



Container Deployment

Table of contents

Preparing the BlueField DPU

Set BlueField to DPU Mode

Determine Your BlueField Variant

Setup DPU Management Access and Update BlueField-Bundle

Port Configuration

Creating SR-IOV Virtual Functions (Host Server)

Scalable Functions (DPU)

Install the DPL Runtime Service on the DPU

Pulling the Container Resources and Scripts from NGC

Running the Preparation Script

Edit the Configuration Files

Setting up the kubelet Pod

Recap, Full Command Sequence

Preparing the BlueField DPU

Set BlueField to DPU Mode

BlueField must run in DPU mode to use the DPL Runtime Service . For details how to change modes, see here: [BlueField Modes of Operation](#).

Determine Your BlueField Variant

Your BlueField may be Installed in a host server or it may be a standalone server.

If your BlueField is a standalone server, please ignore the parts that mention the host server or SR-IOV.

You may still use Scalable Functions (SFs) if your BlueField is a standalone server.

Setup DPU Management Access and Update BlueField-Bundle

These pages provide detailed information about DPU management access and software installation and updates:

- [Host-side Interface Configuration - NVIDIA Docs](#)
- [BF-Bundle Installation and Upgrade](#)
- [NVIDIA DOCA Downloads | NVIDIA Developer](#)

Systems with a Host Server typically use RShim (i.e. the `tmfifo_net0` interface).

Standalone systems will have to use the OOB interface option for management access.

Port Configuration

Creating SR-IOV Virtual Functions (Host Server)

The first step to use SR-IOV is to create Virtual Functions (VFs) on the host server.

VFs can be created using the following sequence:

```
sudo -S # enter sudo shell
echo 4 > /sys/class/net/eth2/device/sriov_numvfs
exit # exit sudo shell
```

Info

Entering sudo shell rather than just issuing a single `sudo` command is necessary because otherwise the `sudo` applies only to the echo command and not the hosting shell and the redirection fails with "Permission denied"

This example creates 4 VFs under Physical Function eth2. Please adjust according to your needs.

If a PF already has VFs and you'd like to change the number of VFs, please set it to 0 before applying the new value.

Scalable Functions (DPU)

For more information, see this: [BlueField Scalable Function User Guide](#)

If you create SFs, refer to their representors in the configuration file.

Install the DPL Runtime Service on the DPU

Pulling the Container Resources and Scripts from NGC

Start by downloading and installing the [ngc-cli](#) tools.

Fetch the configuration files from NGC, this will create a directory named `dpl_rt_service_<version>` .

e.g. `dpl_rt_service_v1.0.0-doca2.10.0`

Commands:

```
wget --content-disposition
https://api.ngc.nvidia.com/v2/resources/nvidia/ngc-
apps/ngc_cli/versions/3.58.0/files/ngccli_arm64.zip -O
ngccli_arm64.zip
unzip ngccli_arm64.zip
./ngc-cli/ngc registry resource download-version
"nvidia/doca/dpl_rt_service"
cd dpl_rt_service_v1.0.0-doca2.10.0
```

Running the Preparation Script

Inside the directory with the scripts and YAML files that you pulled with the ngc-cli tool, you'll find `scripts/dpl_dpu_setup.sh` .

Running this script on the DPU (requires sudo) will allow the usage of SR-IOV Virtual-Function interfaces and create the directory structure of the configuration files in directory `/etc/dpl_rt_service` . In addition, the script will set "hugepages" and call the necessary `mlxconfig` commands to use DPL Runtime Service.

Run the following sequence of commands from the working directory you pulled with the ngc-cli tool:

```
chmod +x ./scripts/dpl_dpu_setup.sh
sudo ./scripts/dpl_dpu_setup.sh
```

```
sudo systemctl restart kubelet.service
sudo systemctl restart containerd.service
```

Restarting the services is necessary for the "hugepages" change to apply to them.

i Info

The following firmware settings are set by the setup script:

- FLEX_PARSER_PROFILE_ENABLE=4
- PROG_PARSE_GRAPH=true
- SRIOV_EN=1

Edit the Configuration Files

Modify your configuration files as they are described here: [Service Configuration](#)

Important: you must create at least one device configuration under `/etc/dpl_rt_service/devices.d/`. It's advisable to start by making a copy of file `/etc/dpl_rt_service/devices.d/NAME.conf.template`.

e.g.

```
cp /etc/dpl_rt_service/devices.d/NAME.conf.template
/etc/dpl_rt_service/devices.d/1000.conf
```

Setting up the kubelet Pod

Now that everything is ready, copy the file `configs/dpl_rt_service.yaml` from the directory that you pulled with the `ngc-cli` into directory `/etc/kubelet.d`.

Please allow a few minutes for the image to be pulled and the pod to be started. you may check the progress with command `sudo journalctl -u kubelet --since -5m`, make sure to scroll down to see the latest log lines.

When the image is pulled, you will see it by using the command `sudo crictl images`.

When the pod is loaded, you will see it by using the command `sudo crictl pods`.

When the DPL Runtime Service is successfully running inside the pod, you will be able to find the log file in `/var/log/doca/dpl_rt_service/dpl_rtd.log`

Recap, Full Command Sequence

```
wget --content-disposition
https://api.ngc.nvidia.com/v2/resources/nvidia/ngc-
apps/ngc_cli/versions/3.58.0/files/ngccli_arm64.zip -O
ngccli_arm64.zip
unzip ngccli_arm64.zip
./ngc-cli/ngc registry resource download-version
"nvidia/doca/dpl_rt_service"
cd dpl_rt_service_v1.0.0v1
chmod +x ./scripts/dpl_dpu_setup.sh
sudo ./scripts/dpl_dpu_setup.sh
sudo systemctl restart kubelet.service
sudo systemctl restart containerd.service

sudo cp /etc/dpl_rt_service/devices.d/NAME.conf.template
/etc/dpl_rt_service/devices.d/1000.conf
## Modify the configuration file /etc/dpl_rt_service/devices.d/1000.conf

sudo cp configs/dpl_rt_service.yaml /etc/kubelet.d/
```

Note

The device ID and version numbers may be different in your case, please adapt as needed.

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.

© Copyright 2025, NVIDIA. PDF Generated on 04/24/2025