OVS Inside BlueField
<table>
<thead>
<tr>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verifying Host Connection on Linux</td>
</tr>
<tr>
<td>Verifying Connection from Host to BlueField</td>
</tr>
<tr>
<td>Verifying Host Connection on Windows</td>
</tr>
</tbody>
</table>
Verifying Host Connection on Linux

When the DPU is connected to another DPU on another machine, manually assign IP addresses with the same subnet to both ends of the connection.

1. Assuming the link is connected to p3p1 on the other host, run:

   ```
   $ ifconfig p3p1 192.168.200.1/24 up
   ```

2. On the host which the DPU is connected to, run:

   ```
   $ ifconfig p4p2 192.168.200.2/24 up
   ```

3. Have one ping the other. This is an example of the DPU pinging the host:

   ```
   $ ping 192.168.200.1
   ```

Verifying Connection from Host to BlueField

There are two SFs configured on the BlueField device, enp3s0f0s0 and enp3s0f1s0, and their representors are part of the built-in bridge. These interfaces will get IP addresses from the DHCP server if it is present. Otherwise it is possible to configure IP address from the host. It is possible to access BlueField via the SF netdev interfaces.

For example:

1. Verify the default OVS configuration. Run:

   ```
   # ovs-vsctl show
   5668f9a6-6b93-49cf-a72a-14fd64b4c82b
   Bridge ovsbr1
   Port pf0hpf
   Interface pf0hpf
   ```
Port ovsbr1
  Interface ovsbr1
    type: internal
Port p0
  Interface p0
Port en3f0pf0sf0
  Interface en3f0pf0sf0
Bridge ovsbr2
  Port en3f1pf1sf0
    Interface en3f1pf1sf0
Port ovsbr2
  Interface ovsbr2
    type: internal
Port pf1hpf
  Interface pf1hpf
Port p1
  Interface p1
ovs_version: "2.14.1"

2. Verify whether the SF netdev received an IP address from the DHCP server. If not, assign a static IP. Run:

```shell
# ifconfig enp3s0f0s0
enp3s0f0s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
  inet 192.168.200.125  netmask 255.255.255.0  broadcast 192.168.200.255
  inet6 fe80::8e:bcff:fe36:19bc  prefixlen 64  scopeid 0x20<link>
    ether 02:8e:bc:36:19:bc  txqueuelen 1000  (Ethernet)
RX packets 3730 bytes 1217558 (1.1 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 22 bytes 2220 (2.1 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

3. Verify the connection of the configured IP address. Run:

```shell
# ping 192.168.200.25 -c 5
64 bytes from 192.168.200.25: icmp_seq=1 ttl=64 time=0.228 ms
64 bytes from 192.168.200.25: icmp_seq=2 ttl=64 time=0.175 ms
64 bytes from 192.168.200.25: icmp_seq=3 ttl=64 time=0.232 ms
```
Verifying Host Connection on Windows

Set IP address on the Windows side for the RShim or Physical network adapter, please run the following command in Command Prompt:

```
PS C:\Users\Administrator> New-NetIPAddress -InterfaceAlias "Ethernet 16" -IPAddress "192.168.100.1" -PrefixLength 22
```

To get the interface name, please run the following command in Command Prompt:

```
PS C:\Users\Administrator> Get-NetAdapter
```

Output should give us the interface name that matches the description (e.g. NVIDIA BlueField Management Network Adapter).

<table>
<thead>
<tr>
<th>Ethernet 2</th>
<th>NVIDIA ConnectX-4 Lx Ethernet Adapter</th>
<th>6 Not Present 24-8A-07-0D-E8-1D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet 6</td>
<td>NVIDIA ConnectX-4 Lx Ethernet Ad...#2</td>
<td>23 Not Present 24-8A-07-0D-E8-1C</td>
</tr>
<tr>
<td>Ethernet 16</td>
<td>NVIDIA BlueField Management Netw...#2</td>
<td>15 Up CA-FE-01-CA-</td>
</tr>
<tr>
<td>FE-02</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Once IP address is set, Have one ping the other.

```
C:\Windows\system32>ping 192.168.100.2
```

Pinging 192.168.100.2 with 32 bytes of data:
Reply from 192.168.100.2: bytes=32 time=148ms TTL=64
Reply from 192.168.100.2: bytes=32 time=152ms TTL=64
Reply from 192.168.100.2: bytes=32 time=158ms TTL=64
Reply from 192.168.100.2: bytes=32 time=158ms TTL=64