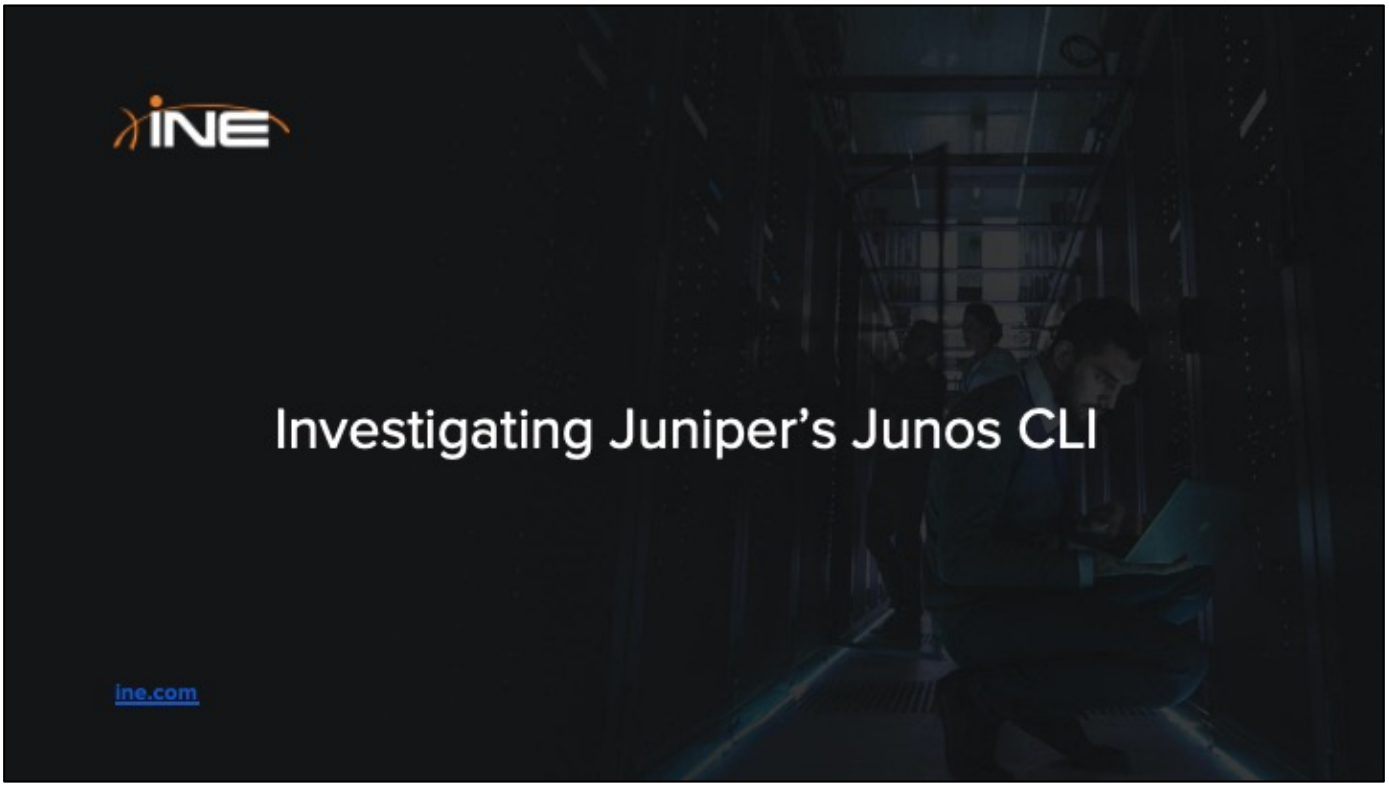
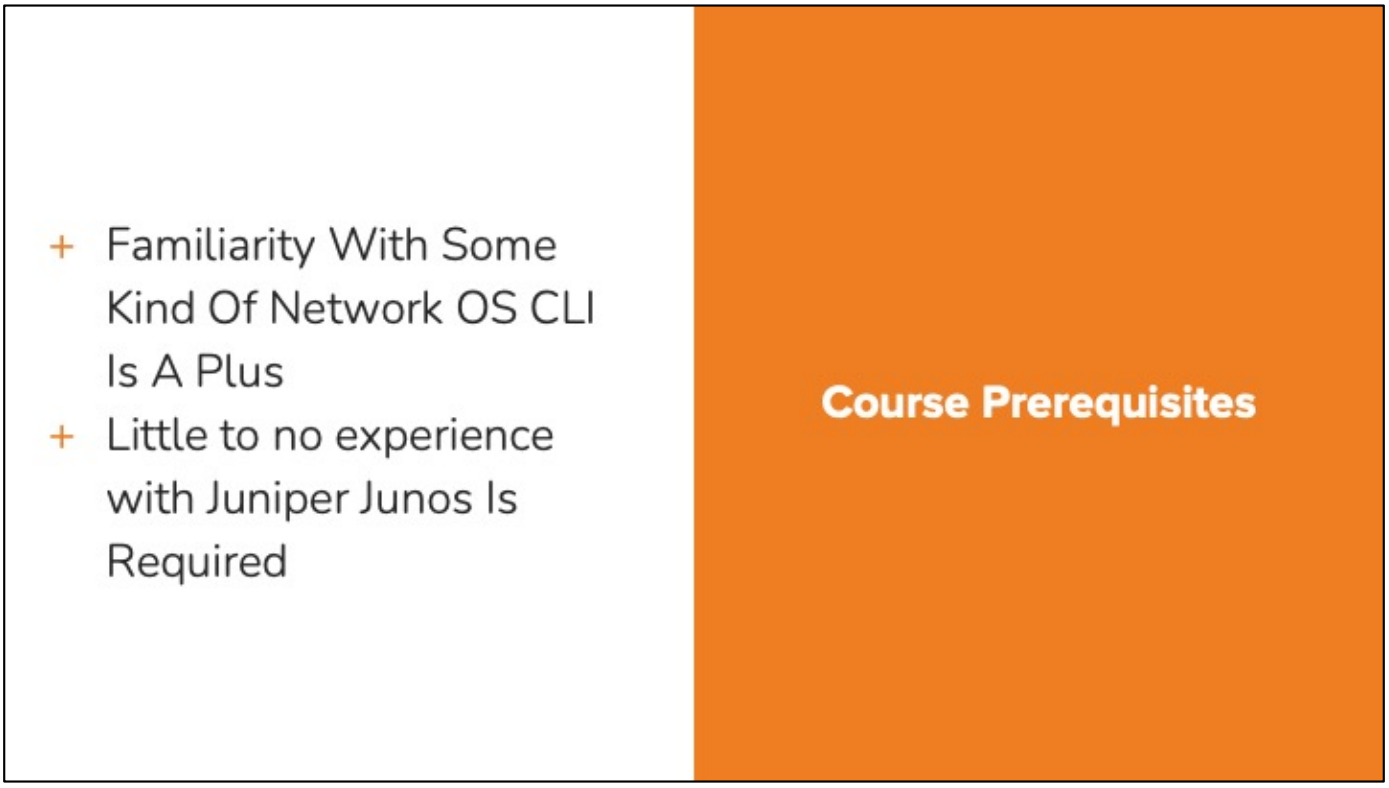
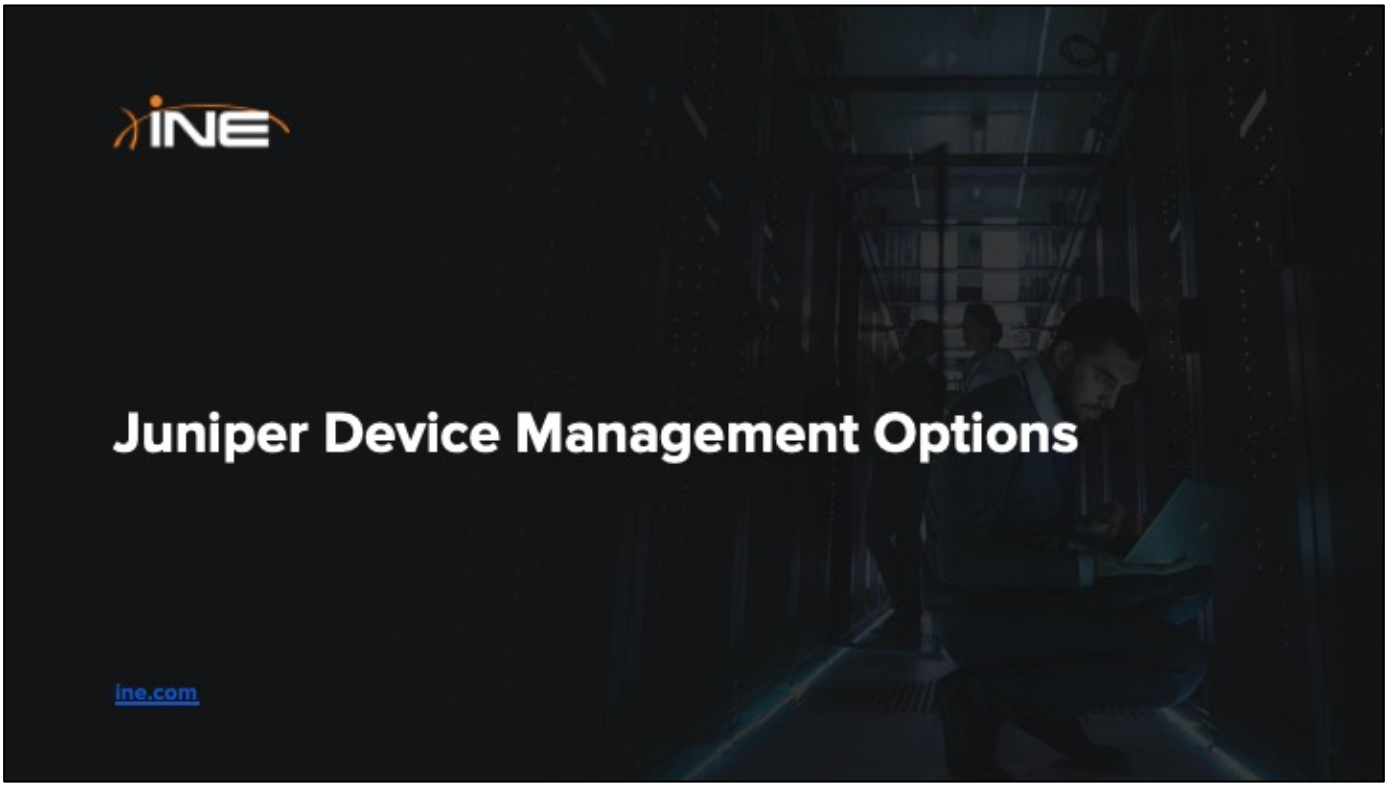




Course Objectives

- To Introduce You To The Junos OS Command Line Interface And How To Use It To Configure And Verify Features And Protocols On Juniper Devices



Topic Overview

- The Console Port
- The Management Port
- SSH & Telnet Access

Juniper Device Management

- There are several ways to manage a Juniper device
- Some require locally-connected access & others are accessible remotely over an IP connection
- Some automatically place you into the FreeBSD or Linux shell...others within Operational Mode

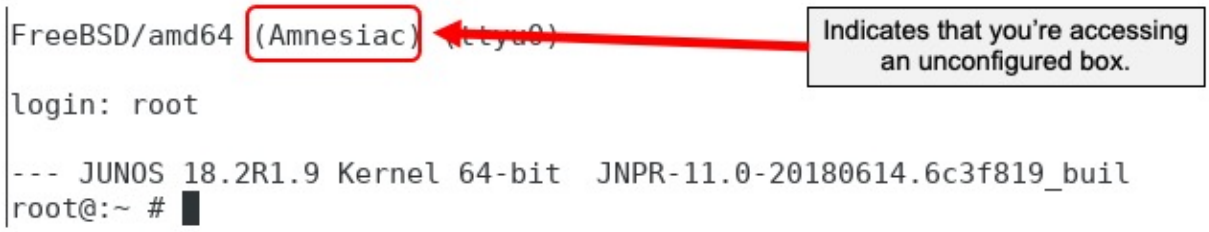
The Console Port

- Used to login when device contains no IP address
- Must be directly-connected to the box

RJ-45 requires a rollover cable (aka a “console cable”, provided with the box)

Console Connections

- Connection via the console port provides immediate access to the Unix shell



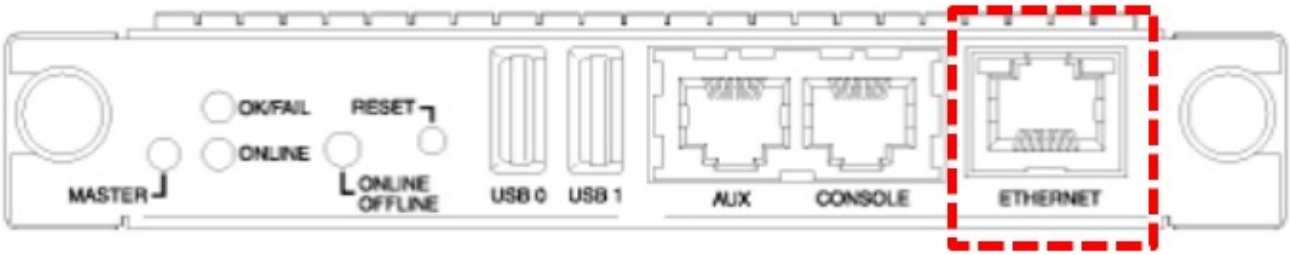
- Command-line invoked with "cli" command

```
root@% cli
root>
```

The Management Port

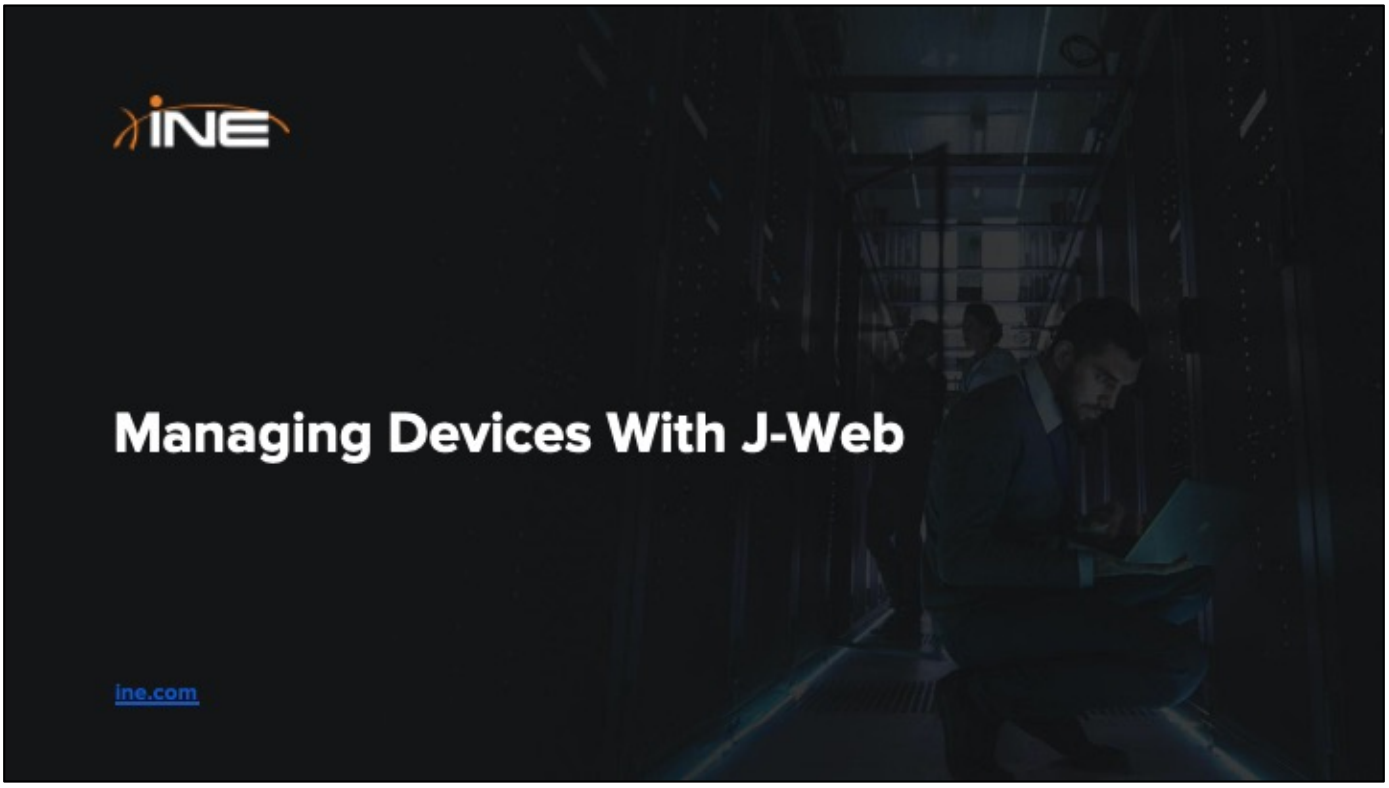
- Provides an RJ-45 interface for IP connectivity (via SSH or Telnet) to the CLI
- Considered an out-of-band (OOB) interface
- Does not provide routable access to other networks (doesn't process transit traffic)
- Typically referenced as:
  - FXP0
  - EM0
  - ME0

The Management Interface



SSH & Telnet Access

- Given an IPv4/IPv6 address, Juniper devices can be managed remotely via SSH or Telnet
  - Telnet
    - TCP based (port 23)
    - Insecure
  - SSH
    - TCP based (port 22)
    - Secure
- Provide access to Junos Operational Mode



Topic Overview

+ Introduction To J-Web

Introduction To J-Web

- Juniper device provide a GUI for management called “J-Web”
- Some Juniper platforms come pre-configured from the factory to support J-Web
- Other devices need minimal configuration from the CLI before J-Web is accessible

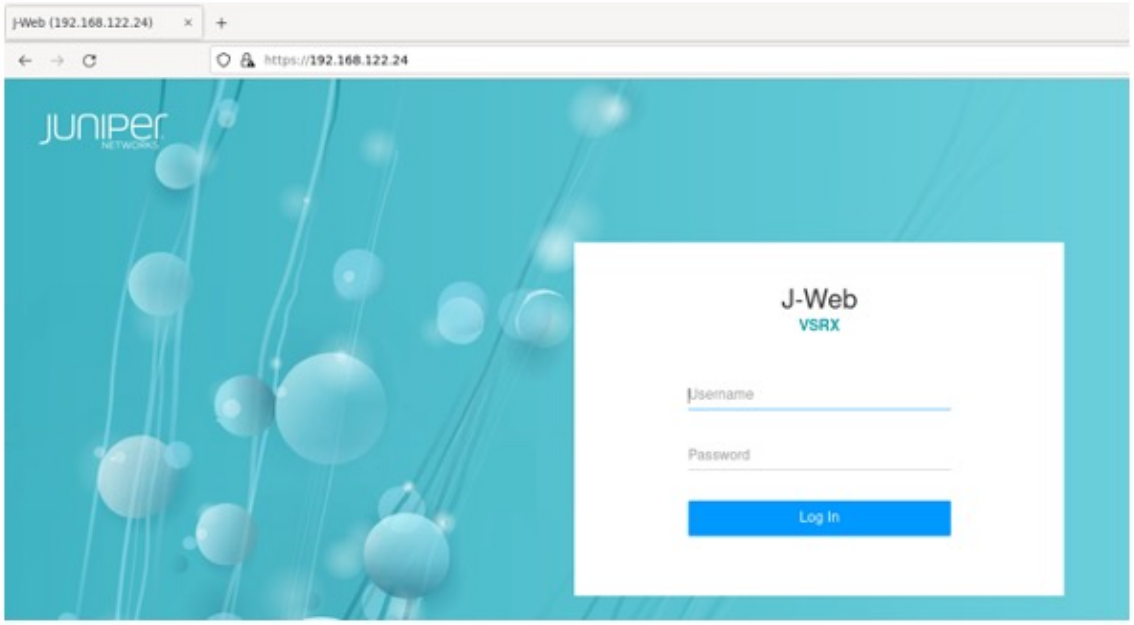
Accessing J-Web

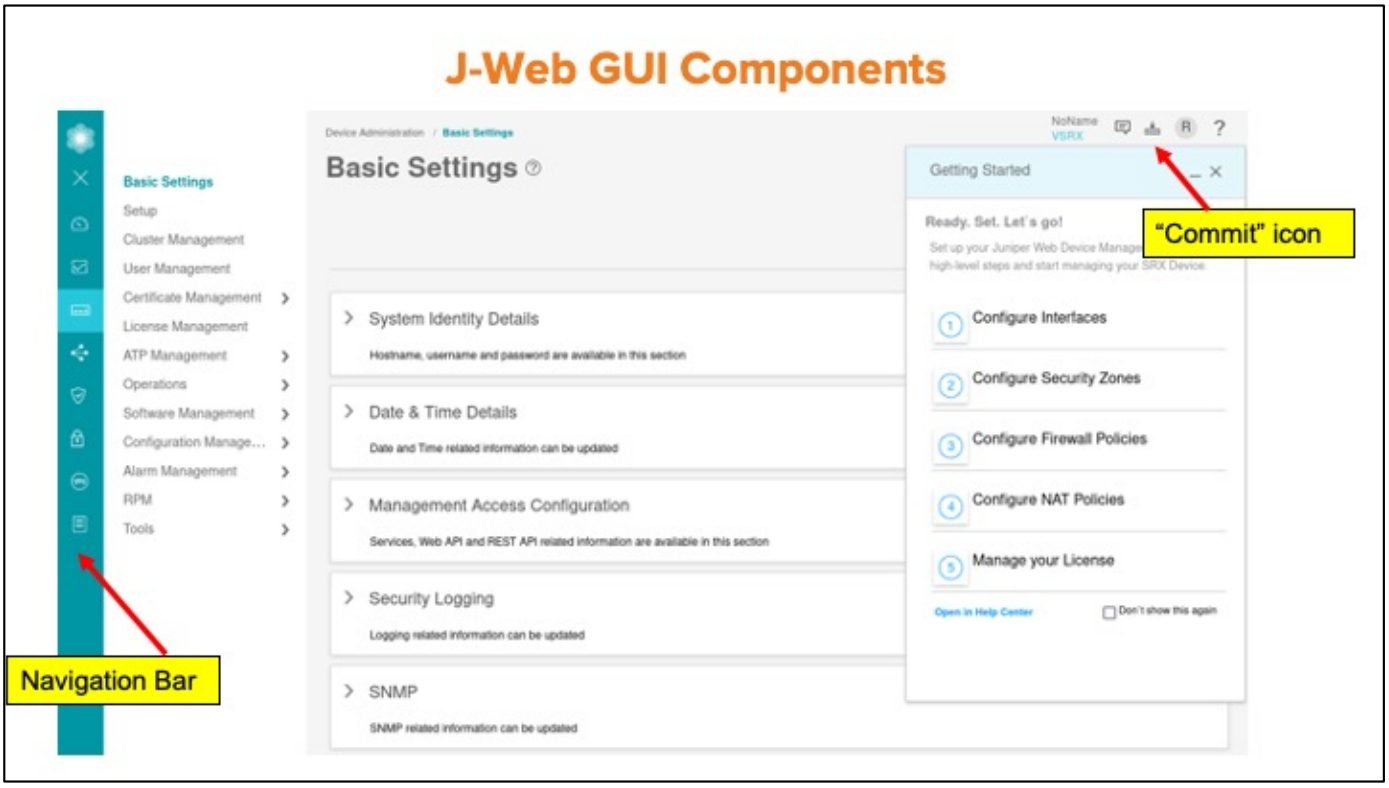
- The following needs to be configured for accessing J-Web:
  - A root authentication password
  - An IP address on a reachable interface
  - Enabling of web-management:

```
root# set system services web-management https interface fxp0.0
```

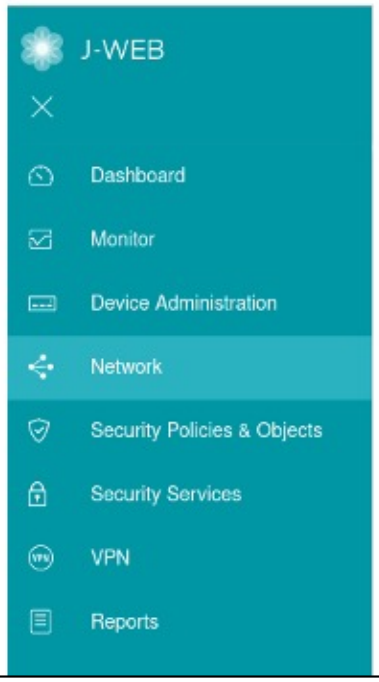
vMX platforms don’t support J-Web.

Initial J-Web Portal





J-Web Navigation Bar



Topic Overview

+ Loopback Interface
+ Types Of Virtual Interfaces
+ Types Of Physical Interface
+ Interface Numbering & Naming Conventions

Juniper Network Interfaces

- Juniper devices support a wide variety of network interfaces
  - Physical
  - Virtual
- Physical network interfaces typically reside in one or two places:
  - Permanent interfaces on the chassis
  - Interfaces that come with removable cards or modules

Juniper Loopback Interfaces

- Juniper devices support a Loopback interface
  - Logical interface
  - Created via configuration
  - Only one Loopback interface ("Lo0") with a single unit (unit 0) supported
  - Lo0.16384 automatically created upon configuration of Lo0 with an IP address of 127.0.0.1
- Juniper documentation states that only a /32 IPv4 address should be configured on Lo0 interface

```
[edit]
root@vMX-4# set interfaces lo0 unit 0 family inet address 4.5.6.7/32
```

This interface DOES support configuration of IP addresses with masks OTHER than /32 if desired.

Unlike Cisco devices, which support the creation of as many Loopback interfaces as you wish…Juniper devices only let you create one.

Other Types Of Virtual Interfaces

- Integrated Routing & Bridging interface (irb)
  - Logical interface on switches and firewalls for routing traffic between VLANs
- Routed VLAN interface (rvi)
  - Special interface for routing packets into and out of private VLANs
- Virtual Chassis Port interface (vcp)
  - Used in virtual chassis environments whose function is to send and receive Virtual Chassis Control Protocol (VCCP) traffic to create, monitor, and maintain the Virtual Chassis
- Multichassis Aggregated Ethernet interface (mc-ae)
  - Logical interface used to represent multiple physical links logically aggregated together when using a MC-LAG topology.

Multichassis Link Aggregation (MC-LAG) is designed so that a host device (i.e. server) with multiple Ethernet NICS can form a LAG with two, different switches. Provides more redundancy than when a LAG is simply point-to-point between two devices.

Types Of Network Interfaces

- Juniper interfaces come as WAN, MAN, LAN and other types
- Interfaces are represented via different acronyms in the output of “show” commands such as:
  - Ae = Aggregated Ethernet Interface
  - Coc1 = Channelized OC1 interface
  - E1= Ethernet interface
  - XE = 10Gigabit Ethernet interface

https://www.juniper.net/documentation/us/en/software/junos/interfaces-fundamentals/topics/topic-map/router-interfaces-overview.html

Module Acronyms

- One should be familiar with the following acronyms which represent card/modules that house network interfaces
  - PIC = Physical Interface Card
    - Provide physical interfaces
    - Plug into an FPC
    - Hot removable and hot insertable
  - FPC = Flexible PIC Concentrator
  - DPC = Dense Port Concentrator
  - MPC = Modular Port Concentrator
  - MIC = Modular Interface Card

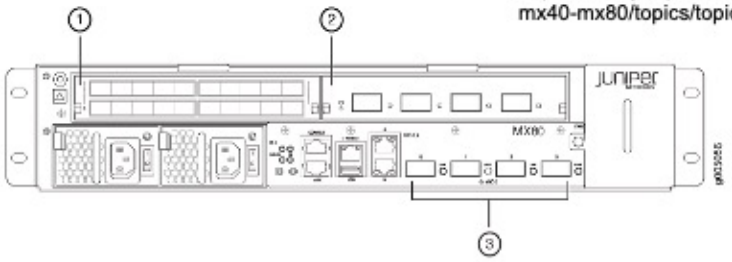
Interface Numbering & Naming

- Interface naming/numbering conventions can vary depending upon Juniper platform
  - Make sure you read the documentation!

Module Naming Example

Figure 2: MX5, MX10, MX40, and MX80 Interface Port Mapping

https://www.juniper.net/documentation/us/en/hardware/mx5-mx10-mx40-mx80/topics/topic-map/mx5-10-40-80-interface-modules.html



1 — MIC slot 1/0 (FPC 1, PIC 0 and PIC 1)

2 — MIC slot 1/1 (FPC 1, PIC 2 and PIC 3)

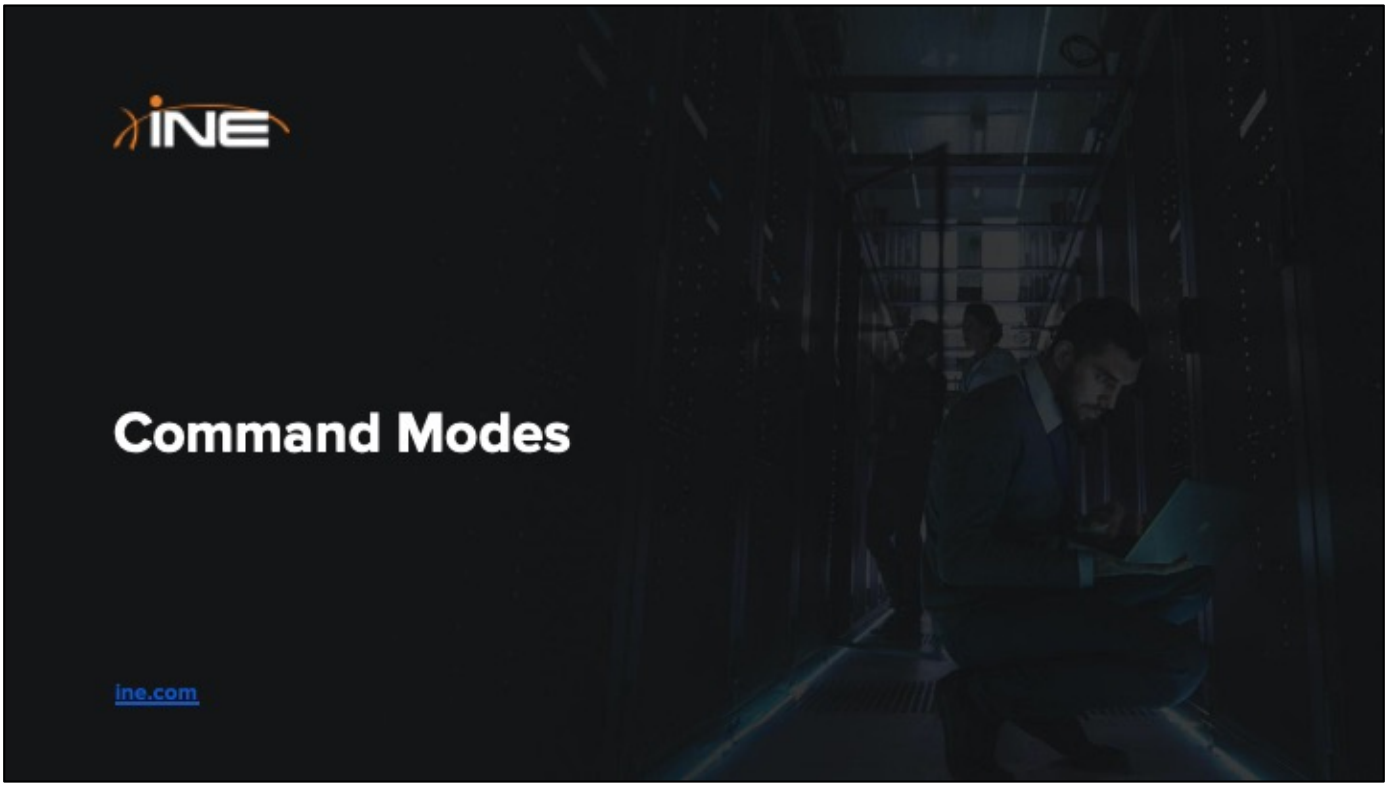
3 — 10-Gigabit Ethernet ports (FPC 0, PIC 0)

The chassis has two built-in MPCs, which are represented in the CLI as `FPC 0` and `FPC 1`.

MPC 0 (`FPC 0`) contains a 4-port 10-Gigabit Ethernet MIC. Both the MPC and the MIC are considered fixed and are built into the front of the chassis. The MIC is represented as `MIC 0` in the CLI and is logically divided into a single PIC, which is represented as `PIC 0`.

MPC 1 (`FPC 1`) has two slots, which accept up to two MICs. The MICs are represented as `MIC 0` and `MIC 1` in the CLI and are logically divided into PICs depending on their type. A MIC installed in MIC slot **1/0** is represented in the CLI as `PIC 0` and `PIC 1`. A MIC installed in MIC slot **1/1** is represented as `PIC 2` and `PIC 3`.

MIC = Modular Interface Card
PIC = Physical Interface Concentrator
FPC = Flexible PIC Concentrator



Topic Overview

- Introducing Junos Shell
- Stopping Daemons
- Junos Command Modes
- Accessing The Command Modes
- Reverting To A Factory Default State

The Shell

- When logging into the Junos OS as root (via the Console port), you are placed into the FreeBSD/Linux shell
- Allows access to normal FreeBSD/Linux shell commands
  - ls, cd, ping, traceroute, netstat, etc.
  - You can also stop and restart protocol daemons from this level if necessary

Stopping Daemons

- One can access the FreeBSD/Linux shell from Operational Mode:

  > start shell user root

  Password:

- Identify the problematic daemon (provide daemon name if already known)

```
root@SRX1:~ # ps -aux | grep alarmd
USER    PID  %CPU %MEM    VSZ   RSS TT  STAT STARTED      TIME COMMAND
root   6644  0.0  1.2 739672 11068  -  S    03:47     0:08.78 /usr/sbin/alarmd -N
```

Stopping Daemons

- Kill daemon via process-id value:

```
root@SRX1:~ # kill -9 6644
```

- Confirm that daemon has restarted with new process-id

```
root@srx1% ps -aux | grep alarmd
root     1161  0.0  0.2  6820  3500  ??  S     4:49AM   0:04.69 /usr/sbin/alarmd -N
```

Introduction To Junos Command Modes

- Junos OS runs on top of a FreeBSD/Linux shell and has two main CLI modes
  - Operational mode
    - root@Router>

  - Configuration mode
    - root@Router#

Introducing Operational Mode

- Used to monitor and troubleshoot device operation
  - Denoted as **user@host>**
- Invoked with **cli** command from shell

```
root@SRX-1% cli
root@SRX-1>
```

Using Operational Mode

- Common operational mode commands...
  - show
  - clear
  - ping
  - Traceroute
  - request
- Junos operational mode is mainly read-only commands

Junos Configuration Mode

- Invoked with **configure** (or "edit") commands from operational mode

```
root@SRX-1> configure
Entering configuration mode
[edit]
root@SRX-1#
```

- Used to make changes to device configuration
  - Denoted as **user@host#**
- Contains hierarchy of all configurations
  - System, interface, protocols, etc.

Common Configuration Mode Commands

- Common Configuration Mode commands...
  - run
  - edit
  - up
  - top
  - set
  - delete
  - copy
  - rename
  - commit
  - rollback
- Supports both read and write commands

Reverting Junos To A Default Config

1. Start the CLI
   + **root@vMX-4% cli**
2. Go to configuration mode
   + **root@vMX-4> configure**
3. Load the default configuration
   + **root@vMX-4# load factory-default**
4. Set a new root password
   + **root@vMX-4# set system root-authentication plain-text-password**

Reverting Junos To A Default Config

5. Save the new config
   + **root@SRX-1# commit**
6. Exit to operational mode
   + **root@SRX-1# exit configuration-mode**
7. Reload
   + **root@SRX-1> request system reboot**

Take note that the command, “load factory-default” can be done from a Telnet or SSH session. HOWEVER, it you “commit” this command it will delete all IP-related config and you will be locked out of your session. Better to re-configure your interface IP information (and any authentication information) prior to issuing “commit”.

Reverting Junos To A Default Config (Alternative)

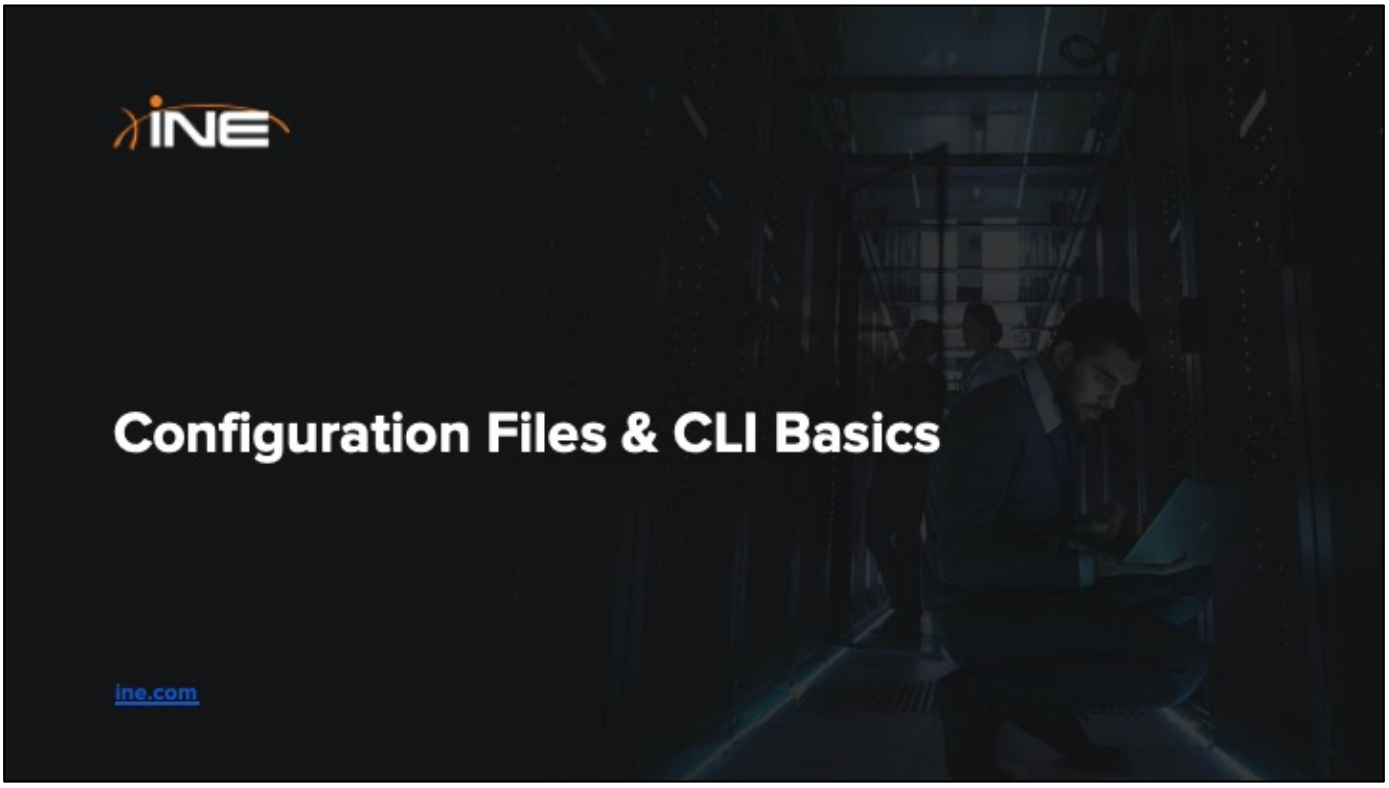
- Go to Operational Mode
  - **root@SRX-1> request system zeroize**
- Note that this command can be done from Operational Mode and automatically reboots the device

```
user@host> request system zeroize
warning: System will be rebooted and may not boot without configuration
Erase all data, including configuration and log files? [yes,no] (no) yes


0 1 1 0 0 0 done

syncing disks... All buffers synced.
Uptime: 5d19h20m26s
recorded reboot as normal shutdown
Rebooting...
```

This command can only be done from a Console connection.

-

If you want to reset your system to a truly factory-default state (including the lack of a root authentication password) then this is the command to use.



Topic Overview

+ Differentiating The Active From The Candidate Config Files
+ Junos Parser Hierarchy
+ Introducing The "Set" Command
+ Hierarchy Navigational Shortcuts

Junos Active vs. Candidate Configuration

- Junos has two configuration files:
  - Active configuration
    - Currently running configuration
  - Candidate configuration
    - Uncommitted changes to be merged with the active config

Junos Active vs. Candidate Config

- In Junos all changes must be explicitly committed
  - `root@SRX-1# commit`
- Junos commits have error checking and recovery built in
  - Syntax errors result in a rejected commit
  - Successful commits have sub-versions automatically saved to the commit database
  - **`rollback`** can be used to revert to previous active configurations

Candidate Configuration File

- Uncommitted changes to the Candidate Configuration persist until:
  - They are committed or…
  - They are deleted
  - Use the Configuration Mode command, "*rollback 0*" to delete all uncommitted changes

```
root@Test-Vmx# set system host-name My-VMX

[edit]
root@Test-Vmx# exit
The configuration has been changed but not committed
Exit with uncommitted changes? [yes,no] (yes)

Exiting configuration mode

root@Test-Vmx> edit
Entering configuration mode
The configuration has been changed but not committed

[edit]
root@Test-Vmx# show | compare
[edit system]
-  host-name Test-Vmx;
+  host-name My-VMX;
```

Junos Parser Hierarchy

- Junos configurations have a defined hierarchy
- Example: Junos IPv4 address
  - **configure**
    - **interface**
      - **ge-0/0/0**
        - **unit 0**
          - **family inet**
            - **address 1.2.3.4/24**

Junos "Set" Command

- **"set"** is used to make a configuration change
  - Available options for "**set**" are relative to your place in the CLI hierarchy
  - **"edit"** can be used to move places in the hierarchy
- Example **set** from root:
  - `root@SRX-1# set system host-name vMX-1`
- Example **set** from lower hierarchy:

```
root@SRX-1# set interfaces ge-0/0/0 unit 0 family inet
address 1.1.1.1/24
```

Navigating Junos Hierarchy

- Junos always tells you where you are in the tree

```
root@SRX-1#
[edit interfaces ge-0/0/1 unit 0 family ethernet-switching vlan]
```

- **edit** command used to move down the hierarchy

```
root@SRX-1# edit protocols ospf
[edit protocols ospf]
root@SRX-1#
```

- **up** command used to move up the hierarchy

```
[edit interfaces ge-0/0/1 unit 0 family ethernet-switching
vlan]
root@SRX-1# up 2
[edit interfaces ge-0/0/1 unit 0]
```

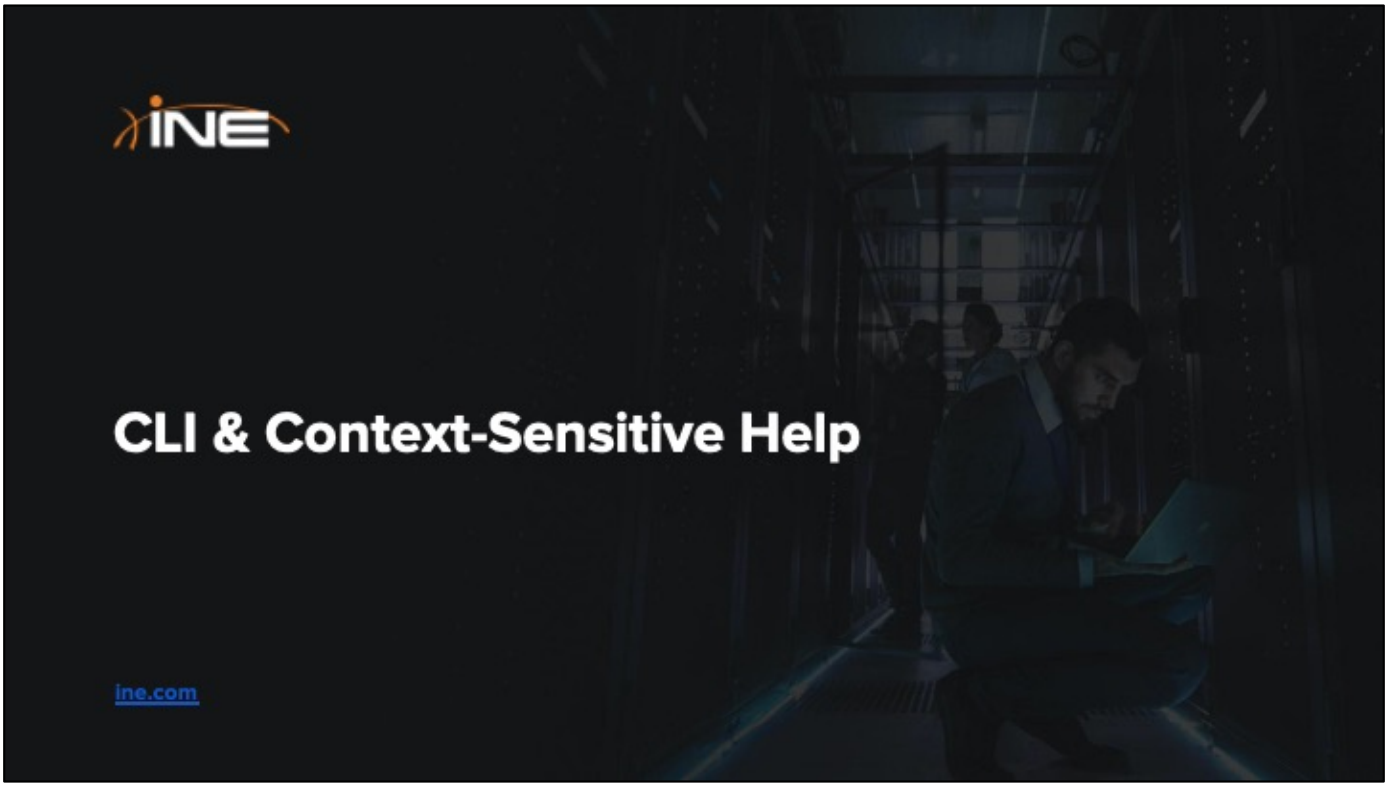
Junos Top Command

- **top** command moves to root of hierarchy

```
[edit interfaces ge-0/0/1 unit 0]
root@SRX-1# top
[edit]
```

- **top** can also be used to run root commands from lower hierarchy

```
[edit interfaces ge-0/0/0 unit 0]
root@SRX-1# top show protocols ospf
```



Topic Overview

+ CLI Help Methods

Discovering Command Options

- Available commands and keywords can be discovered with the question mark "?"
  - Typing "?" directly after one-or-more characters will display all available commands matching that pattern
  - Typing "?" after a space will display all available command options in that position.

Junos Command Completion

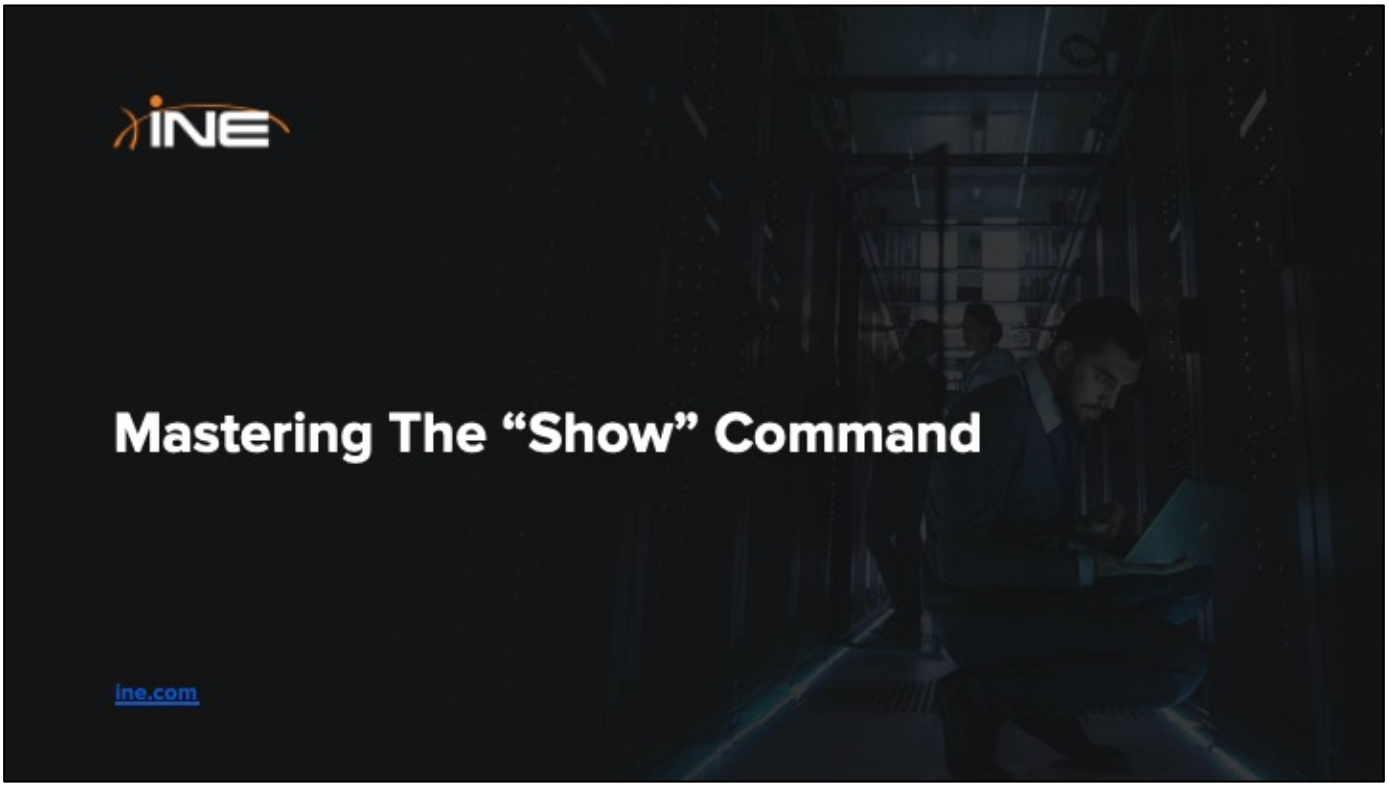
- Command completion
  - Invoked with either space or tab
  - Tab also completes user defined variables

Junos Built-In Help

- Junos has built-in help functions
  - Explicit with **`help`** command
    - **`root@SRX-1> help apropos route`**
      - Performs a simple string search for any command (or command description) containing a matching string
    - **`root@SRX-1# help tip cli`**
      - Displays random interesting tips about CLI usage

Junos Built-In Help

- **`root@SRX-1# help reference ospf area`**
  - Displays summary information about the statement (like a manual) based on the summary descriptions that appear in the Junos OS configuration guides.
- **`root@SRX-1# help topic interfaces family`**
  - Displays usage guidelines for the statement based on information that appears in the Junos OS configuration guides.



Topic Overview

+ Command Mode Differences
+ Controlling Output Detail
+ Filtering Command Output
+ Using The “Refresh” Keyword
+ Displaying Device Configuration

Junos Show Command

- **"Show"** is different in operational mode vs. configuration mode
- **"Show"** in operational mode
  - Used to check state of the system
  - E.g. check the status of ge-0/0/0

```
root@SRX-1> show interfaces ge-0/0/0
Physical interface: ge-0/0/0, Enabled, Physical link is Up
  Interface index: 134, SNMP ifIndex: 508
  Link-level type: Ethernet, MTU: 1514, Link-mode: Full-duplex, Speed: 1000mbps,
  BPDU Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
<snip>
```

- **"Show"** in Configuration Mode is primarily used to show elements of the Candidate Configuration file

Controlling Output Detail

- Junos **show** output has various levels of detail
- Levels from least to most detail are...
  - Terse
  - Brief
  - Detail
  - Extensive
- Example:
  - `root@EX-1> show interfaces terse`

Junos "Show" Command

- **show** in configuration mode
  - Used to check the configuration of the system
  - **show** is relative to your place in the CLI hierarchy
  - **edit** can be used to move places in the hierarchy
- Example **show** from root:
  - `root@SRX-1# show interfaces ge-0/0/0`
- Example **show** from lower hierarchy:

```
root@SRX-1# edit interfaces ge-0/0/0
[edit interfaces ge-0/0/0]
root@SRX-1# show
```

Using The “Run” keyword

- **“run”** allows operational mode commands to be issued from configuration mode
- Example:

```
[edit]
root@vMX-1# show interfaces ge-0/0/0
unit 0 {
    family inet {
        address 1.1.1.1/24;
```

```
[edit]
root@vMX-1# run show interfaces ge-0/0/0
Physical interface: ge-0/0/0, Enabled, Physical link is Up
  Interface index: 148, SNMP ifIndex: 526
  Link-level type: Ethernet, MTU: 1514, MRU: 1522, LAN-PHY mode,
  Speed: 1000mbps, BPDU Error: None, Loop Detect PDU Error: None,
```

Filtering Command Output

- Junos supports using the pipe "|" for filtering output
- Possible keywords include:
  - Match (displays any line in command output including the supplied keyword

```
root@vMX-1> show interfaces ge-0/0/0 | match address
  Current address: 0c:88:73:2d:00:02, Hardware address: 0c:88:73:2d:00:02
      Addresses, Flags: Is-Default Is-Preferred Is-Primary
```

Filtering Command Output

- **Except** (used to exclude entire lines of output that contain supplied keywords)

```
root@vMX-1> show chassis fpc
                     Temp  CPU Utilization (%)   CPU Utilization (%)  Memory
Utilization (%)
Slot State            (C)  Total  Interrupt      1min   5min   15min  DRAM (MB)
Heap     Buffer
  0  Online           Testing  31         0       26     25     24    511
 31          0
  1  Empty
  2  Empty
```

```
root@vMX-1> show chassis fpc | except Online
                     Temp  CPU Utilization (%)   CPU Utilization (%)  Memory
Utilization (%)
Slot State            (C)  Total  Interrupt      1min   5min   15min  DRAM (MB)
Heap     Buffer
  1  Empty
  2  Empty
```

Filtering Command Output

- **Find** (display command output beginning with the first line containing the supplied keyword)

```
root@vMX-1> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
ge-0/0/0                up    up
ge-0/0/0.0              up    up   inet     1.1.1.1/24
                                   multiservice
lc-0/0/0                up    up
lc-0/0/0.32769          up    up   vpls
pfe-0/0/0               up    up
pfe-0/0/0.16383         up    up   inet
                                   inet6
```

```
root@vMX-1> show interfaces terse | find lc-0/0/0
lc-0/0/0                up    up
lc-0/0/0.32769          up    up   vpls
pfe-0/0/0               up    up
pfe-0/0/0.16383         up    up   inet
                                   inet6
```

This is like the “begin” keyword in Cisco IOS.

Introducing The "Refresh" Option

- Some "show" commands are more useful if issued multiple times over a time interval to see any changes in statistics or counters.
- Rather than typing the same command repeatedly one can use the "refresh" option

Using The “Refresh” Option

```
root@vMX4-Router> show interfaces ge-0/0/1 extensive | match "Input packet count" | refresh
---(refreshed at 2022-02-17 19:51:58 UTC)---
    Input packet count                    537
---(refreshed at 2022-02-17 19:52:03 UTC)---
    Input packet count                    537
---(refreshed at 2022-02-17 19:52:08 UTC)---
    Input packet count                    537
---(refreshed at 2022-02-17 19:52:13 UTC)---
    Input packet count                    537
```

- By default, command will be “refreshed” every 5-seconds indefinitely.
- Can be stopped by typing “q” or “Ctrl + C”
- Only some “show” commands contain this option
- Interface statistics can be cleared with “*clear interface <interface> statistics*”

Displaying The Configuration

- Normally, the output of "*show configuration*" is displayed in a hierarchical format
  - This is useful to help visualize how commands must be entered
- The configuration output can be changed to display the "set" commands that were used to create it:
  - root@vMX-1> show configuration | display set

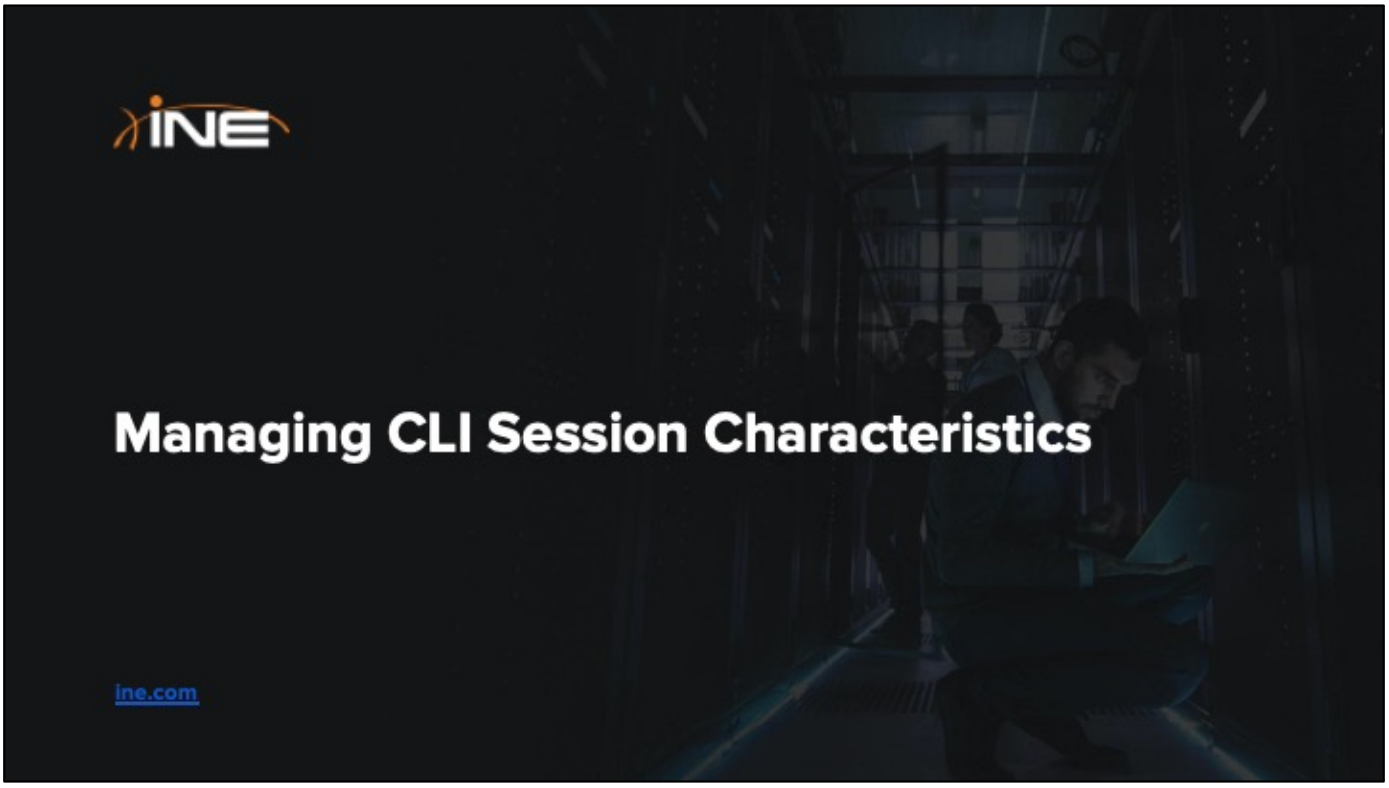
Displaying Additional Details

- **display detail**
  - Shows additional comments about syntax usage

```
root@SRX-1> show configuration | display detail
```

- Junos also supports multiple pipes

```
root@SRX-1> show interfaces terse | except down | match
inet
```

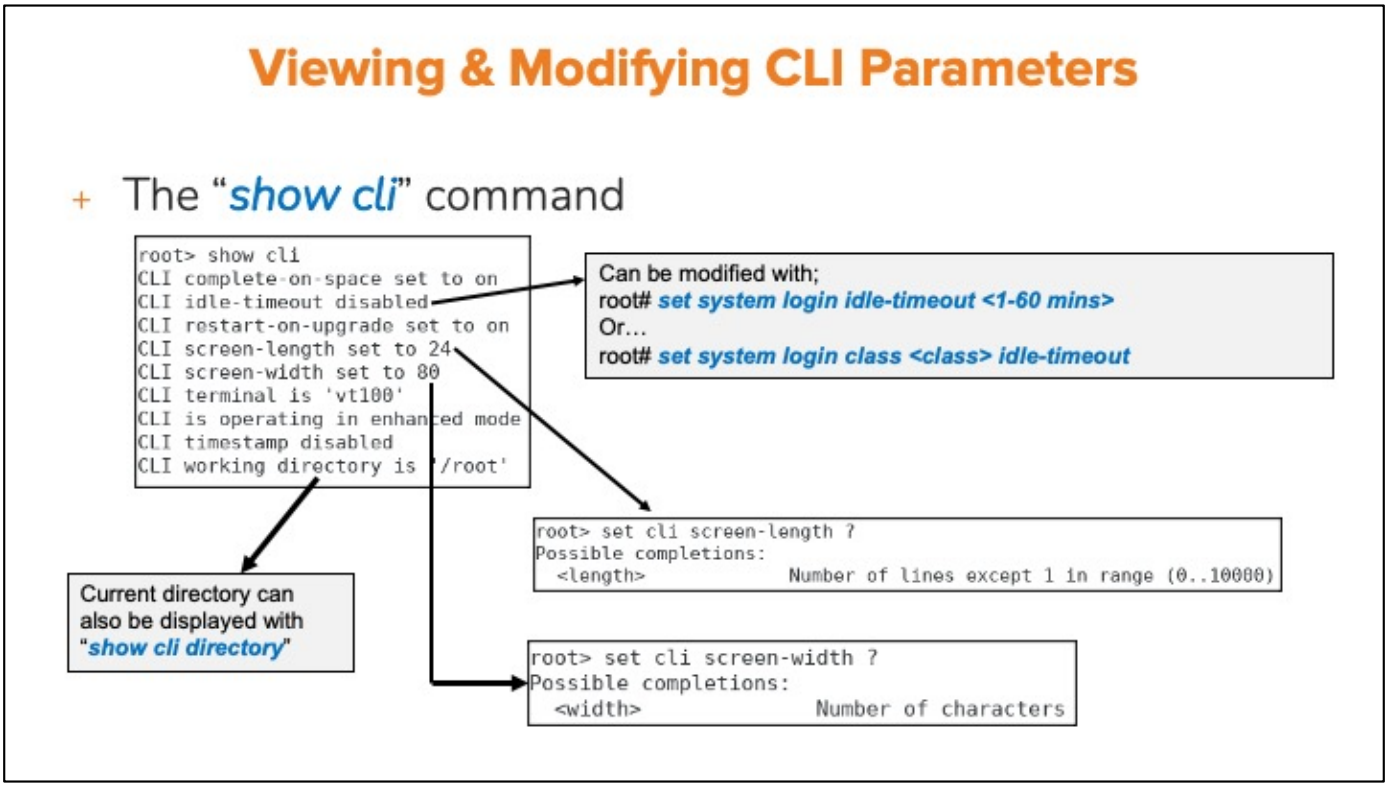


Topic Overview

+ Options For Managing CLI Sessions
+ Viewing The CLI History

Managing The CLI

- Commands exist to help you view elements related to the CLI such as:
  - What permissions do you currently have?
  - What default directory are your files being saved to?
  - What commands have you recently entered?
- Various Junos “show cli” commands exist that answer these questions (and more)
- These commands are performed within Operational Mode



Screen-length dictates how many lines of output are displayed before Junos pauses and displays the "more" output.

Notice that adjusting the screen width or length is done at Operational Mode, which means it only affects your current CLI session.

Viewing Command History

- There are several ways to view your command history
  - Up or Down arrows on the keyboard
  - "*show cli history*"
- Notes about "*show cli history*"
  - Shows the last 24-commands entered at the current level
  - "*show cli history <integer>*" controls the quantity of historical commands displayed
  - "*run show cli history*" can be used to display CLI history from within Configuration Mode

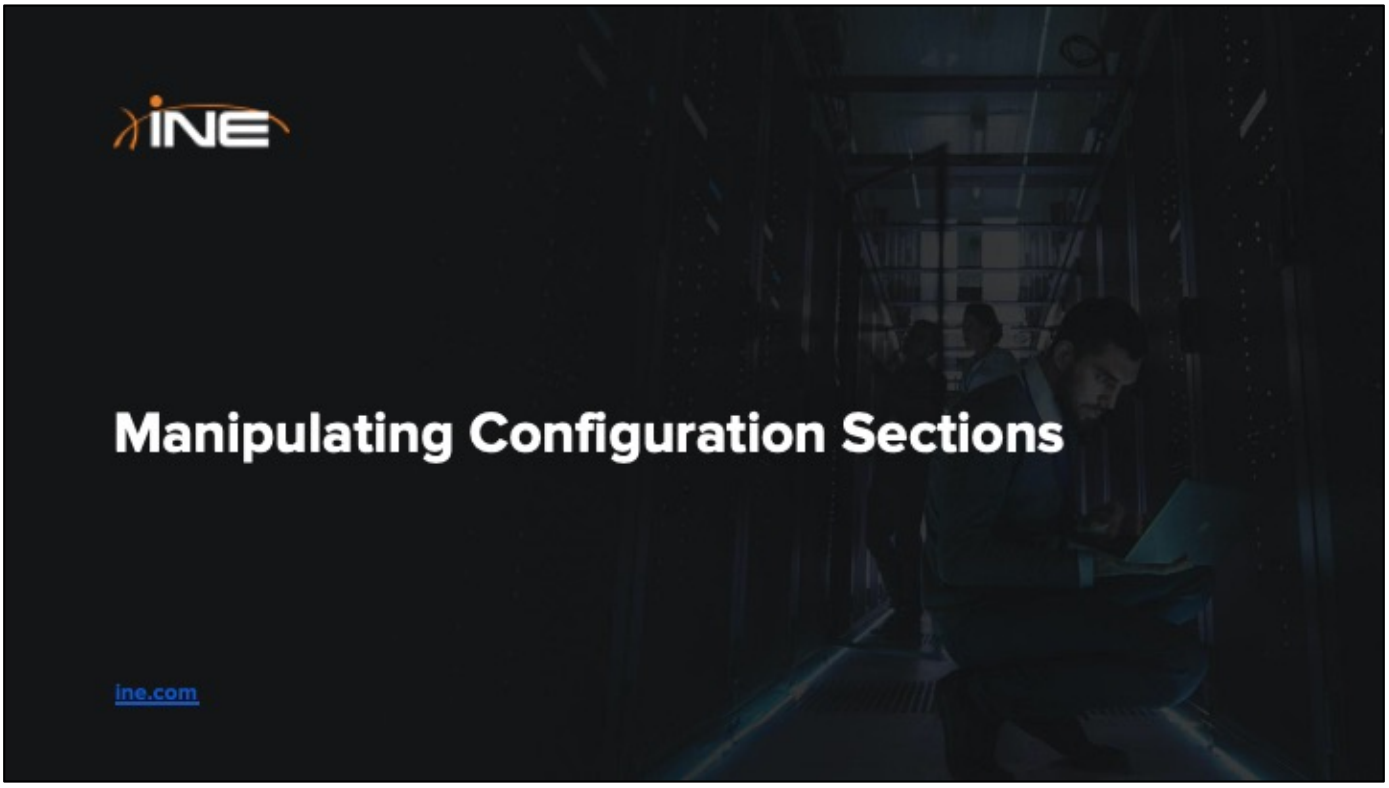
Other Useful CLI Commands

- "*set cli ?*" shows all possible ways to modify CLI for the current session
- Change your working directory to save files to a non-default location:

```
root@vMX> set cli directory ?
Possible completions:
  <directory>          Pathname of working directory
root@vMX> set cli directory
```

- Add a current timestamp after any command is entered:

```
root@vMX> set cli timestamp
Feb 16 15:26:59
CLI timestamp set to: %b %d %T

root@vMX> show interfaces terse
Feb 16 15:27:09
Interface               Admin Link Proto
ge-0/0/0                up    down
```



Topic Overview

- Introducing The “Delete” Command
- Delete Large Configuration Sections
- Junos Deactivate, Rename, Copy, Insert & Replace Commands
- Protecting Configuration Sections

Using The Junos "Delete" Command

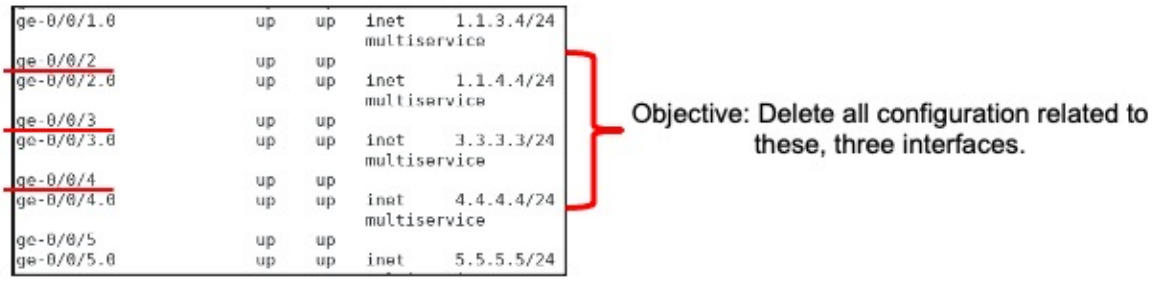
- **"delete"** is used to remove configuration changes
  - Equivalent of IOS **no** command
  - Like other commands, path is relative to place in CLI hierarchy
- Example "**delete**" from root:
  - `root@SRX-1# delete system host-name SRX-1`
- Example "**delete**" from lower hierarchy:

```
root@SRX-1# edit system
[edit system]
root@SRX-1# delete host-name
```

Deleting Large Configuration Sections

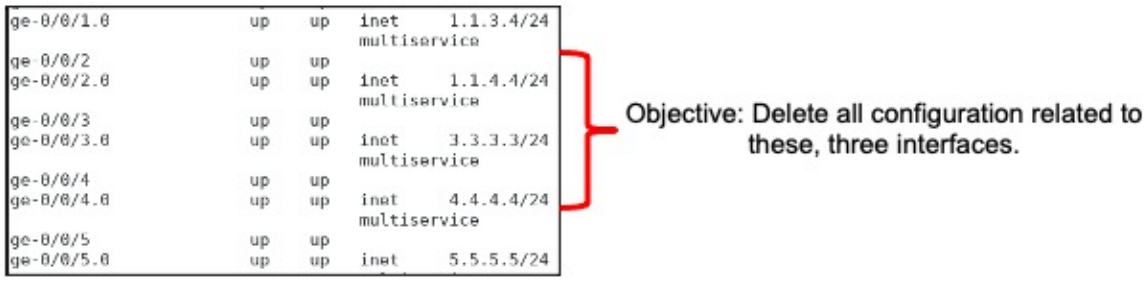
- Wildcard options can be added to the “delete” command
- Enables the deletion of large quantities of the Candidate Configuration
- Enables the addition of regular expressions with the “delete” command

Deleting Large Configuration Sections



- Using the command, "delete" by itself would require using the command three times:
  - delete interfaces ge-0/0/2
  - delete interfaces ge-0/0/3
  - delete interfaces ge-0/0/4

Deleting Large Configuration Sections




```
root@vMX-4# wildcard delete interfaces ge-0/0/[2-4]
  matched: ge-0/0/2
  matched: ge-0/0/3
  matched: ge-0/0/4
Delete 3 objects? [yes,no] (no)
```

- Prefacing the “delete” command with the keyword, “wildcard” allows you to use regular expressions with this command.

Warning About Using "Delete"

- *Note: delete is recursive down the hierarchy!*

```
[edit]
root@SRX-1# delete
This will delete the entire configuration
Delete everything under this level?
[yes,no] (no)
```

Junos "Deactivate" Command

- **"deactivate"** used to temporarily disable configuration sections without deleting them
- Example:
  - **root@SRX-1# deactivate protocols ospf**
- Reversed with the **activate** command
  - **root@SRX-1# activate protocols ospf**

Junos "Rename" Command

- **"rename"** used move configuration snippet from one section to another
  - E.g. move config from one interface to another interface
  - E.g. change an IP address
- Examples:

```
root@SRX-1# rename interfaces ge-0/0/1 to ge-0/1/0
root@SRX-1# rename interfaces ge-0/0/0 unit 0 family inet
address 1.2.3.4/24 to address 5.6.7.8/24
```

- Still requires a "commit" action

Junos "Copy" Command

- **"Copy"** used to replicate configuration snippet from one section to another
  - E.g. copy config from one interface to another interface
- Example:
  - **root@SRX-1# copy interfaces ge-0/0/0 to ge-0/0/1**

Junos “Replace Pattern” Command

- Sometimes a unique pattern is found multiple places within a configuration

```
[edit protocols]
+   mpls {
+       interface ge-0/0/2.0;
+   }
[edit protocols ospf area 0.0.0.0]
      interface ge-0/0/1.0 { ... }
+     interface ge-0/0/2.0;
[edit protocols]
+   ldp {
+       interface ge-0/0/2.0;
+   }
+   lldp {
+       interface ge-0/0/2;
```

- You might need to replace that pattern with a different pattern… “replace pattern” allows for this.

Junos “Replace Pattern” Command

- Scenario: Ge-0/0/2’s NIC is suspected of being faulty. You need to:
  - Swap its cable over to Ge-0/0/3
  - Replicate its config to the new interface
  - Find all instances where Ge-0/0/2 is referenced in the config and replace these with Ge-0/0/3

- `root@Router1# replace pattern ge-0/0/2 with ge-0/0/3`

Junos “Replace Pattern” Command

```
root@Router1# replace pattern ge-0/0/2 with ge-0/0/3
```

Before…

```
+   ge-0/0/2 {
+       unit 0 {
+           family inet {
+               address 10.10.14.1/24;
+           }
+           family inet6 {
+               address 2001:db8:0:14::1/64;
+           }
+       }
+   }
[edit protocols]
+   mpls {
+       interface ge-0/0/2.0;
+   }
[edit protocols ospf area 0.0.0.0]
      interface ge-0/0/1.0 { ... }
+     interface ge-0/0/2.0;
[edit protocols]
+   ldp {
+       interface ge-0/0/2.0;
+   }
+   lldp {
+       interface ge-0/0/2;
+   }
```

After…

```
root@Router1# show | compare
[edit interfaces]
+   ge-0/0/3 {
+       unit 0 {
+           family inet {
+               address 10.10.14.1/24;
+           }
+           family inet6 {
+               address 2001:db8:0:14::1/64;
+           }
+       }
+   }
[edit protocols]
+   mpls {
+       interface ge-0/0/3.0;
+   }
[edit protocols ospf area 0.0.0.0]
      interface ge-0/0/1.0 { ... }
+     interface ge-0/0/3.0;
[edit protocols]
+   ldp {
+       interface ge-0/0/3.0;
+   }
+   lldp {
+       interface ge-0/0/3;
+   }
```

Graphics courtesy of Day One: Beginner's Guide to Learning Junos

Junos "Insert" Command

- Some protocols and features within Junos process commands in a particular sequence
  - i.e. Firewall Filters, Routing Policies
- By default, adding a command places that new command as the last command within its hierarchy
- You may, instead, wish to place this command before (or after) existing commands

Junos “Insert” Command

- Junos “insert” command allows one to insert a new command at a particular location within a hierarchy

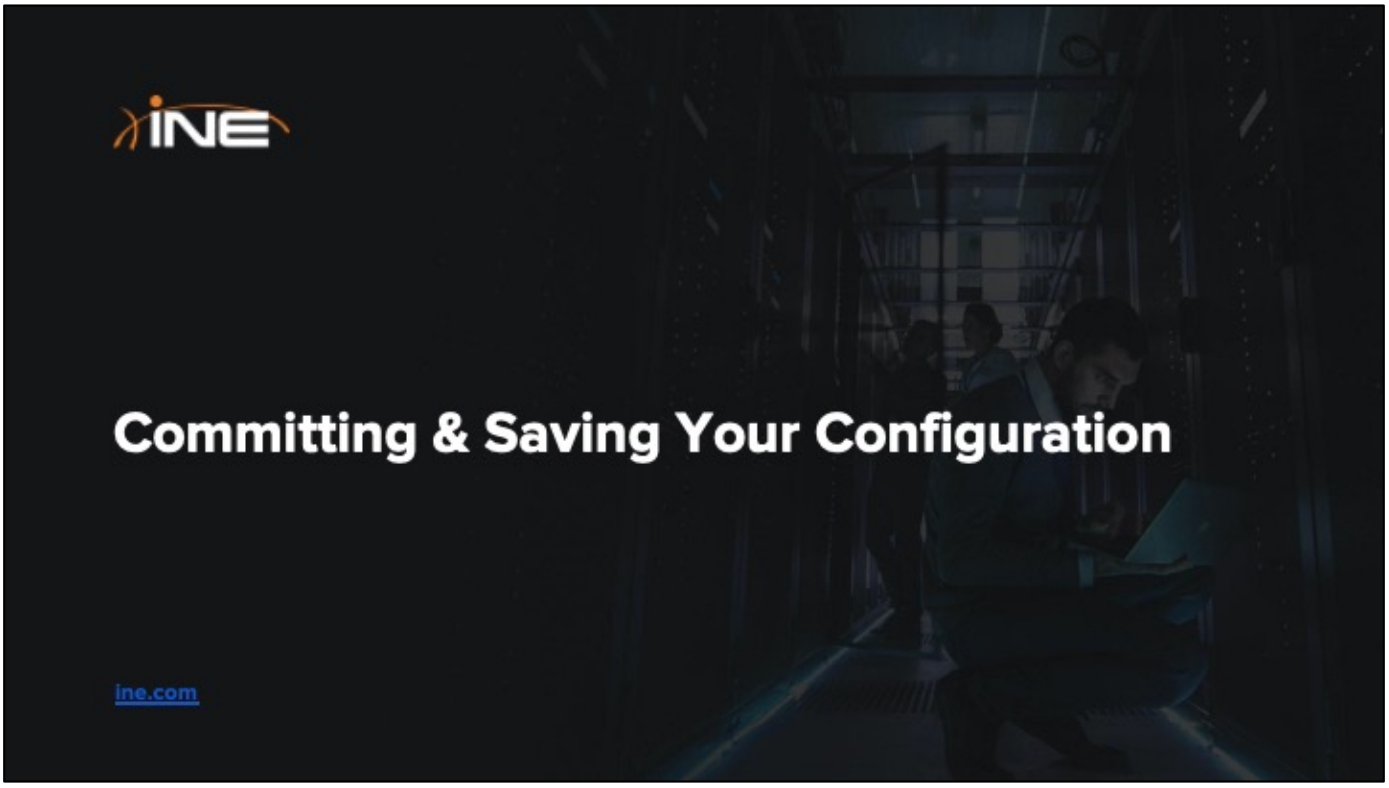
root@vMX-4# ***insert*** <new command> [***before*** | ***after***] <existing command>

Protecting Configuration Sections

- The Junos "*protect*" command can be used to prevent anyone from changing sections of your configuration

```
[edit interfaces]
root@vMX-4# protect ge-0/0/5

[edit interfaces]
root@vMX-4# set ge-0/0/5 unit 0 family inet address 8.8.8.8/24
warning: [interfaces ge-0/0/5] is protected, 'interfaces ge-0/0/5 unit 0 family
inet address 8.8.8.8/24' cannot be created
```

- This command is negated using "*unprotect*"



Topic Overview

+ Committing Configuration Changes
+ Adding Comments To Commits
+ Committing With Redundancy
+ Naming Your Configurations
+ Viewing & Loading Saved Configurations

Committing Configuration Changes

- **`commit and-quit`**
  - Commits changes and exits back to operational mode
- **`commit check`**
  - Checks uncommitted changes against the syntax parser for errors and returns success or failure. Does **not** actually commit anything.
- **`commit at`**
  - Schedules a commit for a later time and date
  - Future time (or date and time) must be enclosed with quotes ("23:01:00")
  - Cleared by "*run clear system commit*"
- **`commit confirmed`**
  - Commits changes, but automatically rolls them back if another **`commit`** is not issued within 10-minutes (configurable)
  - Useful for remotely committing changes that might lock you out

Show system uptime (to display current clock when using "commit at" command).

Adding Comments To Commits

- **Show system commit**
  - Displays a history of system commits
- Output of "show system commit" is more meaningful if comments/descriptions have been supplied

```
root@vMX-4# commit comment "Keith Bogart added addresses to Ge-0/0/5"
commit complete
```

- "***Commit comment***" allows one to add a meaningful description and then commit the Candidate Configuration

```
root@vMX-4> show system commit
0   2022-02-07 18:42:12 UTC by root via cli
    Keith Bogart added addresses to Ge-0/0/5
1   2022-02-07 18:36:08 UTC by root via cli
```

Committing With Redundancy

- Often, Juniper devices contain multiple Routing Engines for redundancy
- The "commit" command does not affect a backup configuration on the redundant RE
- Two ways to ensure backup RE contains the same, committed config as the primary RE:

```
root@vMX-4# commit synchronize
```

Must be done manually after every commit.

Or...

```
root@vMX-4# set system commit ?
Possible completions:
synchronize          Synchronize commit on both Routing Engines by default
```

Saving Named Configuration Files

- Both the Candidate and Active configuration files can be saved with descriptive names using the "**save**" command
  - Saving a named copy of the Candidate Configuration:

```
[edit]
root@vMX-4# save Keith-Candidate-config.txt
Wrote 45 lines of configuration to 'Keith-Candidate-config.txt'
```

  - Saving a named copy of the Active Configuration

```
root@vMX-4> show configuration | save Keith-Working-Config.txt
Wrote 45 lines of output to 'Keith-Working-Config.txt'
```

  - View all saved files

```
root@vMX-4> file list

/root/:
.cshrc@ -> /packages/mnt/os-runtime/root/.cshrc
.login@ -> /packages/mnt/os-runtime/root/.login
.profile@ -> /packages/mnt/os-runtime/root/.profile
Keith-Candidate-config.txt
Keith-Working-Config.txt
```

The ".txt" extension is optional and not required to save, or load, configs.

Saving the Candidate Configuration may be dangerous because it also saves any uncommitted changes.

-

Both of these commands save the configs in a hierarchical format.

Viewing Saved Configuration Files

- The command, "***file show*** <***filename***>" can be used to display the contents of a saved configuration file.

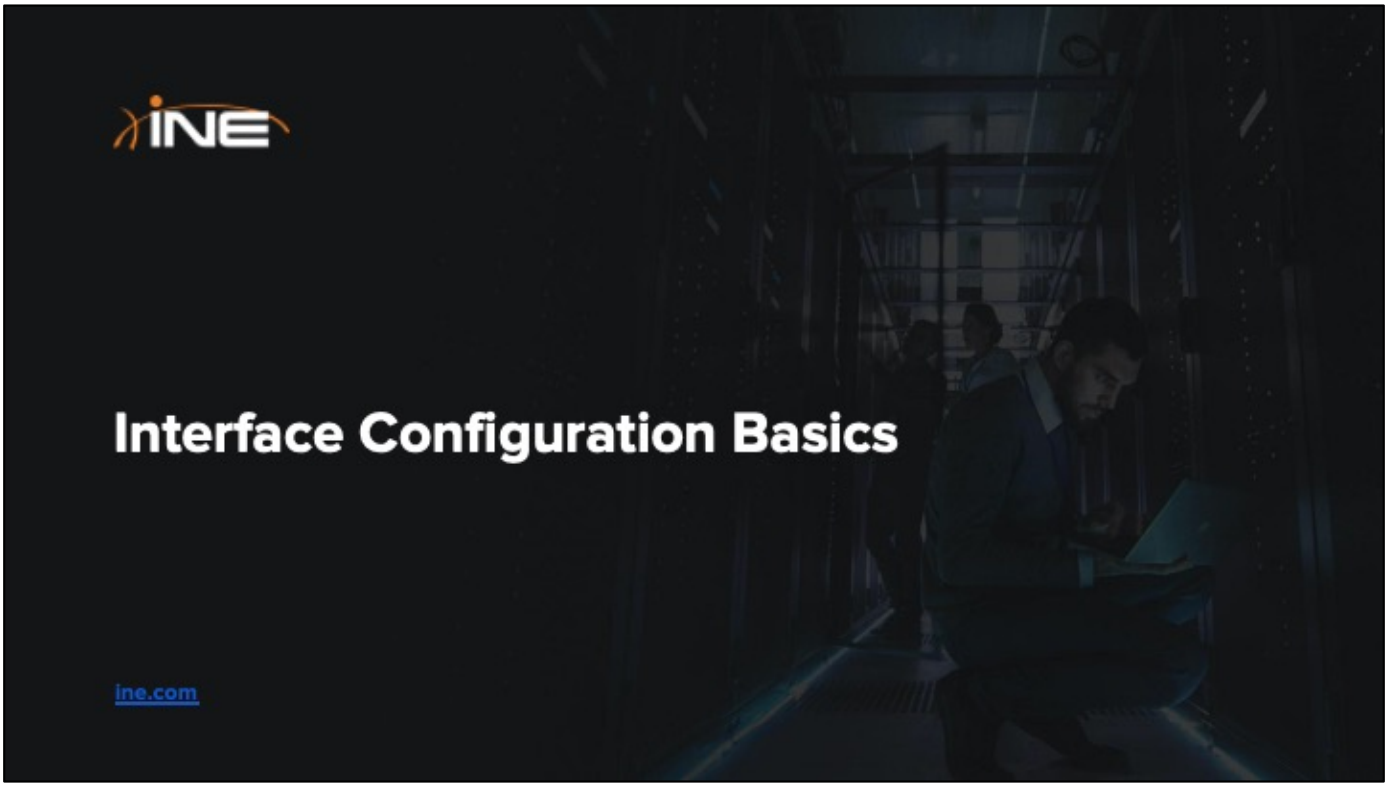
```
root@vMX-4> file show Keith-Working-Config.txt
## Last commit: 2022-01-27 17:05:53 UTC by root
version 18.2R1.9;
system {
    root-authentication {
        encrypted-password "$6$D4ckKWmV$Twd.9ED1
tH/wxphV2C/Ek2hlGKvgl0"; ## SECRET-DATA
    }
    host-name vMX-4;
    syslog {
        user * {
            any emergency;
            pfe none;
        }
        file messages {
            any notice;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
```

Loading Configuration Files

- Saved configurations can be loaded onto your Juniper device
- Done within configuration mode and replaces the current Candidate Configuration.

```
root@vMX-4# load override ?
Possible completions:
  <filename>           Filename (URL, local, remote, or floppy)
  Keith-Candidate-config.txt  Size: 904, Last changed: Feb 07 20:03:20
  Keith-Working-Config.txt  Size: 911, Last changed: Feb 07 20:02:42
  Test-Candidate       Size: 909, Last changed: Feb 07 20:12:31
  json                 Load configuration in JSON format
  terminal             Use login terminal
```

- Exactly the same behavior as ”configure replace” in Cisco IOS devices.
- JSON – Configs can be saved as JSON and subsequent JSON files can be loaded using this option (show configuration | display JSON)
- Terminal – allows you to copy and paste a configuration file saved locally on your laptop in hierarchical format. End with “Ctrl-D” (for “done”)



Topic Overview

- Introduction To Interface Units
- Interface Families
- Relation of Interface Units To VLANs
- Disabling Interfaces

Intro To Interface Configuration

- Interfaces contain both physical and logical configuration elements
  - Physical configuration elements include speed, duplex, and other Layer-1 and Layer-2 characteristics
  - Logical configuration elements include Layer-3 addressing, QoS, VLAN assignment, and other Layer-3 through Layer-7 characteristics.
- All logical configuration elements are configured within interface "Units".
  - Typically, unless an interface will be performing VLAN-Tagging it will only have a single, logical unit assigned, "Unit 0".

Interface Units

- Sub-interfaces in Junos are called “units”
  - Unit 0 is the equivalent of the main interface
- All interfaces require at least unit 0
  - Separates physical interface parameters from logical ones
  - E.g. speed & duplex vs. IP address & VLAN encapsulation
- Some interfaces may have more than one unit
  - Depends on encapsulation
  - E.g. Ethernet VLANs, Frame Relay DLCIs, ATM VPI/VCIs, etc.

Junos Interface Families

- Junos enforces more configuration hierarchy through explicit Address Families at the interface/unit level
- **family** command controls which protocols an interface runs:
  - family ethernet-switching
  - family inet
  - family inet6
  - family mpls
  - family iso
- **family** is defined under the unit
  - `root@SRX-1# edit interface ge-0/0/0 unit 0 family inet`

Junos Router-On-A-Stick

- Junos unit number does not directly relate to VLAN number
  - Unit value is locally significant
  - For clarity of config, numbers typically configured to match VLAN identifiers
- Junos can use 802.1Q units (subinterfaces) to route between VLANs on a routed interface
- Junos router-on-a-stick

```
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 10 vlan-id 10
set interfaces ge-0/0/0 unit 10 family inet address 1.2.3.4/24
```

Junos Interface Units & Ethernet VLANs

- Hierarchical Configuration Example:

```
root@SRX-1# show interfaces
ge-0/0/0 {
    vlan-tagging;
    unit 12 {
        vlan-id 12;
        family inet {
            address 10.1.12.2/24;
        }
        family inet6 {
            address 2001:10:1:12::2/64;
        }
    }
```

Disabling Interfaces

- Juniper interfaces are automatically up if a cable is connected, and electrical connectivity is detected.
- One may wish to administratively disable an interface, or a logical interface unit.
- Disabling a physical interface automatically places all associated logical interfaces into an “Down / Up” state.

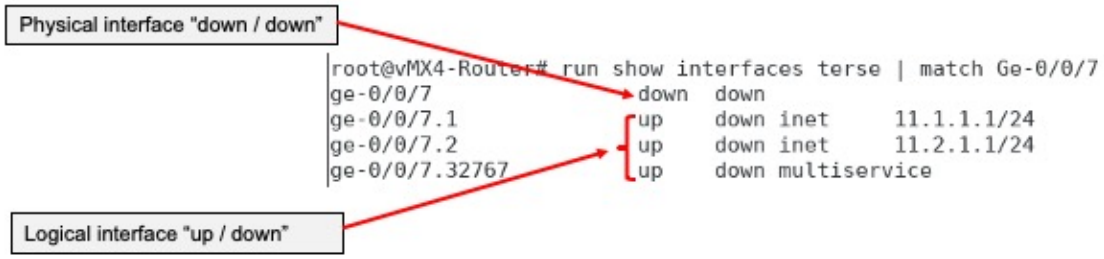
Disabling Interfaces

- There are two methods to disable a physical interface

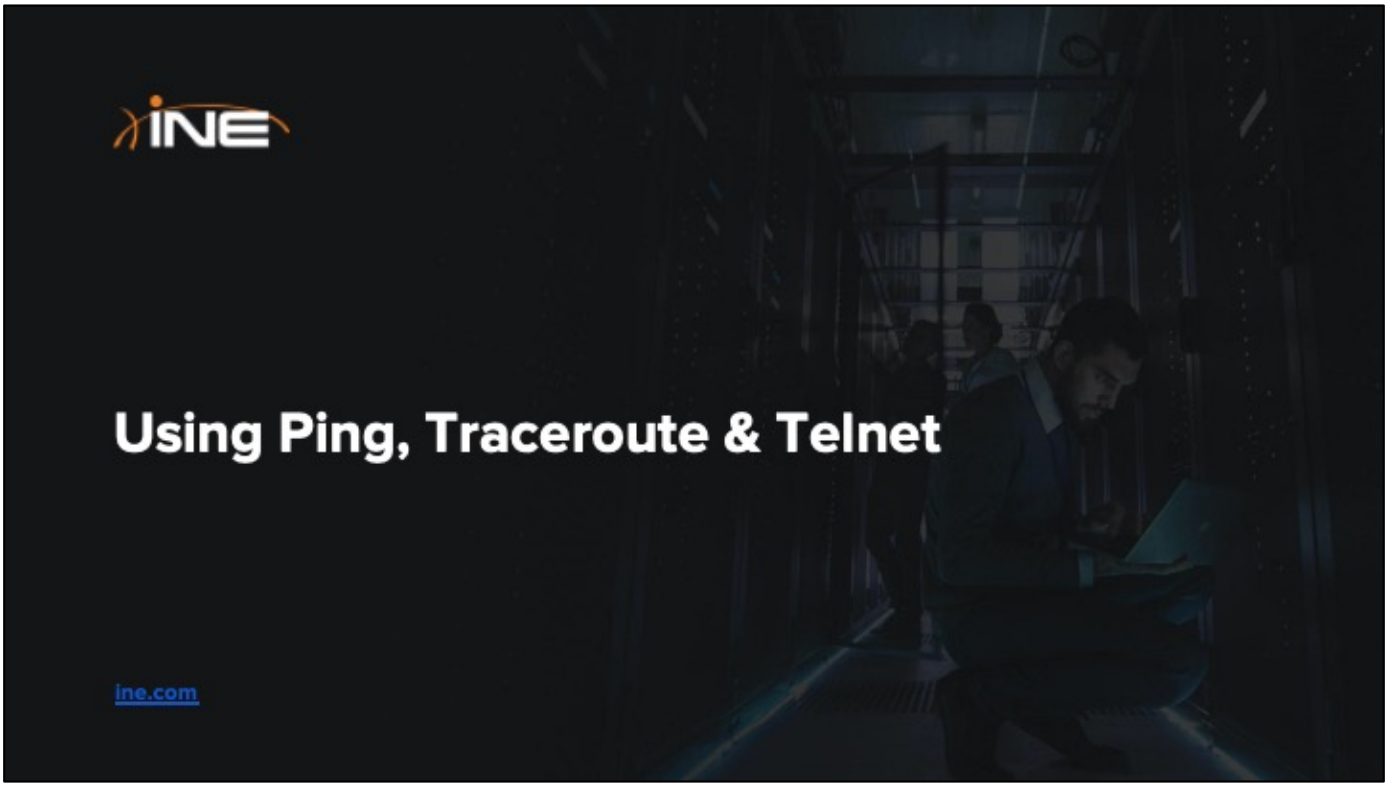
```
[edit]
root@vMX4-Router# set interfaces ge-0/0/7 disable
```

...or

```
root@vMX4-Router> start shell user root
root@vMX4-Router:~ # ifconfig ge-0/0/7 down
```



To bring the interface back up: ”delete interface <interface> disable”
(Yes…in this case a “double negative” is actually a good thing)



Topic Overview

+ Using Ping & Traceroute In Junos
+ Enabling Incoming Telnet Sessions On Your Juniper Device

Ping

- Generates ICMP “Echo” and “Echo Reply” packets
- Several options available with the command:
  - Simple destination ping (continuously creates packets until stopped with Ctrl-C sequence):

```
root@vMX4> ping 4.4.4.1
PING 4.4.4.1 (4.4.4.1): 56 data bytes
64 bytes from 4.4.4.1: icmp_seq=0 ttl=255 time=63.459 ms
64 bytes from 4.4.4.1: icmp_seq=1 ttl=255 time=8.837 ms
```

  - Reduced-count ping with “rapid” keyword

```
root@vMX4> ping 4.4.4.1 rapid
PING 4.4.4.1 (4.4.4.1): 56 data bytes
!!!!!
--- 4.4.4.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 61.013/226.230/353.091/104.995 ms
```

Other Ping Options

Commonly used options

```
root@vMX4> ping ?
Possible completions:
  <host>                Hostname or IP address of remote host
  atm                   Ping remote Asynchronous Transfer Mode node
  bypass-routing        Bypass routing table, use specified interface
  ce-ip                 Ping CE IP address
  clns                  Ping TSO node
  count                 Number of ping requests to send (1..2000000000 packets)
  detail                Display incoming interface of received packet
  do-not-fragment       Don't fragment echo request packets (IPv4)
  ethernet              Ping to an ethernet host by unicast mac address
  inet                  Force ping to IPv4 destination
  inet6                 Force ping to IPv6 destination
  interface             Source interface (multicast, all-ones, unrouted packets)
  interval              Delay between ping requests (seconds)
  logical-system        Name of logical system
+ loose-source          Intermediate loose source route entry (IPv4)
  mac-address           MAC address of the nexthop in xx:xx:xx:xx:xx:xx format
  mpls                  Ping label-switched path
  no-resolve            Don't attempt to print addresses symbolically
  overlay               Ping overlay path
  pattern               Hexadecimal fill pattern
  rapid                 Send requests rapidly (default count of 5)
  record-route          Record and report packet's path (IPv4)
  routing-instance      Routing instance for ping attempt
  size                  Size of request packets (0..65468 bytes)
  source                Source address of echo request
  strict                Use strict source route option (IPv4)
+ strict-source         Intermediate strict source route entry (IPv4)
  tos                   IP type-of-service value (0..255)
  ttl                   IP time-to-live value (IPv6 hop-limit value) (hops)
  verbose               Display detailed output
  vpls                  Ping VPLS MAC address
  wait                  Maximum wait time after sending final packet (seconds)
```

Traceroute

- Generates ICMP “Echo” and “Echo Reply” packets with increasing TTL values
- Several options available with the command:
  - Simple traceroute

```
root@vMX-1> traceroute 15.9.3.1
traceroute to 15.9.3.1 (15.9.3.1), 30 hops max, 52 byte packets
 1  1.1.3.4 (1.1.3.4)  8.079 ms  7.465 ms  10.389 ms
 2  4.4.4.1 (4.4.4.1)  9.909 ms  9.071 ms  9.662 ms
```

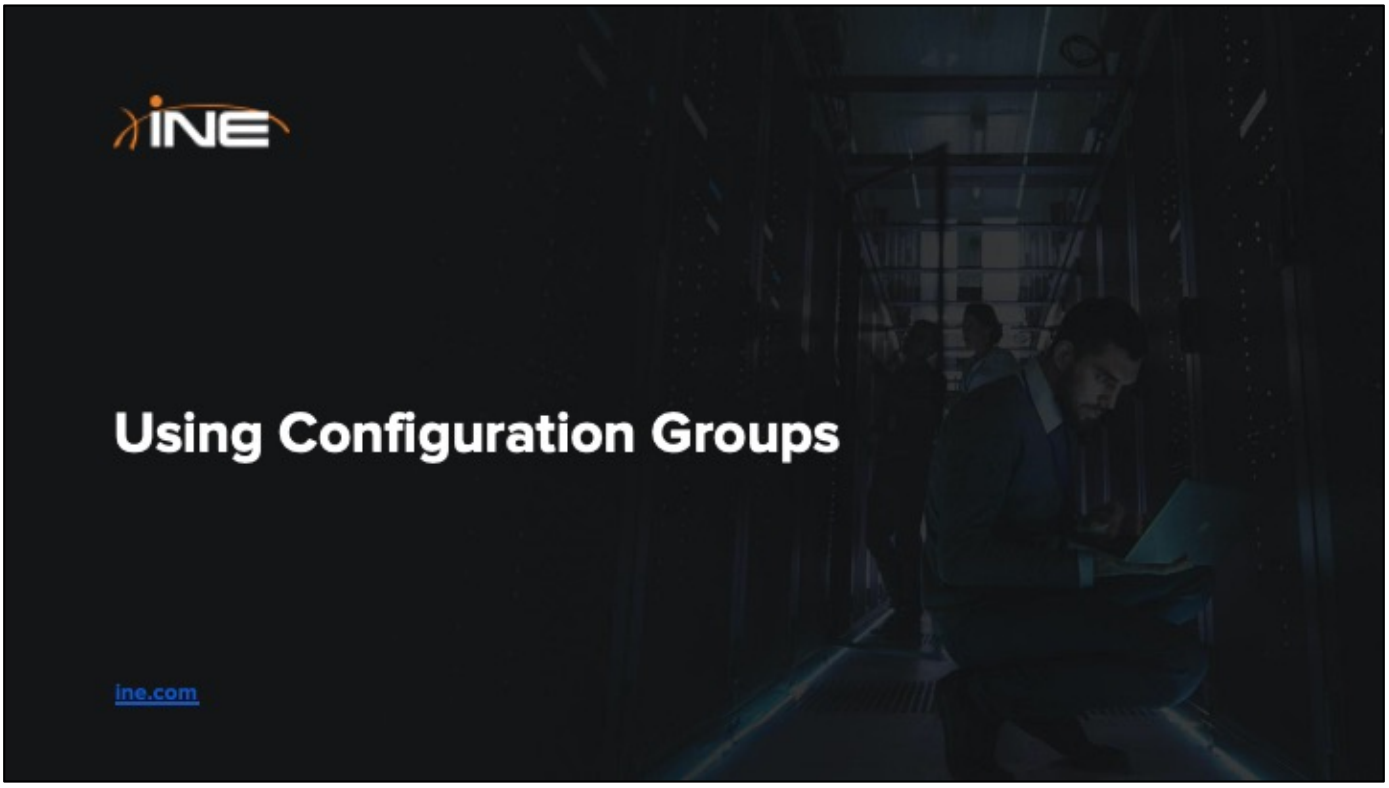
  - Monitored traceroute (continuous with real-time statistics)

```
root@vMX-1> traceroute monitor 15.9.3.1
                                My traceroute  [v0.69]
vMX-1 (8.0.0.8)(tos=0x0 psize=64 bitpattern=0x00)      Mon Feb 21 19:52:03 2022
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                          Packets               Pings
 Host                                  Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. 1.1.3.4                             0.0%     7   18.2 202.3   6.1 818.2 338.3
 2. 15.9.3.1                            0.0%     7    6.7  57.5   6.7 302.4 109.3
```

Telnet

- Telnet allows for a remote management connection across TCP/IP
- Defaults to TCP port 23
- To allow incoming Telnet on Juniper devices two things are required:
  - Creation of local user account(s) and authentication (or use AAA)
  - Allow the telnet service ("*set system services telnet*")
- To telnet from a Juniper device use the "telnet" command

```
root@vMX-1> telnet 15.9.3.1
root@vMX-1> telnet 15.9.3.1 port 334
```



Topic Overview

+ Configuration Groups Use Case
+ Creating Configuration Groups
+ Utilizing Wildcards
+ Applying Configuration Groups
+ Verifying Inheritance

Defining The Problem

- Sometimes there is a need to create a set of configuration commands that you wish to apply to more than one place such as:
  - Generic configuration you are likely to apply everywhere on all your devices (radius-server config, DNS, NTP, etc)
  - Specific configuration that differs from Juniper's defaults but is the common "base" configuration for that type of object in your network.
- Configuration Groups can serve this purpose

Creating Configuration Groups

- Create the configuration group and give it a distinctive name:
  - root@vMX-4# *set groups Common-Services*
- Within the group, create configuration statements

```
[edit groups Common-Services]
root@vMX-4# set system services ssh port 5000 rate-limit 3

[edit groups Common-Services]
root@vMX-4# set system services telnet connection-limit 5
```

Configuration Groups Wildcards

- Regular expressions can be used as wildcards within a configuration group to match multiple elements at once

```
Common-Int-IPv6 {
    interfaces {
        <ge-*> {
            unit 0 {
                family inet6 {
                    nd6-max-cache 50;
                    dad-disable;
```

In this example by using the asterisk after ”ge-” we are matching all GigabitEthernet interfaces
Items that use wildcards or other regular expressions need to be enclosed within angled brackets.

Applying Configuration Groups

- Child elements (protocols, services, interfaces, etc) “inherit” the configuration settings within configuration groups.
- Configuration groups can be applied at any level of the configuration hierarchy:

Applied at the top of the configuration hierarchy →

```
[edit]
root@vMX-4# set apply-groups Common-Int-IPv6

[edit]
root@vMX-4# commit
commit complete
```

Applied lower in the configuration hierarchy →

```
[edit]
root@vMX-4# edit interfaces ge-0/0/1

[edit interfaces ge-0/0/1]
root@vMX-4# set apply-groups Common-Int-IPv6

[edit interfaces ge-0/0/1]
root@vMX-4# commit
```

Viewing Inheritance

- Output of "**show**" or "**show configuration**" only displays configuration that is unique to each section of the configuration…NOT configuration data inherited from Configuration Groups
- Must use the filter of "***display inheritance***" to view inherited configuration elements

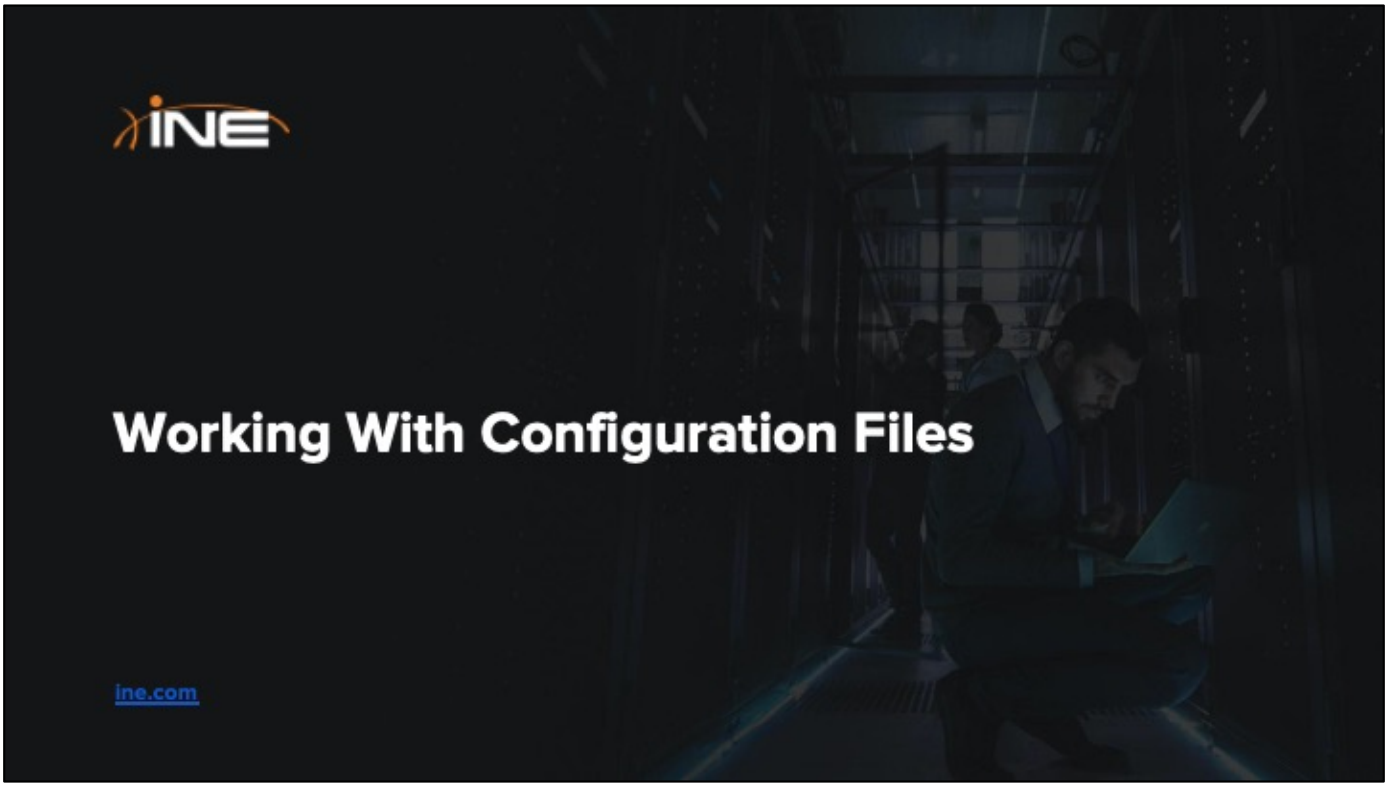
Viewing Inheritance

Candidate Configuration: "show"

```
interfaces {
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 1.1.3.4/24;
            }
        }
    }
    ge-0/0/2 {
        unit 0 {
            family inet {
                address 1.1.4.4/24;
            }
        }
    }
```

Candidate Configuration: "show | display inheritance"

```
interfaces {
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 1.1.3.4/24;
            }
            ##
            ## 'inet6' was inherited from group 'Common-Int-IPv6'
            ##
            family inet6 {
                ##
                ## '50' was inherited from group 'Common-Int-IPv6'
                ##
                nd6-max-cache 50;
                ##
                ## 'dad-disable' was inherited from group 'Common-Int-IPv6'
                ##
                dad-disable;
            }
        }
    }
    ge-0/0/2 {
        unit 0 {
            family inet {
                address 1.1.4.4/24;
            }
            ##
            ## 'inet6' was inherited from group 'Common-Int-IPv6'
            ##
            family inet6 {
                ##
                ## '50' was inherited from group 'Common-Int-IPv6'
                ##
                nd6-max-cache 50;
                ##
                ## 'dad-disable' was inherited from group 'Common-Int-IPv6'
                ##
                dad-disable;
```



Topic Overview

- Configuration File Locations
- Rollback Usage
- Understanding “Show | Compare”
- Comparing Rollbacks
- Viewing CLI Change History
- The Junos Rescue Configuration

Configuration File Locations

- Before any configuration is applied, Juniper devices only have a single configuration file called, "factory.conf" (the Factory Default file)
  - Located in /etc/config/factory.conf
- After a device has been configured at least once it has an "Active Configuration" file.
  - Located in /config/juniper.conf.gz

Rollback Juniper Configuration Files

- As changes are made to the Active Configuration, the previous Active Configuration is saved as a Rollback Configuration
  - First Rollback can be found in /config/juniper.conf.1.gz
  - Second rollback is juniper.conf.2.gz (and so on)
- The software saves the last 50 committed configurations, including the rollback number, date, time, and name of the user who issued the commit configuration command.

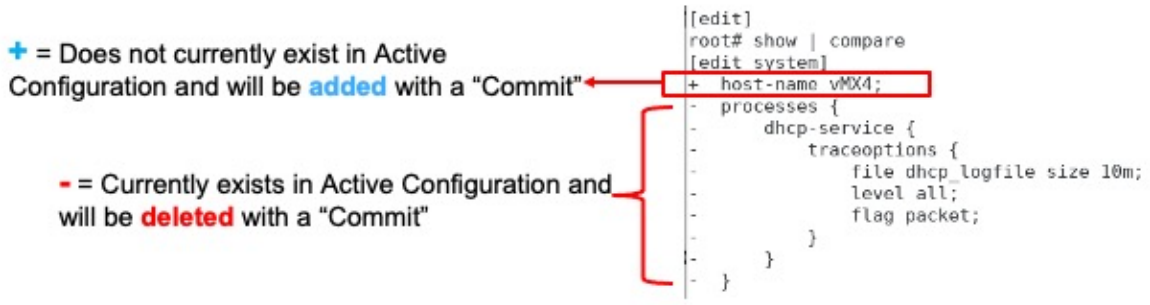
Configuration Rollback

- **# rollback** or **rollback 0**
  - Deletes/reverts all unsaved changes
  - Loads active config as candidate config, reverting uncommitted changes

- **# rollback [num]**
  - Loads previous commits into candidate config
  - Still needs **commit** to apply the rollback to active config
- **# show | compare**
  - Shows the difference between active config and candidate config and uncommitted changes

Understanding "Show | Compare" Output

- When performed within **Configuration Mode**, the command, "*show | compare*" contrasts uncommitted changes with the current Active Configuration
- This shows what WILL happen if you "commit" your Candidate Configuration
- If you wish to delete your uncommitted changes type "*rollback 0*"

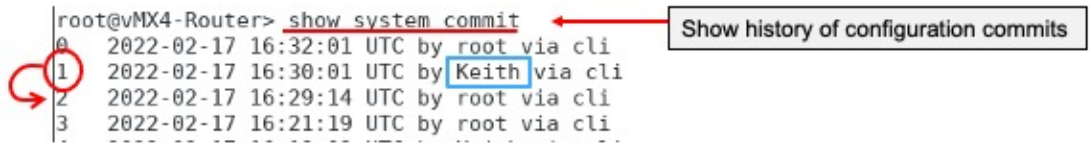


Comparing Rollbacks

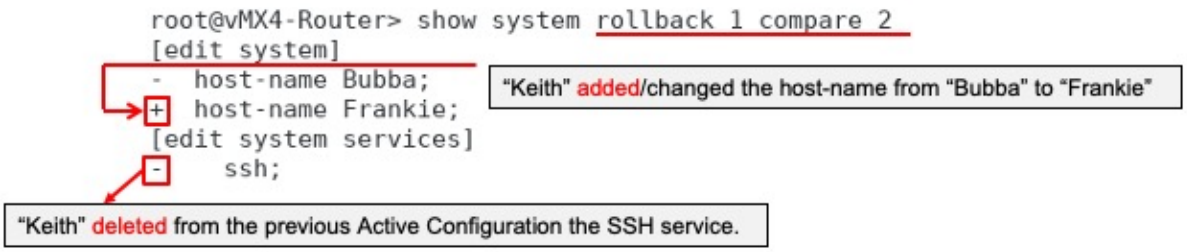
- **`# show | compare rollback x`**
  - Shows the diff between candidate config and rollback x
- **`> show config | compare rollback x`**
  - Shows the diff between active config and rollback x
- **`> show system rollback x`**
  - Shows the full configuration of rollback x
- **`> show system rollback x compare y`**
  - Shows the diff between rollback x and y

Viewing User Change History

- How do you view the configuration changes a previous user made?



- To view changes that user "Keith" made compare his/her rollback value against the ***previous*** rollback value



Rescue Configuration

- As successive commits are made, the configuration may drift from a known, working configuration.
- It can be difficult to remember which Rollback contains the known, working config
- A rescue configuration allows you to define a known, working configuration or a configuration with a known state for recovery, if necessary.
- The rescue configuration can serve as a last resort if your device configuration and the backup configuration files become damaged beyond repair

Creating A Rescue Configuration

- First, ensure your device is operating as expected and is stable.
- Create the rescue configuration:

```
user@host> request system configuration rescue save
```

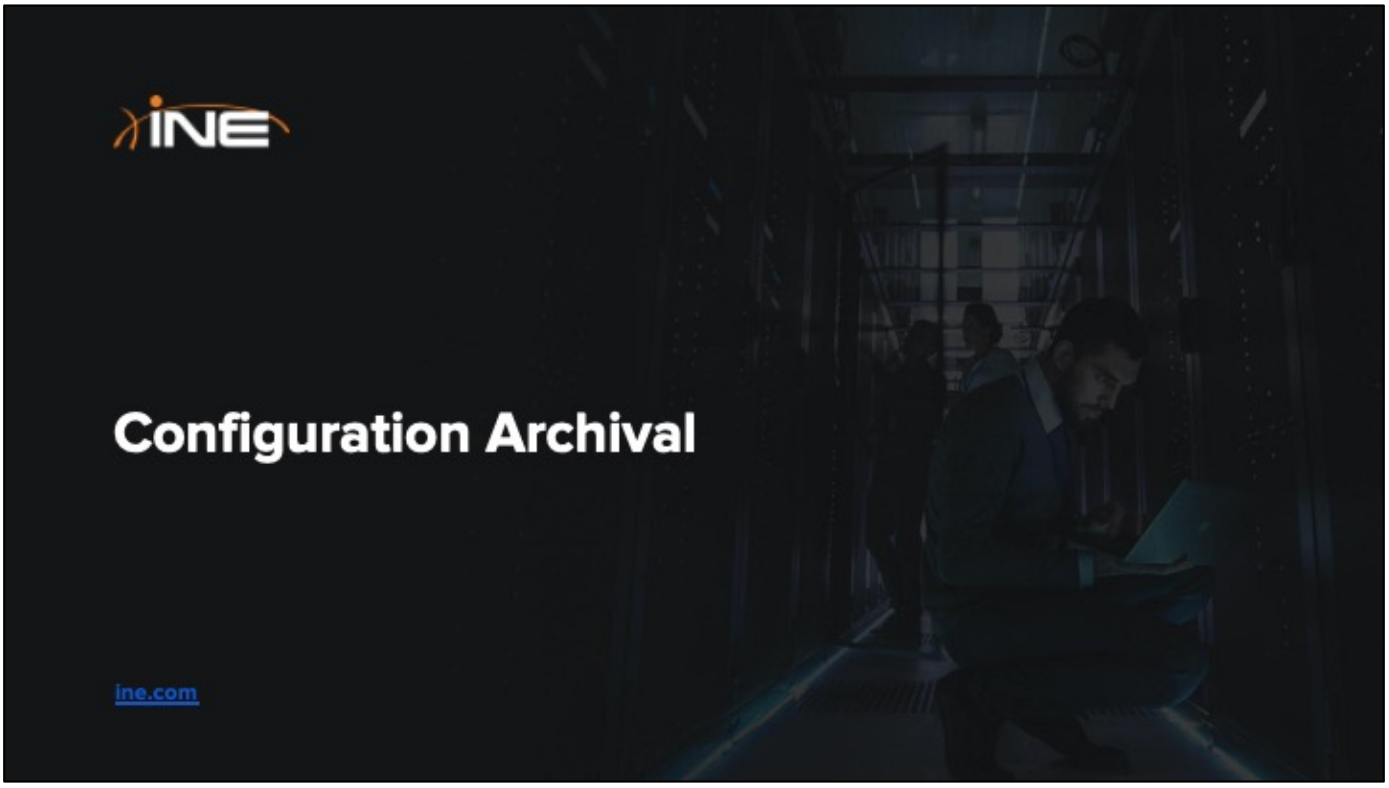
- Rollback to the rescue configuration when needed:

```
[edit]
user@host# rollback rescue
load complete
```


Must still issue the "commit" command

- If desired, you can delete the rescue configuration

```
user@host> request system configuration rescue delete
user@host>
```



Topic Overview

- Introduction To Configuration Archival
- Configuration Archival Options
- Configuration Archival Examples
- Verification

Configuration Archival

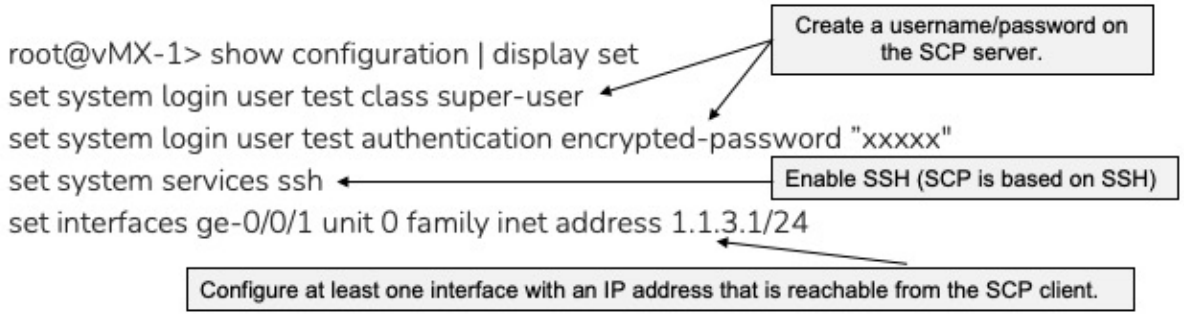
- A good best-practice is to periodically save a copy of your working configuration to an external devices
- Configuration archival performs this automatically
- When should this be done?
  - At periodic, regular intervals or...
  - After each commit

Configuration Archival Options

- Decide when you want Configuration Archival to activate:
  - *Set system archival configuration transfer-on-commit*
  - *Set system archival configuration transfer-interval <mins>*
- Configuration archival location, method and credentials
  - [edit system archival configuration]
  - *Set archive-sites <scp | ftp | http><username@path> password <password>*

Configuration Archival Example

- In this example we'll use another Juniper device as our archive server



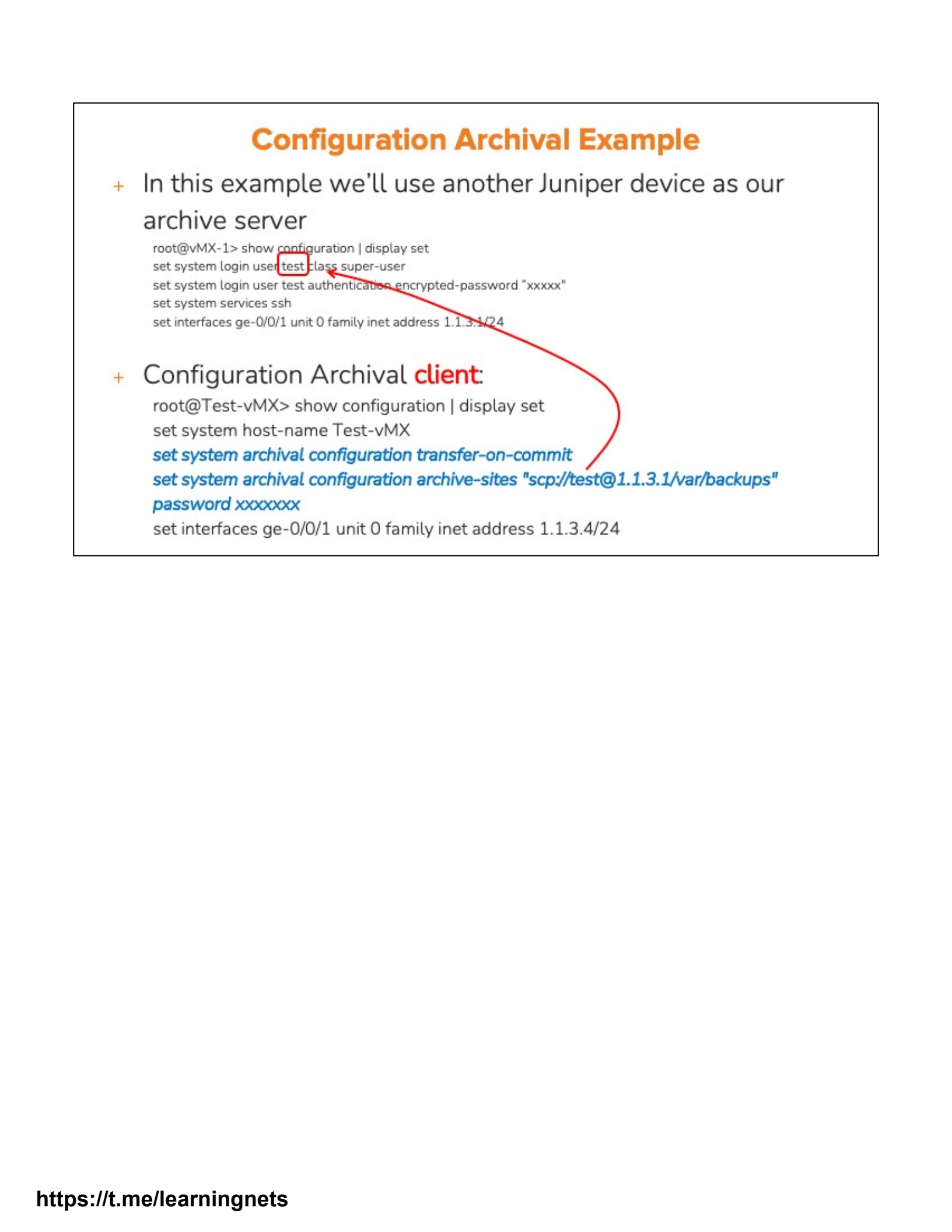
Configuration Archival Example

- In this example we'll use another Juniper device as our archive server

```
root@vMX-1> show configuration | display set
set system login user test class super-user
set system login user test authentication encrypted-password "xxxxx"
set system services ssh
set interfaces ge-0/0/1 unit 0 family inet address 1.1.3.1/24
```

- Configuration Archival **client**:

```
root@Test-vMX> show configuration | display set
set system host-name Test-vMX
set system archival configuration transfer-on-commit
set system archival configuration archive-sites "scp://test@1.1.3.1/var/backups"
password xxxxxxx
set interfaces ge-0/0/1 unit 0 family inet address 1.1.3.4/24
```



Configuration Archival Verification

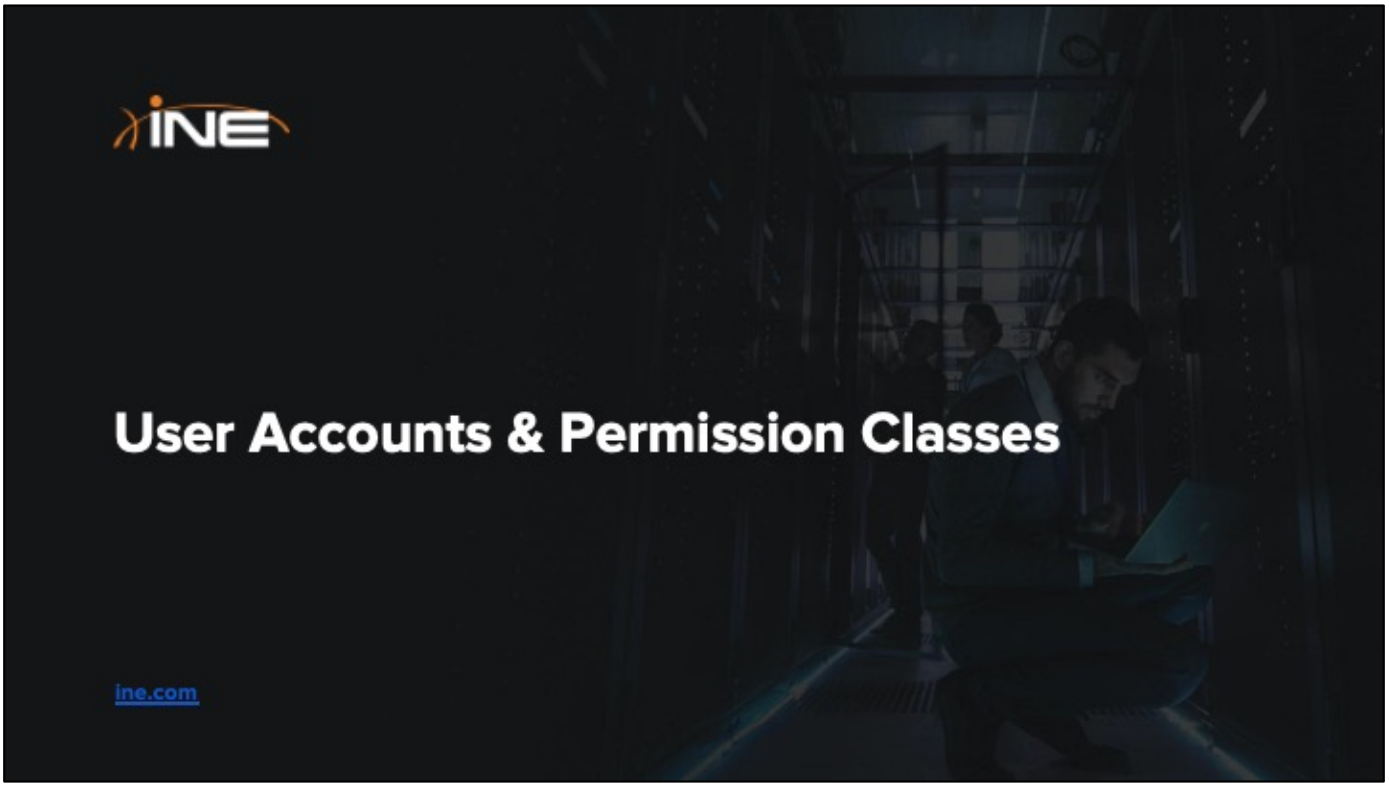
Archival Client

```
root@Test-vMX> edit
Entering configuration mode

[edit]
root@Test-vMX# set system host-name My-vMX-Archival-Test

[edit]
root@Test-vMX# commit
commit complete
```

Archival Server

```
root@vMX-1> file list

/var/backups/:
My-vMX-Archival-Test_20220209_203118_juniper.conf.gz
Test-vMX_20220209_201817_juniper.conf.gz
vMX-4_20220209_201337_juniper.conf.gz
```

host-name_YYYYMMDD_HHMMSS_juniper.conf.gz



Topic Overview

- Junos User Accounts
- Login Classes & Permissions
- Creating & Modifying Login Classes
- Configuring Users

User Accounts

- User accounts must be created for remote access device login
- Users can be authenticated in two different ways:
  - Locally-configured user accounts
  - Remote authentication with Radius or TACACS+
- User accounts consist of the following:
  - Username
  - User's full name (optional)
  - User ID (optional)
  - Class

Required Information For User Accounts

- Username
  - Do not use spaces, colons or commas
  - May be up to 64-characters in length
- User's Full Name (Optional)
  - Enclose in quotation marks if name contains spaces
  - Do not use colons or commas
- User Identifier (Optional)
  - Numeric identifier associated with user
  - In the range of 100 to 64,000
  - Software-assigned if you don't specify one
- User access class
  - Defines privileges for the user

Login Classes

- Define access privileges for users who are logged into a box
- Control such things as:
  - Ability to commit configuration changes
  - What commands a user can utilize
  - Session idle times
- Four classes are predefined, and others can be manually created
- A user can only be assigned to one login class

Classes & Permissions

- Predefined, and user-created classes are assigned one-or-more permission flags

Login Class	Permission Flags
Operator	clear, network, reset, trace, and view
Read-Only	view
Super-user	all
Unauthorized	none

- Predefined classes cannot be changed, copied or renamed.

Class Permissions

- There are a wide range of permissions that can be assigned to user-created classes
- Most permissions have two forms:
  - "Plain" form: provides read-only capability for that permission type
  - "Control" form: provides full read/write capability

Common Permissions

- *Configure*: Can enter configuration mode (using the configure command) and commit configurations (using the commit command).
- *network*: Can access the network by using the ping, ssh, telnet, and traceroute commands.
- *Routing* (and *routing-control*): Can view (or view and configure) general routing, routing protocol, and routing policy configuration information in configuration mode and operational mode.

https://www.juniper.net/documentation/us/en/software/junos/user-access/topics/topic-map/junos-os-login-class.html

Creating Login Classes

- root@vMX-1# set system login class <descriptive-name> idle-timeout <*mins*> permissions [*permission-1 permission-2 permission-3*]

Example:

```
[edit]
root@vMX-1# set system login class Junior-Admins idle-timeout 15 permissions [ configure interface-control network ]
```

Brackets are not required if only assigning a single permission.

Modifying Custom Classes

- Configuring permissions for existing classes is additive:

```
[edit]
root@vMX-1# show system login class Junior-Admins
idle-timeout 15;
permissions [ configure interface-control network ];
```

```
[edit]
root@vMX-1# set system login class Junior-Admins permissions maintenance
```

```
[edit]
root@vMX-1# show system login class Junior-Admins
idle-timeout 15;
permissions [ configure interface-control maintenance network ];
```

Verifying Custom Classes

```
root@vMX-1# show
## Last changed: 2022-02-09 17:05:05 UTC
version 18.2R1.9;
system {
    login {
        class Test-Class {
            permissions network;
        }
        class Test2 {
            permissions [ firewall interface-control ];
        }
        class Test3 {
            permissions [ interface maintenance network ];
        }
        class Junior-Admins {
            idle-timeout 15;
            permissions [ configure interface-control network ];
        }
    }
```

Configuring Users

- Create username and assign a password:

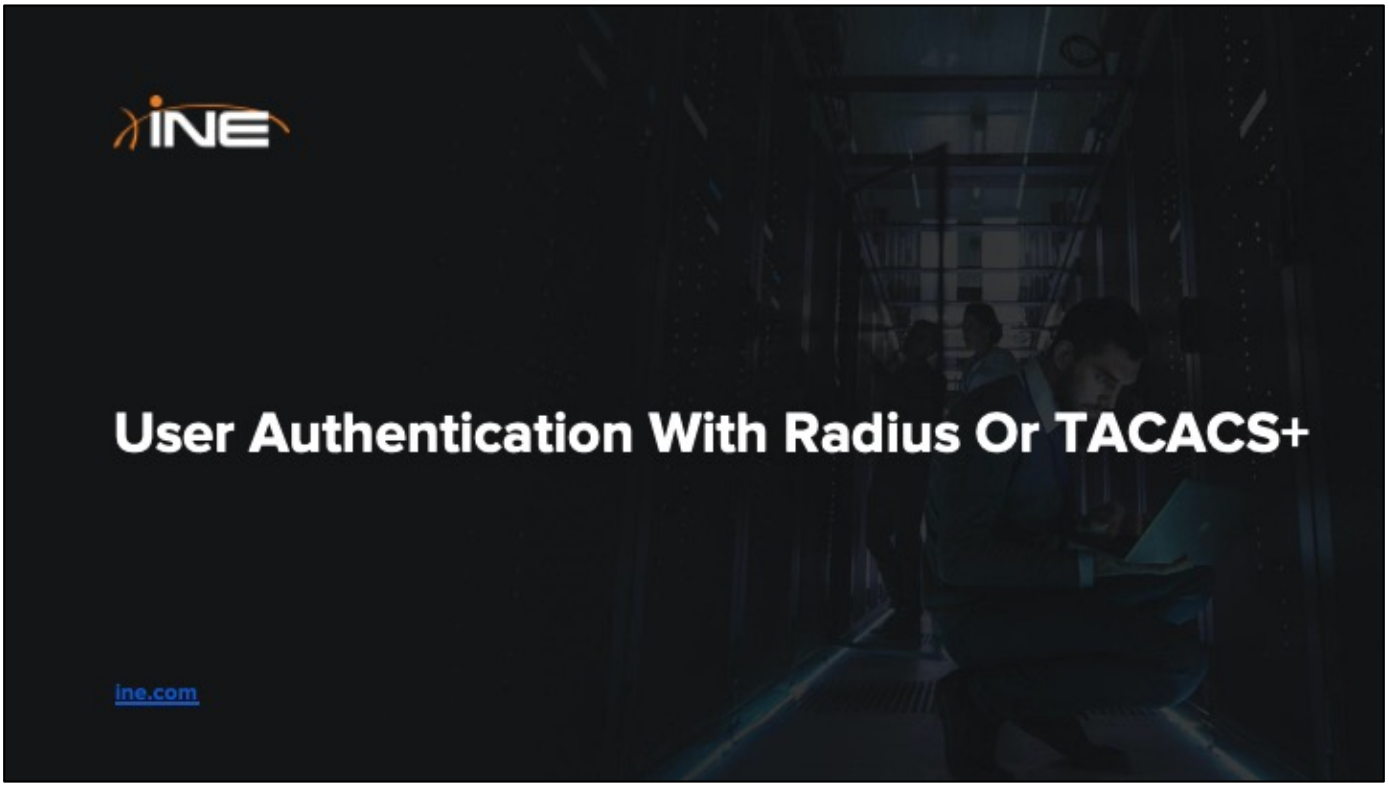
```
root@vMX-1# set system login user Keith authentication plain-text-password
New password:
Retype new password:

[edit]
root@vMX-1#
```

Type (and retype) the user's password here

- Apply a class to the user:

```
root@vMX-1# set system login user Keith class ?
Possible completions:
  <class>              Login class
  Junior-Admins
  Test-Class
  operator             permissions [ clear network reset trace view ]
  read-only            permissions [ view ]
  super-user           permissions [ all ]
  unauthorized         permissions [ none ]
[edit]
```

This can also be done in a single line by specifying the "class" first followed by the authentication password.



Topic Overview

- Using Radius or TACACS+ For Authentication
- User Templates
- Enabling Remote Management With Telnet Or SSH

Radius & TACACS+ Authentication

- If user accounts frequently change, it may make sense to keep all authentication data in a central database.
- Users can be authenticated using AAA with Radius or TACACS+ methods
- Juniper device must be configured to connect with a Radius or TACACS+ server
- Local authentication should still be used for fallback.

Radius Client Example

- In this example, the Juniper device is considered a “client” of the Radius server:
  - [edit] *set system radius-server <ip-addr> secret <password>*
- Ensure that Juniper device is sourcing Radius packets from an address the server is expecting:
  - [edit] *set system radius-server <ip-address> source-address <juniper-ip-address>*
- Configure Juniper device to send authentication requests to Radius and fallback to local authentication

```
root@vMX-4# set system authentication-order [radius password]
```

User Templates Overview

- Radius only provides ***Authentication*** capabilities for users who wish to access device administration
- User Templates must be configured on the box to handle the Authorization portion of AAA
- If an authenticated user is not associated with a User Template, that user fails authentication.
- There are two types of User Templates available:
  - **Remote**: Special-purpose template to apply to all users who do not match with any user-defined template.
  - **User-defined** templates

Configuring User Templates

- User Templates are configured just like configuring a normal user…but without supply a password:
  - Configuring the special "remote" template:
    - [edit system login] ***set user remote class [class]***

```
root@vMX-4# set system login user remote class read-only
```

  - Configuring user-defined templates:
    - [edit system login] ***set user <template-name> class [class]***

```
root@vMX-4# set system login user Admins class super-user
```

- Configure the RADIUS server to map the authenticated user to the appropriate user template
  - Set the "Juniper-Local-User-Name" Juniper VSA (vendor-specific attribute) (Vendor 2636, type 1, string) to the name of a user template configured on the device,

Enabling Remote Management Access

- After user accounts are created, either Telnet or SSH (or both) can be configured to allow remote management access to your device:

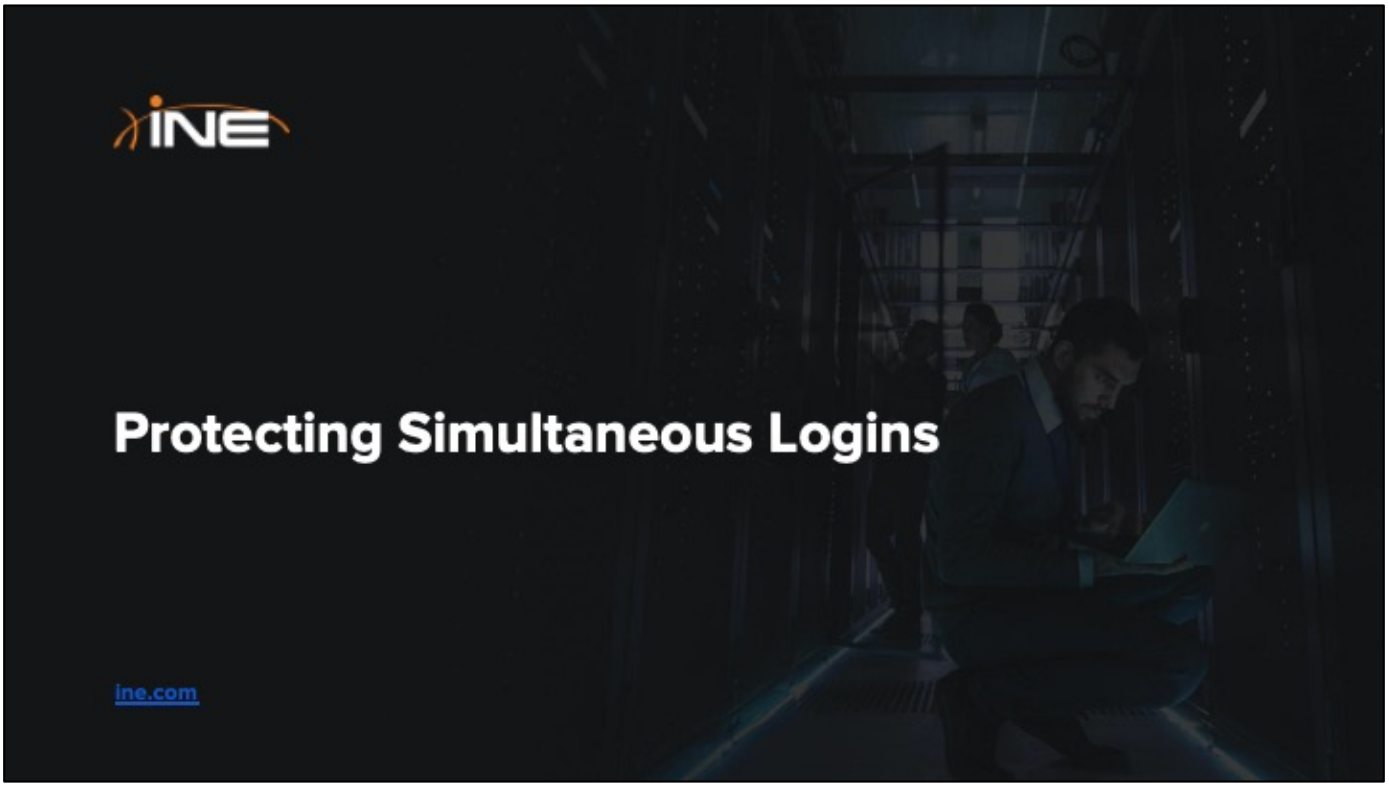
```
system {
    login {
        user Keith {
            uid 2000;
            class super-user;
            authentication {
                encrypted-password "$6$Q/R12sLu$
    services {
        ssh;
        telnet;
```

root@vmx1#*set system services <telnet | ssh>*

SSH Keys

- Once the SSH service is committed, various public/private keypairs are dynamically generated by default such as:
  - RSA
  - SHA256
  - ECDSA

```
root@vMX-1# commit
Generating RSA key /etc/ssh/ssh_host_key
Generating public/private rsa1 key pair.
Your identification has been saved in /etc/ssh/ssh_host_key.
Your public key has been saved in /etc/ssh/ssh_host_key.pub.
The key fingerprint is:
SHA256:BKwHIm2XhJ5VEyTI0U4Z16vNcJFuqXjRdgyryrGcqeg root@vMX-1
The key's randomart image is:
+---[RSA1 2048]---+
| o.*+BBo .       |
|. B Oo.o=        |
| + B o o.B       |
|  o o +.X o      |
|     o %S.       |
|    o = o        |
|   o B           |
| .  B            |
|oE..             |
+----[SHA256]-----+
Generating DSA key /etc/ssh/ssh_host_dsa_key
Generating public/private dsa key pair.
```

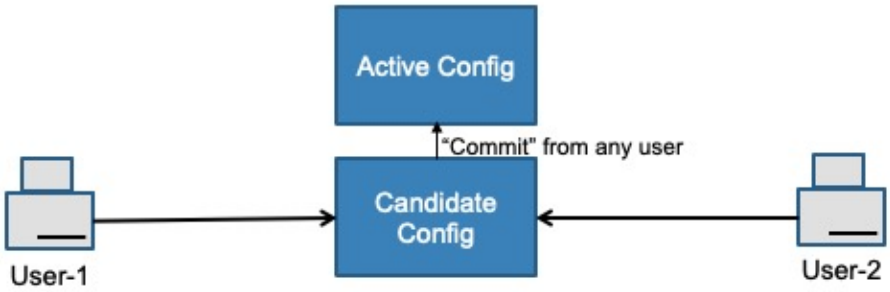


Topic Overview

- Possible Problems With Multiple Simultaneous Logins
- Configure Exclusive
- Configure Private

Protecting The Candidate

- A single copy of the Candidate Configuration is shared among all users who are logged in
- A “commit” from one user commits their changes as well as *any other unsaved changes* from other admins
- This can lead to confusion and configuration errors



Configure Exclusive

- The command "*configure exclusive*" allows only a single admin to commit changes to the Candidate Configuration file

First user…

```
root@vMX-1> configure exclusive
warning: uncommitted changes will be discarded on exit
Entering configuration mode
```

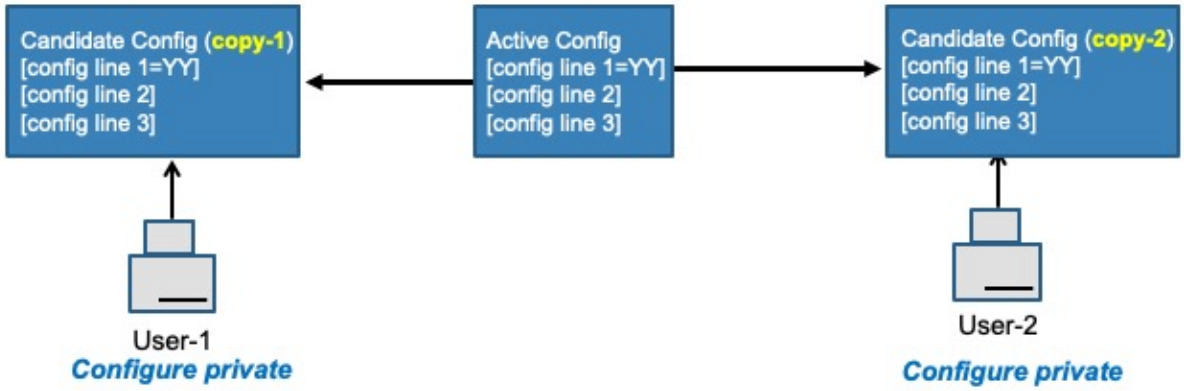
Subsequent users…

```
Keith@vMX-1> configure
Entering configuration mode
Users currently editing the configuration:
  root terminal u0 (pid 6072) on since 2022-02-04 15:49:12 UTC
      exclusive [edit]

[edit]
Keith@vMX-1# set system host-name Test
error: configuration database locked by:
  root terminal u0 (pid 6072) on since 2022-02-04 15:49:12 UTC, idle 00:00:56
      exclusive [edit]
```

- Notice that "configure exclusive" does NOT prevent someone else from entering configuration mode! But once they are there, they can't make any changes.

Configure Private

- Each user who issues a "*Configure Private*" command starts with their own, independent copy of the Candidate Configuration

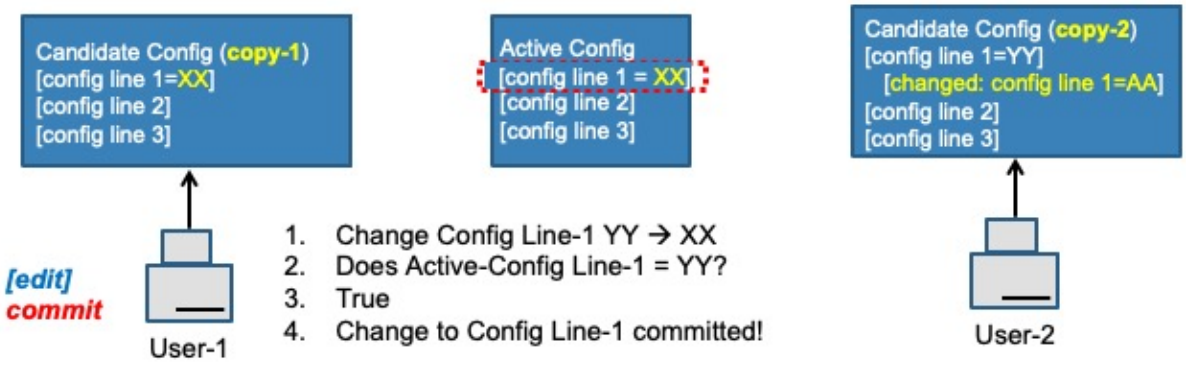


Configure Private

- Uncommitted changes made by a user ONLY affect that user's Candidate Config

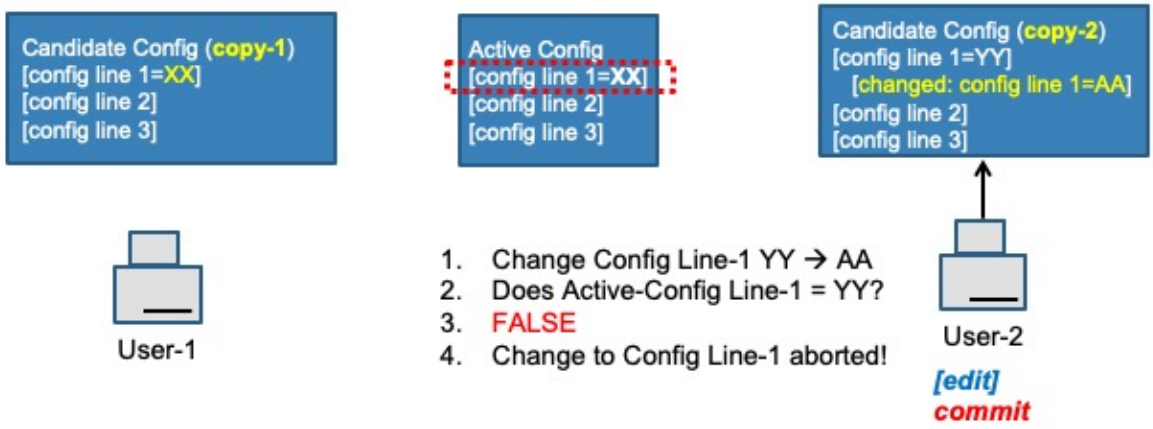
Configure Private

- Commits by a user affect that user's Candidate Configuration as well as the Active Configuration file
- They do NOT affect another user's private Candidate Configuration file.



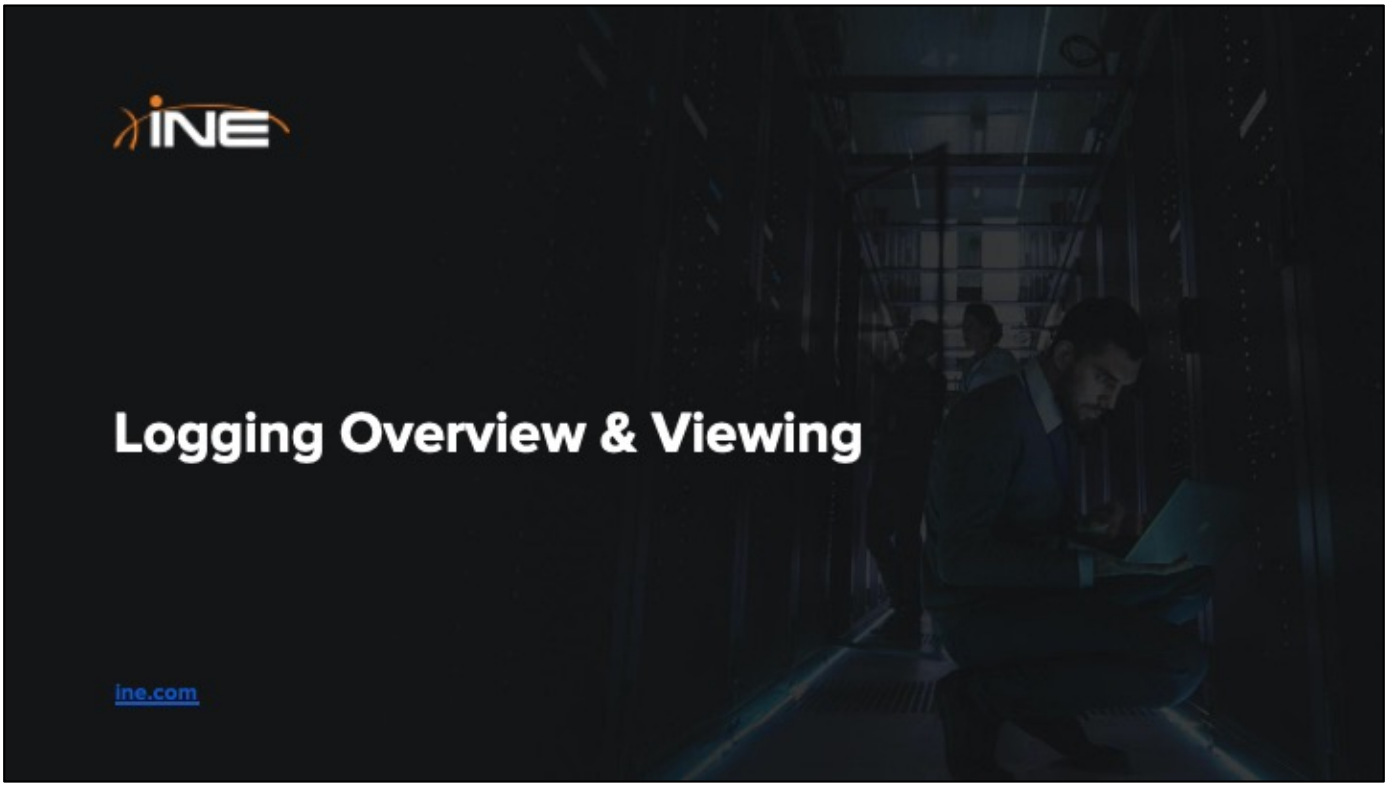
Configure Private

- Further commits to the same section of the config by a different user will fail the logic test.



Configure Private

- When a user (who entered “configure private”) leaves configuration mode with unsaved changes, those changes are lost.
- This is different than the behavior when leaving normal configuration mode (unsaved changes are saved)



Topic Overview

- Overview Of Juniper Logging
- Common Log Files
- Viewing Logs

Overview Of Juniper Logging

- The Junos OS generates system log messages to record events that occur on the switch, including the following:
  - Routine operations, such as a user login into the configuration database.
  - Failure and error conditions, such as failure to access a configuration file.
  - Emergency or critical conditions, such as power-down of the switch due to excessive temperature.
- System log messages are generated due to various reasons
  - SYSLOG events
  - Internally-generated logging messages from daemons

Juniper Logging Overview

- Juniper devices store system logging messages in various files
- System logs can assist with troubleshooting
- Most log files are stored in /var/log directory
  - Contents of this directory can be viewed with the "*file list /var/log*" command

```
root@vMX-1> file list /var/log

/var/log:
aprobed
apsd
authd_enc
authd_libstats
authd_profilelib
authd_sdb.log
bbesmgd
chassisd
<output omitted>
```

The Junos CLI command, "**show log**" will display the same output.

Common Log Locations

- The most common log files that people view are:
  - **Chassisd**: Logs about the chassis process (daemon)
  - **Interactive-commands**: Contains historical list of commands that were entered on the box (among other things)
  - **Messages**: Logs from virtually all messages are stored here

Viewing Common Logs

- There are a couple of commands one can use to view logging messages
  - root@vMX-1>**file show <path>/<filename>**

```
root@vMX-1> file show /var/log/chassisd
Jan 27 15:52:13  trace_log_init: configuring snmp tracing
Jan 27 15:52:13
CHASSISD release 18.2R1.9 built by builder on 2018-06-28 04:38:09 UTC
Jan 27 15:52:13  trace flags 7f00 trace file /var/log/chassisd size 3000000
Jan 27 15:52:13  Opening static database...
```

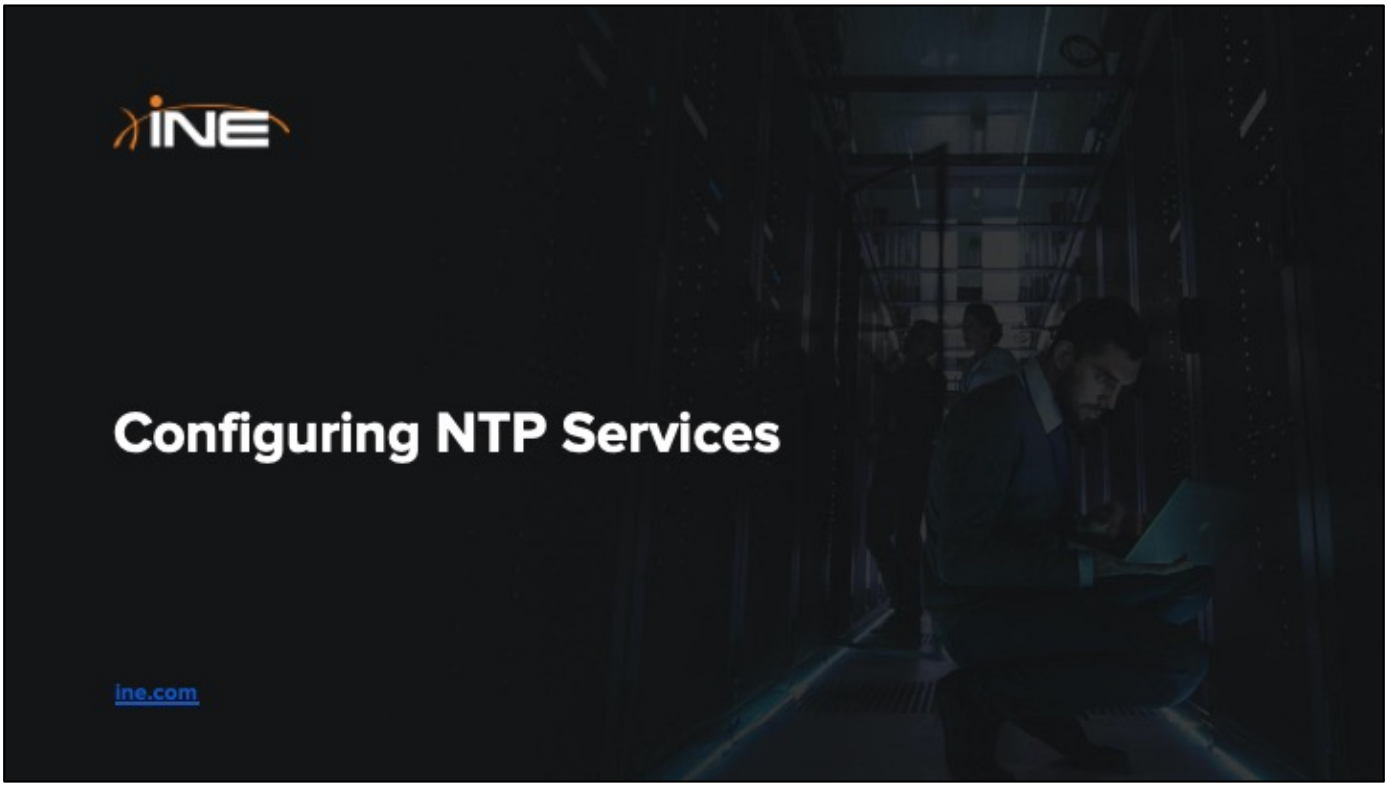
  - root@vMX-1>**show log <filename>**

```
root@vMX-1> show log messages
Jan 27 19:15:00 vMX-1 newsyslog[9334]: logfile turned over due to size>1024K
Jan 27 19:15:09  vMX-1 fpc0 SCHED: Thread 66 (PFE Statistics) ran for 1875 ms without yielding
Jan 27 19:15:13  vMX-1 fpc0 Scheduler Oinker
```

  - root@vMX-1>**monitor <start | stop> <logging filename>**

The “file show” command also supports CLI filters such as:
file show /var/log/interactive-commands | find “User 'root', command” → This filter will display only the commands that “root” entered from the CLI

The “monitor start” command sends to the console in realtime every log file generated by the process you specify:
monitor start messages
monitor start interactive-commands
***Be careful as this could generate a TON of output depending on what is selected.



Topic Overview

+ Defining The Problem Solved By NTP
+ Clock Sources
+ NTP Overview & Device Roles
+ The NTP “Boot Server”
+ NTP Client Configuration
+ NTP Authentication
+ NTP Verification

The Need For Timestamps

- Juniper devices (routers, switches, firewalls, etc) can utilize or display time-and-date information for several purposes:
  - Logging output
  - Debugging/Traceoption output
  - User “show” commands
  - Network Management/Reporting tools
- Having an accurate timestamp on the above features can be critical

Where Do We Get The Clock

- All routers/switches have an internal “system clock”
- Most are battery-driven and maintain the time/date across reloads
- Devices with battery-driven system clocks can also distribute this information to remote devices via NTP

Sources Of Time

- The System Clock can obtain its information via:
  - Manual Configuration
  - NTP (Network Time Protocol)
  - SNTP (Simple Network Time Protocol)
  - VINES Time Service (Virtual Integrated Network Service)
- In this section we'll be concentrating on NTP

NTP Overview

- Network Time Protocol
- IETF Standard: RFC 1305 (version-3) and RFC 5905 (version-4)
- Uses UDP port 123 (source and destination)
- NTP Nodes obtain time from an authoritative source:
  - Atomic Clock
  - GPS
  - Radio Clock
  - Other network devices

NTP Device Roles

- NTP Client = Device that periodically polls a server for time/calendar information
- NTP Server = Provides that information to Client
  - NTP Server also considered an “authoritative source” of time based on its Stratum Level
    - Stratum 1 = Device directly connected to a radio or atomic clock source
    - Stratum 2 = Time Server that is one hop away from Stratum-1 device

A “Stratum 0” is an atomic clock.

NTP Synchronization

- If a Juniper device is configured as an NTP Client it will attempt to synchronize time with a configured NTP server.
- Synchronization will fail if a large discrepancy exists between initial NTP Client and NTP Server values.
- An NTP "Boot Server" can be configured to provide initial time-and-date information at boot time.

Here is a great URL that explains the difference between an NTP Server and an NTP Boot-Server:
https://community.juniper.net/communities/community-home/digestviewer/viewthread?MID=65266

Juniper NTP Client Configuration

- Configure the NTP Boot-Server

```
root@vMX-4# set system ntp boot-server 4.4.4.44
```

- Set the NTP Server address (ensure it is reachable)
  - NTP Server and NTP Boot-Server may be the same device

```
set system ntp server 4.4.4.44
```

- Default NTP version is “4”. Change version if necessary:

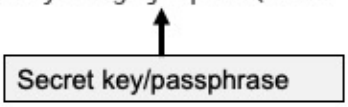
```
root@vMX-4# set system ntp server 4.4.4.44 version ?
Possible completions:
  <version>            NTP version to use (1..4)
```

If ONLY an NTP Boot-Server is configured, the device will get accurate time-and-date info upon booting. However, after this point there is no way to prevent clock drift.

NTP Authentication

- A spoofed NTP server could cause errors in time-and-date settings set by NTP
- One may prefer to authenticate the NTP server to verify authenticity. HMAC with MD5 can provide this

```
set system ntp authentication-key 1 type md5
set system ntp authentication-key 1 value "$9$cGzyeM2gajiqY25Qn6At"
set system ntp server 4.4.4.44 key 1
set system ntp trusted-key 1
```



The value of “authentication-key” (such as “1”) needs to match what has been configured on the NTP Server.

NTP Verification

- Show ntp status

```
root@vMX-4> show ntp status
status=c035 sync_alarm, sync_unspec, 3 events, event_clock_reset,
version="ntpd 4.2.0-a Thu Jun 28 03:52:42  2018 (1)", processor="amd64",
system="FreeBSDJNPR-11.0-20180614.6c3f819_buil", leap=11, stratum=16,
precision=-23, rootdelay=0.000, rootdispersion=0.705, peer=0,
refid=STEP, reftime=00000000.00000000  Thu, Feb  7 2036  6:28:16.000,
poll=4, clock=e5b4fbf9.1c76362b  Mon, Feb 14 2022 16:04:09.111, state=3,
offset=0.000, frequency=0.000, jitter=0.000, stability=0.000
```

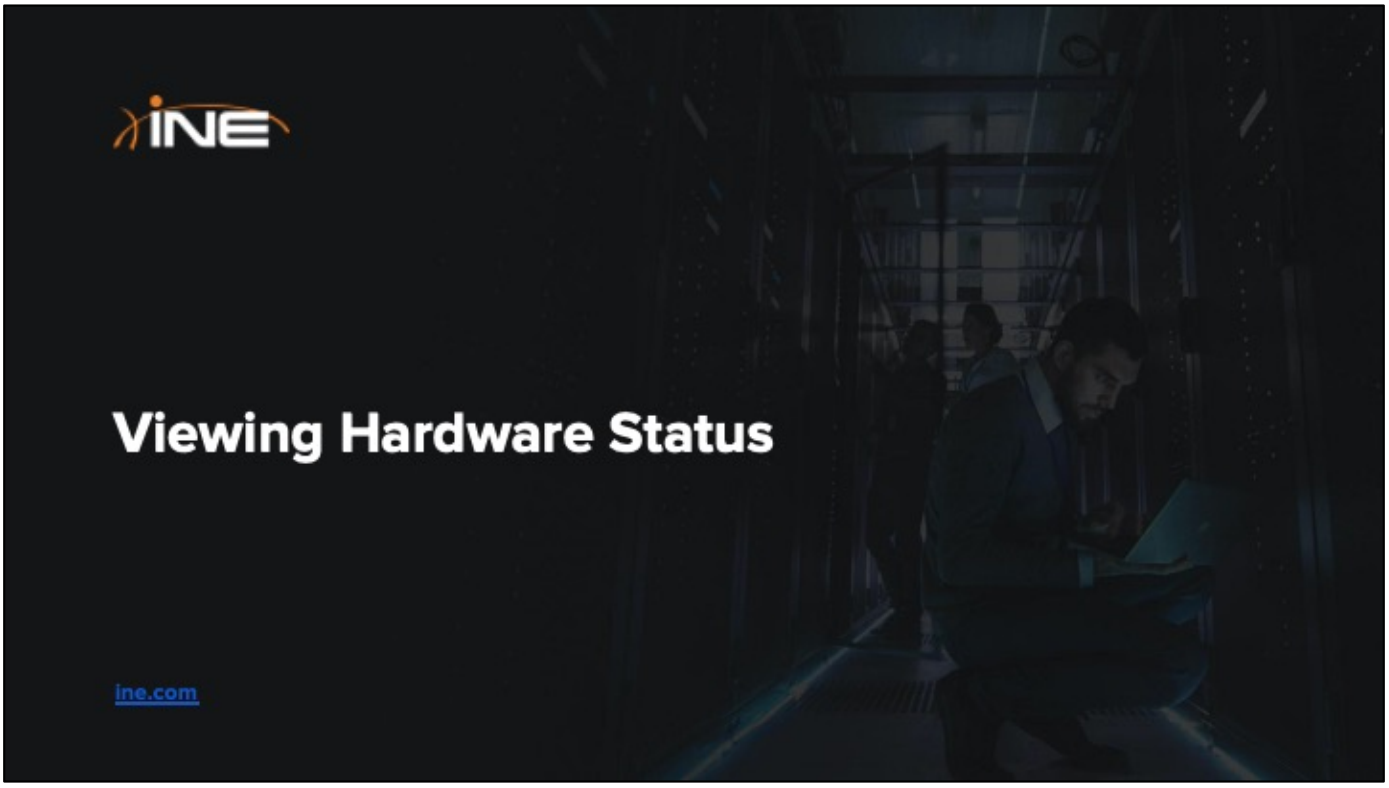
```
root@vMX-4> show ntp status
status=0664 leap_none, sync_ntp, 6 events, event_peer/strat_chg,
version="ntpd 4.2.0-a Thu Jun 28 03:52:42  2018 (1)", processor="amd64",
system="FreeBSDJNPR-11.0-20180614.6c3f819_buil", leap=00, stratum=2,
precision=-23, rootdelay=5.498, rootdispersion=1.485, peer=56196,
refid=4.4.4.44,
reftime=e5b4fc1f.50dfc5f8  Mon, Feb 14 2022 16:04:47.315, poll=6,
clock=e5b4fc2d.269f8f81  Mon, Feb 14 2022 16:05:01.150, state=3,
offset=0.000, frequency=0.000, jitter=64.716, stability=0.000
```

It may take a few minutes (depending on clock drift) for NTP peering to establish

NTP Verification

- *Show system uptime*

```
user@vSRX1-JNCIA> show system uptime
Current time: 2020-04-21 20:41:07 UTC
Time Source:  NTP CLOCK
System booted: 2020-04-20 23:33:34 UTC (21:07:33 ago)
Protocols started: 2020-04-20 23:35:51 UTC (21:05:16 ago)
Last configured: 2020-04-21 13:11:28 UTC (07:29:39 ago) by jncia
 8:41PM  up 21:08, 1 users, load averages: 1.23, 1.33, 1.32
```



Topic Overview

+ An Introduction To Monitor Juniper Device Hardware Status

Show chassis hardware

- Displays hardware information about your chassis

```
user@SRX4200> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                SN1234AB0001      SRX4200
Midplane         REV 07   650-071675   12345678901       SRX4200
Routing Engine 0          BUILTIN      BUILTIN           SRX Routing Engine
FPC 0                     BUILTIN      BUILTIN           FEB
  PIC 0                   BUILTIN      BUILTIN           8x10G-SFP
    Xcvr 0       REV 01   740-031980   AA12345678        SFP+-10G-SR
    Xcvr 4       REV 01   740-021308   AB12345678        SFP+-10G-SR
```

Image courtesy of "Day One: Beginner's Guide to Learning Junos"

The output of this command varies widely depending on your hardware platform.

Show chassis hardware models

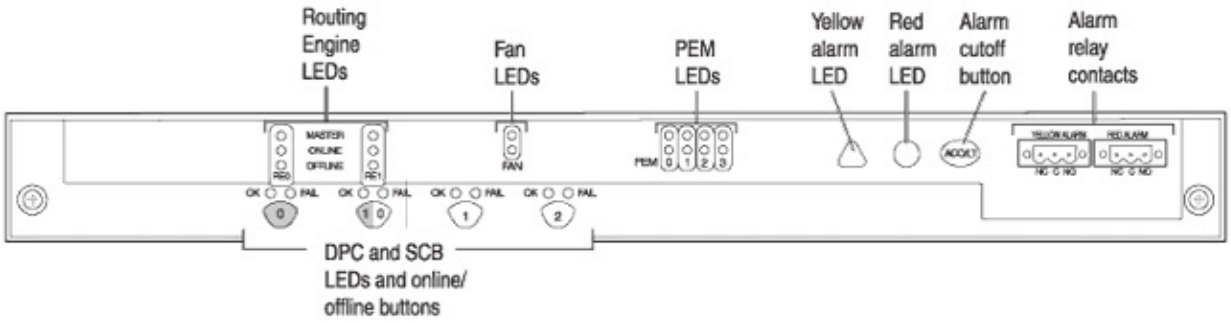
- Shows FRU ("Field Replaceable Unit") model numbers
- Useful when ordering replacement parts from Juniper

```
user_1@mx960-lr4-re1> show chassis hardware models
Hardware inventory:
Item             Version  Part number  Serial number     FRU model number
Midplane         REV 03   710-013698   *******           CHAS-BP-MX960-S
FPM Board        REV 03   710-014974   *******           CRAFT-MX960-S
PEM 0            Rev 05   740-029344   *******           PWR-MX960-4100-DC-S
PEM 1            Rev 05   740-029344   *******           PWR-MX960-4100-DC-S
PEM 2            Rev 05   740-029344   *******           PWR-MX960-4100-DC-S
PEM 3            Rev 05   740-029344   *******           PWR-MX960-4100-DC-S
Routing Engine 1 REV 01   740-051822   *******           RE-S-1800X4-32G-S
```

Image courtesy of "Day One: Beginner's Guide to Learning Junos"

Remotely Accessing LED Status

- LEDs on various cards are useful in determining hardware health...but how does one access those LEDs from a remote connection?

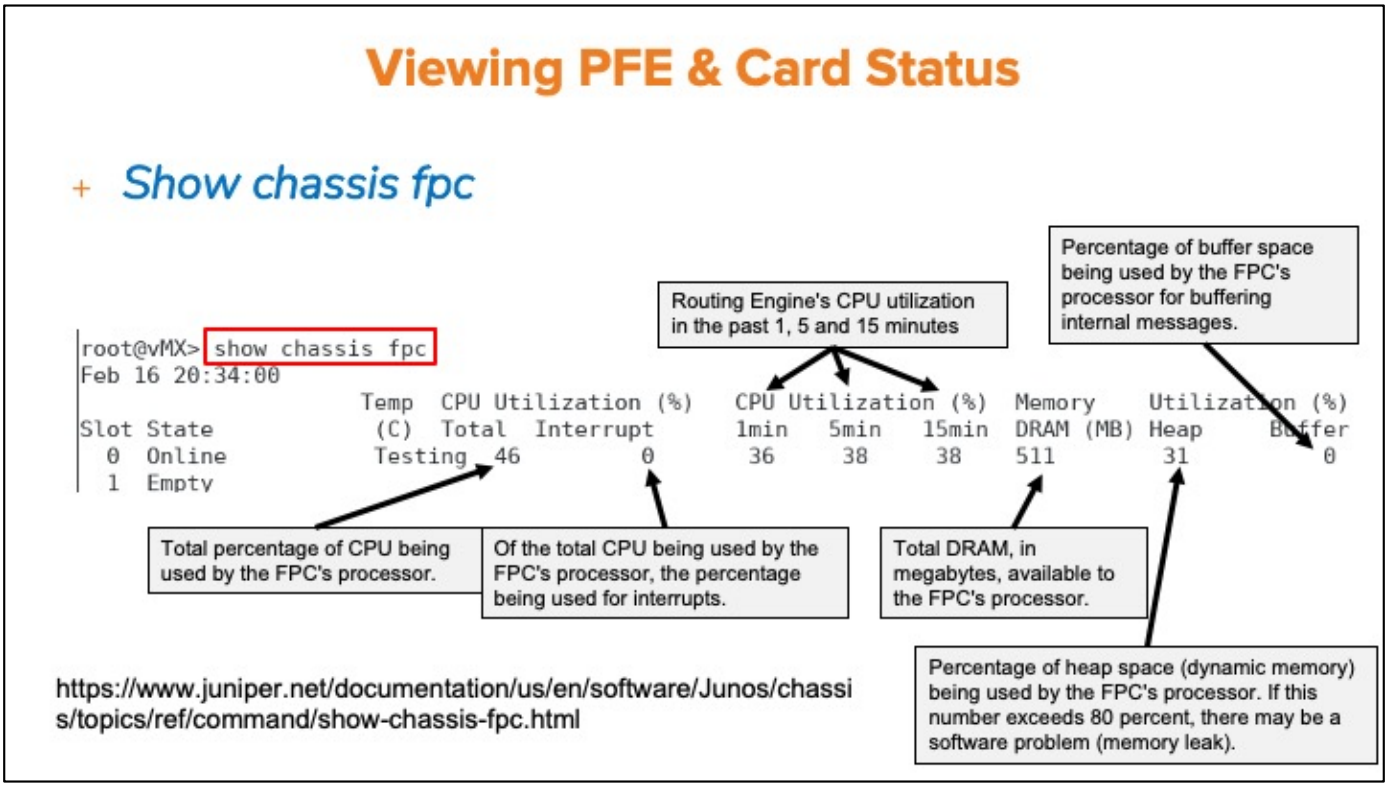


Show chassis craft-interface

- Use this command to see a text representation of the status of various LEDs on your device
- Utilize hardware documentation to explain the purpose of each LED

```
user_1@mx960> show chassis craft-interface
Front Panel System LEDs:
Routing Engine    0    1
-------------------------
OK                *    *
Fail              .    .
Master            *    .

Front Panel Alarm Indicators:
-----------------------------
Red LED       *
Yellow LED    *
Major relay   *
Minor relay   *

Front Panel FPC LEDs:
FPC    0   1   2   3   4   5   6   7   8   9   10   11
-------------------------------------------------------
Red    .   .   .   .   .   .   .   .   .   .   .    .
Green  .   *   .   .   .   .   .   .   .   .   .    *
```

Image courtesy of "Day One: Beginner's Guide to Learning Junos"



In this context, FPC (which technically means Flexible PIC Concentrator) and PFE are synonymous.

Other useful permutations of this command:
show chassis fpc [detail] [pic-status]

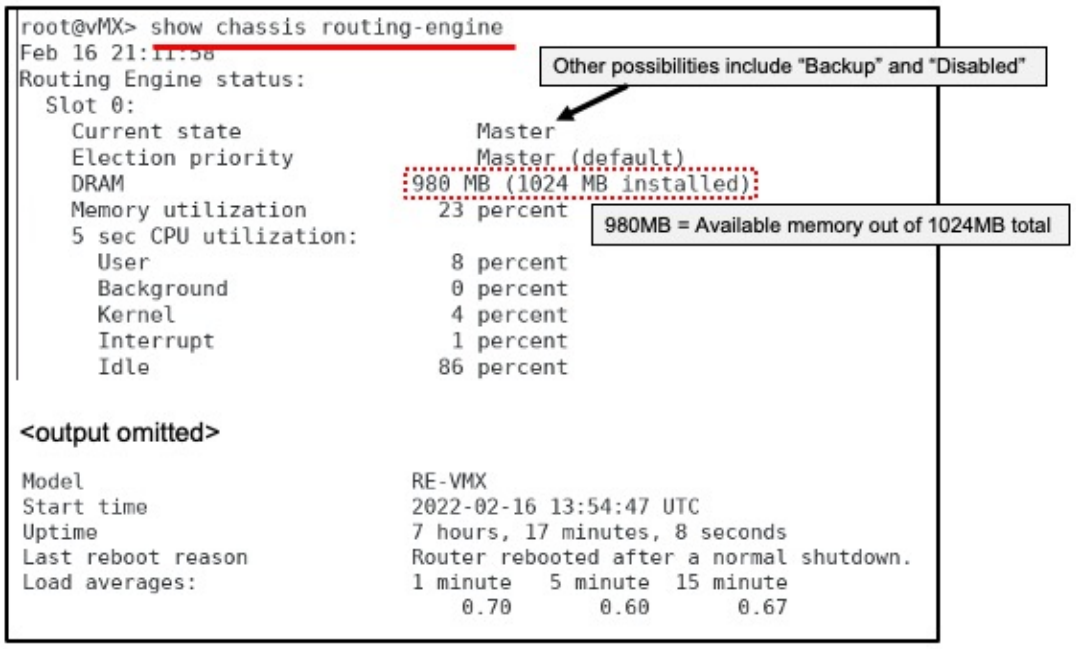
Viewing PIC Status

- PIC = Physical Interface Card

```
user_1@mx960> show chassis fpc pic-status
Apr 27 16:10:42
Slot 1   Online       MPC7E 3D MRATE-12xQSFPP-XGE-XLGE-CGE
  PIC 0  Online       MRATE-6xQSFPP-XGE-XLGE-CGE
  PIC 1  Online       MRATE-6xQSFPP-XGE-XLGE-CGE
Slot 3   Offline      MS-DPC
Slot 11  Online       MPCE Type 2 3D Q
  PIC 0  Online       2x 10GE  XFP
  PIC 1  Online       2x 10GE  XFP
  PIC 2  Online       2x 10GE  XFP
  PIC 3  Online       2x 10GE  XFP
```

Image courtesy of "Day One: Beginner's Guide to Learning Junos"

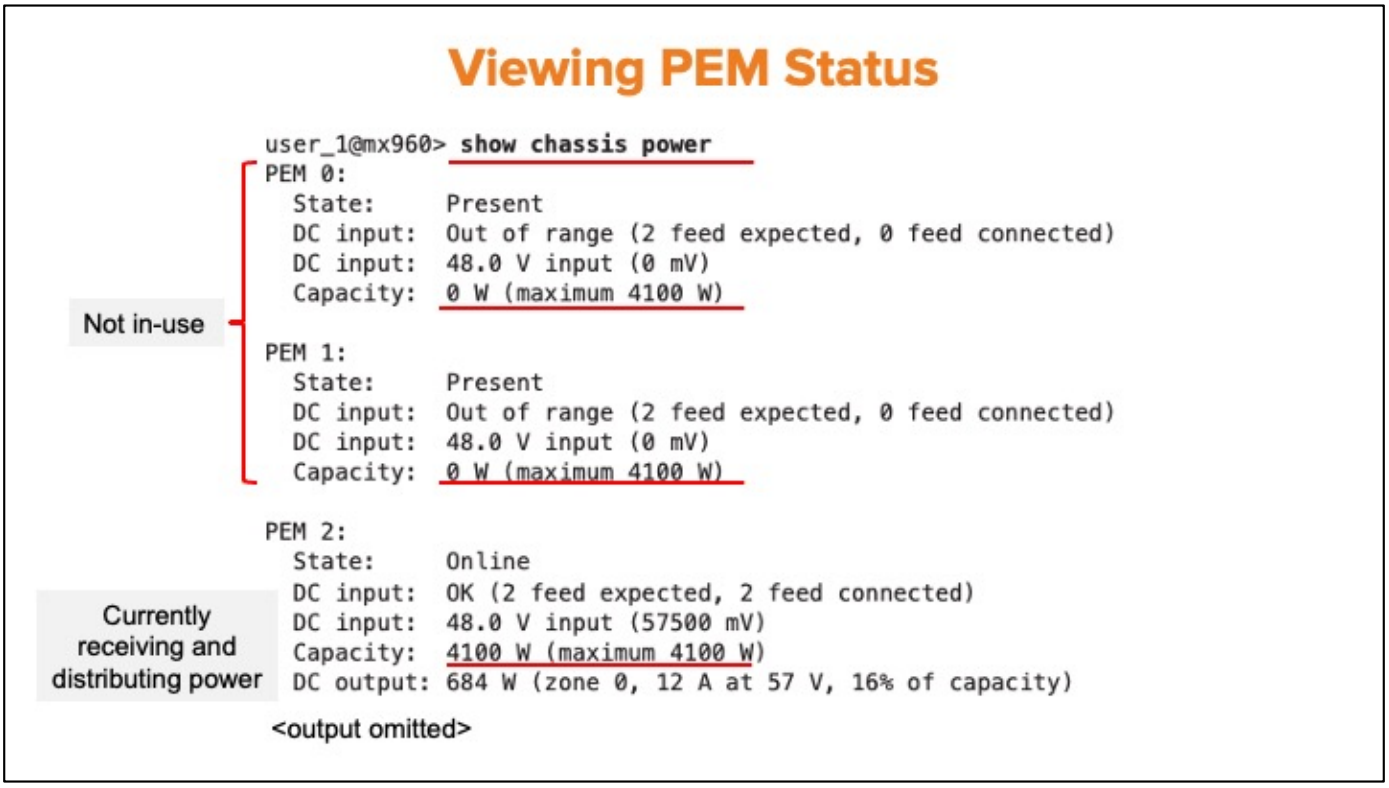
Viewing Routing Engine (RE) Statistics



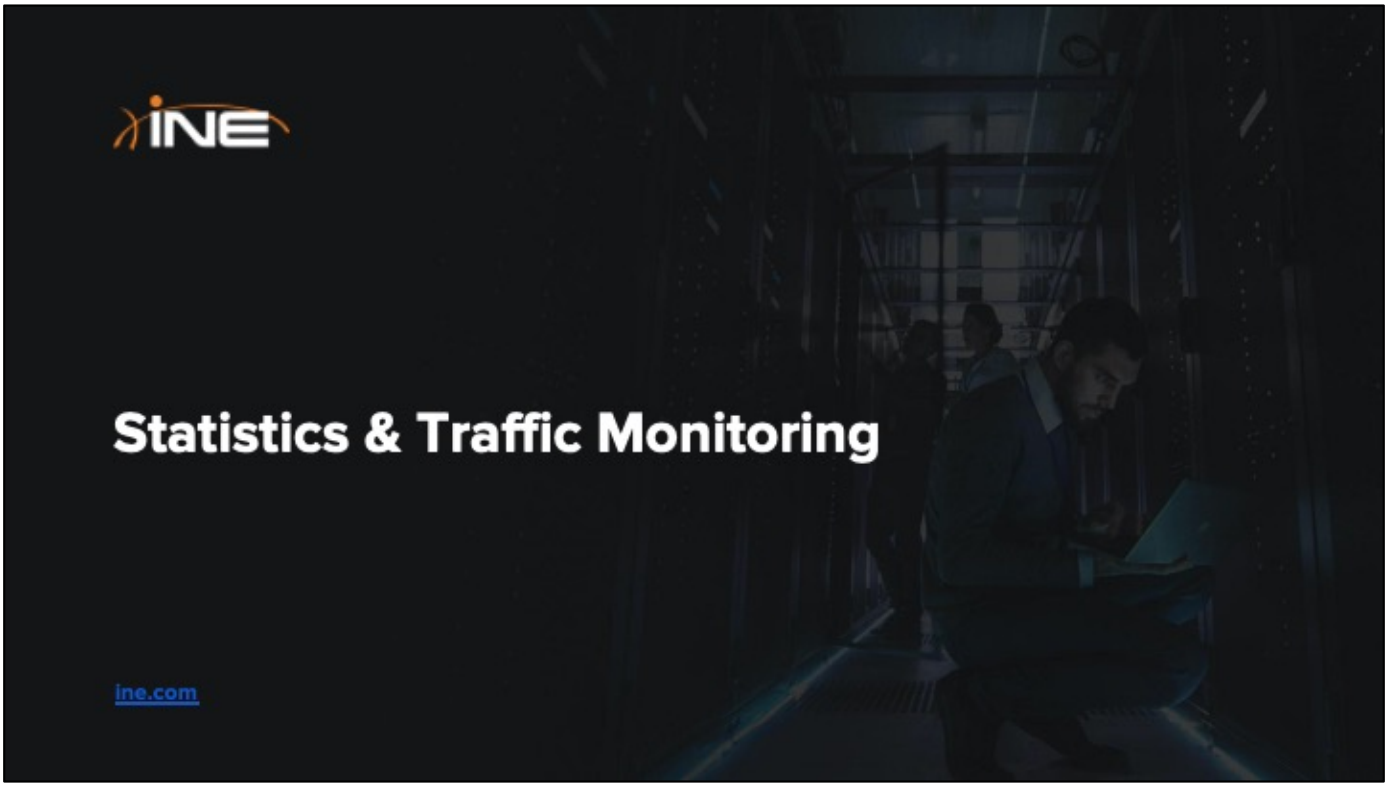
Viewing Chassis Environment

```
user@MX80> show chassis environment
Class Item                          Status     Measurement
Temp PEM 0                          OK
     PEM 1                          OK
     RE 0 Intake                    OK         34 degrees C / 93 degrees F
     RE 0 Front Exhaust             OK         37 degrees C / 98 degrees F
     RE 0 Rear Exhaust              OK         33 degrees C / 91 degrees F
     Routing Engine                 OK         41 degrees C / 105 degrees F
     Routing Engine CPU             OK         50 degrees C / 122 degrees F
     TFEB 0 QX 0 TSen               OK         37 degrees C / 98 degrees F
     TFEB 0 QX 0 Chip               OK         46 degrees C / 114 degrees F
     TFEB 0 LU 0 TSen               OK         37 degrees C / 98 degrees F
     TFEB 0 LU 0 Chip               OK         51 degrees C / 123 degrees F
     TFEB 0 MQ 0 TSen               OK         37 degrees C / 98 degrees F
     TFEB 0 MQ 0 Chip               OK         43 degrees C / 109 degrees F
     TFEB 0 TBB PFE TSen            OK         34 degrees C / 93 degrees F
     TFEB 0 TBB PFE Chip            OK         42 degrees C / 107 degrees F
Fans Fan 1                          OK         Spinning at intermediate-speed
     Fan 2                          OK         Spinning at intermediate-speed
     Fan 3                          OK         Spinning at intermediate-speed
     Fan 4                          OK         Spinning at intermediate-speed
     Fan 5                          OK         Spinning at intermediate-speed
```

PEM = Power Entry Module



Capacity—Actual power input capacity with maximum capacity displayed (in parentheses) in watts.



Topic Overview

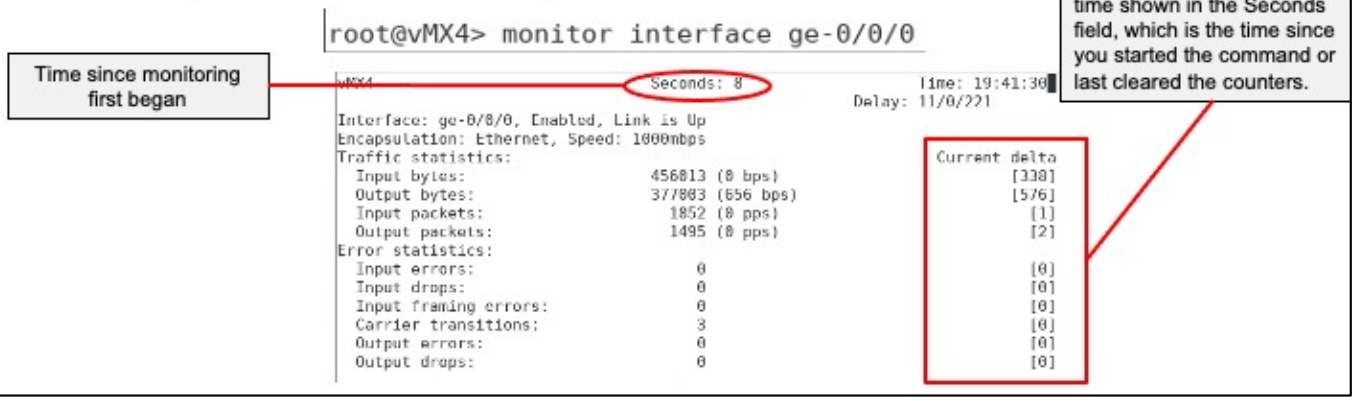
- Real-Time Traffic Statistics Gathering
- Monitoring Traffic Statistics Across Single & Multiple Interfaces

Real-Time Monitoring

- Part of being able to troubleshoot and/or monitor your network device is having visibility to real-time traffic monitoring.
- The Junos “monitor” command can be used to:
  - View real-time traffic statistics on an interface
  - View real-time frames/packets sent to/from the Routing Engine (RE)

Monitoring Interface Statistics

- The command, "*monitor interface <interface-name>*" can be used to view real-time traffic statistics on an interface.
  - Displays traffic statistics (byte and packet counts) as well as error statistics (drops and errors)
  - Updates counters every second



Access to the CLI is removed while this command is in-use. Only after you quit the feature do you regain access to the CLI.

Controlling The Command

- The command “monitor interface” comes with several keyboard shortcuts for controlling it:

```
Next='n', Quit='q' or ESC, Freeze='f', Thaw='t', Clear='c', Interface='i'
```

  - n = moves to the next numbered interface and begins statistical capture there.
  - q = Stops capturing statistics and returns to the CLI prompt
  - f = Pauses the updating of statistics
  - t = Resumes real-time updating of statistics after a freeze
  - c = Restarts the timer and clears the “delta” column values
  - i = allows one to manually switch the statistical capture to a different interface

```
New interface: ge-0/0/9
Next='n', Quit='q' or ESC, Freeze='f', Thaw='t', Clear='c', Interface='i'
```

n = move to the next numbered interface
---If command initiated on physical interface, "n" moves to next physical interface
---If command initiated on a logical interface, "n" moves to next logical interface

Traffic Across All Interfaces

- The command, "*monitor interface traffic*" can be used to get real-time traffic counts across all interfaces
- Allows various toggles for changing the output of each column such as:
  - viewing bytes or packets
  - viewing delta counters since the command was started (or restarted)
- Does NOT divide traffic into logical unit interfaces but collapses all traffic into physical interface statistics

Monitor Interface Traffic

```
root@vMX4> monitor interface traffic
```

Current time

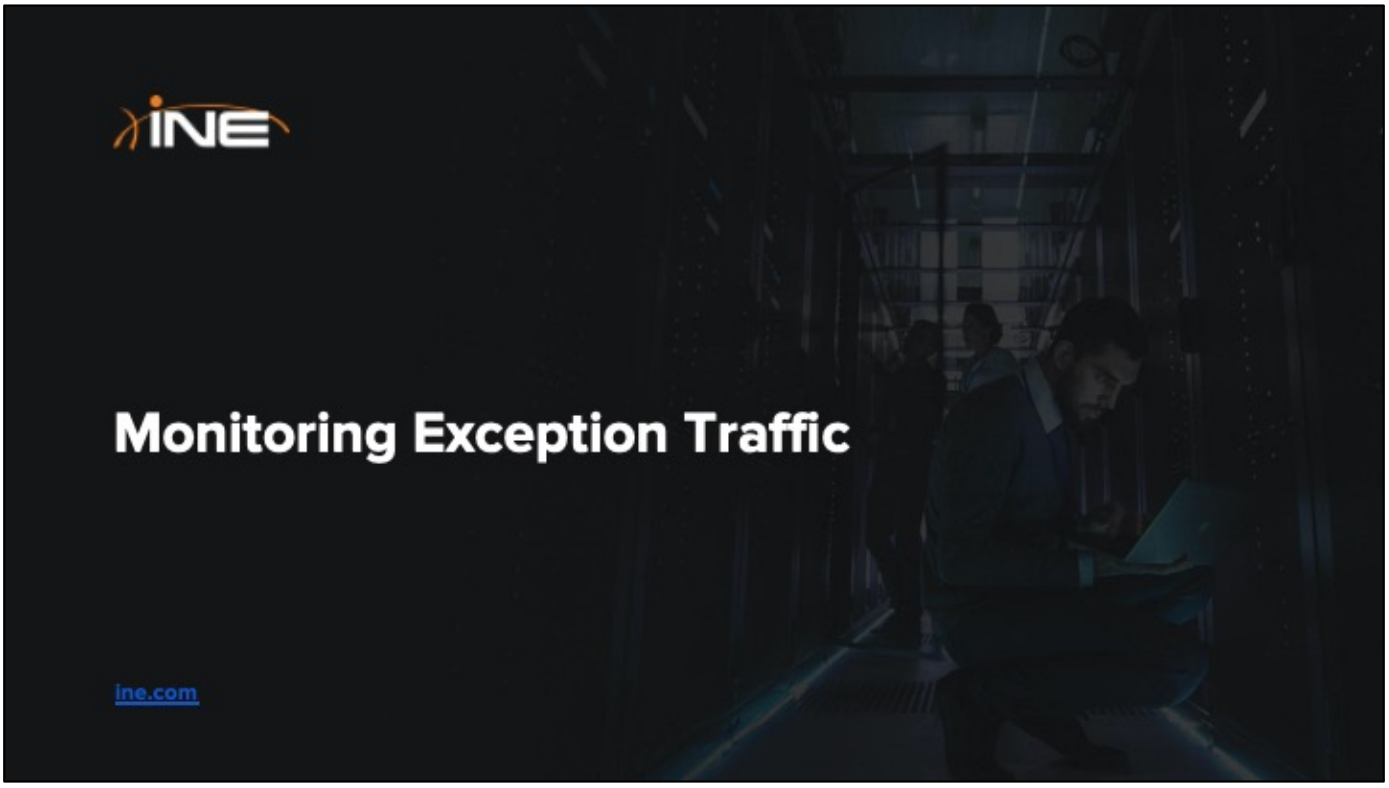
```
vMX4                                Seconds: 56                 Time: 20:46:13

Interface    Link  Input packets        (pps)     Output packets        (pps)
 ge-0/0/0      Up           3222          (0)              2465          (0)
 lc-0/0/0      Up              0                               0
 pfh-0/0/0     Up              0                               0
 ge-0/0/1      Up           5886          (0)              5477          (0)
 ge-0/0/2    Down              0          (0)                 0          (0)
 ge-0/0/3    Down              0          (0)                 0          (0)
 ge-0/0/4    Down              0          (0)                 0          (0)
 ge-0/0/5    Down              0          (0)                 0          (0)
 ge-0/0/6    Down              0          (0)                 0          (0)
 ge-0/0/7    Down              0          (0)                 0          (0)
 ge-0/0/8      Up              0          (0)                 0          (0)
 ge-0/0/9      Up              0          (0)                 0          (0)
 demux0        Up              0                              0
 dsc           Up              0                              0
 eml           Up         534790                         531129
 esi           Up              0                              0
 fti0          Up              0                              0
 fti1          Up              0                              0
```

Controlling The Command

- The command "*monitor interface traffic*" comes with several keyboard shortcuts for controlling it:

```
Bytes=b, Clear=c, Delta=d, Packets=p, Quit=q or ESC, Rate=r, Up=^U, Down=^D
```

  - b = Changes from packet counts to byte counts.
  - c = Restarts the timer and clears the "delta" column values
  - d = Displays "rate" column into a "delta" column
  - p = Changes the Input/Output "bytes" columns into "packets" columns
  - q = Quits the feature and returns the CLI prompt
  - r = Changes the "delta" columns into "pps" columns
  - Up/Down = Displays additional interfaces that don't fit in the screen output



Topic Overview

+ Viewing Exception Traffic With "Monitor Traffic"
+ Filtering Using The "Matching" Keyword

Traffic Debugging

- "*monitor traffic*" is a debugging tool to see exception traffic in real-time
- Only shows exception traffic. Does **not show transit traffic** (need a packet capture for that)
- Comes with various optional keywords to give more detail for output
  - Brief (default)
  - Detail
  - Extensive
- Locks you out of the CLI until you quit the monitoring session

Monitor Traffic

- Juniper documentation states that without specification of an interface the command will default to capturing traffic to/from the Management Interface

```
root@vMX4> monitor traffic
verbose output suppressed, use <detail> or <extensive> for full protocol decode
Address resolution is ON. Use <no-resolve> to avoid any reverse lookup delay.
Address resolution timeout is 4s.
Listening on ge-0/0/0, capture size 96 bytes

21:17:31.280430 Out LLDP, name vMX4, length 60
        [|LLDP]
21:17:36.148005  In LLDP, name Test-Router, length 60
        [|LLDP]
21:17:36.204487 Out LLDP, name vMX4, length 60
        [|LLDP]
Reverse lookup for 4.3.2.8 failed (check DNS reachability).
Other reverse lookup failures will not be reported.
Use <no-resolve> to avoid reverse lookups on IP addresses.

21:17:39.097426  In IP 4.3.2.8 > 224.0.0.5: OSPFv2, Hello, length 56
21:17:40.176099  In IP 4.3.2.8.18261 > 4.3.2.1.telnet: . ack 2593041027 win 4128
21:17:40.176325 Out IP 4.3.2.1.telnet > 4.3.2.8.18261: . ack 1 win 65535
```

On my vMX routers, running in a GNS3 environment it will default to capturing traffic on the lowest numbered physical interface that is up and with an IP address

Controlling The Command

- There are various ways to control what traffic is monitored with the “monitor traffic” command:
  - Match on a specific interface

```
root@vMX4> monitor traffic interface ge-0/0/9
```

  - Increase the level of output detail:

```
root@vMX4> monitor traffic brief
root@vMX4> monitor traffic detail
root@vMX4> monitor traffic extensive
```

  - Include all Layer-2 header fields in output:

```
root@vMX4> monitor traffic layer2-headers
```

The "Matching" keyword

- Often, the "*monitor traffic*" command will display more information than you wish to see.
- The "*matching*" keyword is available to allow filtering of captured traffic.
- Supports regular expressions ***encapsulated with quotes***

Using The “Matching” keyword

- Matching on source or destination IP address:

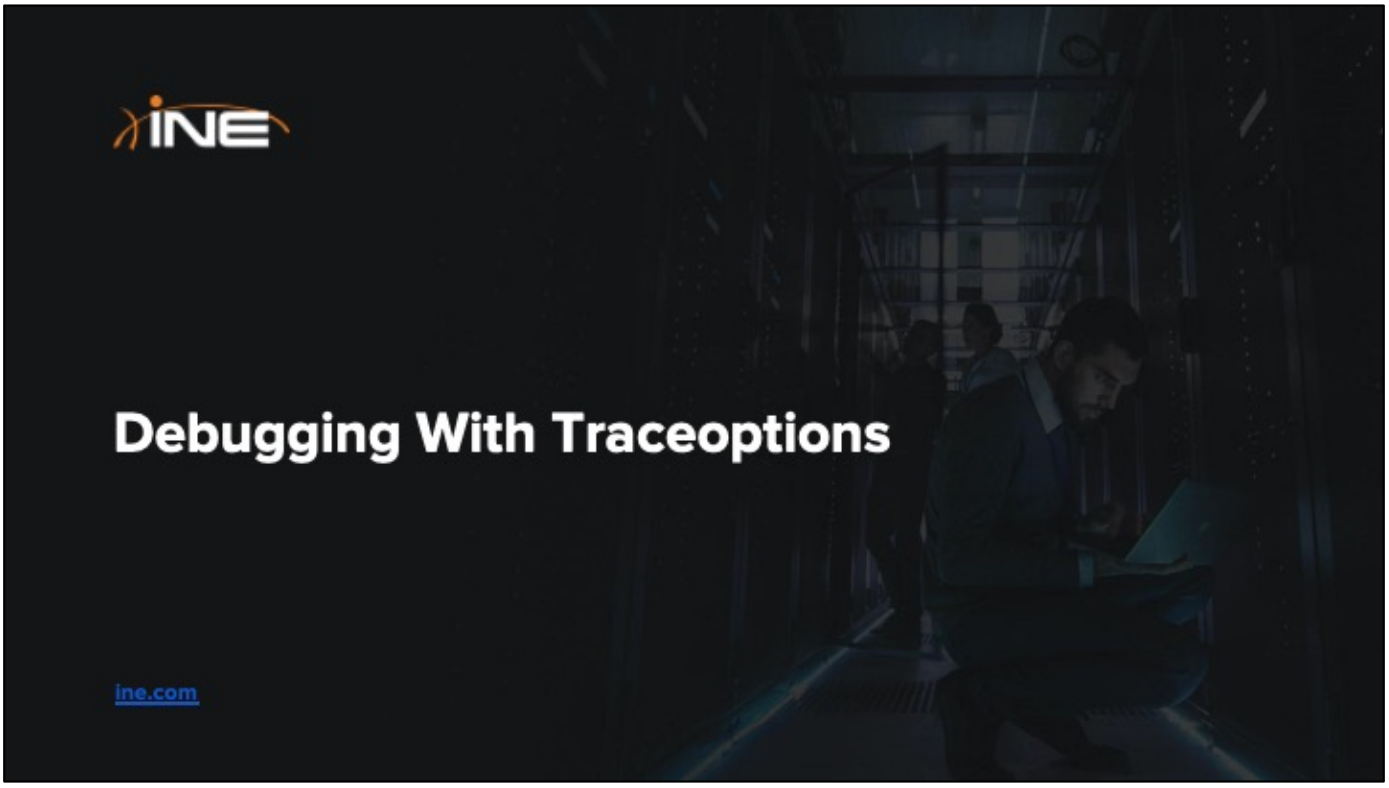
```
root@vMX4> monitor traffic interface ge-0/0/9 matching "host 4.4.4.1"
monitor traffic interface ge-0/0/1 matching "host 2001:1111:2222:3333::4"
monitor traffic interface ge-0/0/9 matching "dst 224.0.0.5"
```

- Matching on IP protocol value:

```
monitor traffic interface ge-0/0/9 matching "ip proto 89"
monitor traffic interface ge-0/0/9 matching "ip proto ospf"
```

- Matching on TCP/UDP port numbers:

```
monitor traffic interface ge-0/0/9 matching "tcp port 22"
monitor traffic interface ge-0/0/9 matching "tcp port ssh"
```



Topic Overview

- Debugging With Traceoptions
- Traceoption Command Syntax
- Traceoption Defaults
- Modifying Traceoption File Usage

Using Juniper Traceoptions

- Traceoptions are the method Juniper uses to enable real-time debugging of protocols
- Increases the CPU load so they should be used sparingly and only for brief durations
- Enabling traceoptions involves
  - Creating a filename where debugs will be stored
  - Specifying one-or-more flags
  - Other optional parameters as needed

Unlike the “monitor traffic” feature which only captures and displays packets sent and received…traceoptions displays events that trigger movement through the FSM (Finite State Machine) of a protocol.

Traceoptions Command Syntax

- Traceoptions are configured within Configuration Mode

Descriptive name of file you are creating

set protocols ospf traceoptions file Keiths-OSPF-Debug

What you wish to capture and send to the traceoptions file

set protocols ospf traceoptions flag hello

CLI help can be used to view available flags

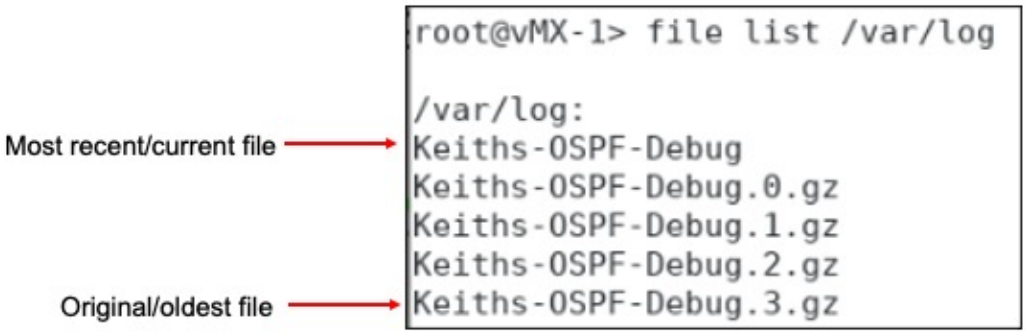
```
root@vMX-1# set protocols ospf traceoptions flag ?
Possible completions:
  all                  Trace everything
  backup-spf           Trace backup SPF (LFA) specific events
  database-description  Trace database description packets
  error                Trace errored packets
  event                Trace OSPF state machine events
```

By default the file you create is stored in the /var/log path.

Traceoptions Defaults

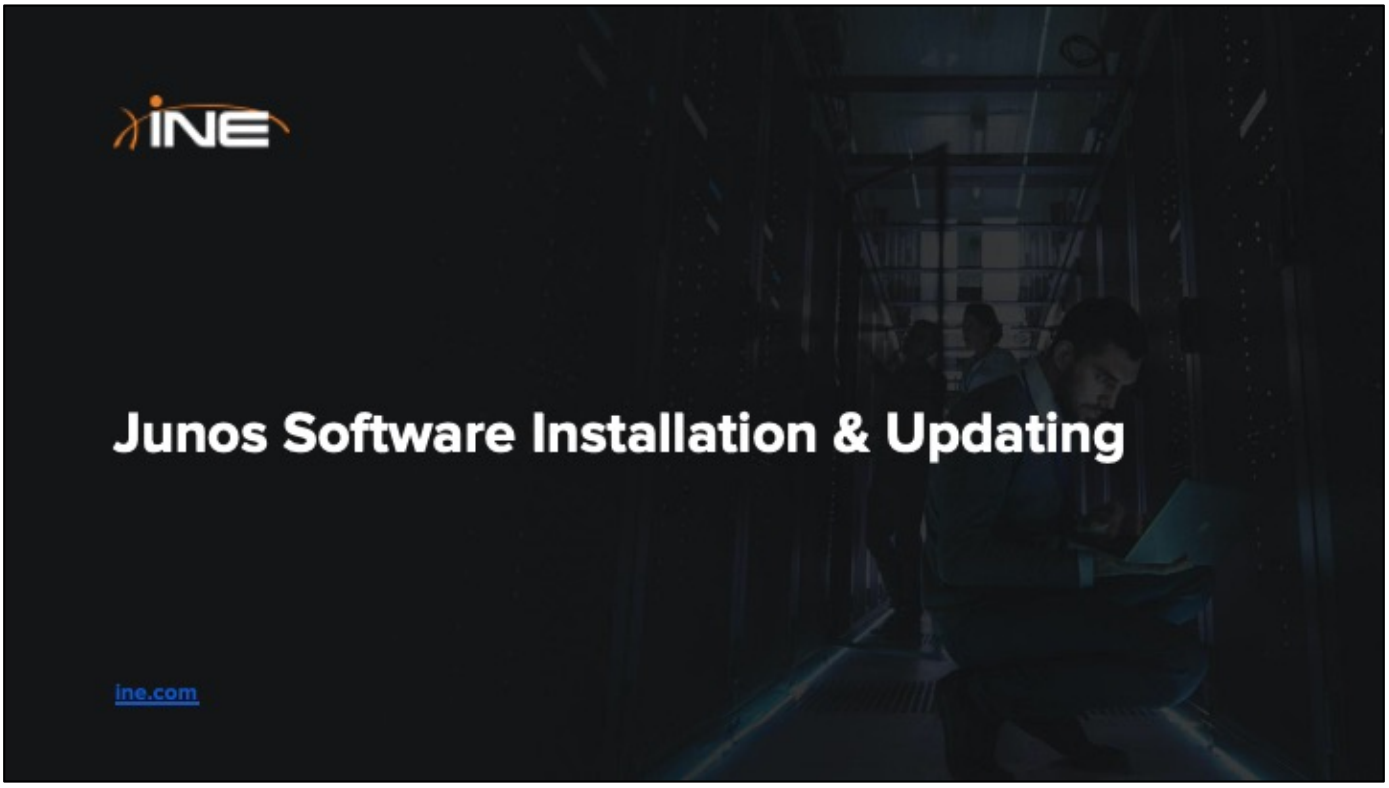
- When you create/name a traceoptions file the following defaults occur:
  - The file is allowed to reach a maximum size of 128KB
  - Once the maximum size is reached:
    - It is renamed <filename>.0
    - A new file is created called <filename>
    - The number after each filename is increased as new files are created.
    - Ten (10) files are created by default
    - Once the maximum number of files is reached, the oldest file is overwritten

Traceoptions Files



Modifying Traceoption File Defaults

- The size of each traceoptions file can be modified:
  - Default is 128KB
  - Configurable range: 10 KB through the maximum file size supported on your system
  - Example: *set protocols ospf traceoptions file size 10k*
- The quantity of files created before the oldest file is overwritten can be modified:
  - Default is 10-files
  - Configurable range: 2 – 1000 files
  - Example: set protocols ospf traceoptions file files 5



Topic Overview

- Overview Of Junos Software Updating
- Junos Installation Media Types
- Contrasting “Install Packages” Vs “Install Media”
- Understanding Junos Package & Release Naming Conventions
- Junos Release Types & Schedule
- Junos Installation Packet Upgrade Process

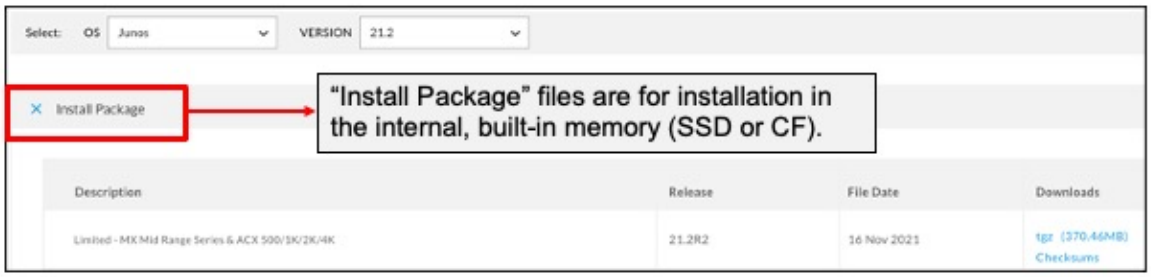
Introduction To Software Upgrading

- Several situations can lead to a desire to update/upgrade your Junos software:
  - Patches for known bugs
  - Enhanced operation of existing features
  - New feature introduction
- Prior to performing a software update, the following things need to be known:
  - What is the appropriate software for your device (name and version)?
  - How much space (i.e. memory) does the new software require?
  - How much free memory does your device contain?

Junos Installation Media

- Juniper devices typically boot software from internal device memory. Either...
  - Internal Compact Flash (*pre 15.1 Junos*) or...
  - An internal SSD (Solid State Drive) (*15.1 and later Junos*)
- Alternatively, one could copy Junos onto a removable USB Flash Drive and boot the device from it.
- Software is downloaded from different sections of Juniper's website depending on if installation will be on the internal memory(CF or SSD)...or a removable USB Flash drive.

Install Package Or Media?



SSD = Solid State Drive
CF = Compact Flash

Junos Package Prefix Names

- Juniper software is downloaded as a "package"
- Junos OS installation packages have the following general pattern:
  - *prefix-release-edition-signed.extension*
- One must find the appropriate "package prefix" that matches your hardware type:

Installation Package Prefix	Description
jinstall*	Junos OS for M Series, MX Series, T Series, TX Matrix, and TX Matrix Plus routers.
jinstall64*	64-bit Junos OS for the JCS1200 Route Reflector, TX Matrix Plus routers with 3D SIBs, and PTX Series Packet Transport Routers.
jinstall-ex*	Junos OS for the EX Series Ethernet Switch portfolio.
jinstall-host-acx5k*	Junos OS for the ACX5000 Series routers.

Full table can be found at: **https://www.juniper.net/documentation/us/en/software/junos/junos-install-upgrade/topics/topic-map/software-install-and-upgrade-overview.html#id-junos-os-installation-package-names**

In a traditional sense, a **software package** is simply multiple applications or code modules that work together to meet various goals and objectives.

Release Naming Conventions

- Junos OS "release" naming convention:

  m.nZb.s

The software release number 17.2R1.13, for example, maps to this format as follows:

- *m* is the main release number of the product, for example, 17.
- *n* is the minor release number of the product, for example, 2.
- *Z* is the type of software release, for example, R for FRS or maintenance release.
  For types of software releases, see Table 2. [See next slide for "Table 2"]
- *b* is the build number of the product, for example, 1, indicating the FRS rather than a maintenance release..
- *s* is the spin number of the product, for example, 13.

https://www.juniper.net/documentation/us/en/software/junos/junos-install-upgrade/topics/topic-map/software-install-and-upgrade-overview.html

Junos Software Release Types

Table 2: Software Release Types

Release Type	Description
R	First revenue ship (FRS) or maintenance release software. R1 is FRS. R2 onward are maintenance releases.
F	Feature velocity release. Feature velocity releases are only in Junos OS Release 15.1.
B	Beta release software.
I	Internal release software. These are private software releases for verifying fixes.
S	Service release software, which are released to customers to solve a specific problem—this release will be maintained along with the life span of the underlying release. The service release number is added after the R number, for example, 14.2R3-S4.4. Here S4 represents the 4th service release on top of 14.2R3 and is the 4th re-spin.
X	Special (eXception) release software. X releases follow a numbering system that differs from the standard release numbering. Starting with Junos OS Release 12.1X44-D10, SRX Series devices follow a special naming convention for Junos OS releases. For more information, refer to the Knowledge Base article KB30092 at https://kb.juniper.net/InfoCenter/index?page=home.

https://www.juniper.net/documentation/us/en/software/junos/junos-install-upgrade/topics/topic-map/software-install-and-upgrade-overview.html

Edition Naming Conventions

- Junos OS installation packages have the following general pattern:
  - *prefix-release-edition-signed.extension*
- Prior to Junos release 15.1 there were three types of "editions"
  - Domestic
  - Limited
  - FIPS
- Starting with 15.1 the editions were simplified
  - Junos OS with a null (empty) edition field is the standard image for Junos OS
  - Limited: no cryptographic support and intended for countries in the Eurasian Customs Union (EACU)

Domestic: Junos OS for customers in the United States and Canada and for all other customers with a valid encryption agreement. This edition includes high-encryption capabilities such as IPsec and SSH for data leaving the router or switch. Later images use a null, or empty, edition field for this category.

Limited: Junos OS for all other customers. This edition does not include any high-encryption capabilities for data leaving the router or switch. Sometimes referred to as the *Export* edition, starting in Junos OS Release 15.1R1, this category is is renamed to the limited edition.

FIPS: Junos OS that provides advanced network security for customers who need software tools to configure a network of Juniper Networks routers and switches in a Federal Information Processing Standards (FIPS) 140-2 environment. For more information about Junos-FIPS, see FIPS 140-2 Security Compliance. In later images, FIPS, instead of being a separate edition, is an option you select on installation.

Release Schedule

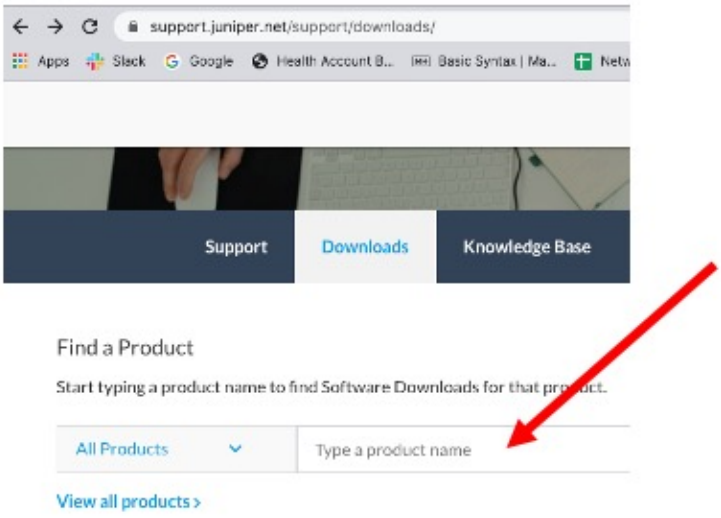
- Junos OS is typically released every quarter
- Releases designated as:
  - Major release (Junos 20.x)
    - Remains consistent across a calendar year
  - Minor release (Junos 20.2)
    - Indicates which quarter the software was released

Bug Fixes

- Some types of Junos OS versions are released more frequently to resolve issues;
  - Maintenance releases
    - Released about every 6-weeks
    - Fix a collection of issues
    - Prefaced with an "R" (14.1**R**2) = second maintenance release for 14.1
  - Service Releases
    - Released on-demand
    - Fix critical issues
    - Prefixed with an "S" (14.1S2)
- New features are released for every major and minor release
- Bug fixes are added to Service and Maintenance releases

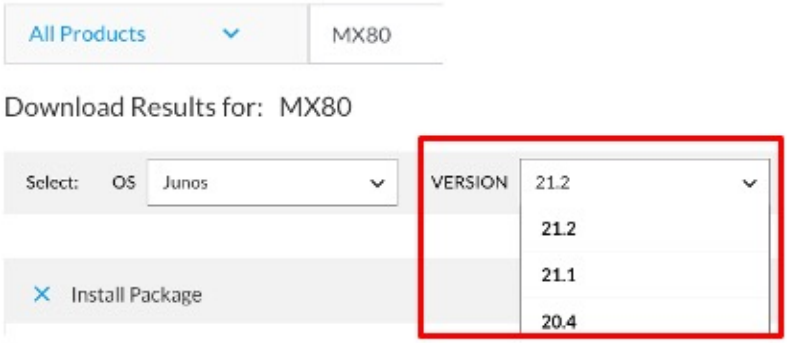
Identifying Memory Requirements

- Start by going to Juniper's Software Download site and enter your product name



Identifying Memory Requirements

+ Select the software release for your hardware



Determining Memory Requirements

+ Determine the amount of memory required to install the new software

Verifying Available Memory

- Use the command, "*show system storage*" to view available memory

System running pre-15.1 Junos OS

```
labroot@optimus-re0> show system storage
Filesystem             Size       Used      Avail  Capacity   Mounted on
/dev/ad0s1a            3.4G       623M      2.5G       20%    /
devfs                  1.0K       1.0K        0B      100%    /dev
/dev/md0               81M        81M         0B      100%    /packages/mnt/jbase
/dev/md1               102M       102M        0B      100%    /packages/mnt/jkernel64-15.1R5-S3.4
```

System running 15.1 or later Junos OS

```
labroot@PE1> show system storage
Filesystem             Size       Used      Avail  Capacity   Mounted on
/dev/md0.uzip          22M        22M         0B      100%    /
devfs                  1.0K       1.0K        0B      100%    /dev
/dev/gpt/junos         21G        11G       8.0G       59%    /.mount
devfs                  1.0K       1.0K        0B      100%    /.mount/dev
/dev/md1.uzip          24M        24M         0B      100%    /.mount/packages/mnt/os-libs-11
```

Starting with Junos OS 15.1, FreeBSD (the underlying host OS) was upgraded from FreeBSD 6.1 to FreeBSD 10. Junos OS with upgraded FreeBSD 10 has two volumes: dev/gpt/Junos and dev/gpt/oam. The /Junos volume refers to SSD and the /oam volume refers to CF.

In Junos OS with upgraded FreeBSD 10, compact flash (CF) is not used to boot a device anymore (dev/gpt/oam). Only if SSD fails will the device boot from CF, that is, SSD and CF will not be used (or displayed in the output) simultaneously.

Cleaning Up Memory

- If you have unnecessary files in memory they can be deleted to free up space.
- Use the command, "request *system storage cleanup*"
  - You can add the option of "*dry-run*" to see what files would be deleted without deleting anything.

```
root@vMX-1> request system storage cleanup dry-run

List of files to delete:

         Size Date         Name
  3767B Feb 23 16:32 /var/log/dfcd_enc.0.gz
  3463B Feb 15 15:51 /var/log/dfcd_enc.1.gz
  3591B Feb 14 16:12 /var/log/dfcd_enc.2.gz
   119B Feb 11 16:02 /var/log/dfcd_enc.3.gz
   311B Jan 27 16:30 /var/log/dfcd_enc.4.gz
   727B Jan 27 16:26 /var/log/dfcd_enc.5.gz
  12.3K Feb 23 16:49 /var/log/interactive-commands.0.gz
```

Overview Of The Upgrade Process

- This is the process to follow when upgrading (or downgrading) your Junos OS:
  - Back up the system, including the configuration.
  - Copy the software installation package to your device
  - Use the CLI to install the new software on the device.
  - Reboot the device and boot from the newly installed software
  - Back up the new software and configuration.

Creating A Backup

- There are several benefits for backing-up your files prior to a software upgrade/downgrade:
  - The device can boot from a backup and come back online in case a component fails or a power failure during an upgrade corrupts the primary boot device.
  - The backup copy of the system saves your active configuration files and log files.
  - The device can recover from a known, stable environment in case of an unsuccessful upgrade.
- Use the Operational Mode command, "*request system snapshot*" to automatically create backup files.

Request System Snapshot

```
user@host> request system snapshot
umount: /altroot: not currently mounted
Copying / to /altroot.. (this may take a few minutes)
umount: /altconfig: not currently mounted
Copying /config to /altconfig.. (this may take a few minutes)

The following filesystems were archived: / /config
```

The “/root” filesystem contains your currently installed version of Junos (among other things) and is copied to /altroot.
The “/config” filesystem contains your Active Config file as well as some of your most recent Rollback configs. This is copied to /altconfig.

Software Download Options

- There are three ways for your Juniper device to retrieve the software install package:
  - Create a temporary URL on the Juniper website and have the device (router, switch, etc) connect directly to the URL and download the package
  - Download the package to a computer directly connected to the Juniper device
  - Download the package to a remote host (i.e Server) and use the Juniper CLI to retrieve the package from that remote host

https://www.juniper.net/documentation/en_US/junos/topics/topic-map/software-download-nfx.html

Downloading The Software Install Package

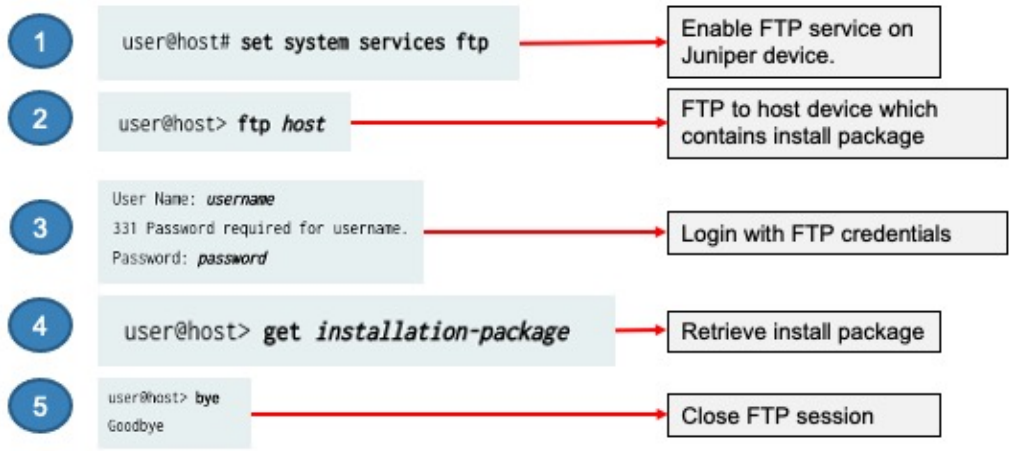
- Install package files can be transferred to the Juniper Router, Switch, etc using any of the following methods:
  - SCP/SSH
  - TFTP
  - FTP
  - USB
  - Directly from Juniper.net via HTTPS

```
user@host> file copy "https://cdn.juniper.net/software/ittest/software_target/agileEcotTest/Dev_Binary_Build.tar?
SM_USER=user1=1507622971_dce164fa854b4a27550c254eef950dd8" /var/tmp
```

Example of using HTTPS and a temporary URL to download install package from Juniper.net

FTP Package Download Steps

- Once the appropriate install package has been downloaded to your local host (i.e. Laptop, PC, etc) you can use FTP to transfer it to your Juniper device:



Ensure that your software package is downloaded to the /var/tmp directory. If you're not sure…login to the shell…change directory to /var/tmp and then log back into the CLI and perform the steps above.

Package Validation

- Once the software has been copied to your Juniper device you can (optionally) validate it to ensure compatibility with your current Active Configuration, hardware, etc
  - Use the Operational Mode command:
    request system software validate <path/image-name>
- What does it do?
  - Initializes instances of the daemons from the new version of code
  - Checks current configuration, etc. against daemons
  - Doesn't perform any major functions (i.e. routing).

Package Validation

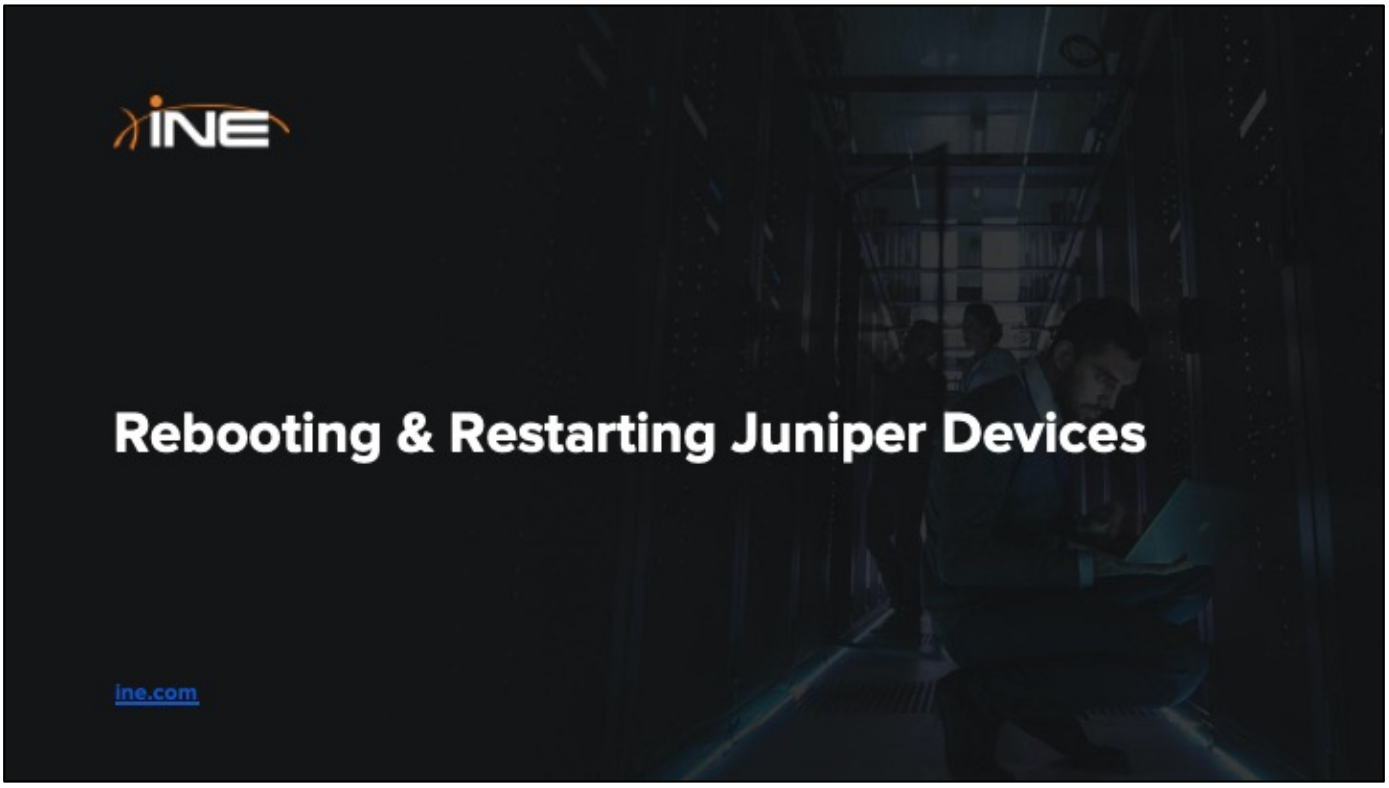
```
user@host> request system software validate /var/sw/pkg/jbundle-5.3I20020124_0520_sjg.tgz
Checking compatibility with configuration
Initializing...
Using /packages/jbase-5.3I20020122_1901_sjg
Using /var/sw/pkg/jbundle-5.3I20020124_0520_sjg.tgz
Using /var/chroot/var/tmp/jbundle/jbase-5.3I20020124_0520_sjg.tgz
Using /var/chroot/var/tmp/jbundle/jkernel-5.3I20020124_0520_sjg.tgz
Using /var/chroot/var/tmp/jbundle/jcrypto-5.3I20020124_0520_sjg.tgz
Using /var/chroot/var/tmp/jbundle/jpfe-5.3I20020124_0520_sjg.tgz
Using /var/chroot/var/tmp/jbundle/jdocs-5.3I20020124_0520_sjg.tgz
Using /var/chroot/var/tmp/jbundle/jroute-5.3I20020124_0520_sjg.tgz
Validating against /config/juniper.conf.gz
mgd: commit complete
```

Software Package Installation

- To install the software package that has already been downloaded to your device use the following command:

```
root@vMX-1> request system software add /var/tmp/<package-name.tgz> reboot
```

  - If you do not specify “reboot” the system will install the software the next time you reboot the device.



Topic Overview

+ Methods For Rebooting Juniper Hardware

Options For Reboots

- There are several situations in which you may want to stop the functioning of your Juniper device
  - Replace faulty equipment
  - Move chassis to different cage or datacenter
- You have the following options available:
  - Halt
  - Power-Off
  - Reboot

Halting The System

- root@vMX1>*request system halt*
  - Stops all processes on the device
  - Stops the forwarding of traffic
  - Keeps the console port active
  - Waits for a keypress to reboot the box

```
root@vMX-1> request system halt
Halt the system ? [yes,no] (no) yes


*** FINAL System shutdown message from root@vMX-1 ***

System going down IMMEDIATELY


Waiting (max 60 seconds) for system process `vnlru' to stop... done
Waiting (max 60 seconds) for system process `bufdaemon' to stop... done
Waiting (max 60 seconds) for system process `syncer' to stop...
Syncing disks, vnodes remaining... 0 0 0 done
All buffers synced.
Uptime: 3h49m26s
Khelp module "jsocket" can't unload until its refcount drops from 7 to 0.

The operating system has halted.
Please press any key to reboot.
```

Turning Off The Power

- You can remotely power-off your system with the following command:
  - root@vMX1>request system power-off
  - Requires physical access to the device to restore power

Additional options include…

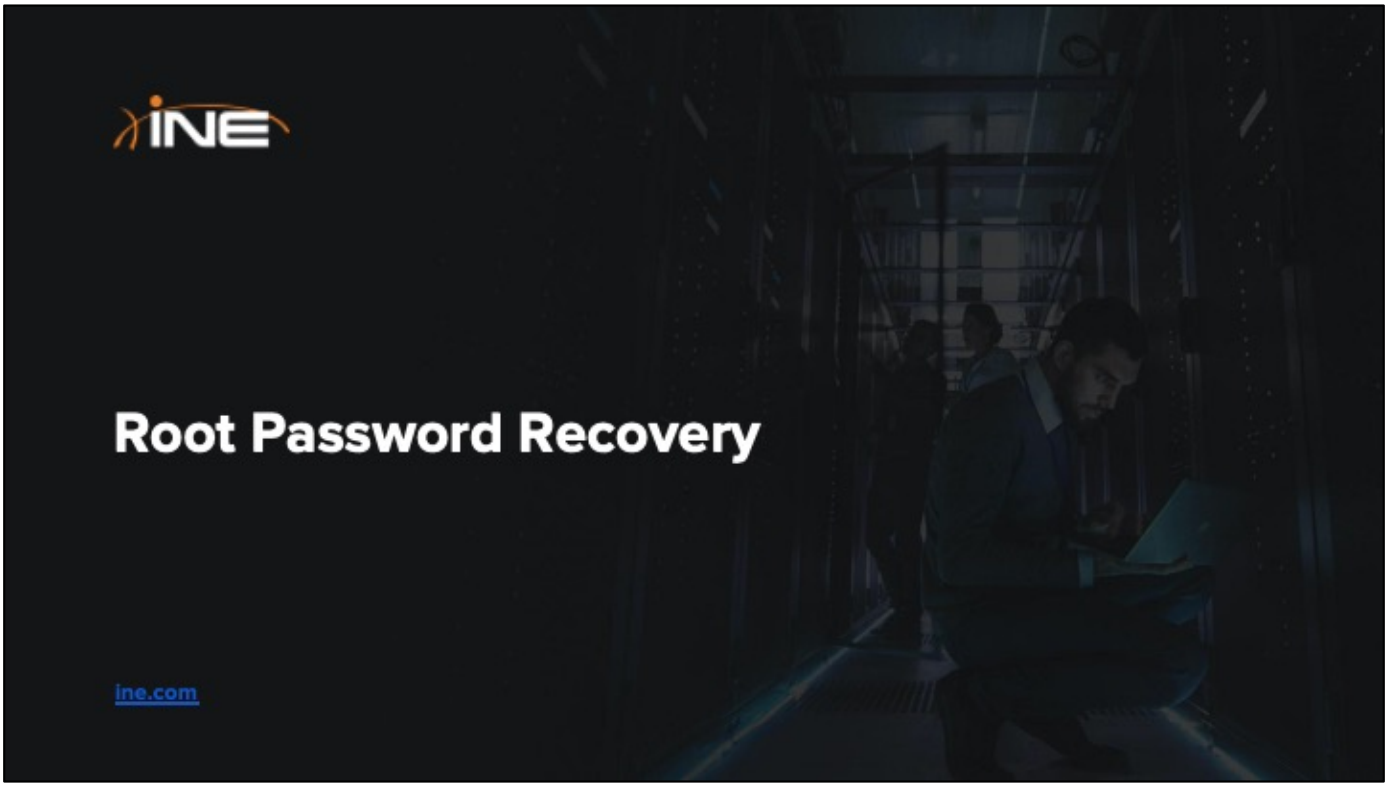
```
root> request system power-off ?
Possible completions:
  <[Enter]>            Execute this command
  at                   Time at which to perform the operation
  both-routing-engines  Power off both Routing Engines
  in                   Number of minutes to delay before operation
  junos                Boot off Junos volume
  message              Message to display to all users
  network              Network boot via PXE
  oam                  Boot off OAM volume
  other-routing-engine  Power off other Routing Engine
  usb                  Boot off USB device
  |                    Pipe through a command
```

Rebooting The Device

- To reboot your device, use the following command:

```
root> request system reboot ?
Possible completions:
  <[Enter]>            Execute this command
  at                   Time at which to perform the operation
  both-routing-engines  Reboot both the Routing Engines
  in                   Number of minutes to delay before operation
  junos                Boot off Junos volume
  message              Message to display to all users
  network              Network boot via PXE
  oam                  Boot off OAM volume
  other-routing-engine  Reboot the other Routing Engine
  usb                  Boot off USB device
  |                    Pipe through a command
```

- Use "*clear system reboot*" to cancel any scheduled reboots.



Topic Overview

+ Junos Root Password Recovery Process

Root Password Recovery Overview

- At minimum, a Juniper device requires a “root” authentication password before any other configuration is applied.
- If no other users designated within the class “Super-User” exist, “root” will be the only one able to commit changes to the config.
- But what if the Root user password is lost/forgotten?

Root Password Recovery Process

- You can follow this process to recover your Root password (Junos using FreeBSD 6.1):
  1. Connect to the device's console port
  2. Powercycle the device
  3. Look for the following prompt

```
Hit [Enter] to boot immediately, or space bar for command prompt.
Booting [kernel] in 9 seconds...
```

  4. Press the spacebar quickly (you may have to do it several times)
  5. At the "ok" prompt type "*boot -s*"
  6. At the following prompt, type the word "*recovery*"

```
Enter full pathname of shell or 'recovery' for root password recovery or RETURN for /bin/sh: recovery
```

Junos 15.1 and more recent uses an upgraded version of FreeBSD (version 10)

Root Password Recovery Process

- (password recovery continued)
  7. Enter Configuration Mode from within the CLI
  8. Configure a new Root password

```
[edit]
user@host# set system root-authentication plain-text-password
```

  9. Commit the configuration
  10. Exit Configuration Mode
  11. Exit Operational Mode
  12. Type “y” to reboot the router

```
Reboot the system? [y/n] y
```

Root Password Recovery Process

- You can follow this process to recover your Root password (Junos using FreeBSD 10):
  1. Connect to the device's console port
  2. Powercycle the device
  3. Look for the following "autoboot" prompt

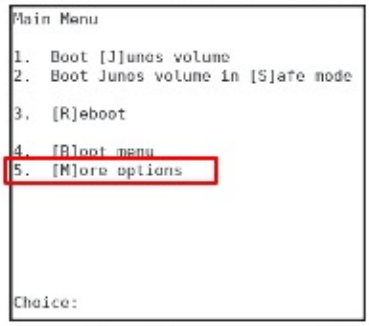
```
FreeBSD/x86 bootstrap loader, Revision 1.1
(builder@feyrith.juniper.net, Sun Feb  4 13:06:24 PST 2018)
/
Autoboot in 1 seconds... (press Ctrl-C to interrupt)
```

  4. Press Ctrl-C to interrupt normal bootup and display the Junos Main Menu

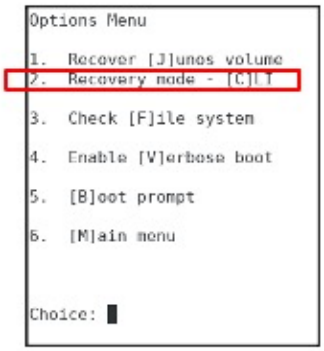
Junos 15.1 and more recent uses an upgraded version of FreeBSD (version 10)

Root Password Recovery Process

5. Select option-5 (More Options) from the Main Menu



6. Select option-2 to enter Recovery Mode



Root Password Recovery Process

- (password recovery continued)
  7. Enter Configuration Mode from within the CLI
  8. Configure a new Root password

```
[edit]
user@host# set system root-authentication plain-text-password
```

  9. Commit the configuration