



**Hewlett Packard**  
Enterprise

# **HPE Cray EX Series System Administration with HPE Performance Cluster Manager**

Lab module networks

# Contents

HSN device detail.....3

Fabric manager fmctl command.....4

Fabric manager fmn\_ commands.....5

Slingshot Topology Tool.....5

cm network show commands.....7

## HSN device detail

The training lab cluster contains compute nodes with Mellanox ConnectX-6 HSN devices. Air-cooled compute nodes will contain HPE Slingshot, Mellanox ConnectX-6, or Mellanox ConnectX-5 adapters. Liquid-cooled compute nodes will contain HPE Slingshot or Mellanox ConnectX-5 adapters in NIC Mezzanine cards. Adjust the commands for the adapters in the cluster.

1. In a terminal session on the admin node, list the servers with ConnectX-6 network interfaces in this classroom (the file `host-cx6` was created by an administrator).

```
cat /root/host-cx6
```

2. Show the speed of the interface in Mb/s.

```
pdsh -w ^host-cx6 ethtool hsn0 | grep Speed
```

3. Group devices that report the same speed.

```
pdsh -w ^host-cx6 ethtool hsn0 | grep Speed | dshbak -c
```

4. Show link detected status.

```
pdsh -w ^host-cx6 ethtool hsn0 | grep Link
```

5. Obtain HSN cable information on one node:

```
ssh x3019c0s29b0n0 ethtool -m hsn0 | head
```

6. Obtain HSN cable vendor detail on one node:

```
ssh x3019c0s29b0n0 ethtool -m hsn0 | grep -i vendor
```

7. Obtain HSN cable vendor detail for nodes in the `/root/host-cx6` file:

```
pdsh -w ^host-cx6 ethtool -m hsn0 | grep -i vendor | dshbak -c
```

8. List Mellanox adapters:

```
pdsh -w ^host-cx6 lspci | grep -i mellanox | dshbak -c
```

9. Show device up/down state, assigned IP address, and netmask:

```
pdsh -w ^host-cx6 ip a show hsn0 | dshbak -c
```

10. Show device up/down state:

```
ssh x3019c0s29b0n0 ip link show
```

11. Show driver and firmware version:

```
ssh x3019c0s29b0n0 ethtool -i hsn0
```

12. Show specified port ACTIVE/INIT/DOWN state and LinkUP state:

```
ssh x3019c0s29b0n0 ibstat mlx5_0 | grep -i state
```

13. From the admin node, login to the fabric manager node.

```
ssh root@fmn2
```

## Fabric manager fmctl command

fmctl is a CLI utility for interacting with the Slingshot Fabric Manager. fmctl has two modes: CLI and interactive. Each command operates on a resource path defined in the OpenAPI specification of the Fabric Manager REST API.

Basic commands:

- **get** displays the current state of a resource or metric
- **create** generates a resource with the given parameters
- **update** changes the value of a specific field
- **replace** sets the entire value of the resource to the new given parameters
- **delete** removes a resource completely

Advanced Commands:

- **interactive** opens an interactive shell session to make repeated queries
- **describe** describes the API at the specified resource path
- **waitfor** allows a user to “pause” until a resource value matches the given condition
- **version** obtains the fmctl version

Each command may have custom flags, please use fmctl <cmd> -h for details.

1. List the Slingshot switches in the fabric:

```
fmctl get switches
```

2. List the Slingshot switches in the fabric in JSON format:

```
fmctl get switches --raw
```

3. List the Slingshot switches in the fabric in JSON format and send the output through jq:

```
fmctl get switches --raw | jq
```

4. Show Slingshot switch port connections (use spacebar to page through output, type q to quit):

```
fmctl get switches/x3019c0r42b0
```

5. Show Slingshot switch port connections by specifying a full resource path (use spacebar to page through output, type q to quit):

```
fmctl get /fabric/switches/x3019c0r42b0
```

6. Increase the verbosity to see additional detail above results and near the top of results:

```
fmctl get switches/x3019c0r42b0 --verbose high
```

7. Set verbosity to debug level to see additional detail above results and near the top of results:

```
fmctl get switches/x3019c0r42b0 --verbose debug
```

8. Run a metrics query with a limit of 2:

```
fmctl get /metrics expand=true limit=2
```

9. Run a metrics query with a limit of 3:

```
fmctl get /metrics expand=true limit=3
```

10. Remove the limit; you will likely receive an errorCode message: "Query returned large number results, please specify resultLimit."

```
fmctl get /metrics expand=true
```

11. Use a filter parameter:

```
fmctl get /metrics limit=20 filter="PhysicalContext eq Configuration.SwitchFirmware"
```

12. Query a specific metric (replace <ID> with a metrics ID shown in a previous command):

```
fmctl get /metrics/<ID>
```

13. Query health status:

```
fmctl get health-engines/template-policy
```

## Fabric manager **fmn\_** commands

Fmn\_ commands exist on the FMN and include bash and Python scripts that support fabric manager configuration and status.

1. Verify that fabric is up:

```
fmn_status
```

When fmn\_status reports issues, collect additional information.

2. Add the --details option to see which links are down.

```
fmn_status --details
```

3. Display fmn version:

```
fmn_version
```

4. Verify cluster configuration and state on the Active/Standby Fabric Managers.

```
fmn_cluster_status
```

5. List commands that begin with fmn\_; these commands are used to configure, reset, manage failover, collect telemetry, update switch firmware, and so on.

```
fmn_ <TAB> <TAB>
```

## Slingshot Topology Tool

Command output is wide; resize terminal session to the width of your monitor.

1. In the fmnn terminal session, use the interactive mode of the Slingshot Topology Tool to query the switch:

**slingshot-topology-tool**

The interactive prompt appears.

(STT)

2. Use show switches to show the switches on the fabric. Output identifies the type of switch: Columbia, or the top-of-rack air-cooled switch, and Colorado, or the liquid-cooled blade switch used in the HPE Cray EX liquid-cooled supercomputer. The switch is identified by its xname

(STT) **show switches**

3. Show firmware version and the high-level health of software components:

(STT) **show switches software**

4. Show all stuck flow summary information.

(STT) **show flow\_timeout summary**

5. Show cables output shows the source and destination locations for each cable, again using xnames. It also identifies the type of connection, fabric or edge, and the status, Connected or Not Connected. It also includes the serial IDs for the cables.

(STT) **show cables**

The output of the STT show cables command should match the contents of the /opt/cray/etc/sct/Shasta\_system\_hsn\_pt\_pt.csv file.

6. Show edge ports.

(STT) **show edge**

The output of the STT show edge command should match the contents of the /opt/cray/fabric\_template.json file.

7. Show all contents of topology file. In STT, a topology is defined as a map of port-port connections for a Slingshot fabric.

(STT) **show topology**

8. Show contents of Point2Point file.

(STT) **show p2p**

9. Show all port info of a switch.

(STT) **show switch ports x3019c0r42**

10. Show all jack info of a switch.

(STT) **show switch jacks x3019c0r42**

11. Show all error flags of a switch port.

(STT) **show switch error\_flags x3019c0r42**

12. Shows list of errors within a switch.

```
(STT) show switch failures x3019c0r42
```

13. Shows rossw\_info of each port of switch.

```
(STT) show switch rossw_info x3019c0r42
```

14. Show the current Slingshot Tools version.

```
(STT) show version
```

15. Diagnose downed links.

```
(STT) show links health
```

16. Run diagnostic that provides link flapping information:

```
(STT) run show-flaps
```

17. Run diagnostic that collects fabric information from the FMN.

```
(STT) run linkdbg
```

18. Run Rosetta diagnostics for error statistics.

```
(STT) run dgrerrstat
```

19. Exit the Slingshot Topology Tool.

```
(STT) exit
```

20. Exit the fmn.

```
exit
```

## **cm network show commands**

The `cm network show` command shows properties of a cluster network. Many of the following commands follow examples shown in `cm network show` online help. These commands introduced in release 1.6.

1. In a terminal session logged into the admin node, show the network ranges for the 'head' network:

```
cm network show -R -w head
```

2. Show all the nodes and controllers on the 'data1' network and condense the names:

```
cm network show -w hsn -n -C -e
```

3. Show all the nodes, controllers, and their details on all networks

```
cm network show -n -C -d
```

4. Show all the details for the head-bmc network on one line (the `-l` option is a lowercase letter L):

```
cm network show -d -w head-bmc -l
```

5. Show all the details for head-bmc in json format:

```
cm network show -d -w head-bmc -j
```

6. Show all the details for head-bmc on one line, no header, with 6 spaces between fields (the -l option is a lowercase letter L):

```
cm network show -d -w head-bmc -l -H -S 6
```

7. Show the networks in a config-file format

```
cm network show -c
```

This completes the networks lab exercise.