OSDx Command Line Interface (CLI) Getting started

bintec-Dm 929-I v2.3.1.x

Copyright© Version 1.0 bintec elmeg

Legal Notice

Warranty

This publication is subject to change.

bintec offers no warranty whatsoever for information contained in this manual.

bintec is not liable for any direct, indirect, collateral, consequential or any other damage connected to the delivery, supply or use of this manual.

Table of Contents

Chapter 1 CLI Access

1.1 Introduction

Some OSDx products have a command line user interface. Whilst this interface is common to all products, the specific commands available in each one may differ as a result of different feature sets.

1.2 CLI access interfaces

OSDx CLI can be accessed through different interfaces. Some commands are available regardless of the interface used.

Local access via physical interfaces, if present:

• keyboard and monitor
• serial port

Remote access via IP connectivity:

• telnet
• ssh

Basic IP connectivity must be in place to remotely connect through telnet or ssh. For this option to work, the product must have default IP configuration as a factory setting or a local CLI access.

1.3 Login and default user

After connecting to the CLI through an interface, the first step is to log in with a user and password.

If a product does not have a specific factory default user and password, "admin" will be used for both in OSDx.

After a successful login, a welcome banner appears and the operational menu prompt is shown ("<user>@<hostname>$"). The operating menu has commands that perform operational tasks and show the system's status.

Example:

```
osdx login: admin
Password:


Welcome to Teldat OSDx v2.3.1.1.


This system includes free software.
Contact Teldat for licenses information and source code.

admin@osdx$
```

1.4 Completion and online help

The TAB key can be used for completion purposes and to show online help regarding the available options where completion is no longer possible. Command abbreviations are allowed as long as the command is not ambiguous.

1.5 Operational and configuration menus

The **configure** command is used to access the configuration menu. Here, the device's configuration can be modified. The prompt changes to bold letters and ends with #. The **exit** command can be used to exit the configuration menu.

Example:

```
admin@osdx$ configure
admin@osdx# exit
admin@osdx$
```

Example:

```
admin@osdx$ configure
admin@osdx# exit
admin@osdx$
```

Chapter 2 Basic operational commands

2.1 show version

Shows summarized information relative to the device: OS, hardware and system status.

Example:

```
admin@osdx$ show version
  OS vendor:              Teldat
  OS name:                OSDx
  OS version:             v2.3.1.1
  OS Linux kernel:        4.19.67
  OS built by:            jenkins@daphne
  OS build date:          Thu Mar 26 16:30:32 UTC 2020
  OS installation:        physical
  OS boot mode:           user

  Hardware vendor:        Teldat
  Hardware model:         SDE
  Hardware S/N:           GWFWPX2
  Hardware UUID:          d5604611-d884-58f7-8ea0-19332083db1c
  Hardware architecture:  amd64
  Hardware fwid:          iso
  Hardware base MAC:      e4:43:4b:6f:f1:00
  Hardware cpu:           1 x Intel(R) Xeon(R) Silver 4110 CPU @ 2.10GHz (16 cores)

  Last reboot reason:     First boot

  Date:                   Mon Mar 30 09:25:43 UTC 2020
  Uptime:                 3 days, 1:02:56
  CPU load (1m, 5m, 15m): 0.00 0.00 0.00
  CPU usage % (1m):       0.02
  Storage usage (kB):     312204/229103988
  Memory usage (kB):      857212/23114428
  Temperature (°C):       30
  Users logged in:        1
  Mode (current/next boot): user/user

  Hostname:               osdx

admin@osdx$
```

2.2 show license

Displays every open-source license linked to the OSDx project.

2.3 show interfaces

Shows basic interface status information.

Example:

```
admin@osdx$ show interfaces

----------------------------------------------------------------
Idx  Name         IP Adress          Admin  Oper  Vrf  Description
----------------------------------------------------------------
 1  lo    127.0.0.1/8               up     up
          ::1/128
 2  eth0  10.0.0.6/16               up     up          MANAGEMENT
```

```
         fe80::5054:ff:fe12:3456/64
 3  eth1                                      down    down

admin@osdx$
```

2.4 show running

Shows the active configuration applied in the system.

Example:

```
admin@osdx# show running
  # Teldat OSDx SDE version v2.3.1.1
  # Tue Mar 24 15:03:08 UTC 2020
  set interfaces ethernet eth0 address 192.168.212.177/22
  set protocols static route 0.0.0.0/0 next-hop 192.168.212.2
  set service dns resolver name-server 192.168.212.3
  set service ssh
  set system login user admin authentication encrypted-password '$6$GSjsCj8gHLv$/VcqU6FLi6CT2Oxn0MJQ2C2tqnRDr
  YKNF8HIYWJp68nvXvPdFccDsT04.WtigUONbKYrgKg8d6rEs8PjljMkH0'
admin@osdx#
```

Additionally, a *path* can be specified, showing the active configuration for a particular configuration subtree.

Example:

```
admin@osdx$ show running path system
  # Teldat OSDx SDE version v2.3.1.1
  # Mon Mar 23 16:38:18 UTC 2020
  set system login user admin authentication encrypted-password '$6$GSjsCj8gHLv$/VcqU6FLi6CT2Oxn0MJQ2C2tqnRDr
  YKNF8HIYWJp68nvXvPdFccDsT04.WtigUONbKYrgKg8d6rEs8PjljMkH0'
```

2.5 show running tree

Shows the configuration that is currently active in a tree structure with brackets.

Example:

```
admin@osdx# show running tree
  interfaces {
      ethernet eth0 {
          address 192.168.212.177/22
      }
  }
  protocols {
      static {
          route 0.0.0.0/0 {
              next-hop 192.168.212.2 {
              }
          }
      }
  }
  service {
      dns {
          resolver {
              name-server 192.168.212.3
          }
      }
      ssh {
      }
  }
  system {
      login {
          user admin {
              authentication {
                  encrypted-password $6$GSjsCj8gHLv$/VcqU6FLi6CT2Oxn0MJQ2C2tqnRDrYKNF8HIYWJp68nvXvPdFccDsT04.
```

```
          WtigUONbKYrgKg8d6rEs8PjljMkH0
                }
            }
        }
    }
admin@osdx#
```

The **path** can also be specified, so that only a particular region of the configuration template is shown.

Example:

```
admin@osdx$ show running tree path system
    login {
        user admin {
            authentication {
                encrypted-password $6$GSjsCj8gHLv$/VcqU6FLi6CT2Oxn0MJQ2C2tqnRDrYKNF8HIYWJp68nvXvPdFccDsT04.
    WtigUONbKYrgKg8d6rEs8PjljMkH0
            }
        }
    }
admin@osdx$
```

2.6 show interfaces ethernet <ethN> identify

Helps identify the interface with a physical ethernet port by flashing the port's LEDs. This command is only available for devices that support it.

Example:

```
admin@osdx$ show interfaces ethernet eth0 identify
Interface eth0 should be blinking now.
Press Enter to stop...
```

2.7 show history

Shows the CLI command history with the timestamp for each command.

Example:

```
admin@osdx$ show history
 0 2020-03-31 11:20:06  show running
 1 2020-03-31 11:20:57  show interfaces
 2 2020-03-31 11:21:00  reboot now
 3 2020-03-31 11:22:03  clear terminal
 4 2020-03-31 11:22:18  show history
admin@osdx$
```

2.8 set date

Manually sets the date.

Example:

```
admin@osdx$ set date 2018-06-29 12:23:00
admin@osdx$ set date ntp pool.ntp.org
```

2.9 poweroff now

Powers off the device.

2.10 reboot now

Reboots the device.

2.11 factory-reset

Resets the device to its factory configuration.

Example:

```
admin@osdx$ factory-reset
This will destroy all data in the current image and restore factory defaults.
Continue? (Yes/No) [No]: y
Removing current data
Rebooting...
```

2.12 Pipes

CLI allows for pipes to be used with commands.

2.12.1 file <path>

Saves the command output to a file.

Example:

```
admin@osdx$ show running | file running://running_output
  Command's output saved under "running_output"
  Filesize: 628.000 B
admin@osdx$
```

2.12.2 grep

Prints lines matching a pattern.

Example:

```
admin@osdx$ show version | grep UUID
    Hardware UUID:           d45cddc3-ecc6-5776-877e-87749826dc82
admin@osdx$
```

2.12.3 head

Shows the first part of a process output.

Example:

```
admin@osdx$ show license | head
Open source common licenses included in OSDx
********************************************************************************


********************************************************************************
License: Apache-2.0
********************************************************************************


                            Apache License
                      Version 2.0, January 2004
                   http://www.apache.org/licenses/
admin@osdx$
```

2.12.4 less

Processes output.

Example:

```
admin@osdx$ show license | less
Open source common licenses included in OSDx
********************************************************************************

********************************************************************************
License: Apache-2.0
********************************************************************************


                              Apache License
                        Version 2.0, January 2004
                     http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership of such entity.

:
```

2.12.5 tail

Shows the last part of a process output.

Example:

```
admin@osdx$ show license | tail
      claim that you wrote the original software. If you use this software
      in a product, an acknowledgment in the product documentation would be
      appreciated but is not required.
   2. Altered source versions must be plainly marked as such, and must not be
      misrepresented as being the original software.
   3. This notice may not be removed or altered from any source distribution.

   Jean-loup Gailly        Mark Adler
   jloup@gzip.org          madler@alumni.caltech.edu

admin@osdx$
```

2.12.6 time

Displays the execution time of a process.

Example:

```
admin@osdx$ image show | time
Teldat OSDx Installed images:
```

```
    v2.3.1.1 (running) (boot)
    v1.11

real    0m0.036s
user    0m0.024s
sys     0m0.012s
admin@osdx$
```

2.12.7 wc

Prints newline, word, and byte counts.

Example:

```
admin@osdx$ show license | wc
  97944  523027 4015026
admin@osdx$
```

Chapter 3 Configuration menu

The configuration menu is accessed via the **configure** command.

3.1 set

Sets one configuration path or value in the device configuration. Use TAB for completion and to obtain online help on the available options. Configuration changes do not take effect until **commit** is run.

Example:

```
admin@osdx# set interfaces ethernet eth0 address 192.168.212.177/22
```

3.2 delete

Deletes the configuration subtree that belongs to the path selected. Configuration changes do not take effect until **commit** is run.

Example:

```
admin@osdx# delete interfaces ethernet eth0
```

3.3 show

The configuration menu displays information relative to the OSDx configuration. A path can be added to show only part of the configuration. Several options are available:

3.3.1 changes

Shows the changes made to the running configuration, displaying only CLI operations that modify the current configuration. These changes do not apply unless they are committed.

Example:

```
admin@osdx# set interfaces ethernet eth1 address 2.2.2.2/24
admin@osdx# show changes
  set interfaces ethernet eth1 address 2.2.2.2/24
```

If the **tree** option is used, the whole configuration tree will be shown (with + and - prefixes highlighting additions and deletions, and the > prefix highlighting modifications). A path can be added to show only that part of the configuration.

Example:

```
admin@osdx# set interfaces ethernet eth1 address 2.2.2.2/24
admin@osdx# set system console keymap in
admin@osdx# show changes tree
  interfaces {
      ethernet eth0 {
          address 192.168.213.28/22
      }
+     ethernet eth1 {
+         address 2.2.2.2/24
+     }
  }
  protocols {
      static {
          route 0.0.0.0/0 {
              next-hop 192.168.212.2 {
              }
          }
```

```
        }
    }
    service {
        dns {
            resolver {
                name-server 192.168.212.3
            }
        }
        ssh {
        }
    }
    system {
        console {
>           keymap in
        }
        login {
            user admin {
                authentication {
                    encrypted-password $6$GSjsCj8gHLv$/VcqU6FLi6CT2Oxn0MJQ2C2tqnRDrYKNF8HIYWJp68nvXvPdFccDsT04.
    WtigUONbKYrgKg8d6rEs8PjljMkH0
                }
            }
        }
    }
}
admin@osdx#
```

3.3.2 running

Displays the active configuration without any uncommitted changes. The output of this command is the same as the one displayed in operational mode.

Example:

```
admin@osdx# set interfaces ethernet eth1 address 2.2.2.2/24
admin@osdx# show running
  # Teldat OSDx SDE version v2.3.1.1
  # Tue Mar 24 08:05:24 UTC 2020
  set interfaces ethernet eth0 address 192.168.212.177/22
  set protocols static route 0.0.0.0/0 next-hop 192.168.212.2
  set service dns resolver name-server 192.168.212.3
  set service ssh
  set system login user admin authentication encrypted-password '$6$GSjsCj8gHLv$/
VcqU6FLi6CT2Oxn0MJQ2C2tqnRDrYKNF8HIYWJp68nvXvPdFccDsT04.WtigUONbKYrgKg8d6rEs8PjljMkH0'
  set system login user admin level admin
admin@osdx# show running tree
  interfaces {
      ethernet eth0 {
          address 192.168.212.177/22
      }
  }
  protocols {
      static {
          route 0.0.0.0/0 {
              next-hop 192.168.212.2 {
              }
          }
      }
  }
  service {
      dns {
          resolver {
              name-server 192.168.212.3
          }
      }
      ssh {
      }
  }
```

```
    system {
        login {
            user admin {
                authentication {
                    encrypted-password $6$GSjsCj8gHLv$/
                    VcqU6FLi6CT2Oxn0MJQ2C2tqnRDrYKNF8HIYWJp68nvXvPdFccDsT04.WtigUONbKYrgKg8d6rEs8PjljMkH0
                }
            }
        }
    }
}
```

3.3.3 working

Offers a preview of what the resulting configuration looks like after applying any uncommitted changes.

Example:

```
admin@osdx# set interfaces ethernet eth1 address 2.2.2.2/24
admin@osdx# show working path interfaces
  set interfaces ethernet eth0 address '192.168.212.177/22'
  set interfaces ethernet eth1 address '2.2.2.2/24'
admin@osdx# show working tree
  interfaces {
      ethernet eth0 {
          address 192.168.212.177/22
      }
      ethernet eth1 {
          address 2.2.2.2/24
      }
  }
  protocols {
      static {
          route 0.0.0.0/0 {
              next-hop 192.168.212.2 {
              }
          }
      }
  }
  service {
      dns {
          resolver {
              name-server 192.168.212.3
          }
      }
      ssh {
      }
  }
  system {
      login {
          user admin {
              authentication {
                  encrypted-password $6$GSjsCj8gHLv$/
                  VcqU6FLi6CT2Oxn0MJQ2C2tqnRDrYKNF8HIYWJp68nvXvPdFccDsT04.
      WtigUONbKYrgKg8d6rEs8PjljMkH0
              }
          }
      }
  }
```

3.4 run

Allows any command from the operational menu to be entered whilst in the configuration menu.

Example:

```
admin@osdx# run show version
```

3.5 commit

Dynamically applies current configuration changes to the device (through the **set** and **delete** commands).

3.6 save

Saves the configuration that is currently active to the disk startup configuration (*config.boot* file). Optionally, *save file <filename>* saves the active configuration to a custom-named file.

3.7 load

Loads a configuration to the working configuration session. Several options are available:

- *load boot-config:* Uses the disk startup configuration file that was last saved.
- *load factory-config:* Automatically loads the factory configuration settings onto the working configuration tree.
- *load file <filename>:* Loads a specific file containing a saved configuration (e.g., some file saved using the *save file <filename>* option).

After loading the new configuration, the *commit* command must be run for the configuration to apply.

3.8 discard

Discards changes made in the current configuration (**set** or **delete**) that have not been committed yet.

3.9 exit

Exits the CLI's configuration menu and jumps to the operational menu. Alternatively, press the Ctrl+D key combination.

Chapter 4 Basic Configuration commands

4.1 set interfaces ethernet <ethX> address

Sets a fixed IP address or dhcp client on an ethernet interface.

Example:

```
admin@osdx# set interfaces ethernet eth0 address 192.168.212.171/22
admin@osdx# set interfaces ethernet eth0 address dhcp
```

4.2 set interfaces ethernet <ethX> vif <vlan-id> address

Sets a fixed IP address or dhcp client on an ethernet VLAN interface.

Example:

```
admin@osdx# set interfaces ethernet eth0 vif 10 address 192.168.212.171/22
admin@osdx# set interfaces ethernet eth0 vif 10 address dhcp
```

4.3 set protocols static route

Configures a static route.

Example:

```
admin@osdx# set protocols static route 0.0.0.0/0 next-hop 192.168.212.2
```

4.4 set service dns

Configures DNS servers.

4.4.1 set service dns resolver

Configure the DNS resolver. Only resolves local router resolutions.

Example:

```
admin@osdx# set service dns resolver name-server 192.168.212.3
```

4.4.2 set service dns forwarding

Configures DNS forwarding. Acts like a DNS server, forwarding the requests received to the servers configured. If no resolver is configured, local resolutions use forwarding servers.

Example:

```
admin@osdx# set service dns forwarding name-server 192.168.212.4
```

4.5 set system ntp server address

Configures NTP servers.

Example:

```
admin@osdx# set system ntp server address pool.ntp.org
```

4.6 set system vrf

Configures Virtual Routing and Forwarding domains (VRF). The VRF needs to be created before it is applied to an interface.

Example:

```
admin@osdx# set system vrf vrf1
admin@osdx# set interfaces ethernet eth1 vrf vrf1
```

4.7 set system time-zone

Sets the local time zone.

Example:

```
admin@osdx# set system time-zone Europe/Madrid
```

4.8 set service ssh

Enables the SSH server.

Example:

```
admin@osdx# set service ssh
```

4.9 set service cnm

Configures the CNM management platform.

Example:

```
admin@osdx# set service cnm role router
admin@osdx# set service cnm url cnm3.networkcloudmanager.com
```

4.10 set system console keymap

Configures keymap for local keyboard access.

Example:

```
admin@osdx# set system console keymap es
```

4.11 set system syslog global

Configures the syslog level.

Example:

```
admin@osdx# set system syslog global level debug
```

Chapter 5 Firmware upgrade

5.1 image add

Installs a new image.

Example:

```
admin@osdx$ image add http://someurl.com/os_iso.iso
using firmware update url: http://someurl.com/os_iso.iso
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  243M  100  243M    0      0  10.9M      0  0:00:22  0:00:22 --:--:-- 11.2M


Starting upgrade to Teldat OSDx 202003231635.5449b31-master-snapshot

Size check
Copying files
Copying current configuration
Setting up bootloader
Execute additional upgrade steps from new version
Cleaning up

Successfull upgrade to Teldat OSDx 202003231635.5449b31-master-snapshot

Run "reboot now" to boot from new version
admin@osdx$
```

5.2 image show

Shows installed images.

Example:

```
admin@osdx$ image show
Teldat OSDx Installed images:

    v2.3.1.1 (running) (boot)
    v1.11
```

5.3 image boot

Selects an image to use in the next boot.

Example:

```
admin@osdx$ image boot v1.11
Teldat OSDx Boot image set to v1.11
admin@osdx$ image show
Teldat OSDx Installed images:

    v1.13
    v1.12 (running)
    v1.11 (boot)
```

5.4 image delete

The *image delete* command allows you to delete images in the device. A single image can be deleted, as well as multiple images at once.

5.4.1 image delete <image_version>

Deletes an old image stored in the device.

Example:

```
admin@osdx$ image delete v1.13
This will destroy Teldat OSDx v1.13.
Continue? (Yes/No) [No]: Yes
Teldat OSDx image v1.13
admin@osdx$
```

Warning: The running image cannot be deleted, as shown in the following example.

Example:

```
admin@osdx$ image show
Teldat OSDx Installed images:
    v2.3.1.1 (running) (boot)
    v1.12
    v1.11
admin@osdx$ image delete v2.3.1.1
Error: Running image cannot be deleted.
Aborting...
Cleaning up
  CLI Error: Command error
admin@osdx$
```

5.4.2 image delete all

Deletes all images (except the one that is running and the one configured for the next boot).

Example:

```
admin@osdx$ image delete all
This will destroy Teldat OSDx old images.
Continue? (Yes/No) [No]: y
Teldat OSDx image v1.13 deleted
admin@osdx$
```

Chapter 6 Additional operational commands

6.1 clear

This section shows a list of additional clear commands found in the operational menu, used to clear information.

6.1.1 clear console

Clears the screen. Executing this command is the same as running "clear" at the Linux terminal.

6.1.2 clear interfaces counters

Resets interface counters to 0.

6.1.3 clear terminal

Clears the terminal. Executing this command is the same as running "reset" at the Linux terminal.

6.2 file

File operations. The following root paths are available:

- **<version>://** if supported, different image storage folder installed.
- **running://** running image storage folder.
- **usb://** plugged in usb drives. There is a folder for each plugged drive.
- **<url>:** remote url with multiple protocol support (curl).

6.2.1 file copy

Copies files. The source or destination may be a remote URL.

Example:

```
admin@osdx$ file copy running://config.boot usb://USB/
admin@osdx$
```

6.2.2 file delete

Deletes a local file.

Example:

```
admin@osdx$ file delete running://user-data/capture.pcap
Do you want to delete the file? (y/n): y
admin@osdx$
```

6.2.3 file diff

Shows changes between files.

Example:

```
admin@osdx$ file diff running://config.boot running://config2.boot

 --- /opt/vyatta/etc/config/config.boot 2020-03-24 09:38:49.928000000 +0000
+++ /opt/vyatta/etc/config/config2.boot 2020-03-24 11:03:09.444000000 +0000
```

```
@@ -3 +3 @@
-set interfaces ethernet eth0 address 192.168.212.177/22
+set interfaces ethernet eth0 address 192.168.212.178/22

admin@osdx$
```

6.2.4 file move

Moves path.

Example:

```
admin@osdx$ file move running://config.boot running://config2.boot
Path successfully moved
admin@osdx$
```

6.2.5 file show

Shows the contents of a local file or folder.

6.3 nslookup

Tests the DNS domain lookup.

Example:

```
admin@osdx$ nslookup www.teldat.com
Server:        192.168.212.3
Address:       192.168.212.3#53

Non-authoritative answer:
Name:   www.teldat.com
Address: 85.223.148.162
```

6.4 ping

Sends a ping to a host. To abort ping sending, press **Ctrl+C**.

Example:

```
admin@osdx$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=57 time=3.55 ms
^C
--- 8.8.8.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.550/3.550/3.550/0.000 ms
```

6.5 show

This section shows a list of additional "show" commands belonging to the operational menu.

6.5.1 show log

Shows system log messages. If messages do not fit due to the size of the screen, the linux-based **less** viewer is used. It makes it possible to scroll through messages using the arrow keys, search with '/', etc.

To exit the **less** view mode, press '**q**'.

6.5.2 show interfaces counters

Prints interface counters.

Example:

```
admin@osdx$ show interfaces counters


--------------------------------------------------------------------------------
Idx  Name  Oper  Rx Packets  Rx Bytes   Rx Errors  Tx Packets  Tx Bytes  Tx Errors
--------------------------------------------------------------------------------
  1  lo    up             0          0          0           0         0          0
  2  eth0  up        191513  267670033          1       90395   6046659          0
  3  eth1  down           0          0          0           0         0          0
  4  eth2  down           0          0          0           0         0          0
  5  eth3  down           0          0          0           0         0          0

admin@osdx$
```

6.5.3 show system journal

Shows journalctl in interactive mode

6.5.4 show tech-support

Shows the tech-support report. This command is an easy way to get an overview of the device and send the information to tech support.

6.6 traceroute

Analyzes packet hops.

Example:

```
admin@osdx$ traceroute 172.24.0.98
traceroute to 172.24.0.98 (172.24.0.98), 30 hops max, 60 byte packets
 1  192.168.212.2 (192.168.212.2)  0.728 ms  0.654 ms  0.643 ms
 2  172.24.0.98 (172.24.0.98)  0.349 ms  0.311 ms  0.276 ms
```

Chapter 7 Monitoring

Interactive operational commands. To exit any of the monitor commands, press *Ctrl+C*.

7.1 monitor log

Accesses a real-time system log viewer that prints new messages as they occur.

7.2 monitor system journal

Monitors the last lines of journalctl.

7.3 monitor test connection

Establishes client or server connection in TCP or UDP.

7.3.1 monitor test connection server <port> <tcp/udp>

Listens for an incoming connection on a certain port using a transport layer protocol (TCP or UDP).

7.3.2 monitor test connection server <port> <tcp/udp> local-address <address>

Listens for an incoming connection using a specified local address.

7.3.3 monitor test connection server <port> <tcp/udp> local-address <address> local-interface <interface>

Listens for an incoming connection through a specified interface.

7.3.4 monitor test connection client <port> <tcp/udp>

Initiates a connection with a remote host using a certain protocol.

7.3.5 monitor test connection client <port> <tcp/udp> local-interface <interface>

Initiates a connection with a remote host through a specified interface.

7.4 monitor test performance

Performs a bandwidth test. This command needs additional options to work, which are specified below.

7.4.1 monitor test performance server

Sets up a server to wait for bandwidth test connections. Additional options are:

- **port (default value: 5001):** Specifies a custom port to listen from.
- **local-address:** Binds the server to a specific address.
- **udp:** Changes default behavior (tcp protocol).

7.4.2 monitor test performance client <address>

Initiates a bandwidth test with a remote host. Options are:

- **port:** Initiate the bandwidth-test in a certain port.
- **local-address:** Bind the client to a specific address.
- **udp:** Use UDP protocol instead of TCP.
- **bytes:** Specify the maximum amount of bytes to transmit in the test.
- **duration (default value: 10s):** Specify the time limit in seconds for the bandwidth-test.
- **ttl :** Specify the time-to-live (TTL).

7.5 monitor traffic

Monitors network traffic features.

7.5.1 monitor traffic bandwidth

Runs a real-time bandwidth monitor (bmon).

7.5.2 monitor traffic bandwidth interface <interface>

Monitors a specific interface.

7.5.3 monitor traffic dump

Dumps traffic on a network using tcpdump.

Example:

```
admin@osdx$ monitor traffic dump detail packets 2 filter 'host 192.168.213.18'
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:24:13.377925 IP (tos 0x10, ttl 64, id 50844, offset 0, flags [DF], proto TCP (6), length 96)
    192.168.213.18.22 > 192.168.213.17.58656: Flags [P.], cksum 0x2bc8 (incorrect -> 0x29a7),
    seq 207437627:207437671, ack 3562983015, win 629, options [nop,nop,TS val 2977290 ecr 521067523],
    length 44
12:24:13.378035 IP (tos 0x10, ttl 64, id 27331, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.213.17.58656 > 192.168.213.18.22: Flags [.], cksum 0x077b (correct), seq 1, ack 44, win 1690,
 options [nop,nop,TS val 521067563 ecr 2977290], length 0
2 packets captured
22 packets received by filter
13 packets dropped by kernel
```

Data packets can be stored into data files using the *save <filename>* option. File size and rotation number can also be specified.

7.5.4 monitor traffic flow

Monitors top traffic flows and executes the *iftop* command.

7.5.5 monitor traffic flow interface <interface>

Monitors captured data flow through a specific interface.

Chapter 8 Advanced troubleshooting

This chapter divides commands by layer to help the user deal with some problems.

8.1 Physical Level

First of all, we must check all defined interfaces are working at the physical level. To do this, there are several useful commands:

- *show interfaces:* checks global information.

    Example:

    ```
    admin@osdx$ show interfaces


    ----------------------------------------------------------------------
    Idx  Name           IP Address         Admin  Oper  Vrf  Description
    ----------------------------------------------------------------------
      1  lo    127.0.0.1/8                  up     up
               ::1/128
      2  eth0  192.168.212.177/22           up     up
               fe80::5054:ff:fe12:3450/64
      3  eth1                               down   down

    admin@osdx$
    ```

- *show interfaces detail:* checks global information in greater detail.

    Example:

    ```
    admin@osdx$ show interfaces detail


    ---------------------------------------------------------------------------------------------------------
    Idx  Name           IP Address         Admin  Oper  Link  MTU   Vrf  Upper  Lower  Type      Phys addr
    ---------------------------------------------------------------------------------------------------------
      1  lo    127.0.0.1/8                  up     up    up    65536                    loopback  00:00:00:00:00:00
               ::1/128
      2  eth0  192.168.212.177/22           up     up    up    1500                     ethernet  52:54:00:12:34:50
               fe80::5054:ff:fe12:3450/64
      3  eth1                               down   down  down  1500                     ethernet  52:54:00:12:34:51

    admin@osdx$
    ```

- *show interfaces counters:* checks all interface counters.

    Example:

    ```
    admin@osdx$ show interfaces counters


    -------------------------------------------------------------------------------
    Idx  Name  Oper  Rx Packets  Rx Bytes  Rx Errors  Tx Packets  Tx Bytes  Tx Errors
    -------------------------------------------------------------------------------
      1  lo    up             1        33          0           1        33          0
      2  eth0  up          3580    306075         15       15231   8402214          0
      3  eth1  down           0         0          0           0         0          0

    admin@osdx$
    ```

- *show interfaces <intf_type>:* checks the global information pertaining to a given interface type.

    Example:

    ```
    admin@osdx$ show interfaces ethernet


    ----------------------------------------------------------------------
    Idx  Name           IP Address         Admin  Oper  Vrf  Description
    ----------------------------------------------------------------------
    ```

```
 2  eth0  192.168.212.177/22              up    up
           fe80::5054:ff:fe12:3450/64
 3  eth1                                 down   down

admin@osdx$
```

8.2 Link Level

Next, we will look at information at the link level. Different commands can be used for this task:

- *show arp:* checks global ARP information.

  Example:

```
admin@osdx$ show arp
Address                HWtype  HWaddress          Flags Mask          Iface
192.168.212.2          ether   00:a0:26:b8:c6:e0  C                   eth0
192.168.213.26         ether   70:85:c2:88:d2:25  C                   eth0
admin@osdx$
```

- *show arp <interface>:* checks arp information per interface.

  Example:

```
admin@osdx$ show arp eth0
Address                HWtype  HWaddress          Flags Mask          Iface
192.168.215.9          ether   00:0c:29:88:b1:75  C                   eth0
192.168.213.26         ether   70:85:c2:88:d2:25  C                   eth0
admin@osdx$
```

8.3 Network Level

Now we are going to check if the routing information is OK. The following commands are useful:

- *show ip route:* checks the main VRF routing table.

  Example:

```
admin@osdx$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

S>* 0.0.0.0/0 [1/0] via 192.168.212.2, eth0, 00:36:20
C>* 192.168.212.0/22 is directly connected, eth0, 00:36:21
admin@osdx$
```

- *show ip route <type>:* checks routing table entries per type.

  Example:

```
admin@osdx$ show ip route static
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

S>* 0.0.0.0/0 [1/0] via 192.168.212.2, eth0, 00:41:43
admin@osdx$
```

- *show ip route summary:* checks the summary of routing table entries.

  Example:

```
admin@osdx$ show ip route summary
```

```
Route Source        Routes              FIB  (vrf default)
connected           1                   1
static              1                   1
------
Totals              2                   2

admin@osdx$
```

- *show ip route vrf <vrf name>:* checks a given VRF routing table.

    Example:

```
admin@HUB1-OSDX$ show ip route vrf WAN1
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR,
       > - selected route, * - FIB route


VRF WAN1:
S>* 0.0.0.0/0 [1/0] via 20.20.1.251, eth2, 17:12:27
K * 0.0.0.0/0 [255/8192] unreachable (ICMP unreachable), 17:12:37
C>* 20.20.1.0/24 is directly connected, eth2, 17:12:30
admin@HUB1-OSDX$
```

- *show ip route vrf <vrf name> <type>:* checks selected VRF routing table entries by type.

    Example:

```
admin@HUB1-OSDX$ show ip route vrf WAN1 connected
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR,
       > - selected route, * - FIB route


VRF WAN1:
C>* 20.20.1.0/24 is directly connected, eth2, 17:17:30
admin@HUB1-OSDX$
```

- *show ip route vrf <vrf name> summary:* checks the counter of selected VRF routing table entries.

    Example:

```
admin@HUB1-OSDX$ show ip route vrf WAN1 summary
Route Source        Routes              FIB  (vrf WAN1)
kernel              1                   0
connected           1                   1
static              1                   1
------
Totals              3                   2

admin@HUB1-OSDX$
```

8.3.1 IPSEC protocol

Check whether the information related to the IPSEC protocol is correct.

- *show vpn ipsec policy:* checks the information available on kernel crypto policies.

```
admin@HUB1-OSDX$ show vpn ipsec policy
src 10.10.1.4/32 dst 20.20.2.1/32 proto gre
dir in priority 183488 ptype main
tmpl src 0.0.0.0 dst 0.0.0.0
proto esp reqid 15 mode transport
src 20.20.2.1/32 dst 10.10.1.4/32 proto gre dev WAN2
dir out priority 183488 ptype main
tmpl src 0.0.0.0 dst 0.0.0.0
```

```
proto esp reqid 15 mode transport
src 0.0.0.0/0 dst 0.0.0.0/0
socket in priority 0 ptype main
admin@HUB1-OSDX$
```

- *show vpn ipsec sa:* checks information related to IPSEC SA.

```
admin@HUB1-OSDX$ show vpn ipsec sa
vpnprof-dmvpn-NHRPHUB: #1165, ESTABLISHED, IKEv2, 276a467090c6fd2c_i 7fcc481678d5cabc_r*
  local  '20.20.2.1' @ 20.20.2.1[4500]
  remote '10.10.1.3' @ 10.10.1.3[4500]
  AES_GCM_16-256/PRF_HMAC_SHA2_256/ECP_256
  established 57s ago, rekeying in 564s
  NHRPHUB: #3507, reqid 7, INSTALLED, TRANSPORT, ESP:AES_GCM_16-256/ECP_256
    installed 360s ago, rekeying in 146s, expires in 300s
    in  c45d9e8d,  87744 bytes,  1810 packets,     0s ago
    out ccd642f6,  88104 bytes,  1810 packets,     0s ago
    local  20.20.2.1/32[gre]
    remote 10.10.1.3/32[gre]

admin@HUB1-OSDX$
```

- *show vpn ipsec sa local-peer <peer_addr>:* checks information related to IPSEC SA in a selected local peer.

```
admin@HUB1-OSDX$ show vpn ipsec sa local-peer 20.20.1.1
vpnprof-dmvpn-NHRPHUB: #1163, ESTABLISHED, IKEv2, ef3e7b3d67cd785e_i* d57039647d6a53ce_r
  local  '20.20.1.1' @ 20.20.1.1[4500]
  remote '10.10.1.1' @ 10.10.1.1[4500]
  AES_GCM_16-256/PRF_HMAC_SHA2_256/ECP_256
  established 226s ago, rekeying in 507s
  NHRPHUB: #3508, reqid 12, INSTALLED, TRANSPORT, ESP:AES_GCM_16-256/ECP_256
    installed 371s ago, rekeying in 6s, expires in 289s
    in  cd9844e3, 216576420 bytes, 533690 packets,     0s ago
    out c07c95dc,  90744 bytes,  1865 packets,     0s ago
    local  20.20.1.1/32[gre]
    remote 10.10.1.1/32[gre]
admin@HUB1-OSDX$
```

- *show vpn ipsec sa peer <peer_addr>:* checks information related to IPSEC SA in a selected peer.

```
admin@HUB1-OSDX$ show vpn ipsec sa peer 10.10.1.1
vpnprof-dmvpn-NHRPHUB: #1166, ESTABLISHED, IKEv2, 585f4b9f89e53fab_i* e39a9a18f6b97c43_r
  local  '20.20.2.1' @ 20.20.2.1[4500]
  remote '10.10.1.1' @ 10.10.1.1[4500]
  AES_GCM_16-256/PRF_HMAC_SHA2_256/ECP_256
  established 90s ago, rekeying in 986s
  NHRPHUB: #3533, reqid 11, INSTALLED, TRANSPORT, ESP:AES_GCM_16-256/ECP_256
    installed 12s ago, rekeying in 393s, expires in 648s
    in  c302bdee,   2976 bytes,    61 packets,     0s ago
    out c36413c4,   2996 bytes,    61 packets,     0s ago
    local  20.20.2.1/32[gre]
    remote 10.10.1.1/32[gre]

vpnprof-dmvpn-NHRPHUB: #1163, ESTABLISHED, IKEv2, ef3e7b3d67cd785e_i* d57039647d6a53ce_r
  local  '20.20.1.1' @ 20.20.1.1[4500]
  remote '10.10.1.1' @ 10.10.1.1[4500]
  AES_GCM_16-256/PRF_HMAC_SHA2_256/ECP_256
  established 275s ago, rekeying in 458s
  NHRPHUB: #3532, reqid 12, INSTALLED, TRANSPORT, ESP:AES_GCM_16-256/ECP_256
    installed 43s ago, rekeying in 290s, expires in 617s
    in  c94a2bb7, 23552306 bytes, 39531 packets,     0s ago
    out c3f4461f,  10552 bytes,   217 packets,     0s ago
    local  20.20.1.1/32[gre]
    remote 10.10.1.1/32[gre]

admin@HUB1-OSDX$
```

- *show vpn ipsec state:* checks the kernel cryptostate.

```
admin@HUB1-OSDX$ show vpn ipsec state
```

```
src 20.20.2.1 dst 10.10.2.3
proto esp spi 0xcaa27144 reqid 5 mode transport
replay-window 0
aead rfc4106(gcm(aes)) 0x06042c28354c3434b42616563350bf554189f0bf8906d619b22d6e849779f84cc4f3a84b 128
anti-replay context: seq 0x0, oseq 0x98, bitmap 0x00000000
sel src 20.20.2.1/32 dst 10.10.2.3/32 dev WAN2
src 10.10.2.3 dst 20.20.2.1
proto esp spi 0xcacf0bc4 reqid 5 mode transport
replay-window 0
aead rfc4106(gcm(aes)) 0x188294b165a8b4d79a0be4e90b7232eb85cd9794e53ac13ee1a6c10404685d70c87b1630 128
anti-replay context: seq 0x0, oseq 0x0, bitmap 0x00000000
sel src 10.10.2.3/32 dst 20.20.2.1/32

admin@HUB1-OSDX$
```

• *show vpn ipsec status:* checks the IPSEC process status.

Example:

```
admin@HUB1-OSDX$ show vpn ipsec status
IPSec Process Running PID: 2094

16 Active IPsec Tunnels

IPsec Interfaces :
admin@HUB1-OSDX$
```

8.3.2 NHRP protocol

Check whether the information regarding the NHRP protocol is correct.

• *show ip nhrp:* checks the status of tunnel interfaces.

Example:

```
admin@HUB1-OSDX$ show ip nhrp
Iface    Type     Protocol    NBMA        Private    Expires(s)   Flags   Identity
tun1     dynamic  10.1.0.1    10.10.1.4   -          45           UT      10.10.1.4
tun1     dynamic  10.2.0.1    10.10.2.4   -          42           UT      10.10.2.4
tun1     dynamic  10.2.0.3    10.10.2.1   -          47           UT      10.10.2.1
tun1     local    10.1.255.1  -           -          -                    -
tun1     dynamic  10.1.0.4    10.10.1.3   -          48           UT      10.10.1.3
tun1     dynamic  10.2.0.4    10.10.2.3   -          48           UT      10.10.2.3
tun1     dynamic  10.2.0.2    10.10.2.2   -          52           UT      10.10.2.2
tun1     dynamic  10.1.0.2    10.10.1.2   -          55           UT      10.10.1.2
tun1     dynamic  10.1.0.3    10.10.1.1   -          48           UT      10.10.1.1
tun2     local    10.2.255.1  -           -          -                    -
tun2     dynamic  10.1.0.1    10.10.1.4   -          42           UT      10.10.1.4
tun2     dynamic  10.2.0.1    10.10.2.4   -          42           UT      10.10.2.4
tun2     dynamic  10.2.0.3    10.10.2.1   -          47           UT      10.10.2.1
tun2     dynamic  10.1.0.4    10.10.1.3   -          48           UT      10.10.1.3
tun2     dynamic  10.2.0.4    10.10.2.3   -          48            T      10.10.2.3
tun2     dynamic  10.2.0.2    10.10.2.2   -          49            T      10.10.2.2
tun2     dynamic  10.1.0.2    10.10.1.2   -          51           UT      10.10.1.2
tun2     dynamic  10.1.0.3    10.10.1.1   -          48           UT      10.10.1.1
admin@HUB1-OSDX$
```

8.3.3 DMVPN protocol

Check whether the information related to the DMVPN protocol is correct.

• *show vpn dmvpn:* checks the information on dmvpn connections.

Example:

```
admin@HUB1-OSDX$ show vpn dmvpn
Src                      Dst                      Flags  SAs  Identity
```

```
20.20.1.1                 10.10.2.2              n      1     10.10.2.2
20.20.1.1                 10.10.1.2              n      1     10.10.1.2
20.20.2.1                 10.10.2.3              n      1     10.10.2.3
20.20.2.1                 10.10.1.3              n      1     10.10.1.3
20.20.2.1                 10.10.1.4              n      1     10.10.1.4
20.20.1.1                 10.10.1.1              n      1     10.10.1.1
20.20.1.1                 10.10.1.3              n      1     10.10.1.3
20.20.2.1                 10.10.1.2              n      1     10.10.1.2
20.20.1.1                 10.10.2.4              n      1     10.10.2.4
20.20.2.1                 10.10.2.4              n      1     10.10.2.4
20.20.1.1                 10.10.1.4              n      1     10.10.1.4
20.20.2.1                 10.10.2.1              n      1     10.10.2.1
20.20.2.1                 10.10.2.2              n      1     10.10.2.2
20.20.1.1                 10.10.2.1              n      1     10.10.2.1
20.20.2.1                 10.10.1.1              n      1     10.10.1.1
20.20.1.1                 10.10.2.3              n      1     10.10.2.3
admin@HUB1-OSDX$
```

8.3.4 BGP protocol

Check whether the information related to the BGP protocol is correct.

• *show ip bgp:* to check info of BGP-learned routes.

Example:

```
admin@HUB1-OSDX$ show ip bgp
BGP table version is 26, local router ID is 100.100.2.1, vrf id 0
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 30.30.1.0/24     0.0.0.0                  0         32768 ?
*=i100.100.1.1/32   10.2.0.1                20    100      0 ?
* i                 10.2.0.1                20    100      0 ?
* i                 10.1.0.1                20    100      0 ?
*>i                 10.1.0.1                20    100      0 ?
* i100.100.1.3/32   10.2.0.3                 0    100      0 ?
*>i                 10.2.0.3                 0    100      0 ?
*=i172.24.11.0/24   10.2.0.1                20    100      0 ?
* i                 10.2.0.1                20    100      0 ?
* i                 10.1.0.1                20    100      0 ?
*>i                 10.1.0.1                20    100      0 ?
*=i172.24.12.0/24   10.2.0.1                20    100      0 ?
* i                 10.2.0.1                20    100      0 ?
* i                 10.1.0.1                20    100      0 ?
*>i                 10.1.0.1                20    100      0 ?
* i172.24.31.0/24   10.2.0.3                 0    100      0 ?
*>i                 10.2.0.3                 0    100      0 ?
* i172.24.32.0/24   10.2.0.3                 0    100      0 ?
*>i                 10.2.0.3                 0    100      0 ?
*> 192.168.212.0/22 30.30.1.251              0         32768 ?

Displayed  8 routes and 20 total paths
admin@HUB1-OSDX$
```

• *show ip bgp summary:* checks general information regarding the BGP protocol.

Example:

```
admin@HUB1-OSDX$ show ip bgp summary

IPv4 Unicast Summary:
BGP router identifier 100.100.2.1, local AS number 10 vrf-id 0
BGP table version 26
RIB entries 15, using 2400 bytes of memory
```

```
Peers 12, using 246 KiB of memory
Peer groups 1, using 64 bytes of memory

Neighbor        Local            V      AS MsgRcvd MsgSent  TblVer  InQ OutQ  Up/Down State/PfxRcd
*10.1.0.1       10.1.255.1       4      10     415     424       0    0    0 06:51:41           3
*10.1.0.1       10.2.255.1       4      10     415     423       0    0    0 06:51:37           3
*10.1.0.2       10.2.255.1       4      10    1380    1392       0    0    0 22:57:57           0
*10.1.0.2       10.1.255.1       4      10    1380    1392       0    0    0 22:57:44           0
*10.2.0.1       10.1.255.1       4      10     415     420       0    0    0 06:51:33           3
*10.2.0.1       10.2.255.1       4      10     415     418       0    0    0 06:51:29           3
*10.2.0.2       10.2.255.1       4      10    1380    1392       0    0    0 22:57:56           0
*10.2.0.2       10.1.255.1       4      10    1380    1392       0    0    0 22:57:52           0
*10.2.0.3       10.1.255.1       4      10    1380    1389       0    0    0 22:54:26           3
*10.2.0.3       10.2.255.1       4      10    1380    1389       0    0    0 22:54:26           3
*10.2.0.4       10.1.255.1       4      10    1378    1391       0    0    0 22:55:03           0
*10.2.0.4       10.2.255.1       4      10    1378    1391       0    0    0 22:55:03           0

Total number of neighbors 12
* - dynamic neighbor
12 dynamic neighbor(s), limit 65535
admin@HUB1-OSDX$
```

- *show ip bgp neighbors:* checks detailed information on BGP connected neighbors.

- *show ip bgp neighbors <address> routes:* checks the routes received and accepted from a neighbor.

    Example:

```
admin@HUB1-OSDX$ show ip bgp neighbors 10.1.0.3 routes
BGP table version is 33, local router ID is 100.100.2.1, vrf id 0
Default local pref 100, local AS 20
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 100.100.1.3/32   10.1.0.3               0             0 10 ?
*> 172.24.31.0/24   10.1.0.3               0             0 10 ?
*> 172.24.32.0/24   10.1.0.3               0             0 10 ?

Displayed  3 routes and 6 total paths
admin@HUB1-OSDX$
```

8.3.5 OSPF protocol

Check whether the information related to the OSPF protocol is correct.

- *show ip ospf:* checks the general information on OSPF.

    Example:

```
admin@HUB1-OSDX$ show ip ospf
 OSPF Routing Process, Router ID: 100.100.2.1
 Supports only single TOS (TOS0) routes
 This implementation conforms to RFC2328
 RFC1583Compatibility flag is disabled
 OpaqueCapability flag is disabled
 Initial SPF scheduling delay 0 millisec(s)
 Minimum hold time between consecutive SPFs 50 millisec(s)
 Maximum hold time between consecutive SPFs 5000 millisec(s)
 Hold time multiplier is currently 1
 SPF algorithm last executed 23h03m22s ago
 Last SPF duration 120 usecs
 SPF timer is inactive
 LSA minimum interval 5000 msecs
 LSA minimum arrival 1000 msecs
 Write Multiplier set to 20
```

```
 Refresh timer 10 secs
 This router is an ASBR (injecting external routing information)
 Number of external LSA 14. Checksum Sum 0x00067226
 Number of opaque AS LSA 0. Checksum Sum 0x00000000
 Number of areas attached to this router: 1
 Area ID: 0.0.0.1
   Shortcutting mode: Default, S-bit consensus: no
   Number of interfaces in this area: Total: 1, Active: 1
   Number of fully adjacent neighbors in this area: 2
   Area has no authentication
   Number of full virtual adjacencies going through this area: 0
   SPF algorithm executed 4 times
   Number of LSA 4
   Number of router LSA 3. Checksum Sum 0x0001ba80
   Number of network LSA 1. Checksum Sum 0x0000fb11
   Number of summary LSA 0. Checksum Sum 0x00000000
   Number of ASBR summary LSA 0. Checksum Sum 0x00000000
   Number of NSSA LSA 0. Checksum Sum 0x00000000
   Number of opaque link LSA 0. Checksum Sum 0x00000000
   Number of opaque area LSA 0. Checksum Sum 0x00000000


admin@HUB1-OSDX$
```

• *show ip ospf border-routers:* checks the information on OSPF border routers.

   Example:

```
admin@HUB1-OSDX$ show ip ospf border-routers
============= OSPF router routing table =============
R    100.100.2.2          [100] area: 0.0.0.1, ASBR
                          via 30.30.1.2, eth1


admin@HUB1-OSDX$
```

• *show ip ospf database:* checks OSPF routing database information.

   Example:

```
admin@HUB1-OSDX$ show ip ospf database

       OSPF Router with ID (100.100.2.1)

             Router Link States (Area 0.0.0.1)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
30.30.1.251     30.30.1.251     1472 0x800002af 0xd5e7 1
100.100.2.1     100.100.2.1      730 0x80000035 0x307a 1
100.100.2.2     100.100.2.2      383 0x80000e5e 0xb41f 1

             Net Link States (Area 0.0.0.1)

Link ID         ADV Router      Age  Seq#       CkSum
30.30.1.251     30.30.1.251      218 0x80000105 0xfb11

             AS External Link States

Link ID         ADV Router      Age  Seq#       CkSum  Route
100.100.1.1     100.100.2.1     1060 0x8000000f 0x899a E2 100.100.1.1/32 [0x0]
100.100.1.2     100.100.2.2      505 0x80000036 0x8f61 E2 100.100.1.2/32 [0x0]
100.100.1.3     100.100.2.1      360 0x80000031 0x31ce E2 100.100.1.3/32 [0x0]
100.100.1.4     100.100.2.2       42 0x8000002f 0xc049 E2 100.100.1.4/32 [0x0]
100.100.2.1     100.100.2.1      590 0x80000033 0x9a5a E2 100.100.2.1/32 [0x0]
100.100.2.2     100.100.2.2     1245 0x80000e46 0x7b69 E2 100.100.2.2/32 [0x0]
172.24.11.0     100.100.2.1      950 0x8000000f 0x0d12 E2 172.24.11.0/24 [0x0]
172.24.12.0     100.100.2.1     1010 0x8000000f 0x021c E2 172.24.12.0/24 [0x0]
172.24.21.0     100.100.2.2      505 0x80000036 0xae34 E2 172.24.21.0/24 [0x0]
172.24.22.0     100.100.2.2      505 0x80000036 0xa33e E2 172.24.22.0/24 [0x0]
```

```
172.24.31.0     100.100.2.1     480 0x80000031 0xebfc E2 172.24.31.0/24 [0x0]
172.24.32.0     100.100.2.1     490 0x80000031 0xe007 E2 172.24.32.0/24 [0x0]
172.24.41.0     100.100.2.2      42 0x8000002f 0x17d2 E2 172.24.41.0/24 [0x0]
172.24.42.0     100.100.2.2      42 0x8000002f 0x0cdc E2 172.24.42.0/24 [0x0]


admin@HUB1-OSDX$
```

- *show ip ospf interface <intf_name>:* checks information on OSPF interfaces.

  Example:

```
admin@HUB1-OSDX$ show ip ospf interface eth1
eth1 is up
  ifindex 3, MTU 1500 bytes, BW 1000 Mbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 30.30.1.1/24, Area 0.0.0.1
  MTU mismatch detection: enabled
  Router ID 100.100.2.1, Network Type BROADCAST, Cost: 100
  Transmit Delay is 1 sec, State DROther, Priority 1
  Backup Designated Router (ID) 100.100.2.2, Interface Address 30.30.1.2
  Multicast group memberships: OSPFAllRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 2.381s
  Neighbor Count is 2, Adjacent neighbor count is 2

admin@HUB1-OSDX$
```

- *show ip ospf neighbor <intf_name>:* checks information on OSPF neighbors.

  Example:

```
admin@HUB1-OSDX$ show ip ospf neighbor eth1


Neighbor ID     Pri State          Dead Time Address       Interface        RXmtL RqstL DBsmL
100.100.2.2       1 Full/Backup     37.587s 30.30.1.2       eth1:30.30.1.1       0     0     0
30.30.1.251       1 Full/DR         34.180s 30.30.1.251     eth1:30.30.1.1       0     0     0

admin@HUB1-OSDX$
```

- *show ip ospf route:* checks OSPF routing information.

  Example:

```
admin@HUB1-OSDX$ show ip ospf route
============ OSPF network routing table ============
N    30.30.1.0/24        [100] area: 0.0.0.1
                         directly attached to eth1


============ OSPF router routing table ============
R    100.100.2.2         [100] area: 0.0.0.1, ASBR
                         via 30.30.1.2, eth1


============ OSPF external routing table ===========
N E2 100.100.1.2/32      [100/20] tag: 0
                         via 30.30.1.2, eth1
N E2 100.100.1.4/32      [100/0] tag: 0
                         via 30.30.1.2, eth1
N E2 100.100.2.2/32      [100/1] tag: 0
                         via 30.30.1.2, eth1
N E2 172.24.21.0/24      [100/20] tag: 0
                         via 30.30.1.2, eth1
N E2 172.24.22.0/24      [100/20] tag: 0
                         via 30.30.1.2, eth1
N E2 172.24.41.0/24      [100/0] tag: 0
                         via 30.30.1.2, eth1
N E2 172.24.42.0/24      [100/0] tag: 0
                         via 30.30.1.2, eth1


admin@HUB1-OSDX$
```

8.4 Transport Level

This section shows the commands used to obtain information regarding the transport layer.

8.4.1 CONNTRACK info

Check whether the conntrack information is correct.

• *show system conntrack:* checks the conntrack table.

Example:

```
admin@HUB1-OSDX$ show system conntrack
tcp      6 9 TIME_WAIT src=172.24.31.1 dst=192.168.214.56 sport=36441 dport=80 packets=272 bytes=14415 src=192.168.214.56 dst
tcp      6 7 TIME_WAIT src=172.24.32.1 dst=192.168.214.56 sport=60581 dport=80 packets=68 bytes=3863 src=192.168.214.56 dst=1
tcp      6 3 TIME_WAIT src=172.24.31.1 dst=192.168.214.56 sport=34755 dport=80 packets=43 bytes=2437 src=192.168.214.56 dst=1
tcp      6 2 CLOSE src=172.24.31.1 dst=192.168.214.56 sport=42727 dport=80 packets=5 bytes=535 src=192.168.214.56 dst=172.24.
tcp      6 4 TIME_WAIT src=172.24.31.1 dst=192.168.214.56 sport=37441 dport=80 packets=17 bytes=1255 src=192.168.214.56 dst=1
tcp      6 2 CLOSE src=172.24.31.1 dst=192.168.214.56 sport=55139 dport=80 packets=5 bytes=533 src=192.168.214.56 dst=172.24.
udp      17 179 src=10.1.0.1 dst=10.1.255.1 sport=50321 dport=60000 packets=43 bytes=2580 src=10.1.255.1 dst=10.1.0.1 sport=6
icmp     1 10 src=10.1.0.2 dst=10.1.255.1 type=8 code=0 id=37957 packets=1 bytes=64 src=10.1.255.1 dst=10.1.0.2 type=0 code=0
admin@HUB1-OSDX$
```