



NVIDIA DOCA Troubleshooting

Guide

Table of Contents

Chapter 1. Compiling DOCA Applications from Source.....	1
1.1. Meson Complains About Missing Dependencies.....	1
1.2. Error.....	1
1.3. Solution.....	2
Chapter 2. DOCA Infrastructure (PFs, VFs, SFs, OVS, etc.).....	3
2.1. Tmfifo_net0/RShim is Missing.....	3
2.1.1. Error.....	3
2.1.2. Solution.....	3
2.2. Trouble with Connection between Host and DPU.....	4
2.2.1. Error.....	4
2.2.2. Solution.....	4
2.3. DPU Name is localhost.....	5
2.3.1. Error.....	5
2.3.2. Solution.....	5
Chapter 3. DOCA Applications.....	6
3.1. SFT Error.....	6
3.1.1. Error.....	6
3.1.2. Solution.....	6
3.2. Mlx-regex Error.....	6
3.2.1. Error.....	6
3.2.2. Solution.....	6
3.3. EAL Initialization Failure.....	7
3.3.1. Error.....	7
3.3.2. Solution.....	7
3.4. DOCA Apps using DPDK in Parallel Issue.....	8
3.4.1. Error.....	8
3.4.2. Solution.....	8
Chapter 4. DOCA Libraries.....	9
4.1. DOCA Flow Error.....	9
4.1.1. Error.....	9
4.1.2. Solution.....	9
Chapter 5. Cross-compiling DOCA and CUDA.....	10
5.1. Application Build Error.....	10
5.1.1. Error.....	10
5.1.2. Solution.....	10

Chapter 1. Compiling DOCA Applications from Source

This chapter deals with troubleshooting issues related to compiling DOCA libraries (e.g., missing dependencies).

1.1. Meson Complains About Missing Dependencies

As part of DOCA's installation, a basic set of environment variables are defined so that projects (such as DOCA applications) could easily compile against the DOCA SDK, and to allow users easy access to the various DOCA tools. In addition, the set of DOCA applications sometimes rely on various 3rd party dependencies, some of which require specific environment variables so to be correctly found by the compilation environment (meson).

1.2. Error

There are multiple forms this error may appear in, such as:

- ▶ DOCA libraries are missing:

```
Dependency doca-argp found: NO (tried pkgconfig and cmake)
meson.build:13:1: ERROR: Dependency "doca-argp" not found, tried pkgconfig and
cmake
```

- ▶ DPDK definitions are missing:

```
Dependency libdpdk found: NO (tried pkgconfig and cmake)
meson.build:41:1: ERROR: Dependency "libdpdk" not found, tried pkgconfig and
cmake
```

- ▶ gRPC definitions are missing (when gRPC support is activated):

```
Dependency protobuf found: NO (tried pkgconfig and cmake)
meson.build:47:1 ERROR: Dependency "protobuf" not found, tried pkgconfig and
cmake
```

- ▶ gRPC compiler definitions are missing (when gRPC support is activated):

```
Dependency protobuf found: YES 3.15.8.0
Dependency grpc++ found: YES 1.39.0
Program protoc found: NO
meson.build:50:1: ERROR: Program(s) ['protoc'] not found or not executable
```

1.3. Solution

All the dependencies mentioned above are installed as part of DOCA's installation, and yet it is recommended to check that the packages themselves were installed correctly. The packages that install each dependency define the environment variables needed by it and may require a restart to the user session (logon and logoff) after installation.



Note: All the following examples use the required environment variables for the DPU. For the host, the values should be adjusted accordingly (aarch64 is for the DPU and x86 is for the host):

```
aarch64-linux-gnu → x86_64-linux-gnu
```

DOCA Libraries & Tools:

- ▶ For Ubuntu:

```
export PKG_CONFIG_PATH=${PKG_CONFIG_PATH}:/opt/mellanox/doca/lib/aarch64-linux-  
gnu/pkgconfig  
export PATH=${PATH}:/opt/mellanox/doca/tools
```

- ▶ For CentOS:

```
export PKG_CONFIG_PATH=${PKG_CONFIG_PATH}:/opt/mellanox/doca/lib64/pkgconfig  
export PATH=${PATH}:/opt/mellanox/doca/tools
```

DPDK:

- ▶ For Ubuntu:

```
export PKG_CONFIG_PATH=${PKG_CONFIG_PATH}:/opt/mellanox/dpdk/lib/aarch64-linux-  
gnu/pkgconfig
```

- ▶ For CentOS:

```
export PKG_CONFIG_PATH=${PKG_CONFIG_PATH}:/opt/mellanox/dpdk/lib64/pkgconfig
```

gRPC:

- ▶ For Ubuntu:

```
export PKG_CONFIG_PATH=${PKG_CONFIG_PATH}:/opt/mellanox/grpc/lib/pkgconfig  
export PATH=${PATH}:/opt/mellanox/grpc/bin
```

- ▶ For CentOS:

```
export PKG_CONFIG_PATH=${PKG_CONFIG_PATH}:/opt/mellanox/grpc/lib/pkgconfig:/opt/  
mellanox/grpc/lib64/pkgconfig  
export PATH=${PATH}:/opt/mellanox/grpc/bin
```

Chapter 2. DOCA Infrastructure (PFs, VFs, SFs, OVS, etc.)

This chapter deals with troubleshooting issues related to infrastructure required for using DOCA, like setting up PFs, VFs and SFs, configuring OVS bridges and more.



Note: Even though PFs, VFs, SFs, OVS bridges, etc. are used with DOCA applications, this section does not cover such use cases. For DOCA applications issues, refer to [DOCA Applications](#).

2.1. Tmfifo_net0/RShim is Missing

The `tmfifo_net0` and `rshim` interfaces are missing.

2.1.1. Error

The `tmfifo_net0` interface is not listed when running `ifconfig`.

2.1.2. Solution

Restart the `rshim` service:

```
systemctl restart rshim
ip addr add dev tmfifo_net0 192.168.100.1/24
ip link set dev tmfifo_net0 up
```

If `tmfifo_net0` is still missing:

- ▶ Make sure RShim is installed on the host server (`systemctl status rshim`)
- ▶ Check if RShim is disabled using the following commands executed on the Arm side:

```
mlxprivhost -d /dev/mst/mt41686_pciconf0 q
```

If this command comes back as `disabled`, try enabling it using the following command:

```
mlxprivhost -d /dev/mst/mt41686_pciconf0 p
```
- ▶ Send a `SW_RESET` to the DPU

2.2. Trouble with Connection between Host and DPU

When trying to send packets (e.g., ping) from the host to the DPU using the PF and PF representor, the ping does not work.

2.2.1. Error

There are several possible manifestations for this error. A very common scenario is if there are two DPUs (back-to-back or on the same switch) trying to communicate with each other using the OVS bridge.

For example, assuming the OVS bridge on both DPUs includes `p0` and `pf0hpf`:

```
Bridge br0
  Port p0
    Interface p0
  Port pf0hpf
    Interface pf0hpf
  Port br0
    Interface br0
      type: internal
ovs_version: "2.15.1"
```

Trying to ping the PF0 representor on one of the hosts from the other host results in 100% packet loss:

```
host> ping -c1 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.

--- 1.1.1.1 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

2.2.2. Solution

Most likely, one (or more) of the ports are down (on the host or on the DPU).

For the host, using `ibdev2netdev`, make sure the port you are using (0 or 1) is up:

```
host> ibdev2netdev
mlx5_0 port 1 ==> ens7f0 (Down)
mlx5_1 port 1 ==> ens7f1 (Down)
```

Here you see that both ports are down.



Note: If using `ibdev2netdev` did not display the ports like the example shows, then use this command:

```
host> /etc/init.d/openibd restart
```

To solve that, bring the port you are using up and (optionally) give it an IP address:

```
host> ifconfig ens7f0 1.1.1.1/24 up
```

For the DPU, using `ifconfig`, make sure the ports you are using are up (in the example they are `p0` and `pf0hpf`):

```
dpu> ifconfig
```

If the ports do not appear, then you should bring them up and (optionally) give them an IP address like you did for the host's ports.

2.3. DPU Name is localhost

When connecting to the DPU, the name of the DPU setup would be `root@localhost` and not the name of the DPU (e.g. `1dev-doca-40-BlueField`).

2.3.1. Error

The issue here is that the OOB interface is not working. This means that there is no IP address for the `oob_net0`. You can verify this by running `ifconfig` on the DPU.

2.3.2. Solution

The solution is to give `oob_net0` an IP address in the DPU. For example, on the DPU you can run:

```
ifconfig oob_net0 10.237.53.61/24 up
```

Chapter 3. DOCA Applications

This chapter deals with troubleshooting issues related to DOCA applications.

3.1. SFT Error

An SFT error appears when running an application that requires SFs.

3.1.1. Error

This error may appear in many applications. For example, when running URL filter, the error you get is as follows:

```
Forward to SFT IPV4-UDP failed, error=SFT was not initialized
```

The error here is because the SFs you are using are not set as trusted.

3.1.2. Solution

Delete the SFs and create them again as trusted. See section "SF Configuration" in [Scalable Function Setup Guide](#).

3.2. Mlx-regex Error

When running an application that depends on a RegEx device, a RegEx device error may appear.

3.2.1. Error

This error may appear in many applications that use a RegEx device. The error is:

```
mlx5_regex: Rules program failed 22 mlx5_regex: Failed to program rxp rules.
```

The error here is `mlx-regex` is not running.

3.2.2. Solution

1. Make sure that `mlx-regex` is running. On the DPU, run:

```
DPU> systemctl status mlx-regex
```


2. You will probably see the `Active` line as `Failed` or `inactive`. To fix this, on the DPU, run:

```
DPU> systemctl restart mlx-regex
```

3. Make sure that the RegEx device is active. Run:

```
DPU> systemctl status mlx-regex
```

You should see the `Active` line as `active (running)`.

4. If the `Active` line is still `Failed`, you probably need to restart the InfiniBand (RDMA) driver. On the DPU, run:

```
DPU> /etc/init.d/openibd restart
```

5. Restart the RegEx device again. Run:

```
DPU> systemctl restart mlx-regex
```

6. This should fix the issue. Verify that the RegEx device is active again. Run:

```
DPU> systemctl status mlx-regex
```

3.3. EAL Initialization Failure

EAL initialization failure is a common error that may appear while running applications like URL Filter, Application Recognition, or others.

3.3.1. Error

The error looks like this:

```
[DOCA][ERR][ARGP]: EAL initialization failed
```

There may be many causes for this error. Some of them are as follows:

- ▶ The application requires a `.cdo` file and you gave a wrong path to the file or you did not create the file
- ▶ The application requires huge pages, and you did not allocate huge pages
- ▶ The application requires root privileges to run, and you did not run it as root

3.3.2. Solution

The following solutions are respective to the possible causes listed above:

- ▶ Check that the `.cdo` file exists and that the path that you provided is correct. If the `.cdo` path does not exist, create one using `doca-dpi-compiler`. Refer to [NVIDIA DOCA DPI Compiler](#) for more information.
- ▶ Allocate huge pages. For example, run (on the host or the DPU, depending on where you are running the application):

```
echo 2048 > /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages
```

- ▶ Run the application using `sudo` (or as root):

```
sudo <run_command>
```

3.4. DOCA Apps using DPDK in Parallel Issue

When running two DOCA apps in parallel that use DPDK, the first app runs but the second one fails.

3.4.1. Error

In this example, the first application is Application Recognition, and the second is URL Filter. The following error is received:

```
Failed to start URL Filter with output: EAL: Detected 16 lcore(s)
EAL: Detected 1 NUMA nodes
EAL: RTE Version: 'MLNX_DPDK 20.11.4.0.3'
EAL: Detected shared linkage of DPDK
EAL: Cannot create lock on '/var/run/dpdk/rte/config'. Is another primary process
running?
EAL: FATAL: Cannot init config
EAL: Cannot init config
[15:01:57:246339][DOCA][E][ARGP]: EAL initialization failed
```

The cause of the error is that the second application is using `/var/run/dpdk/rte/config` when the first application is already using it.

3.4.2. Solution

To run two applications in parallel, the second application needs to be run with DPDK EAL option `--file-prefix <name>`.

In this example, after running Application Recognition (without adding the eal option), to run URL Filter, the EAL option must be added. Run:

```
/opt/mellanox/doca/applications/url_filter/bin/doca_url_filter --file-prefix second
-a 0000:01:00.0,class=regex -a 0000:01:00.6,sft_en=1 -a 0000:01:00.7,sft_en=1 -v -c
0xff -- -p
```

Chapter 4. DOCA Libraries

This chapter deals with troubleshooting issues related to DOCA libraries.

4.1. DOCA Flow Error

When trying to add new entry to the pipe, an error is received.

4.1.1. Error

The error happens after trying to add new entry function. The error message would look similar to the following:

```
mlx5_common: Failed to create TIR using DevX
mlx5_net: Port 0 cannot create DevX TIR.
[10:26:39:622581][DOCA][ERR][dpdk_engine]: create pipe entry fail on index:1,
error=Port 0 create flow fail, type 1 message: cannot get hash queue, type=8
```

The issue here seems to be caused by SF/ports configuration.

4.1.2. Solution

To fix the issue, apply the following commands on the DPU:

```
DPU> /opt/mellanox/iproute2/sbin/devlink dev eswitch set pci/0000:03:00.0 mode
legacy
DPU> /opt/mellanox/iproute2/sbin/devlink dev eswitch set pci/0000:03:00.1 mode
legacy
DPU> echo none > /sys/class/net/p0/compat/devlink/encap
DPU> echo none > /sys/class/net/p1/compat/devlink/encap
DPU> /opt/mellanox/iproute2/sbin/devlink dev eswitch set pci/0000:03:00.0 mode
switchdev
DPU> /opt/mellanox/iproute2/sbin/devlink dev eswitch set pci/0000:03:00.1 mode
switchdev
```

Chapter 5. Cross-compiling DOCA and CUDA

This chapter deals with troubleshooting issues related to DOCA-CUDA cross-compilation.

5.1. Application Build Error

When trying to build with meson, an architecture-related error is received.

5.1.1. Error

The error may happen when trying to build DOCA or DOCA-CUDA applications.

```
ccl: error: unknown value 'corei7' for -march
```

It indicates that some dependency (usually `libdpdk`) is not taken from the host machine (i.e., the machine the executable file should be running on). This dependency should be taken from the Arm dependencies directories (the path is specified in the cross file) but is skipped if the host's `PKG_CONFIG_PATH` environment variable is used instead.

5.1.2. Solution

Make sure that the cross file contains the following `PKG_CONFIG` related definitions:

```
[built-in options]
pkg_config_path = ''
[properties]
pkg_config_libdir = ... // Some content here
```

In addition, verify that `pkg_config_libdir` properly points to all `pkgconfig`-related directories under your cross-build root directory, and that the dependency reported in the error is not missing.

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation nor any of its direct or indirect subsidiaries and affiliates (collectively: "NVIDIA") make no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assume no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

Trademarks

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

Copyright

© 2022 NVIDIA Corporation & affiliates. All rights reserved.