfish server (if operational) is in idle state • After a reboot of BMC, or restart/recovery of the Redfish server, the Redfish server is in idle state A new firmware update can be attempted by the Redfish client.

Connection to BMC breaks during firmware update	• Redfish task URI previously returned by the Redfish server is no longer accessible
	• The Redfish server (if operational) is in one of the following states:
	   ○ In idle state, if the firmware update has completed
	   ○ In update state, if the firmware update is still ongoing
	• After a BMC reboot, or the restart/recovery of the Redfish server, the Redfish server is in idle state
	
	A new firmware update can be attempted by the Redfish client.

Two firmware update requests are initiated	The Redfish server blocks the second firmware update request and returns the following: • HTTP code 400 "Bad Request" • Redfish message based on standard registry entry UpdateInProgress Check the status of the ongoing firmware update by looking at the TaskCollection resource.
Redfish task hangs	• Redfish task URI that previously returned by the Redfish server is no longer accessible • PLDM-based firmware update progresses • After a reboot of BMC, or restart/recovery of the Redfish server, the Redfish server us in idle state

Fault	Diagnosis and Possible Solution
	A new firmware update can be attempted by the Redfish client.
BMC-EROT communication failure during image transfer	The Redfish task monitoring the firmware update indicates a failure: • TaskState is set to Exception • TaskStatus is set to Warning • Messages array in the task includes an entry based on the standard registry Update.1.0.0.TransferFailed indicating the components that failed during image transfer The Redfish client may retry the firmware update.
Firmware update fails	The Redfish task monitoring the firmware update indicates a failure: • TaskState is set to Exception • TaskStatus is set to Warning • Messages array in the task includes an entry describing the error The Redfish client may retry the firmware update.
ERoT failure (not responding)	The Redfish task monitoring the firmware update indicates a failure: • TaskState is set to Canceled • TaskStatus is set to Warning • Messages array in the task includes an entry describing the error • The Redfish client reports the error The Redfish client may retry the firmware update.
Firmware image validation failure	The Redfish task monitoring the firmware update indicates a failure: • TaskState is set to Exception • TaskStatus is set to Warning • Messages array in the task includes an entry based on the standard registry Update.1.0.0.VerificationFailed to indicate the component for which verification failed • The Redfish client reports the error

Fault	Diagnosis and Possible Solution
	The Redfish client might retry the firmware update.
Power loss before activation command is sent	• The Redfish server is in idle state A new firmware update can be attempted by the Redfish client.
Firmware activation failure	The Redfish task monitoring the firmware update indicates a failure: • TaskState is set to Exception • TaskStatus is set to Warning • Messages array in the task includes an entry based on the standard registry Update.1.0.ActivateFailed The Redfish client may retry the firmware update.
Push to BMC firmware package greater than 200 MB	• No Redfish task is created • Messages array in the task includes an entry based on the standard registry Base.1.8.1.ResourceExhaustion and a request to retry the operation is given

Step 5 – Upgrade BlueField Firmware Components and BSP

Upgrade the BlueField firmware components (i.e., ATF, UEFI, NIC-firmware) and the BSP using the BFB image.

> ⓘ **Info**
>
> Make sure to download the latest DOCA image (BFB file) available from the BlueField Runtime and Driver Downloader.

The included page could not be found.

Step 6 – Verify Software Component Versions

Verify BlueField BSP, BlueField BMC and BlueField NIC firmware versions are up to date according to the *NVIDIA BlueField BMC Software User Manual* and *NVIDIA BlueField BSP Release Notes*.

1. Use the Redfish `FirmwareInventory` schema over the 1GbE OOB interface to the BlueField's BMC:

```
[redfish_scripts] $ curl -k -u root:<password> -H "Content-
Type: application/octet-stream" -X GET https://<BF-BMC-
IP>/redfish/v1/UpdateService/FirmwareInventory
{
  "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory",
  "@odata.type":
"#SoftwareInventoryCollection.SoftwareInventoryCollection",
  "Members": [
    {
      "@odata.id":
"/redfish/v1/UpdateService/FirmwareInventory/9f7ec75a_BMC_Firm
    },
    {
      "@odata.id":
"/redfish/v1/UpdateService/FirmwareInventory/Bluefield_FW_ERoT
    },
    {
      "@odata.id":
"/redfish/v1/UpdateService/FirmwareInventory/DPU_ATF"
    },
    {
      "@odata.id":
"/redfish/v1/UpdateService/FirmwareInventory/DPU_BOARD"
    },
    {
      "@odata.id":
"/redfish/v1/UpdateService/FirmwareInventory/DPU_BSP"
```

```
      },
      {
        "@odata.id":
"/redfish/v1/UpdateService/FirmwareInventory/DPU_NIC"
      },
      {
        "@odata.id":
"/redfish/v1/UpdateService/FirmwareInventory/DPU_NODE"
      },
      {
        "@odata.id":
"/redfish/v1/UpdateService/FirmwareInventory/DPU_OFED"
      },
      {
        "@odata.id":
"/redfish/v1/UpdateService/FirmwareInventory/DPU_OS"
      },
      {
        "@odata.id":
"/redfish/v1/UpdateService/FirmwareInventory/DPU_SYS_IMAGE"
      },
      {
        "@odata.id":
"/redfish/v1/UpdateService/FirmwareInventory/DPU_UEFI"
      }
   ],
   "Members@odata.count": 11,
   "Name": "Software Inventory Collection"
}
```

Response example for `DPU_ATF`:

```
> curl -k -u root:<password> -H "Content-Type:
application/octet-stream" -X GET https://<BF-BMC-
```

```
IP>/redfish/v1/UpdateService/FirmwareInventory/DPU_ATF
{
  "@odata.id":
"/redfish/v1/UpdateService/FirmwareInventory/DPU_ATF",
  "@odata.type":
"#SoftwareInventory.v1_4_0.SoftwareInventory",
  "Description": "Host image",
  "Id": "DPU_ATF",
  "Members@odata.count": 1,
  "Name": " "Software Inventory",
  "RelatedItem": [
    {
      "@odata.id": "/redfish/v1/Systems/Bluefield/Bios"
    }
  ],
  "SoftwareId": "",
  "Status": {
    "Health": "OK",
    "HealthRollup": "OK",
    "State": "OK",
  },
  "Updateable": true,
  "Version": "v2.2(release):4.0.2-33-gd9f4ad5"
```

> **ⓘ Info**
>
> This request may also be used to query some of the other previously mentioned components (e.g., `9f7ec75a_BMC_Firmware`, `Bluefield_FW_ERoT`).

2. If the versions are not as expected, upgrade as needed:

1. Download the latest DOCA (BFB file) versions from the downloader at the bottom of the [DOCA product page](#).

2. DOCA (BFB) upgrade options (upgrading UEFI, ATF, Arm OS, NIC firmware components):

   - Recommended—BFB upgrade from remote management controller using Redfish `UpdateService` schema over 1GbE to BlueField BMC:

     ```
     export token=`curl -k -H "Content-Type: application/json" -X POST
     https://<bmc_ip>/login -d '{"username":"root", "password":"<password>"}' | grep
     token | awk '{print $2;}' | tr -d ""`
     ```

     For more information on deploying BlueField software from the BMC, refer to the "Deploying BlueField Software Using BFB from BMC" page of the [NVIDIA BlueField BSP](#) document.

Step 7 – Relate BlueField to BlueField BMC and NIC Data Ports on Same Machine

1. Get the BlueField's BMC MAC address using the following Redfish command over the 1GbE OOB port to the BlueField BMC:

```
curl -k -u root:<password> -H 'Content-Type:
application/json' -X GET https://<BF-BMC-
IP>/redfish/v1/Managers/Bluefield_BMC/EthernetInterfaces/eth0
 {
    "@odata.id":
"/redfish/v1/Managers/Bluefield_BMC/EthernetInterfaces/eth0",
    "@odata.type":
"#EthernetInterface.v1_6_0.EthernetInterface",
    "DHCPv4": {
       "DHCPEnabled": true,
       "UseDNSServers": true,
       "UseDomainName": true,
       "UseNTPServers": true
```

```
    },
    "DHCPv6": {
      "OperatingMode": "Stateful",
      "UseDNSServers": true,
      "UseDomainName": true,
      "UseNTPServers": true
    },
    "Description": "Management Network Interface",
    "FQDN": "dpu-bmc",
    "HostName": "BlueField-bmc",
    "IPv4Addresses": [
      {
        "Address": "10.237.40.179",
        "AddressOrigin": "DHCP",
        "Gateway": "0.0.0.0",
        "SubnetMask": "255.255.0.0"
      }
    ],
    "IPv4StaticAddresses": [],
    "IPv6AddressPolicyTable": [],
    "IPv6Addresses": [
      {
        "Address": "fdfd:fdfd:10:237:966d:aeff:fe17:9f5f",
        "AddressOrigin": "DHCPv6",
        "AddressState": null,
        "PrefixLength": 64
      },
      {
        "Address": "fe80::966d:aeff:fe17:9f5f",
        "AddressOrigin": "LinkLocal",
        "AddressState": null,
        "PrefixLength": 64
      }
    ],
    "IPv6DefaultGateway": "fe80::445b:ed80:5f97:8900",
    "IPv6StaticAddresses": [],
```

```
    "Id": "eth0",
    "InterfaceEnabled": true,
    "LinkStatus": "LinkUp",
    "MACAddress": "94:6d:ae:17:9f:5f",
    "MTUSize": 1500,
    "Name": "Manager Ethernet Interface",
    "NameServers": [
      "fdfd:fdfd:7:77:250:56ff:fe8b:e4f9"
    ],
    "SpeedMbps": 0,
    "StaticNameServers": [],
    "Status": {
      "Health": "OK",
      "HealthRollup": "OK",
      "State": "Enabled"
    },
    "VLANs": {
      "@odata.id":
"/redfish/v1/Managers/Bluefield_BMC/EthernetInterfaces/eth0/VL
    }
  }
```

2. Get the BlueField's high-speed port's MAC addresses using the following Redfish command over the 1GbE OOB port to the BlueField BMC:

```
curl -k -u root:<password> -H "Content-Type:
application/octet-stream" -X GET
https://<bmc_ip>/redfish/v1/Chassis/Card1/NetworkAdapters/Nvid
{
  "@odata.id":
"/redfish/v1/Chassis/Card1/NetworkAdapters/NvidiaNetworkAdapte
  "@odata.type":
"#NetworkDeviceFunction.v1_9_0.NetworkDeviceFunction",
  "Ethernet": {
```

```
      "MACAddress": "02:b1:b6:12:39:05",
      "MTUSize": 1500
    },
    "Id": "eth0f0",
    "Links": {
      "OffloadSystem": {
        "@odata.id": "/redfish/v1/Systems/Bluefield"
      },
      "PhysicalPortAssignment": {
        "@odata.id":
"/redfish/v1/Chassis/Card1/NetworkAdapters/NvidiaNetworkAdapte
      }
    },
    "Name": "NetworkDeviceFunction",
    "NetDevFuncCapabilities": [
      "Ethernet"
    ],
    "NetDevFuncType": "Ethernet"
}
```

Step 8 – Change Mode of Operation to Zero-trust Mode

Unless it is explicitly desired for the host to be trusted, make sure to disable the host PCIe RShim to protect the BlueField from potential security threats from the host:

1. Use Redfish BIOS settings schema over the 1GbE OOB to the BlueField BMC:

```
curl -k -X PATCH -d '{"Attributes":{"Internal CPU Model":
"Restricted"}}' -u root:<password> https://<BF-BMC-
IP>/redfish/v1/Systems/<SystemID>/Bios/Settings | python3 -m
json.tool
```

The available BlueField host privilege levels are `Restricted` and `Privileged`. The default is `Privileged`, where the host has access to BlueField.

2. Change the privilege level to `Restricted`.

> **(i) Note**
>
> Changing host privilege level requires BlueField reset for the change
> to take effect.

> **(i) Info**
>
> For more information on BlueField modes of operation, refer to <u>this</u>
> page.

Step 9 – (Optional) Change Mode of Operation from DPU Mode to NIC Mode

To change from <u>DPU mode to NIC mode</u> (or vice versa):

1. To enable NIC mode:

   ```
   curl -k -u root:<password> -H 'content-type:
   application/json' -d '{ "Attributes": { "NicMode": "NicMode"
   } }' -X PATCH https://<BF-BMC-
   IP>/redfish/v1/Systems/Bluefield/Bios/Settings
   ```

2. To disable NIC mode:

   ```
   curl -k -u root:<password> -H 'content-type:
   application/json' -d '{ "Attributes": { "NicMode": "DpuMode"
   ```

```
} }' -X PATCH https://<BF-BMC-
IP>/redfish/v1/Systems/Bluefield/Bios/Settings
```

3. To check that the BMC recorded the change for the next UEFI reboot to apply it:

```
curl -k -u root:<password> -H 'content-type:
application/json' -X GET https://<BF-BMC-
IP>/redfish/v1/Systems/Bluefield/Bios/Settings
```

> ℹ️ **Note**
>
> Reset the BlueField (Arm and NIC) for the mode change to take effect.

4. To verify that the NIC mode has updated accordingly:

```
curl -k -u root:<password> -H 'content-type:
application/json' -X GET https://<BF-BMC-
IP>/redfish/v1/Systems/Bluefield/Bios/
```

Step 10 – (Optional) Disable Secure Boot

As part of the default settings of the BlueField, UEFI Secure Boot is enabled and requires no special configuration to use it with the bundled Ubuntu OS shipped with the BlueField device. Disabling UEFI Secure Boot may be necessary when running an unsigned Arm OS image, such as a customer OS. Using Redfish Secure Boot schema over 1GbE to BlueField BMC, run:

The included page could not be found.

For more information on user management, review <u>this</u> page.

Installation for NIC Mode

Contents:

The following sections detail the procedure for installing BlueField software when the BlueField networking platform (DPU or SuperNIC) is running in NIC mode.

> ### ⓘ Note
>
> NIC mode is the default mode for BlueField SuperNICs, while BlueField DPUs are shipped with DPU mode as their default. To switch between the modes, see NVIDIA BlueField Modes of Operation . To check which mode your BlueField is currently running, refer to section "Identifying Which Mode BlueField is Currently Operating In" on the same page.

> ### ⓘ Note
>
> In the out-of-box state of the BlueField the host is assumed to be trusted. Later in this procedure, after performing BFB Bundle update, a step is provided to disable the host RShim which the user may perform to protect the BlueField from potential security threats from the host.

Upgrade BlueField Firmware Components and BSP Using BFB Image

Upgrade the BlueField firmware components (i.e., ATF, UEFI, NIC-firmware, BlueField BMC firmware) and the BSP using the BFB image.

Tip

Make sure to download the latest bf-fwbundle image (BFB file) available from the DOCA-Host and BlueField Bundle Runtime Downloads.
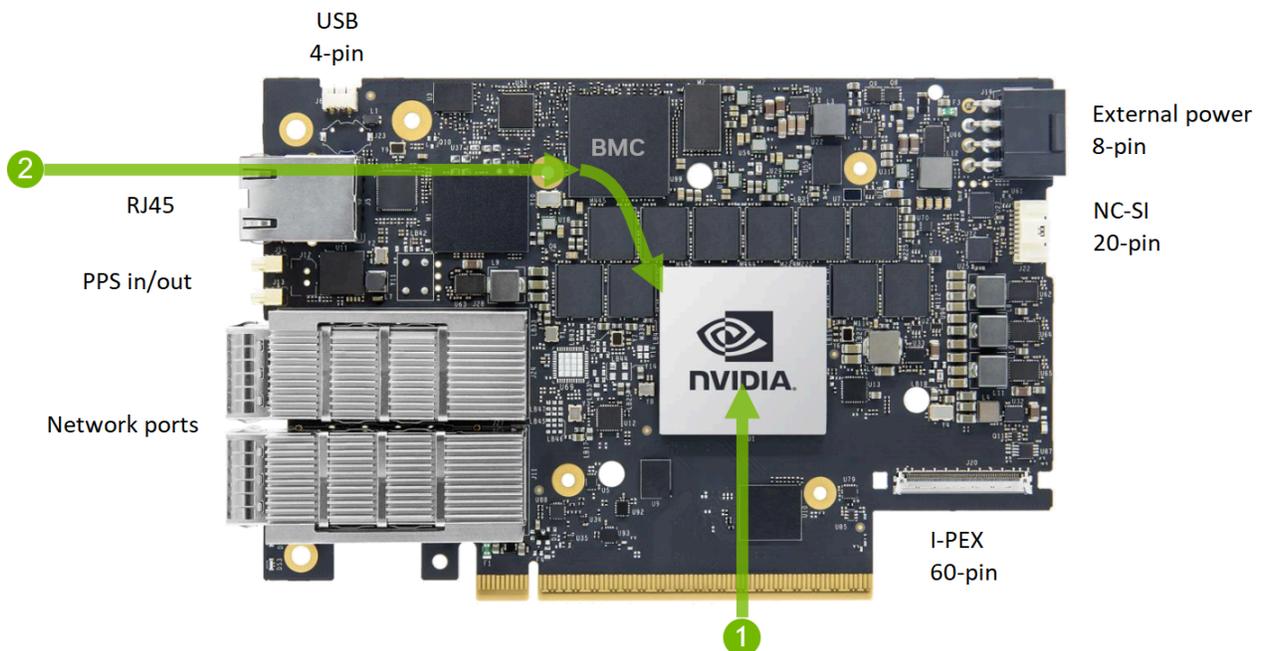
This can be performed using one of the following methods:

1. From the host x86, which should be considered as trusted during this maintenance window, follow the installation procedure here.

2. If a DPU BMC connected to the ToR switch over 1GbE is available, follow the installation procedure here.

PCIe (in-band, via host OS)
① RShim over PCIe, BFB upgrade

RJ45 (out-of-band)
② BlueField-BMC, RedFish, BFB upgrade

Changing UEFI and BMC Password Using bf.cfg

- To change the UEFI password, add the current UEFI password under parameter UEFI_PASSWORD and define the new UEFI password under NEW_UEFI_PASSWORD inside the

`bf.cfg` configuration file.

- To change the BMC root password, add the current BMC root password under parameter `BMC_PASSWORD` and define the new BMC root password under `NEW_BMC_PASSWORD` inside the `bf.cfg` configuration file.

Change Mode of Operation to Zero-trust Mode

Unless it is explicitly desired for the host to be trusted, make sure to disable the host RShim to protect the BlueField from potential security threats from the host by running the following NC-SI command from the host BMC:

Set RShim State Command Format

Byte/Bit	31:24	23:16	15:8	7:0
0...15	NC-SI Header (OEM Command)			
16:19	NVIDIA Manufacture ID (IANA) = 0x8119			
20:23	Command rev=0x00	MLNX Cmd ID= 0x12	Parameter=0x1B	Reserved
24:27	Reserved			Host_RT_Access_State
28:31	Checksum 31:0			

Set RShim State Command Parameters

Field	Bytes	Offset in NC-SI Command	Description
Host_RT_Access_State	1	27	RShim state: • 0 – Enabled • 1 – Locked • Other – reserved

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.

Trademarks

NVIDIA and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.