Cisco 2 JunOS

2022. 06. 17.

1

Course Contents

- Tartalom:
  - JunOS operációs rendszer alapjai
  - CLI felhasználói interfész sajátosságai
  - Kezdeti konfiguráció
  - Alap konfigurációs beállítások
  - Másodlagos beállítások
  - Hardver információk
  - Liszenszelési információk

2

2

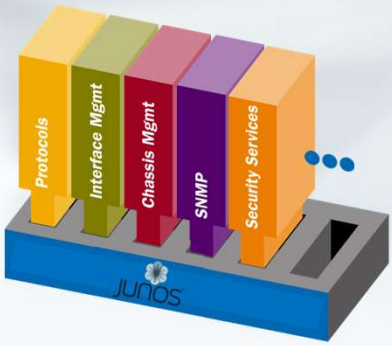
JunOS operációs rendszer alapjai

3

3

The Junos OS

- Robust, modular operating system
  - Provides industry-leading performance and scalability
  - Built on open-source operating systems



4

4

Single Software Train (1 of 3)

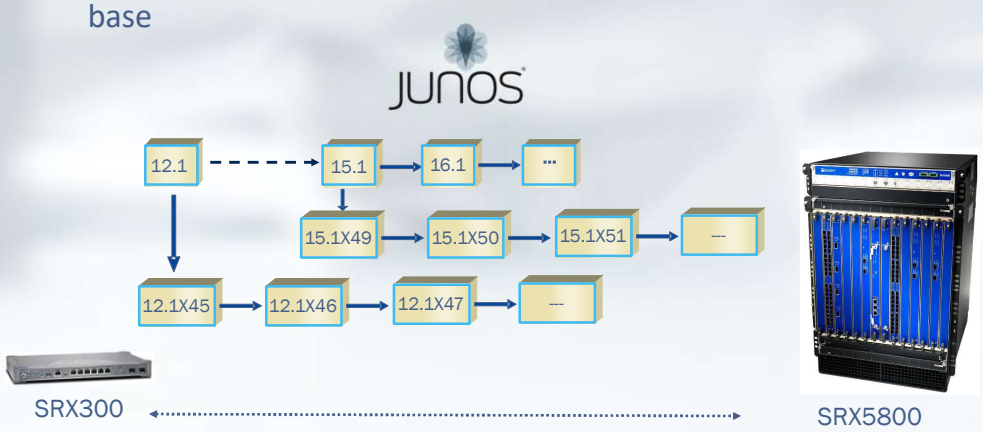
- A single software train for all platforms running the Junos OS
  - Eases management overhead by providing a consistent set of features that are implemented in a consistent manner

5

5

Single Software Train (2 of 3)

- X Releases
  - Use a code base that is updated at a slower pace
  - New features introduced at a faster cadence within that code base

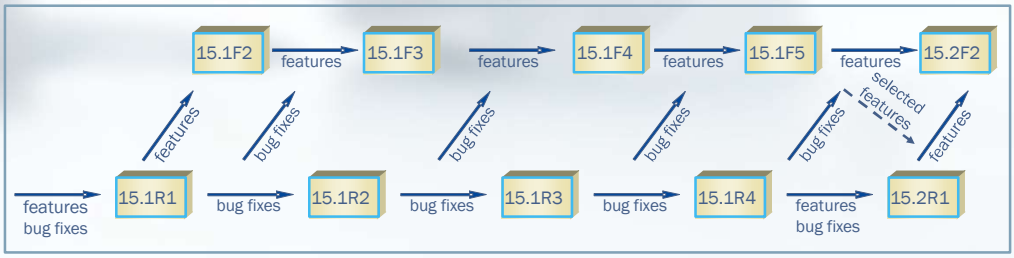


6

6

Single Software Train (3 of 3)

- R and F Releases
  - Within a main line version there are maintenance ("R") and feature velocity ("F") releases.
  - R releases deliver bug fixes only. F releases release new features and content as well as bug fixes within a single version of Junos.
  - F releases started with version 15.1 of Junos.

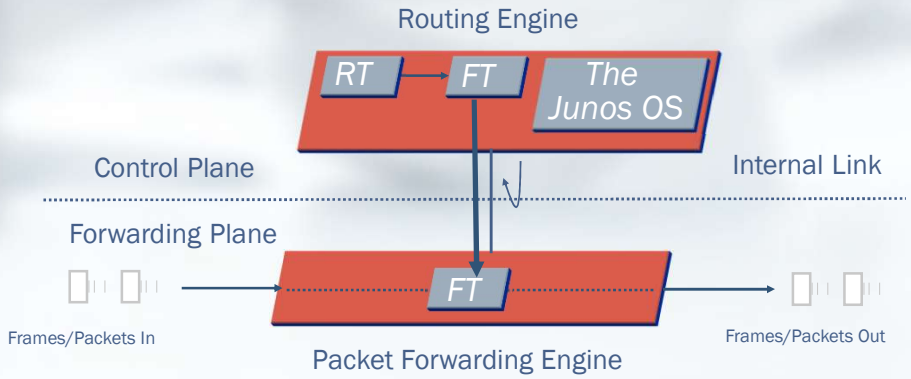


7

7

Separation of Control and Forwarding

- All platforms running the Junos OS share a common design goal:
  - Clean separation of control and forwarding functions

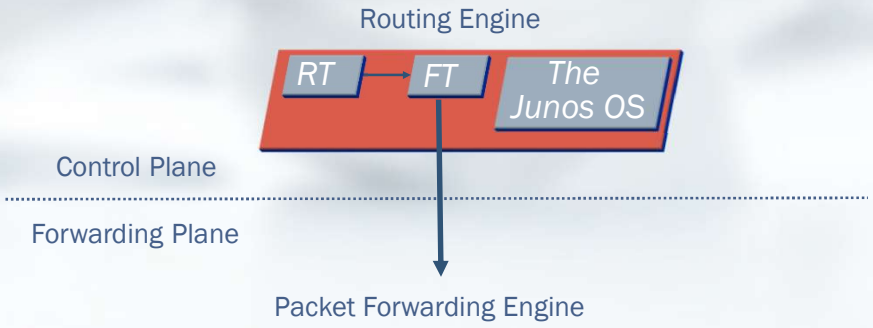


8

8

Routing Engine

- Maintains routing and forwarding tables
- Controls and monitors the chassis
- Manages the PFE

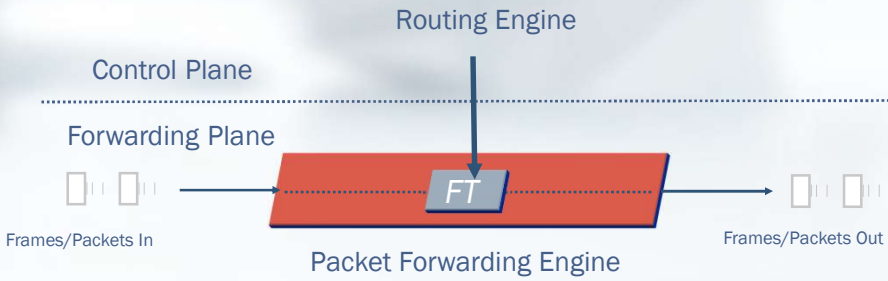


9

9

Packet Forwarding Engine

- Uses Layer 2 and Layer 3 forwarding tables, provided by the RE, to forward traffic toward its destination
- Implements various services such as policing, stateless firewall filtering, and class of service

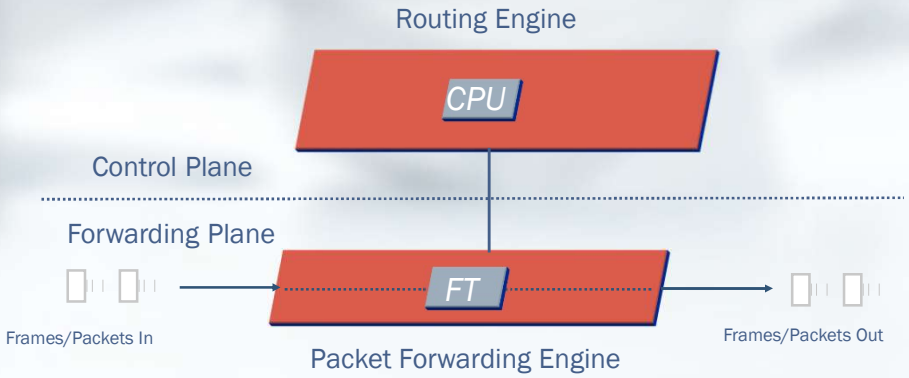


10

10

Transit Traffic Processing

- Transit traffic is forwarded through the local system
  - PFE uses the forwarding table provided by the RE
  - Examples of transit traffic include unicast and multicast traffic

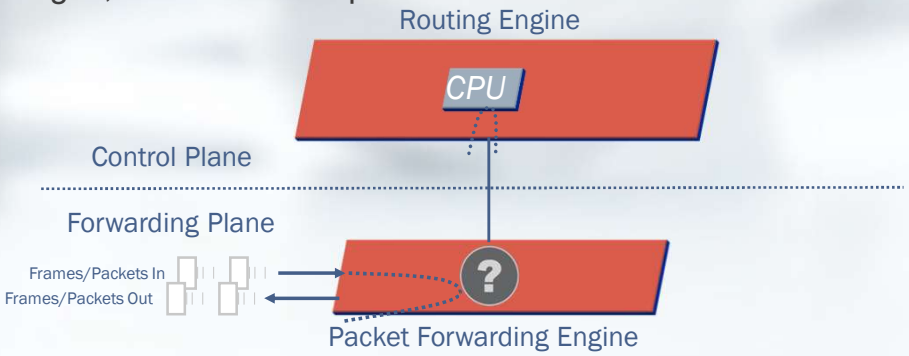


11

11

Exception Traffic Processing (1 of 2)

- Exception traffic is processed by the local system
  - Traffic destined for the local system is processed by RE CPU
  - In most cases, the PFE processes traffic requiring the generation of ICMP messages, such as TTL expired

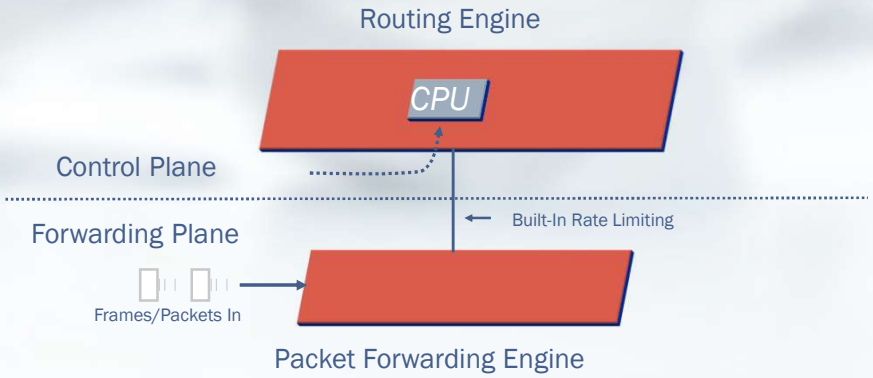


12

12

Exception Traffic Processing (2 of 2)

- Exception traffic is rate-limited on the internal link to protect the RE from potential DoS attacks
  - Control traffic is given preference when congestion exists



13

13

CLI felhasználói interfész sajátosságai

14

14

- CLI alapok
- Operational mód, Configuration mód áttekintése
- Active vs. Candidate konfiguráció közti különbségek áttekintése
- Commit lehetőségek átnézése

15

15

- CLI alapok
- Operational mód, Configuration mód áttekintése
- Active vs. Candidate konfiguráció közti különbségek áttekintése
- Commit lehetőségek átnézése

16

16

Logging In

- When logging in:
  - Nonroot users are placed into the CLI automatically

```
router (ttyu0)

login: user
Password:

--- JUNOS 15.1X49-D100.6 built 2017-06-28 07:33:31 UTC
user@router>
```

  - The root user must start the CLI from the shell
    - Remember to exit the root shell after logging out of the CLI!

```
router (ttyu0)

login: root
Password:

--- JUNOS 15.1X49-D100.6 built 2017-06-28 07:33:31 UTC
root@router% cli
root@router>
```

Shell Prompt

CLI Prompt

17

17

CLI Modes

- Operational mode:
  - Monitor and troubleshoot the software, network connectivity, and hardware

```
user@router>
```

The > character identifies operational mode

- Configuration mode:
  - Configure the device, including interfaces, protocols, user access, and system hardware properties

```
[edit]
user@router#
```

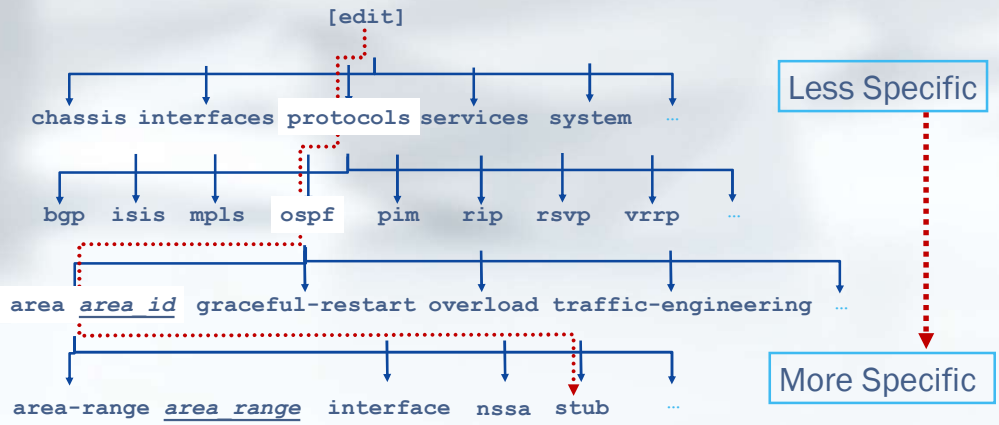
The # character identifies configuration mode

18

18

Configuration Statement Hierarchy

```
[edit]
user@router# edit protocols ospf area 51 stub

[edit protocols ospf area 0.0.0.51 stub]
user@router#
```



19

19

Configuration File Is Hierarchical

- Enter CLI commands without curly brackets:

```
[edit system]
user@router# set services web-management http port 8080
```

- The result is a hierarchical configuration file, complete with curly brackets

```
[edit system]
user@router# show services
web-management {
    http {
        port 8080;
    }
}
```

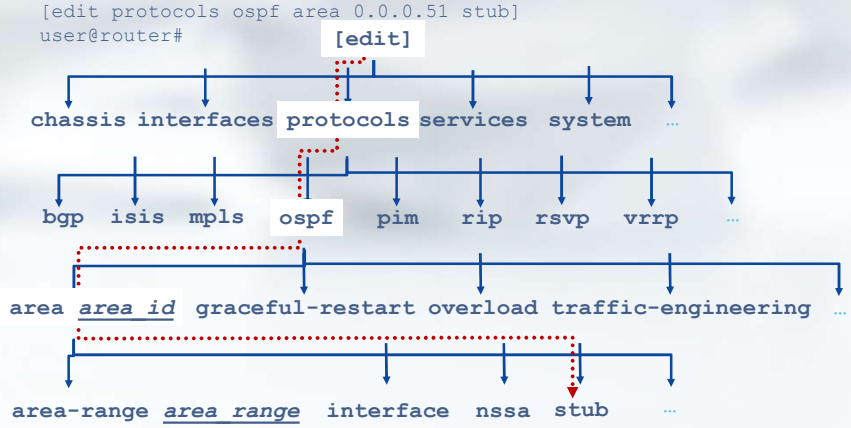
20

20

Moving Between Levels (1 of 6)

- **`edit`** functions like a *change directory* command:

```
[edit]
user@router# edit protocols ospf area 51
stub

[edit protocols ospf area 0.0.0.51 stub]
user@router#
```



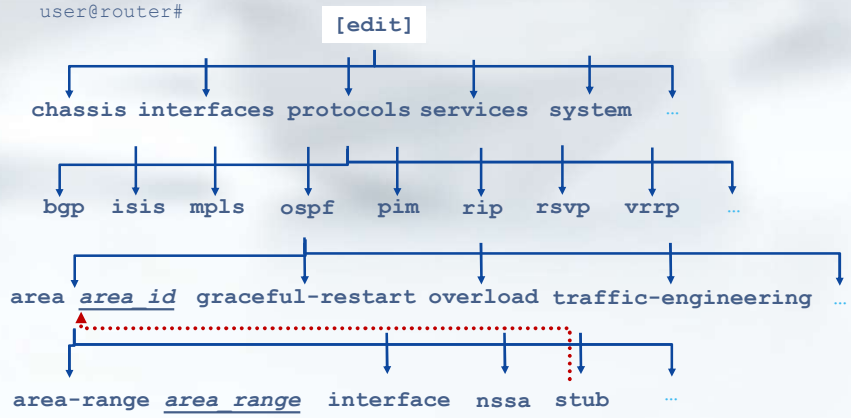
21

21

Moving Between Levels (2 of 6)

- **`up`** moves up one level in the hierarchy:

```
[edit protocols ospf area 0.0.0.51 stub]
user@router# up

[edit protocols ospf area 0.0.0.51]
user@router#
```



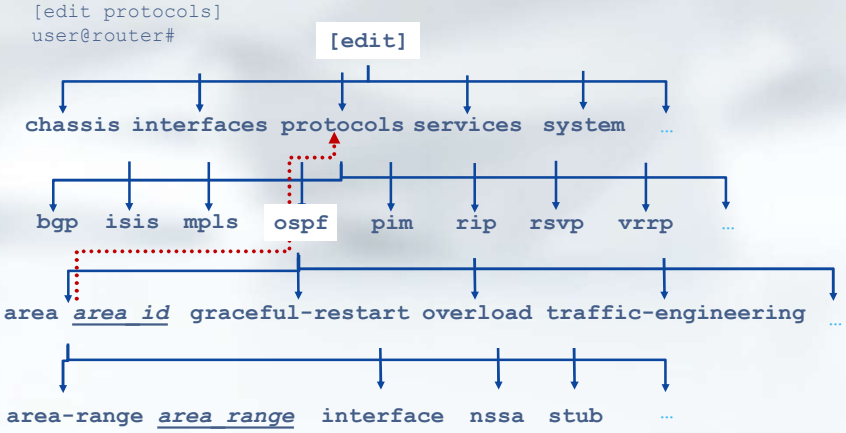
22

22

Moving Between Levels (3 of 6)

- **up** *n* moves up *n* levels in the hierarchy:

```
[edit protocols ospf area 0.0.0.51]
user@router# up 2

[edit protocols]
user@router#
```



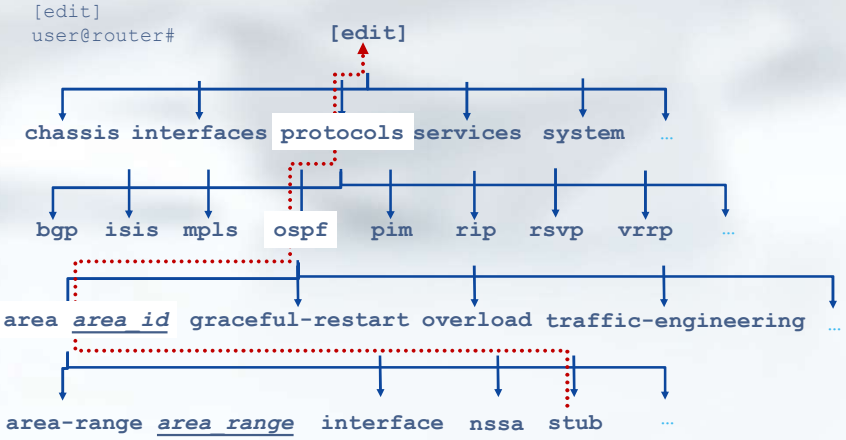
23

23

Moving Between Levels (4 of 6)

- **top** moves to the top of the hierarchy:

```
[edit protocols ospf area 0.0.0.51 stub]
user@router# top

[edit]
user@router#
```



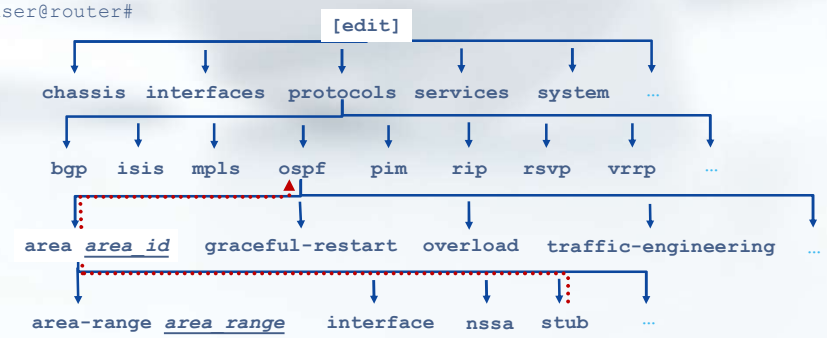
24

24

Moving Between Levels (5 of 6)

- **exit** moves to the *previous*, *higher* level in hierarchy:

```
[edit protocols ospf]
user@router# edit area 51 stub

[edit protocols ospf area 0.0.0.51 stub]
user@router# exit

[edit protocols ospf]
user@router#
```



25

25

Moving Between Levels (6 of 6)

- Summary of moving between levels:
  - **edit** functions like a CD command
  - **up** moves up one level
  - **up** *n* moves up *n* levels
  - **top** moves to the top of the hierarchy
  - **exit** moves to the previous, higher level in the hierarchy or exits configuration mode if at the top level of the hierarchy

```
[edit]
user@router# edit protocols ospf area 51 stub
[edit protocols ospf area 0.0.0.51 stub]
user@router# up
[edit protocols ospf area 0.0.0.51]
user@router# up 2
[edit protocols]
user@router# top
[edit]
user@router# exit
The configuration has been changed but not committed
Exit with uncommitted changes? [yes,no] (yes)
```

26

26

Adding Configuration Statements

- Use **`set`** to add configuration statements:

```
[edit system services]
user@router# show
ssh;
telnet;

[edit system services]
user@router# set ftp

[edit system services]
user@router# show
ftp;
ssh;
telnet;
```

FTP service added

27

27

Removing Configuration Statements

- Use the **`delete`** command to remove statements
  - Removes everything from the specified hierarchy down

```
[edit system services]
user@router# show
ftp;
ssh;
telnet;

[edit system services]
user@router# delete telnet

[edit system services]
user@router# show
ftp;
ssh;
```

Telnet service removed

28

28

Command and Variable Completion

Spacebar completes a command

```
user@host> sh<space>ow i<space>
'i' is ambiguous.
Possible completions:
  igmp                 Show Internet Group Management Protocol...
  ike                  Show Internet Key Exchange information
  interfaces           Show interface information
  ipsec                Show IP Security information
  isis                 Show Intermediate System-to-Intermediate...

user@host> show i
```

Enter a space to complete a command

Use the Tab key to complete an assigned variable

```
[edit policy-options]
user@host# show policy-statement t<tab>his-is-my-policy
then accept;

[edit policy-options]
user@host#
```

Use Tab to complete assigned variables

29

29

Context-Sensitive Help

- Type **?** anywhere on the command line to get help:

```
user@router> ?
Possible completions:
  clear                Clear information in the system
  configure            Manipulate software configuration information
  file                 Perform file operations
  help                 Provide help information
  . . .

user@router> clear ?
Possible completions:
  amt                  Show AMT Protocol information
  arp                  Clear address resolution information
  auto-configuration   Clear auto-configuration action
  bfd                  Clear Bidirectional Forwarding Detection information
  . . .
```

30

30

Topical Help

- **`help topic`** provides topical information:

```
user@router> help topic interfaces ?
Possible completions:
  accept-data           Accept packets destined for virtual address
  accept-source-mac     Policers for specific source MAC addresses
  access-profile-chap   CHAP profile associated with physical interface
  accounting            Packet counting for transit traffic
  accounting-profile    Accounting profile
  acfc                  Compression of Address and Control fields in PPP header
  ...

user@router> help topic interfaces address
                        Configuring the Interface Address

   You assign an address to an interface by specifying the address when
   configuring the protocol family. For the inet family, configure the
   interface's IP address. For the iso family, configure one or more
   addresses for the loopback interface. For the ccc, tcc, mpls, tnp, and
   vpls families, you never configure an address.
   ...
```

31

31

Help with Configuration Syntax

- **`help reference`** offers configuration syntax help:

```
user@router> help reference interfaces address
address

 Syntax

     address address {
         arp ip-address (mac | multicast-mac) mac-address <publish>;
         broadcast address;
       ...
  Hierarchy Level

     [edit interfaces interface-name unit logical-unit-number family family],
     [edit logical-routers logical-router-name interfaces interface-name unit
     logical-unit-number family family]

...
```

32

32

Command and Variable Completion

- Use the Spacebar to complete commands:

```
user@router> sh<space>ow i<space>
'i' is ambiguous.
Possible completions:
  iccp                 Show Inter Chassis Control Protocol...
  igmp                 Show Internet Group Management Protocol...
  igmp-snooping        Show IGMP snooping information
  interfaces           Show interface information
  ipv6                 Show IP version 6 information
  isdn                 Show Integrated Services Digital
  isis                 Show Intermediate System-to-Intermediate...

user@router> show i
```

Press the Spacebar to complete a command

- Use the Tab key to complete commands and variables:

```
[edit policy-options]
user@router# show policy-statement t<tab>his-is-my-policy
then accept;

[edit policy-options]
user@router#
```

Press Tab to complete assigned variables

33

33

Using | (Pipe)

- Use | (pipe) to filter and manipulate command output:

```
user@router> show route | ?
Possible completions:
  count                Count occurrences
  display              Show additional kinds of information
  except               Show only text that does not match a pattern
  find                 Search for first occurrence of pattern
  hold                 Hold text without exiting the --More-- prompt
  last                 Display end of output only
  match                Show only text that matches a pattern
  no-more              Don't paginate output
  request              Make system-level requests
  resolve              Resolve IP addresses
  save                 Save output text to file
  trim                 Trim specified number of columns from start of line
```

34

34

CLICO

Commands in Configuration Mode - 1

Command	Function
set	Assigns a value to a configuration parameter.
delete	Deletes a configuration parameter. If, after deleting the parameter, the configuration statement is empty, that empty statement is removed from the configuration.
show	Displays the configuration from the current configuration hierarchy level and below. Issuing the show command from the top of the configuration hierarchy displays the entire configuration.

35

35

CLICO

Commands in Configuration Mode - 2

Command	Description
copy	• Copies the target configuration statement to a new configuration statement with a different name. • Especially useful when you have many similar configuration statements. • The copy command duplicates the statement and the entire hierarchy of statements under that statement. • The copy command only works with configuration statements that have user-defined names.
rename	• For user-defined parameter names (such as interface names, policy statements, or firewall filters), the rename command assigns a new name to the parameter. • In most cases the rename command allows you use to change a value. • Many use the delete command to remove the statement and then use the set command to add the new value.

36

36

Navigating in the Configuration - 1

- Although you can edit the configuration from the root of the hierarchy, it is often easier to navigate to the area within the configuration you are changing before adding and removing commands.

```
[edit]
user@juniper-router# set system services finger
user@juniper-router# set system services ftp
user@juniper-router# set system services ssh
user@juniper-router# set system services telnet
```

OR

```
[edit system services]
user@juniper-router# set finger
user@juniper-router# set ftp
user@juniper-router# set ssh
user@juniper-router# set telnet
```

```
[edit]
system {
    services {
        finger;
        ftp;
        ssh;
        telnet;
    }
}
```

37

37

Navigating in the Configuration - 2

- The CLI provides four commands for navigation in configuration mode: **edit**, **up**, **top**, and **exit**.
- Use the **edit** command to jump to a specific location within the candidate configuration. The configuration mode banner changes to indicate your new location in the hierarchy

```
[edit]
user@juniper-router# edit system syslog host log

[edit system syslog host log]
user@juniper-router#
```

```
[edit]
user@juniper-router# edit system

[edit system]
user@juniper-router# edit syslog

[edit system syslog]
user@juniper-router# edit host log

[edit system syslog host log]
user@juniper-router#
```

38

38

Navigating in the Configuration - 3

- The **up** command allows you to move up levels within the hierarchy.

```
[edit interfaces fe-1/3/1 unit 0 family inet address 10.0.10.1]
user@juniper-router# up

[edit interfaces fe-1/3/1 unit 0 family inet]
user@juniper-router#
```

39

39

Navigating in the Configuration - 4

- The **top** command allows you to move to the first hierarchy level.
- The **exit** command returns you to the highest hierarchy location from which you previously entered an edit command. If you issue this command from the top level of the configuration hierarchy, you exit configuration mode.

40

40

CLICO

Navigating in the Configuration - 5

- Shortcut: You can combine navigation commands together to move through the hierarchy. For example, you can use **top** and **edit** together or you can use **top** with **show** to display a portion of the configuration from another section of the hierarchy

```
[edit protocols ospf area]
user@juniper-router# top edit system login

[edit system login]
user@juniper-router#
```

```
[edit protocols ospf area]
user@juniper-router# top show system services
web-management {
    http {
        port 8080;
    }
}
```

41

41

CLICO

Editing the Configuration - 1

- You can create or change the candidate configuration by entering a series of commands, including commands to add and remove configuration statements.
- The **set** command inserts a statement and values into the candidate configuration.

```
[edit]
user@juniper-router# set system services ftp
```

The following lines are added to the configuration file:

```
system {
    services {
        ftp;
    }
}
```

42

42

Editing the Configuration - 2

- Or use the **set** command to add statement values when required.

```
[edit]
ser@juniper-router# set system host-name juniper1
```

The following lines will be added to the configuration file:

```
system  {
    host-name juniper1;
}
```

43

Editing the Configuration - 3

- The **delete** command removes statements from your candidate configuration. Deleting a statement effectively returns the device, protocol, or service to an unconfigured state. Deleting a container statement removes everything under that level of the hierarchy.
- **ALERT!** The **delete** command removes all subordinate statements and identifiers.

```
[edit]
user@juniper-router# delete protocols
```

44

Helpful Configuration Mode Commands - 1

- **rename** a configuration statement

```
[edit interfaces]
user@juniper-router# show
fe-4/0/2 {
  unit 0 {
    family inet {
      address 10.73.24.103/24;
    }
  }
}
```

Use the rename command to change the value to 10.73.24.143/24:

```
[edit interfaces]
user@juniper-router# rename fe-4/0/2 unit 0 family inet address
10.73.24.103/24 to address 10.73.24.143/24
```

Check to see that the change is quickly completed:

```
[edit interfaces]
user@juniper-router# show
fe-4/0/2 {
  unit 0 {
    family inet {
      address 10.73.24.143/24;
    }
  }
}
```

45

45

Helpful Configuration Mode Commands - 2

- **replace pattern** of configuration statements

- [edit system]
  user@host# show static-host-mapping
  Test1 inet 192.0.2.4 ;
  Test2 inet 192.0.2.5 ;
  Test3 inet 198.51.100.1 ;
- **Issue the command: replace patterm <string1> with <string2> (enter)**
  [edit system]
  user@host# **replace pattern 192.0.2. with 198.51.100.**
- **Verify the change using the 'show' command**
  [edit system]
  user@host# show static-host-mapping
  Test1 inet 198.51.100.4 ;
  Test2 inet 198.51.100.5 ;
  Test3 inet 198.51.100.1 ;

46

46

Helpful Configuration Mode Commands - 3

- **copy** a configuration statement to another statement

```
[edit]
user@router# copy interfaces ge-0/0/10 to ge-0/0/11
```

47

47

Logging Out

```
[edit protocols ospf]
mike@jnpr1# exit configuration-mode
Exiting configuration mode

mike@jnpr1> exit
logout
Connection closed by foreign host.
$
```

48

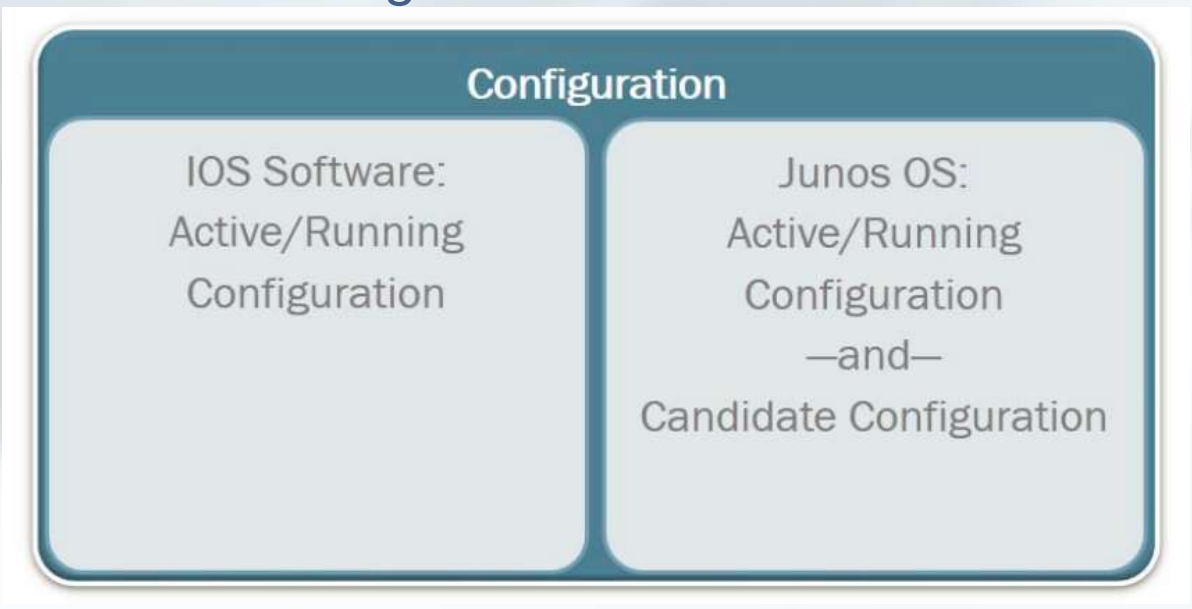
48

- CLI alapok
- Operational mód, Configuration mód áttekintése
- **Active vs. Candidate konfiguráció közti különbségek áttekintése**
- Commit lehetőségek átnézése

49

49

Candidate Configuration



50

50

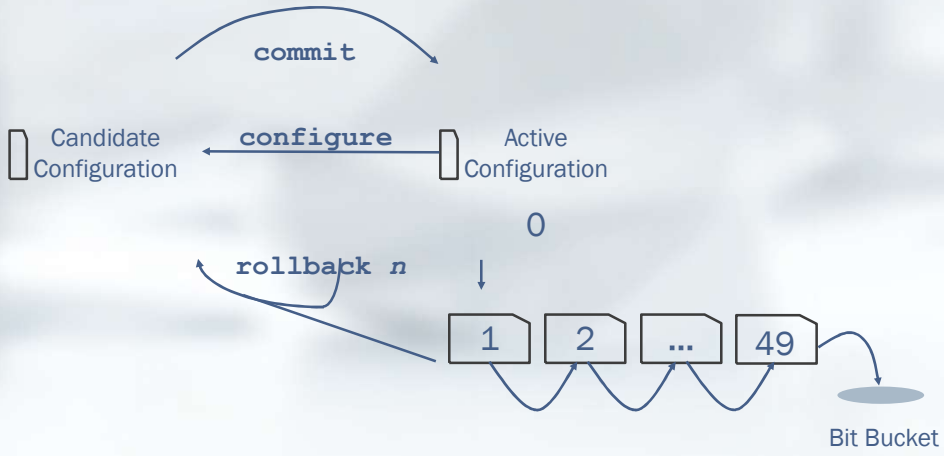
Active Versus Candidate Configuration

- Batch configuration model:
  - Must commit configuration changes
- Active configuration:
  - Current operational configuration
  - Boot-up configuration
- Candidate configuration:
  - A working copy for configuration changes
  - Initialized with the active configuration
  - Becomes active configuration upon commit

51

51

Overview: The Life of a Configuration File



52

52

Viewing the Candidate Configuration

```
[edit]
user@router# show system services
ssh;
web-management {
    http {
        port 8080;
    }
}

[edit]
user@router# edit system services

[edit system services]
user@router# show
ssh;
web-management {
    http {
        port 8080;
    }
}
```

You can display the portions that concern you from the root of the hierarchy...

...or use **edit** to park yourself at a specific subhierarchy

Hint: To view the `set` commands used to build the configuration, use the **show | display set** command.

53

53

Entering Configuration Mode (1 of 2)

- Type **configure** at the operational mode prompt to enter configuration mode:

```
user@router> configure
Entering configuration mode

[edit]
user@router#
```

- Use **configure exclusive** to exclude other users from editing the configuration
  - Any uncommitted changes are discarded when users exit:

```
user@router> configure exclusive
warning: uncommitted changes will be discarded on exit
Entering configuration mode

[edit]
user@router#
```

54

54

Entering Configuration Mode (2 of 2)

- Use **`configure private`** to allow users to edit private copies of candidate configuration concurrently
  - When users issue a **`commit`**, their private changes merge back into the global configuration
  - Any uncommitted changes are discarded when users exit
  - If two users make competing changes, the first user's **`commit`** succeeds, and the second user receives a warning
    - The second user must issue a second **`commit`** to activate the change

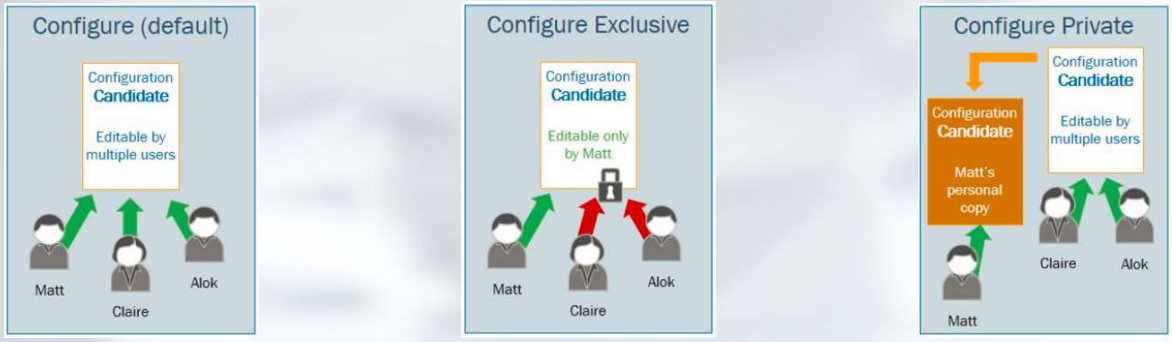
```
walter@router> configure private
warning: uncommitted changes will be discarded on exit
Entering configuration mode
Users currently editing the configuration:
  nancy terminal p0 (pid 9935) on since 2016-12-20 17:11:22 UTC
       private [edit]

[edit]
walter@router#
```

Allows other users to edit private copies of the candidate configuration

55

55

Differences Between Configuration Modes



56

56

Best Practice

- Use **configure exclusive** or **configure private** whenever multiple user accounts can make changes to the configuration. This best practice protects everyone from inadvertent errors.

57

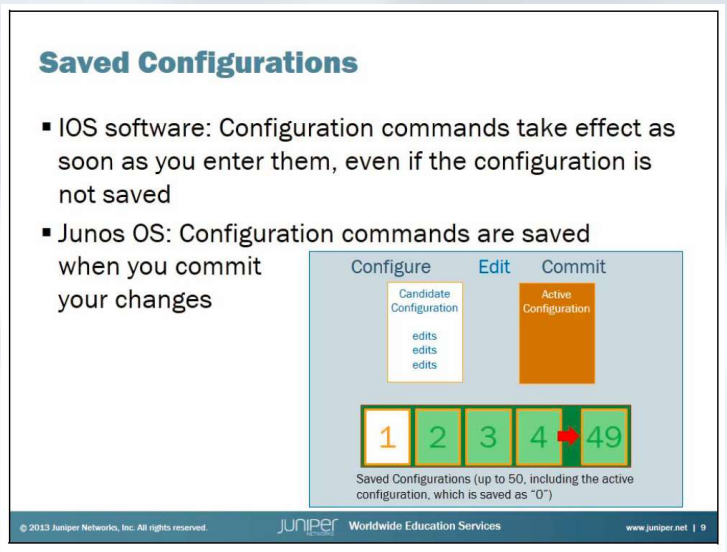
57

- CLI alapok
- Operational mód, Configuration mód áttekintése
- Active vs. Candidate konfiguráció közti különbségek áttekintése
- **Commit lehetőségek átnézése**

58

58

Saved Configurations



59

59

Committing a Configuration (1 of 2)

- Use **`commit`** to activate configuration changes:

```
[edit]
user@router# commit
commit complete
```

  - If multiple REs are installed, use **`commit synchronize`**
- Use **`commit check`** to confirm syntax:

```
[edit]
user@router# commit check
[edit interfaces ge-0/0/10 unit 0]
  'family'
    When ethernet-switching family is configured on an interface, no
other family type can be configured on the same interface.
error: configuration check-out failed
```

- Use **`commit confirmed`** to temporarily activate:

```
[edit]
user@router# commit confirmed
commit confirmed will be automatically rolled back in 10 minutes
unless confirmed
commit complete
```

60

60

Committing a Configuration (2 of 2)

- Use **commit at** to schedule a future commit:

```
[edit]
user@router# commit at 21:00:00
     configuration check succeeds
     commit at will be executed at 2016-12-20 21:00:00 UTC
     Exiting configuration mode
```

- Use **commit comment** to add comments:

```
[edit]
user@router# commit comment "Changed OSPF
configuration"
commit complete

user@router> show system commit
0   2016-12-20 15:32:42 UTC by user via cli
    Changed OSPF configuration
```

- Use **commit and-quit** to save time:

```
[edit]
user@router# commit and-quit
commit complete
Exiting configuration mode

user@router>
```

61

61

Comparing Configuration File Differences

- Compare candidate and active configurations:

```
[edit system services]
user@router# show | compare
[edit system services]
+   ftp;
-   telnet;
```

- Compare active and historical configurations:

```
user@router> show configuration | compare rollback number
user@router> show configuration | compare filename
```

- Compare arbitrary files:

```
user@router> file compare files filename_1 filename_2
```

62

62

Kezdeti konfiguráció

63

63

- Factory-default konfiguráció visszaállítása
- Konfiguráció mentése és betöltése

64

64

Reset Methods

The following methods can be used to reset the device to Factory Default

- Method 1: Reset via Reset PIN
- Method 2: Load Factory Default configuration
- Method 3: Wipe Configuration Files and load Default configuration
- Method 4: Single User Boot Procedure
- Method 5: Install Factory Default Snapshot from Boot monitor
- Method 6: Zeroize

The following method can be used to recover the root password

- Method 4: Single User Boot Procedure

Important Note for Branch SRX:
To recover a Branch SRX which is in cluster mode you must first turn it back into non cluster mode (set chassis cluster disable reboot).
If you don't have a password any more, you can only use Method 4 or Method 5

65

65

Reset Method 1: Reset via Reset Button

- Use the Reset Button
  - On J-Series: Press Configuration Pin for 15sec. to load the factory default
  - On SRX: Press the Reset PIN for 15 sec. follow LED color changes
  - On EX-Switches: Use LCD Menu to load factory default configuration
- Notes
  - You have to exit the shell first
  - The node name in the shell prompt appears to be unchanged, but this will change with the next reboot
  - If you have a Branch SRX which is still in Cluster mode, the factory default configuration can not commit as it includes switching configuration. You then should use method 5 (USB Snapshot) or 4 (Single User Mode)

66

66

Reset Method 2: Load Factory Default Configuration from CLI

If Login is still possible you can use commands to load the factory-default configuration.

You have to set a root password to get the configuration committed

```
Remote Management Console
login: user
password: <none>
root@J2300> configure
root@J2300# load factory-default
# You have to set at least the root password, otherwise you can
not commit
root@J2300# set system root-authentication plain-text-password
New password:
Retype new password:
root@J2300# commit and-quit
root@J2300>
```

67

67

Reset Method 3: Wipe Configuration Files

If Login is still possible and you have shell access you can erase the current configuration file(s) and reboot. This will be equal to a reboot with default configuration

```
root> start shell
root@J6350% cd /config
root@J6350% su
root@J6350% rm juniper.conf.gz
root@J6350% reboot


# Remark on JUNOS 11.2 (or probably earlier)
# You also have to wipe the rescue configuration.
# Otherwise the system will boot the rescue config
# if the normal configuration file has disappeared
```

68

68

Reset Method 4: Single User Boot Procedure

- Single User Mode, from the Boot monitor

```
1. Reboot the device
2. When message <Press space bar> appears --> Interrupt boot process
2. boot -s  --> Device boots in single user mode
4. login as root ,  enter "recover" to load factory default
5. enter cli as user root
6. enter configure mode
7. set system login user authorization plaintext --> Enter <Password>
8. Commit
9. If the unit was still in cluster mode, you have to remove interface
   configuration and interface assignments to security zones to commit
10. request system reboot
11. If the units was in cluster mode, then disable chassis cluster
    and reboot once more.
```

69

69

Reset Methods 5: Boot and Copy Snapshot

- Boot from a Snapshot USB Stick

```
# First you must copy a snapshot from an existing System to a USB Stick
# Keyword factory means, we copy factory default instead of running config
srx> request system snapshot partition media usb factory

# Now move the USB Stick to the System you want to recover and power it up
# Interrupt the Boot Process to get access to the Boot loader prompt
loader> nextboot usb
Setting next boot dev usb
Un-Protected 1 sectors
writing to flash...
Protected 1 sectors
loader> reboot

# Once the system has booted from the USB Stick, copy the image
# with the default configuration back to the internal Flash
srx> request system snapshot factory partition media internal
```

Notes:

- The USB Stick must have at least size of internal Flash (SRX100=1GB)
- This procedure also reformats and partitions flash and copies the software from the stick. All existing information is overwritten

70

70

Reset Method 6: Zeroize System

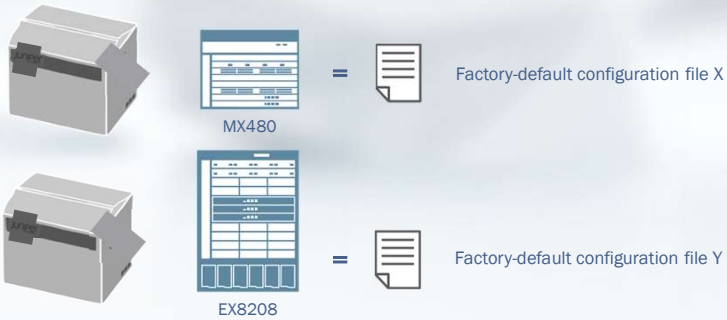
If Login is still possible and you have shell access you can completely wipe anything which is not part of the factory default configuration by zeroizing the media.

```
lab@bnlx-srx220-1> request system zeroize media
warning: System will be rebooted and may not boot without configuration
Erase all data, including configuration and log files? [yes,no] (no)
```

71

Factory-Default Configuration

- Factory-default configurations:
  - Allow access through root account (no password)
  - Include system logging to track system events
  - Contain additional parameters that are platform dependent



72

Loading a Factory-Default Configuration

- Use **`load factory-default`** to load a system's factory-default configuration
  - Must set root password to activate configuration:

```
[edit]
user@router# load factory-default
warning: activating factory configuration

[edit]
user@router# set system root-authentication plain-text-password
New password:
Retype new password:

[edit]
user@router# commit
commit complete
```

Required to activate configuration

73

Backing Out of Configuration Changes

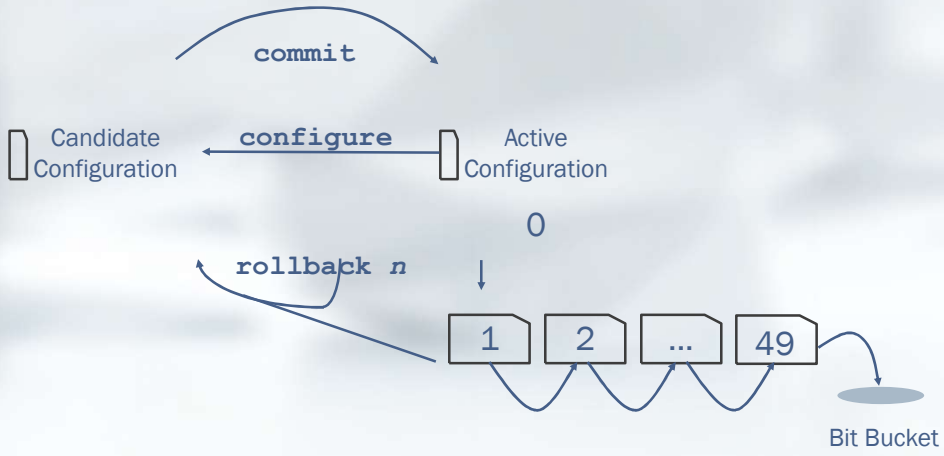
- Use **`rollback`** to restore a previous configuration:
  - **`rollback`** (or **`rollback 0`**) resets the candidate configuration to the currently active configuration
  - **`rollback 1`** loads the previously active configuration
  - **`rollback n`** loads referenced rollback version

```
[edit]
user@router# rollback ?
Possible completions:
  <[Enter]>            Execute this command
  0                    2016-12-20 00:55:48 UTC by user via cli
  1                    2016-11-28 00:16:27 UTC by lab via cli
  ...
  49                   2016-02-05 03:11:00 UTC by lab via cli
```

- Modifies only the candidate configuration
  - Do not forget to commit the changes!

74

Review: The Life of a Configuration File



75

75

Saving Configuration Files

- Use **save** to save the current configuration:
  - Saves only from the current hierarchy down
  - Saves to user's working directory by default

```
[edit]
user@router# save filename
Wrote 101 lines of configuration to 'filename'
```

  - You can also specify a full path and filename or a URL (FTP and SCP)

```
[edit]
user@router# save path/filename

[edit]
user@router# save
ftp://user:password@router/path/filename

[edit]
user@router# save scp://user@router/path/filename
```

76

76

Loading Configuration Files

- Use the **load** command to load a configuration file:

```
[edit]
user@router# load ?
Possible completions:
  factory-default      Override existing configuration with factory default
  merge                Merge contents with existing configuration
  override             Override existing configuration
  patch                Load patch file into configuration
  replace              Replace configuration data
  set                  Execute set of commands on existing configuration
  update               Update existing configuration
```

  - Use **terminal** to input from terminal capture buffer:

```
user@router# load (replace | merge | override) terminal
```

  - Use **relative** to load from current configuration hierarchy:

```
user@router# load (replace | merge) (filename | terminal) relative
```

- Use **commit** to activate the candidate configuration

77

77

Using the **run** Command

- Use **run** to execute operational mode CLI commands while in configuration mode
  - Can save time when testing the effects of a recent change

```
[edit interfaces ge-0/0/12]
user@router# set unit 0 family inet address 10.250.0.141/16

[edit interfaces ge-0/0/12]
user@router# commit
commit complete

[edit interfaces ge-0/0/12]
user@router# run ping 10.250.0.149 count 1
PING 10.250.0.149 (10.250.0.149): 56 data bytes
64 bytes from 10.250.0.149: icmp_seq=0 ttl=255 time=0.967
ms

--- 10.250.0.149 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.967/0.967/0.967/0.000 ms
```

Use **run** to test configuration changes without leaving configuration mode

78

78

Alap konfigurációs beállítások

79

79

- Jelszó beállítás
- Új felhasználók felvétele
- IP címzés
- VLAN-ok kezelése
- Access port/Trunk port beállítások
- LACP beállítás

80

80

Powering On and Off a Junos Device

- Follow safety guidelines when powering on devices
  - Automatic power-on feature when power is interrupted
- Gracefully shut down the Junos OS before removing power
  - Use **`request system halt`** to gracefully halt the Junos OS and help ensure file system integrity
  - When the Junos OS has been halted, system power is maintained
    - Reboot with console activity

```
user@router> request system halt ?
Possible completions:
  <[Enter]>            Execute this command
  at                   Time at which to perform the operation
  in                   Number of minutes to delay before operation
  message              Message to display to all users
  power-off            Power-off the software on RE
  |                    Pipe through a command
```

81

81

Initial Configuration Checklist

- Initial configuration:
  - Must include root password (restrictions exist):

```
[edit]
user@router# set system root-authentication plain-text-password
New password: ***
error: minimum password length is 6
error: require change of case, digits or punctuation
```

  - Typically also includes:
    - Hostname
    - System time
    - Remote access protocols to be used (Telnet, SSH)
    - Management interface and static route for management traffic

82

82

Initial Configuration (1 of 4)

- Log in as root with a null password:

```
Amnesiac (ttyu0)

login: root

--- JUNOS 15.1X49-D100.6 built 2017-06-28 07:33:31 UTC
root@%
```

Amnesiac prompt indicates a factory-default configuration

- Start the CLI:

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

UNIX shell prompt

Operational mode prompt

- Enter configuration mode:

```
root> configure
Entering configuration mode

[edit]
root#
```

Configuration mode prompt

83

83

Initial Configuration (2 of 4)

- Set the identification parameters:
  - Hostname
  - Root password

```
[edit]
root# edit system

[edit system]
root# set host-name router

[edit system]
root# set root-authentication plain-text-password
New password:
Retype new password:
```

Passwords entered must match and meet minimum requirements or an error will be displayed

84

84

Initial Configuration (3 of 4)

- Set the time parameters:
  - Time zone
  - Current time

```
[edit system]
root# set time-zone America/Los_Angeles

[edit system]
root# run set date 201701031244.00
Tue Jan  3 12:44:00 UTC 2017
```

- Set the management access parameters:
  - Telnet or SSH

```
[edit system]
root# set services telnet

[edit system]
root# set services ssh
```

85

85

Initial Configuration (4 of 4)

- Set the management network parameters:
  - Management interface address
  - Static route for management traffic

Management interface name varies between Junos devices

```
[edit system]
root# top

[edit]
root# set interfaces interface_name unit 0 family inet address 10.0.1.131/27

[edit]
root# set routing-options static route 10.0.1.0/24 next-hop 10.0.1.129
```

- Commit the changes!

```
[edit]
root# commit and-quit
commit complete
Exiting configuration mode

root@router>
```

Evidence that configuration changes have taken effect

86

86

Viewing the Results

- Use **`show configuration`** to view the results:

```
root@router> show configuration
## Last commit: 2017-08-01 14:44:01 UTC by lab
version 15.1X49-D100.6;
system {
    host-name host;
    time-zone America/Los_Angeles;
    root-authentication {
        encrypted-password "$1$e/FUEOVo$JF6NiAZxuufGFxDs1OMAr/";
## SECRET-DATA
    }
    services {
        ssh;
        telnet;
    }
    syslog {
...
```

87

87

Viewing the Results - sample

- Use **`show configuration | match system | display set`** to view the results:

```
{master:0}
root@testdevice> show configuration | match system | display set
set system host-name testdevice
set system root-authentication encrypted-password
"$6$L2nlSoeo$MQQxLy3vDMbpcf60rR4Oh1fY1wW/1aj3m5CIdGuS7ZC8gWbfV1hJkoe87ivElxL4GTJ3sQ8l1QfsC87XevQEU/"
set system services ssh
set system time-zone Europe/Budapest
```

88

88

The Rescue Configuration

- A rescue configuration is designed to restore basic connectivity in the event of configuration problems
  - Contents of configuration are user-defined and, by default, no rescue configuration exists

```
root@router> request system configuration rescue
save
```

Saves active configuration as the rescue configuration

```
root@router> request system configuration rescue
delete
```

Deletes the current rescue configuration

```
[edit]
root@router# rollback rescue
load complete

[edit]
root@router# commit
commit complete
```

Loads and activates the current rescue configuration

89

89

Adding New User

root@S1> **configure**

root@S1# **set system login user "USERNAME" class "SUPER-USER" full-name "FIRST LAST" authentication plain-text-password**

New password: ######

Retype new password: ######

root@S1# **commit and-quit**

The following (default) access privileges are available for the account:

operator	permissions	[clear network reset trace view]
read-only	permissions	[view]
super-user	permissions	[all]
unauthorized	permissions	[none]

90

90

Adding New User – show sample 1

{master:0}
root@testdevice> configure
Entering configuration mode

{master:0}[edit]
root@testdevice# set system login user Testember class super-user authentication plain-text-password
New password:######
Retype new password:######

91

91

Adding New User – show sample 2 (before commit)

{master:0}[edit]
root@testdevice# show | compare
[edit system]
+ login {
+ user Testember {
+ class super-user;
+ authentication {
+ encrypted-password "6dEYkmihk$W01UYZm04e6rSY5QhZePMGqg6z5juASMuP.GJpwmrJ2RFQ1fl/xGsJdCUHAeY0U4miF7zbOLAp7YsQ6jJ/ba5/"; ## SECRET-DATA
+ }
+ }
+ }

92

92

Adding New User – show sample 3 (after commit)

root@testdevice# run show configuration | match system | display set

set system login user Testember uid 2000

set system login user Testember class super-user

set system login user Testember authentication encrypted-password "6dEYkmihk$W01UYZm04e6rSY5QhZePMGqg6z5juASMuP.GJpwmrJ2RFQ1fl/xGsJdCUHAeY0U4miF7zbOLAp7YsQ6jJ/ba5/"

s

93

93

Set the Login Class – User Access Privileges

[edit system login]

user@host# **set class operator-and-boot allow-commands "request system reboot"** – set the name of the login class and allow the use of the reboot command

user@host# **set class operator-and-boot permissions [clear network reset trace view]** – set the permission bits for the login class

The system administrator grant users access or permissions to commands and configuration hierarchy levels and statements. Users can execute only those commands and view and configure only those statements for which they have access privileges.

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

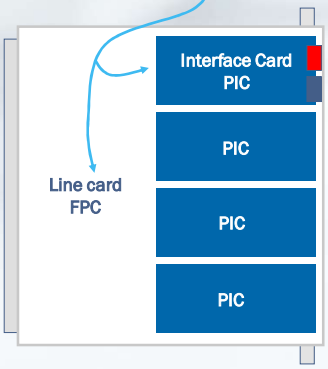
94

94

Interface Naming

- Most interfaces are named according to:
  - Interface media type (ge, so, at, and so forth)
  - Line card (FPC) slot number
  - Interface card (PIC) slot number
  - Port number

Note: While different platforms use different names for line cards and interface cards, the CLI almost always uses FPC and PIC



Interface naming example:

`ge-0/2/3` = port 3 of a Gigabit Ethernet PIC in slot 2 on FPC 0

Note: Slot and port numbering begins with zero (0) rather than one (1)

- Other interface name designations exist, such as `lo0`, `vlan`, `ae`, and so forth

95

95

Logical Units

- Similar to subinterfaces used by other vendors
  - In the Junos OS, a logical unit is *always* required
- Some encapsulations support only one logical unit
  - Unit number must be zero for these encapsulations
- Logical unit numbers are separate in meaning from circuit identifiers and do not need to match
  - We suggest keeping them the same
- Support multiple protocol addresses
  - Watch for multiple addresses when correcting mistakes!

`ge-0/0/14.51`

96

96

Interface Properties (1 of 2)

- Physical properties settings include:
  - Data Link Layer protocol
  - Link speed and duplex
  - Physical MTU
- Logical properties settings include:
  - Protocol family:
    - `inet`
    - `inet6`
    - `iso`
    - `mpls`
    - `bridge`
  - Addresses (IPv4 or IPv6 address and ISO NET address)
  - Virtual circuits (VLAN tag, DLCI, and VPI or VCI)

97

97

Interface Properties (2 of 2)

- Physical and logical interface properties are configured at their respective levels:

```
interfaces {
  interface-name {
        physical-properties;
        […]
        unit unit-number {
              logical-properties;
              […]
        }
  }
}
```

Physical properties are configured under the interface-name

Logical properties are configured under the unit-number

98

98

Interface Configuration Example

- Layer 3 interface configuration example:

```
[edit]
user@router# show interfaces
ge-0/0/2 {
    unit 0 {
        family inet {
            address 172.19.102.1/24;
            address 172.19.102.2/24 {
                preferred;
            }
        }
        family inet6 {
            address 3001::1/64;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.100.1/32;
            address 192.168.200.1/32 {
                primary;
            }
        }
    }
}
```

Note: Multiple addresses supported on a single unit

Use `preferred` option to select preferred address for interface

Note: Multiple protocol families supported on same logical unit (`family inet` is used for IPv4 and `family inet6` is used for IPv6

Use `primary` option to select primary address for interface

99

99

Tracking Interface State

- Use **`show interfaces terse`** to quickly verify the state of interfaces
  - Specify interface name to filter generated output:

```
user@router> show interfaces ge-0/0/2 terse
Interface               Admin Link Proto    Local                 Remote
ge-0/0/2                up    up
ge-0/0/2.0              up    up   inet     10.15.173.1/28
                                            172.19.102.1/24
                                  inet6    3001::1/64
                                            fe80::217:cbff:fe4e:a282/64
```

Admin and link status

Protocol family details

Address information

100

100

IP Addressing – IPv4

- The best way to learn how to configure interfaces in Junos OS is to present an example:

```
root@juniper1# set interfaces ge-0/0/1 unit 0 family inet address 192.168.100.1/30
```

  - **ge-0/0/1** is the name of the physical interface (Gigabit Ethernet)
  - **unit 0** is a logical unit configured within the physical interface
  - **family inet** identifies the protocol used by the logical interface
  - **address 192.168.100.1/30** is the address of the logical interface

101

101

Interface Configuration – show sample

```
{master:0}[edit]
root@testdevice# set interfaces lo0 unit 0 family inet address 10.10.10.10/24

{master:0}[edit]
root@testdevice# commit
configuration check succeeds
commit complete

{master:0}[edit]
Root@testdevice# show interfaces
…
lo0 {
    unit 0 {
        family inet {
            address 10.10.10.10/24;
        }
    }
}
```

102

102

IP Addressing – IPv6

- Configure IPv6 as with IPv4, except use **family inet6** and an IPv6 format address

```
[edit interfaces ge-0/0/0 unit 0]
lab@vSRX-1#set family inet6 address xxxx:xxxx:…xxxx::x /n
```

128-bit address prefix

  - Prefix length typically /64 for a multi-access network
  - Latest best practice is to use /127 prefix on pt-pt links.
  - Must be /128 for loopback addresses.
- OR to simply enable link-local addressing

```
[edit interfaces ge-0/0/0 unit 0]
lab@vSRX-1#set family inet6
```

- OR to use eui-64 addressing

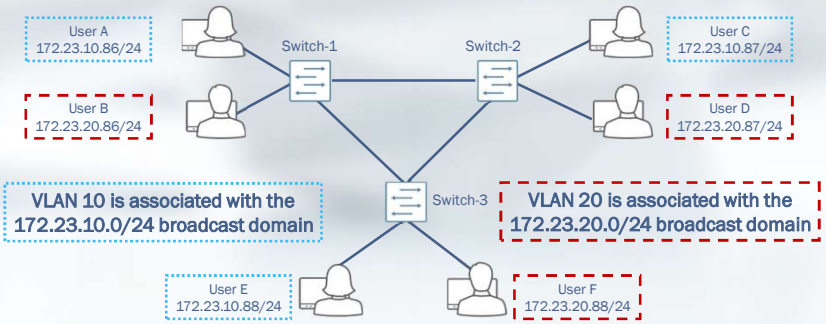
```
[edit interfaces ge-0/0/0 unit 0]
lab@vSRX-1#set family inet6 addres xxxx:xxxx…::/64 eui-64
```

103

103

What Is a VLAN?

- A logical LAN that allows you to assign users to a common broadcast domain based on business needs and regardless of physical location

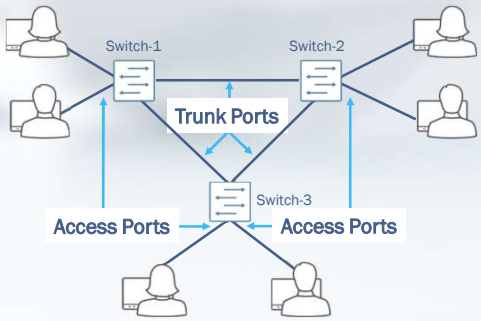


104

104

Switch Port Designations

- Switch ports operate in either access or trunk mode
  - By default all switch ports are access ports and the factory default configuration associates these interfaces with the default VLAN, which is tagged with VLAN 1

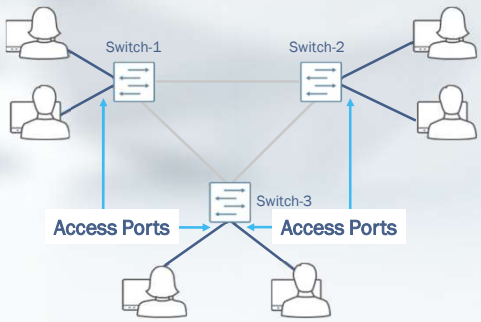


105

105

Access Ports

- Access ports typically connect to end-user devices such as computers, IP phones, and printers
  - Access ports *typically* carry untagged traffic

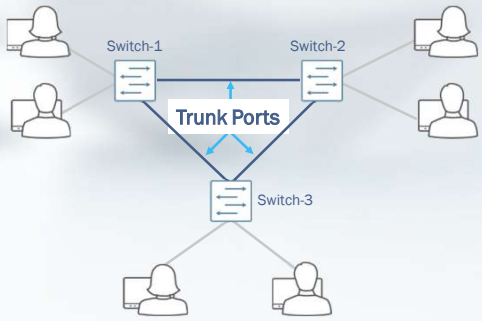


106

106

Trunk Ports

- Trunk ports typically connect switches to other switches or a router with VLAN tagging configured
  - Trunk ports *typically* carry tagged traffic



107

107

The `default` VLAN

- Factory default configuration facilitates plug-and-play implementation by enabling all switch ports for Layer 2 operations and associating them with the configured `default` VLAN
  - The default VLAN is defined with a VLAN ID of 1.

```
{master:0}
root> show vlans

Routing instance        VLAN name             Tag          Interfaces
default-switch          default               1
                                                             ge-0/0/0.0*
                                                             ge-0/0/1.0*
                                                             ge-0/0/2.0*
                                                             ge-0/0/3.0*
                                                             ge-0/0/4.0*
                                                             ge-0/0/5.0*
                                                             ...
```

The asterisk indicates that the interface is active

108

108

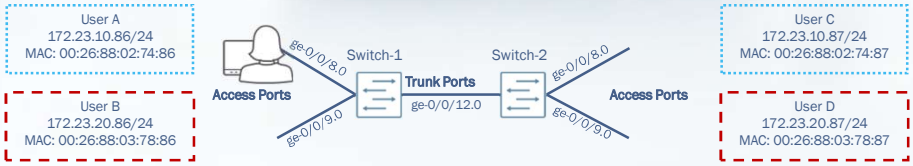
Configuring VLANs

Note: All captures are taken from Switch-1. Switch-2 should have a similar configuration.

```
{master:0}[edit]
user@Switch-1# show vlans
default {
    vlan-id 1;
}
v10 {
    vlan-id 10;
}
v20 {
    vlan-id 20;
}
```

v10 (VLAN ID 10) is associated with the 172.23.10.0/24 broadcast domain

v20 (VLAN ID 20) is associated with the 172.23.20.0/24 broadcast domain



109

109

Configuring VLANs

 # set vlans <VLAN NAME> **vlan-id** <VLAN NUMBER>

set vlans SAMPLEVLAN vlan-id 123

110

110

Show VLANs - sample

```
{master:0}
root@testdevice> show vlans

Routing instance        VLAN name       Tag         Interfaces
default-switch          default         1
                                                    ge-0/0/0.0
                                                    ge-0/0/1.0
                                                    ge-0/0/10.0
                                                    ge-0/0/11.0
                                                    ge-0/0/12.0
                                                    ge-0/0/13.0
                                                    ge-0/0/14.0
                                                    ge-0/0/15.0
                                                    ge-0/0/16.0
                                                    ge-0/0/17.0
                                                    ge-0/0/18.0
                                                    ge-0/0/19.0
                                                    ge-0/0/2.0
```

111

111

Configuring VLANs - sample

```
{master:0}
root@testdevice> configure
Entering configuration mode

{master:0}[edit]
root@testdevice# set vlans SAMPLEVLAN vlan-id 123

{master:0}[edit]
root@testdevice# commit
configuration check succeeds
commit complete
```

112

112

Show VLANs - sample

```
{master:0}[edit]
root@testdevice# show vlans
SAMPLEVLAN {
    vlan-id 123;
}
default {
    vlan-id 1;
    l3-interface irb.0;
}
```

113

113

Private VLANs

PVLANs provide Layer 2 isolation between ports within a VLAN, splitting a broadcast domain into multiple discrete broadcast subdomains by creating secondary VLANs (community VLANs and an isolated VLAN) inside a primary VLAN.

Ports within the same community VLAN can communicate with each other. Ports within an isolated VLAN can communicate **only** with a single uplink port.

PVLANs require a promiscuous port connection with a router or a routed VLAN interface (RVI) to route Layer 3 traffic among the secondary VLANs.

114

114

Private VLANs

PVLANs are useful for restricting the flow of broadcast and unknown unicast traffic and for limiting the communication between known hosts. Service providers use PVLANs to keep their customers isolated from each other. Another typical use for a PVLAN is to provide per-room Internet access in a hotel.

115

115

Private VLANs – port types

- **Promiscuous Port:** trunk port on a switch that is connected uplink to Router or Firewall or servers. Promiscuous port can communicate with all other private VLAN ports within a private VLAN. The port is assigned member of primary VLAN and must be associated with 802.1Q tag. Trunk ports that are member of private VLANs are promiscuous port.
- **Community Port:** It is a private VLAN where hosts connected to ports in a same community VLAN can communicate with each others and can also communicate with promiscuous port of the same private VLAN. It is a secondary VLAN and the port is assigned member of primary VLAN.

116

116

Private VLANs – port types

- **Isolated Port:** The isolated port can't communicate with other hosts connected to other isolated ports or community ports within a same private VLAN. Isolated port can communicate with promiscuous port and private VLAN trunk ports. If you want an Isolated port in a single switch then you don't need to create VLAN for Isolated vlan. In Juniper switches, we have another flavor of Isolated port called inter-switch Isolated VLAN. This VLAN is used to pass traffic from one Isolated port of a switch to Isolated port of another switch through a PVLAN trunk. Inter-switch isolated VLAN must have secondary VLAN ID associated with it.

117

117

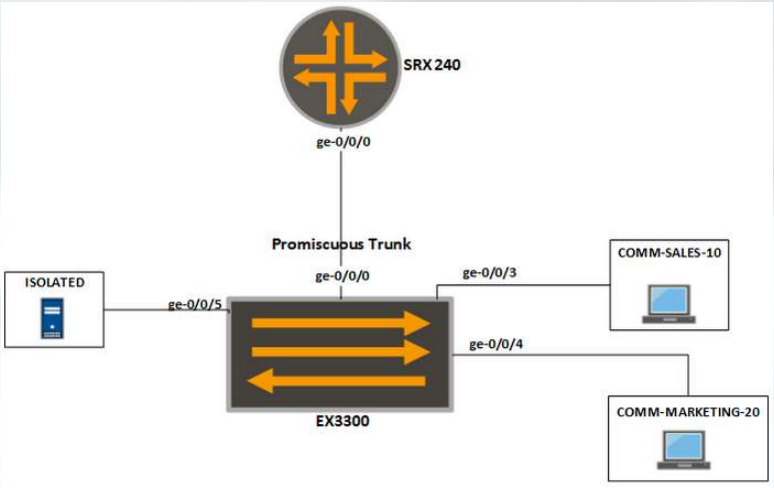
Private VLANs – port types

- **PVLAN Trunk Port:** It is the trunk port which is used to connect two or more switches when PVLAN is configured in all of these switches. The trunk port is member of all the private VLAN, the primary VLAN, community VLAN and inter-switch Isolated VLAN. Trunk ports that are member of private VLANs with ***pvlan-trunk*** command are PVLAN trunk ports.

118

118

Private VLANs - configuration



119

119

Private VLANs – configuration

SRX config

Set the ge-0/0/0 port of SRX to understand the tagged frames sent by the switch:

root# set interfaces ge-0/0/0 vlan-tagging unit 0 vlan-id 1

root# set interfaces ge-0/0/0 vlan-tagging unit 100 vlan-id 100 family inet address 192.168.10.1/24

120

120

Private VLANs – configuration

Switch config

Step 1: Configure primary VLAN name and VLAN-ID of 100:

root# set vlans PVLAN vlan-id 100 no-local-switching

Step 2: Configure the promiscuous trunk port:

root# set interfaces ge-0/0/0 unit 0 family ethernet-switching port-mode trunk

root# set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members PVLAN

Step 3: Assign promiscuous trunk port in primary VLAN:

root# set vlans PVLAN interface ge-0/0/0

Step 4: Configure Access Ports. All community ports and isolated ports must be in access port mode:

root# set interfaces ge-0/0/3 unit 0 family ethernet-switching port-mode access

root# set interfaces ge-0/0/4 unit 0 family ethernet-switching port-mode access

root# set interfaces ge-0/0/5 unit 0 family ethernet-switching port-mode access

121

121

Private VLANs – configuration

Switch config – cont.

Step 5: Configure Community VLANs and assign ports to the community PVLANs:

root# set vlans COMM-SALES-10 vlan-id 10

root# set vlans COMM-SALES-10 primary-vlan PVLAN

root# set vlans COMM-SALES-10 interface ge-0/0/3

root# set vlans COMM-MARKETING-20 vlan-id 20

root# set vlans COMM-MARKETING-20 primary-vlan PVLAN

root# set vlans COMM-MARKETING-20 interface ge-0/0/4

Step 6: Assign port to Isolated PVLAN

root# set vlans PVLAN interface ge-0/0/5.0 isolated

122

122

Private VLANs – checking conf.

Switch config – check

Root> show vlans
Root> show vlans pvlan extensive
Root> show vlans extensive

123

123

Private VLANs – checking conf.

```
{master:0}[edit vlans]
root# show
COMM-MARKETING-20 {
    vlan-id 20;
    interface {
        ge-0/0/4.0;
    }
    primary-vlan PVLAN;
}
COMM-SALES-10 {
    vlan-id 10;
    interface {
        ge-0/0/3.0;
    }
    primary-vlan PVLAN;
}
PVLAN {
    vlan-id 100;
    interface {
        ge-0/0/0.0; //This is promiscuous port. See step 2 and 3 above.
        ge-0/0/5.0; //This is ISOLATED port. See step 6 above.
    }
    no-local-switching;
}
```

124

124

Configuring Access Ports

```
{master:0}[edit]
user@Switch-1# show interfaces ge-0/0/8
unit 0 {
    family ethernet-switching {
        interface-mode access;
        vlan {
            members v10;
        }
    }
}

{master:0}[edit]
user@Switch-1# show interfaces ge-0/0/9
unit 0 {
    family ethernet-switching {
        interface-mode access;
        vlan {
            members v20;
        }
    }
}
```

125

125

Configuring Access Ports

set interfaces <interface_id> unit <unit_id> family ethernet-switching interface-mode access vlan members <VLAN_name>

set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode access vlan members SAMPLEVLAN

126

126

Configuring Access Ports - sample

```
{master:0}[edit]
root@testdevice# set interfaces ge-0/0/0 unit 0 family ethernet-switching
interface-mode access vlan members SAMPLEVLAN
{master:0}[edit]
root@testdevice# show | compare
[edit interfaces ge-0/0/0 unit 0 family ethernet-switching]
+     interface-mode access;
+     vlan {
+         members SAMPLEVLAN;
+     }
```

127

127

Configuring Access Ports – show sample

```
{master:0}
root@testdevice> show vlans

Routing instance        VLAN name          Tag         Interfaces
default-switch          SAMPLEVLAN         123
                                                       ge-0/0/0.0
default-switch          default            1
                                                       ge-0/0/1.0
                                                       ge-0/0/10.0
                                                       ge-0/0/11.0
                                                       ge-0/0/12.0
                                                       ge-0/0/13.0
                                                       ...
```

128

128

Configuring Trunk Ports

```
{master:0}[edit]
user@Switch-1# show interfaces ge-0/0/12
unit 0 {
    family ethernet-switching {
        interface-mode trunk;
        vlan {
            members [ v10 v20 ];
        }
    }
}
```

129

129

Configuring Trunk Ports

set interfaces <interface_id> unit <unit_id> family ethernet-switching interface-mode trunk vlan members [<VLAN_name>]

set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk vlan members [SAMPLEVLAN1 SAMPLEVLAN2]

130

130

Configuring Trunk Ports - sample

set interfaces ge-0/0/2 unit 0 family ethernet-switching interface-mode trunk

set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members tesztvlan10

set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members tesztvlan20

131

131

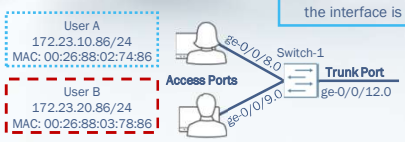
Verifying VLAN Assignments

- Use the **`show vlans`** command to verify VLAN assignments

```
{master:0}
user@Switch-1> show vlans

Routing instance        VLAN name             Tag          Interfaces
default-switch          default               1
                                                           ge-0/0/0.0*
                                                           ...
default-switch          v10                   10
                                                           ge-0/0/12.0*
                                                           ge-0/0/8.0*
default-switch          v20                   20
                                                           ge-0/0/12.0*
                                                           ge-0/0/9.0*
```

VLAN name and tag value

The asterisk indicates that the interface is active



132

132

Verifying VLAN Assignments - sample

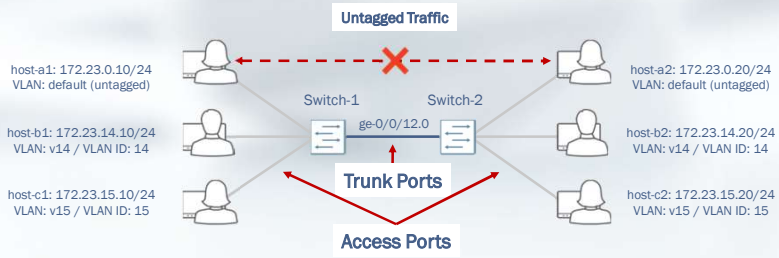
```
{master:0}
root@testdevice> show vlans

Routing instance        VLAN name             Tag          Interfaces
default-switch          SAMPLEVLAN            123
                                                           ge-0/0/0.0
default-switch          default               1
                                                           ge-0/0/1.0
                                                           ge-0/0/10.0
                                                           ge-0/0/11.0
                                                           ge-0/0/12.0
                                                           ge-0/0/13.0
                                                           ge-0/0/14.0
                                                           ge-0/0/15.0
                                                           ge-0/0/16.0
                                                           ge-0/0/17.0
                                                           ge-0/0/18.0
                                                           ge-0/0/19.0
                                                           ge-0/0/20.0
                                                           ge-0/0/21.0
                                                           ge-0/0/22.0
                                                           ge-0/0/23.0
                                                           ge-0/0/3.0
                                                           ge-0/0/4.0
                                                           ge-0/0/5.0
                                                           ge-0/0/6.0
                                                           ge-0/0/7.0
                                                           ge-0/0/8.0
                                                           ge-0/0/9.0
default-switch          tesztvlan10           10
                                                           ge-0/0/2.0
default-switch          tesztvlan20           20
                                                           ge-0/0/2.0
default-switch          tesztvlan30           30
```

133

133

What If …?

- The default behavior for trunk ports is to only send and receive tagged traffic. What if you needed to pass untagged Layer 2 traffic through trunk ports?

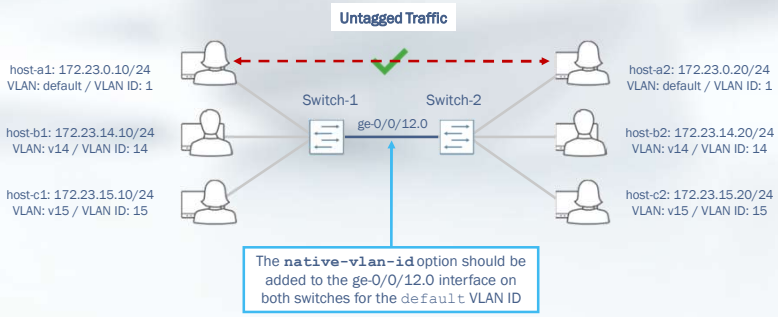


134

134

The `native-vlan-id` Option

- The **`native-vlan-id`** option enables trunk ports to accept untagged traffic in addition to tagged traffic
  - Configured on trunk ports of all switches expected to process untagged traffic



135

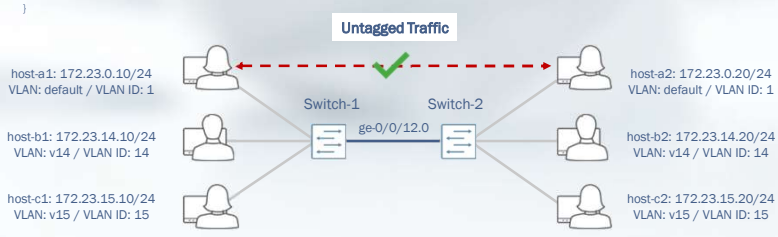
135

A Configuration Example

Note: Capture is taken from Switch-1. Switch-2 should have a similar configuration

```
{master:0}[edit interfaces]
user@Switch-1# show ge-0/0/12
native-vlan-id 1;
unit 0 {
    family ethernet-switching {
        interface-mode trunk;
        vlan {
            members [ 1 v14 v15 ];
        }
    }
}
```

Define the VLAN ID used as the native VLAN and then add the VLAN ID as a member of the trunk interface.



136

136

A Configuration Example - sample

- **For platform without ELS (Enhanced L2 Software)**

```
[edit interfaces interface-name unit 0 family ethernet-switching]
# set native-vlan-id vlan-id
```

- **For platform with ELS**

```
[edit interfaces interface-name]
# set native-vlan-id vlan-id
```

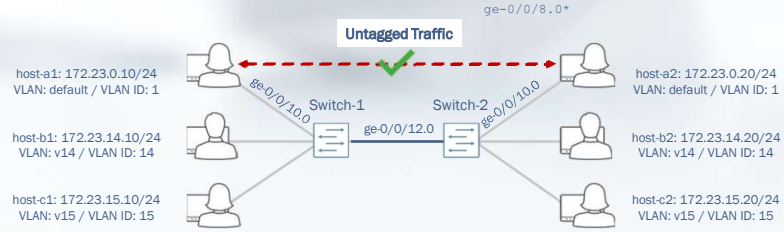
137

137

Monitoring the Native VLAN Assignment

```
{master:0}
user@Switch-1> show vlans

Routing instance        VLAN name             Tag          Interfaces
default-switch          default               1            ge-0/0/0.0*
                                                           ge-0/0/1.0*
                                                           ge-0/0/10.0*
                                                           ge-0/0/11.0*
                                                           ge-0/0/12.0*
                                                           ...
default-switch          v14                   14           ge-0/0/12.0*
                                                           ge-0/0/6.0*
default-switch          v15                   15           ge-0/0/12.0*
                                                           ge-0/0/8.0*
```

The access and trunk ports should now be assigned with the `default` VLAN and untagged traffic should be permitted across the trunk ports

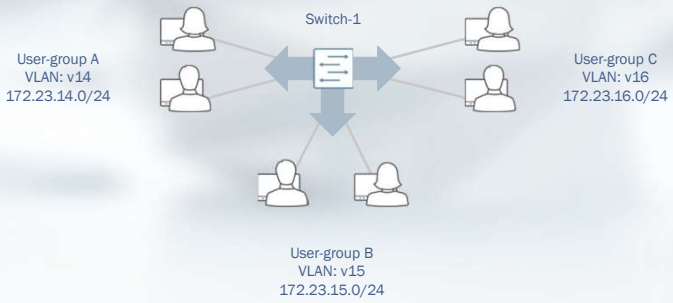


138

138

What Is It?

- An integrated routing and bridging (IRB) interface is a logical Layer 3 interface defined on an EX Series switch that facilitates inter-VLAN routing



Note: Host devices require a default gateway which points to IRB defined on the switch

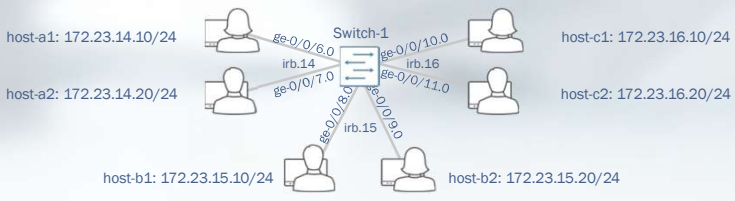
139

139

Configuring IRBs

```
{master:0}[edit]
user@Switch-1# show interfaces irb
unit 14 {
    family inet {
        address 172.23.14.1/24;
    }
}
unit 15 {
    family inet {
        address 172.23.15.1/24;
    }
}
unit 16 {
    family inet {
        address 172.23.16.1/24;
    }
}
```

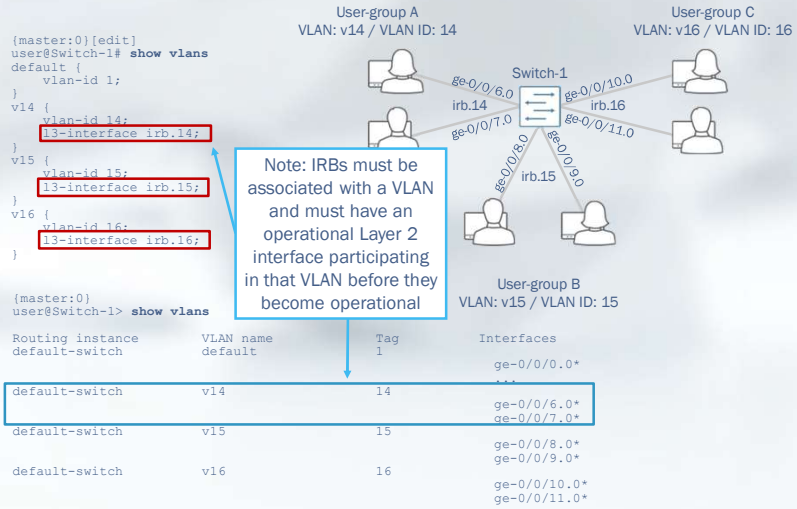
IRB addresses function as gateway address for hosts needing inter-VLAN routing services



140

140

Associating IRBs with VLANs



141

141

Configuring IRBs and with VLANs

NEW syntax – with ELS

set interfaces irb unit <unit_id> family inet address <ipv4 address/mask>

set interfaces irb unit <unit_id> family inet6 address <ipv6 address/mask>

set vlans <VLAN_name> l3-interface irb.<unit_id>

set interfaces irb unit 23 family inet address 10.10.10.45/24

set vlans SAMPLEVLAN l3-interface irb.23

OLD syntax

set interfaces vlan unit 23 family inet address 10.10.10.45/24

set interfaces vlan unit 23 family inet6 address 1234::1/126

set vlans SAMPLEVLAN l3-interface vlan.23

142

142

Configuring IRBs and with VLANs - sample

```
# set interfaces irb unit 20 family inet address 10.10.20.1/24
# set vlans tesztvlan20 l3-interface irb.20

{master:0}[edit]
root@testdevice# show | compare
[edit interfaces irb]
+    unit 20 {
+        family inet {
+            address 10.10.20.1/24;
+        }
+    }
[edit vlans tesztvlan20]
+    l3-interface irb.20;
```

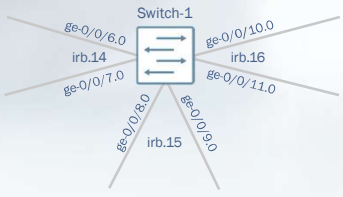
143

143

Verifying Interface State

```
{master:0}
user@Switch-1> show interfaces terse irb
Interface               Admin Link Proto    Local                 Remote
irb                     up    up
irb.14                  up    up   inet     172.23.14.1/24
irb.15                  up    up   inet     172.23.15.1/24
irb.16                  up    up   inet     172.23.16.1/24

{master:0}
user@Switch-1> show interfaces terse ge-* | match eth
ge-0/0/0.0              up    up   eth-switch
ge-0/0/6.0              up    up   eth-switch
ge-0/0/7.0              up    up   eth-switch
ge-0/0/8.0              up    up   eth-switch
ge-0/0/9.0              up    up   eth-switch
ge-0/0/10.0             up    up   eth-switch
ge-0/0/11.0             up    up   eth-switch
```

Verify that the interfaces associated with the VLANs are operational and configured as Layer 2

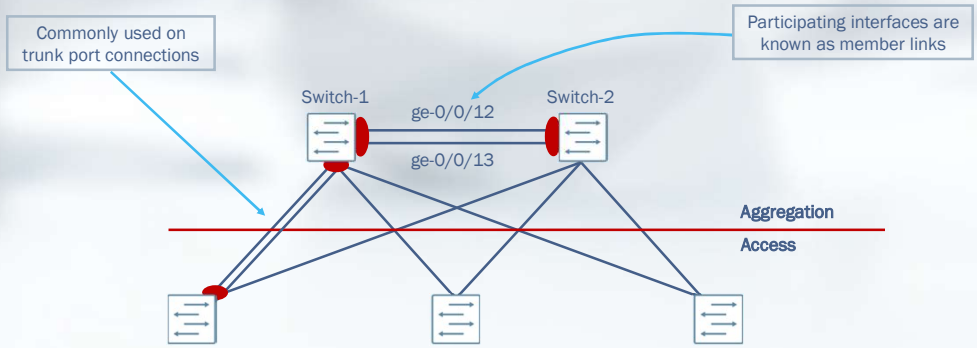


144

144

What Is It?

- Link aggregation combines multiple Ethernet interfaces in to a single link layer interface, also known as a link aggregation group (LAG) or bundle
  - Defined in the 802.3ad standard

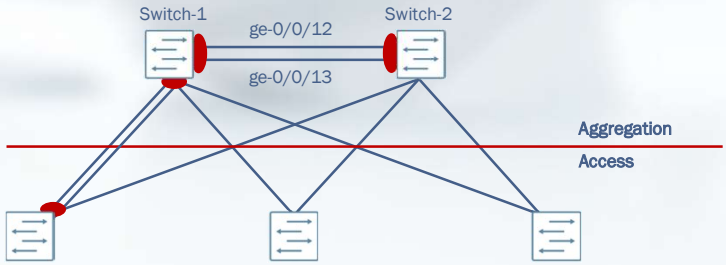


145

145

Benefits of 802.3ad Link Aggregation

- Benefits of 802.3ad link aggregation include:
  - Increases bandwidth
  - Provides link efficiency
  - Creates physical layer redundancy

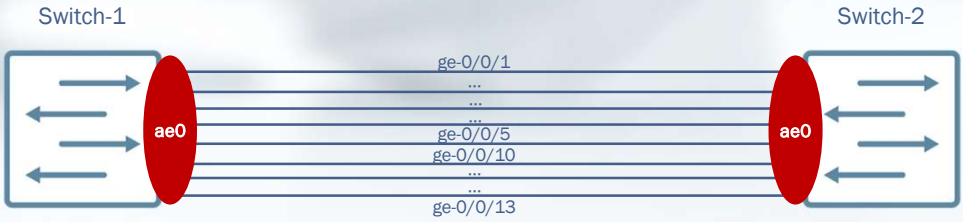


146

146

Link Requirements and Considerations

- Interface requirements and considerations include:
  - Full duplex and link speed must match
  - Up to 16 member links per LAG
  - Member links do not need to be contiguous ports nor must they be on the same switch when part of a Virtual Chassis

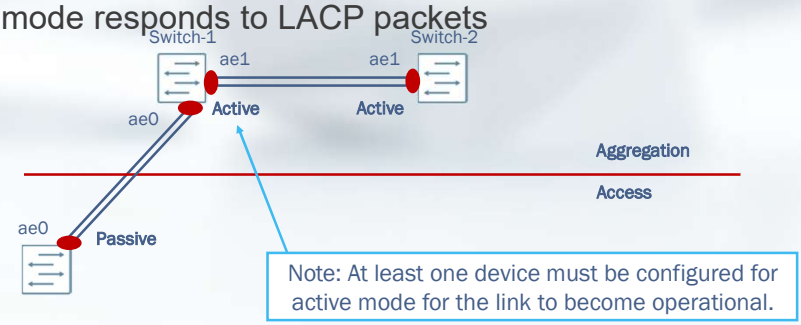


147

147

Link Aggregation Control Protocol

- LACP performs link monitoring and controls the member links that form a single logical channel
- You can set the LACP mode as active or passive:
  - Active mode initiates transmission of LACP packets
  - Passive mode responds to LACP packets



148

148

Configuring Aggregated Ethernet LACP - 1

To create a LAG

1. Create a logical aggregated Ethernet interface.
2. Define the parameters associated with the logical aggregated Ethernet interface, such as a logical unit, interface properties, and Link Aggregation Control Protocol (LACP).
3. Define the member links to be contained within the aggregated Ethernet interface.
4. Configure LACP for link detection.
5. The LAG must be configured on both sides of the link!
6. The interfaces on either side of the link must be set to the same speed and be in full-duplex mode!

149

149

Configuring Aggregated Ethernet LACP - 2

- By default, Ethernet links do not exchange LACP PDU.
- LACP mode can be active or passive. The transmitting link is the *actor*, the receiving link is the *partner*.
- If the actor and partner are both in passive mode, they do not exchange LACP packets, and the aggregated Ethernet links do not come up. Passive is the default mode!

150

150

Configuring Aggregated Ethernet LACP - 3

1. Create aggregated Ethernet device(s):
   user@host# **set chassis aggregated-devices ethernet device-count 1**
2. Create the LAG (Layer 2):
   user@host# **set interfaces ge-0/0/12 ether-options 802.3ad ae0**
   user@host# **set interfaces ge-0/0/13 ether-options 802.3ad ae0**
3. Deactivate logical unit from member links:
   user@host# **deactivate interfaces ge-0/0/12 unit 0**
   user@host# **deactivate interfaces ge-0/0/13 unit 0**
4. Define logical unit underneath the aggregated interface:
   user@host# **set interfaces ae0 unit 0 family ethernet-switching**
5. Define LACP active (or passive) on ae0 interface:
   user@host# **set interfaces ae0 aggregated-ether-options lacp active**
6. Set trunk mode on ae0:
   user@host# **set interfaces ae0 unit 0 family ethernet-switching interface-mode trunk**
   user@host# **set interfaces ae0 unit 0 family ethernet-switching vlan members all**

151

151

Configuring Aggregated Ethernet LACP - 4

1. Create aggregated Ethernet device(s):
   user@host# **set chassis aggregated-devices ethernet device-count 2**
2. Define LACP active (or passive) on ae0 interface:
   user@host# **set interfaces ae1 aggregated-ether-options lacp active/passive**
3. Create the LAG (Layer 3):
   user@host# **set interfaces ae1 unit 0 family inet address x.x.x.x/y**
   user@host# **set interfaces ae1 unit 0 family inet6 address 2001:db8:0:1::/64 eui-64**
4. Associate member links with LAG (ae0):
   user@host# **set interfaces ge-0/0/14 ether-options 802.3ad ae1**
   user@host# **set interfaces ge-0/0/15 ether-options 802.3ad ae1**
5. Deactivate logical unit from member links:
   user@host# **deactivate interfaces ge-0/0/14 unit 0**
   user@host# **deactivate interfaces ge-0/0/15 unit 0**

152

152

Monitoring LAGs

```
{master:0}[edit]
root# run show interfaces terse | match ae0
ge-0/0/12.0             up    up   aenet    --> ae0.0
ge-0/0/13.0             up    up   aenet    --> ae0.0
ae0                     up    up
ae0.0                   up    up   eth-switch
```

```
root> show interfaces extensive ae0.0 | find "LACP Statistics:"
    LACP Statistics:       LACP Rx     LACP Tx   Unknown Rx   Illegal Rx
      ge-0/0/12.0             1143        1140            0            0
      ge-0/0/13.0             1144        1141            0            0
    Marker Statistics:   Marker Rx     Resp Tx   Unknown Rx   Illegal Rx
      ge-0/0/12.0                0           0            0            0
      ge-0/0/13.0                0           0            0            0
    Protocol eth-switch, Generation: 179, Route table: 0
      Flags: Trunk-Mode

{master:0}
```

153

153

Másodlagos beállítások

154

154

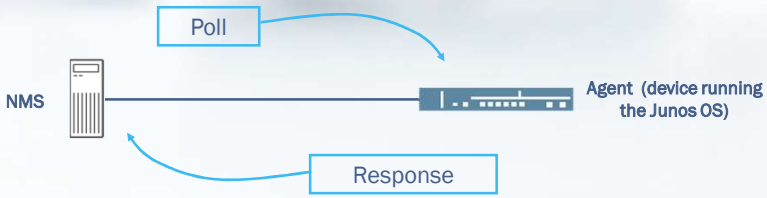
- **SNMP használata**

155

155

SNMP Overview (1 of 2)

- SNMP facilitates communication between an SNMP agent and a network management system
  - NMS and agent communication:
    - Get, GetBulk, and GetNext requests
    - Set requests
    - Notifications (*traps*—SNMP v2c or *informs*—SNMP v3)
  - Agents respond to requests from NMS and send notifications of network events (traps and informs)



156

156

SNMP Overview (2 of 2)

- MIB:
  - Used to define managed objects in a network device
  - Designed in hierarchical tree structure
  - Standard or enterprise specific
  - Consists of object identifiers
- Junos SNMP support:
  - Versions 1, 2c, and 3
  - Remote monitoring events, alarms, and history

List of Juniper Networks enterprise-specific supported MIBs: https://www.juniper.net/documentation/us/en/software/junos/network-mgmt/topics/topic-map/snmp-mibs-and-traps-supported-by-junos-os.html#id-enterprise-specific-snmp-mibs-supported-by-junos-os

157

157

SNMP minimal configuration

- Set the system community and authorization level:

  user@host# set snmp community "*community_string*" authorization read-only
- Authorize the network where the collector will be located to communicate with the SNMP agent:

  user@host# set snmp community "*community_string*" clients 192.168.1.0/32
- **user@host# commit**

158

158

SNMP configuration steps

- Configure the SNMP system name:
  user@host# set snmp name "*name*"
- Specify description (This string is placed into the MIB II sysDescription object):
  user@host# set snmp description "*description*"
- Specify the physical location of the device (This string is placed into the MIB II sysLocation object):
  user@host# set snmp location "*location*"
- Specify an administrative contact for the SNMP system (This name is placed into the MIB II sysContact object):
  user@host# set snmp contact "*admincontact*"
- Specify the unique SNMP community name and the read-only/read-write authorization level
  user@host# set snmp community "*name*" authorization *read-only/read-write*

159

159

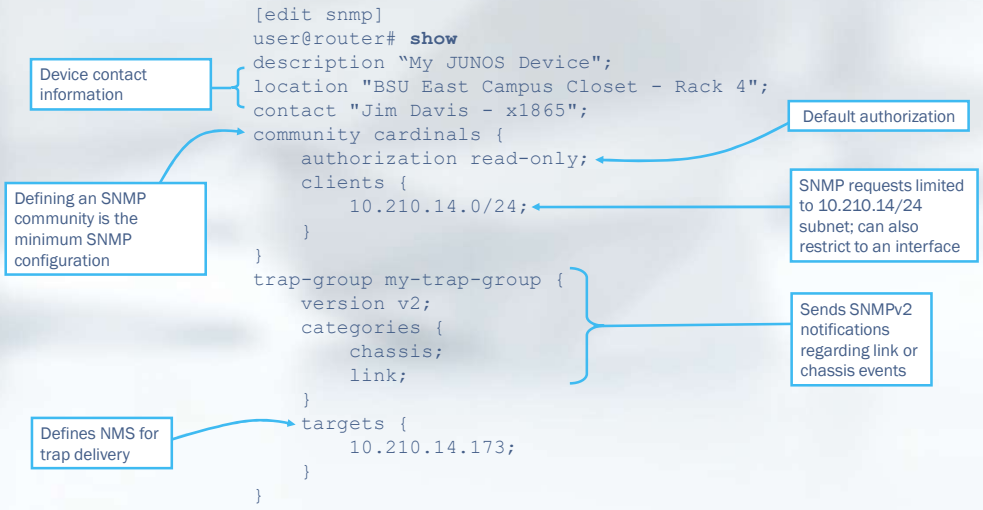
SNMP configuration steps

- Create a client list with a set of IP addresses that can use the SNMP community:
  user@host# set snmp client-list "*list_name*" 192.168.0.0/24
  user@host# set snmp community "*name*" client-list-name "*list_name*"
- Specify IP addresses of clients that are restricted from using the community:
  user@host# set snmp community "*name*" clients 192.170.0.0/24 restrict
- Configure a trap group, destination port, and target to receive the SNMP traps in the trap group:
  user@host# set snmp trap-group "*trap_group_name*" destination-port 155 targets 192.168.0.100
  You do not need to include the destination-port statement if you use the default port 162!

160

160

Example: Configuring SNMP



161

161

Monitoring SNMP Operation

- Operation:
  - Monitor the SNMP agent with NMS tools
  - Monitor SNMP protocol using traceoptions, syslog, and **show** commands
  - MIB walks and gets are available from the CLI:

```
user@router> show snmp mib walk jnxOperatingDescr
jnxOperatingDescr.1.1.0.0 = midplane
jnxOperatingDescr.2.1.1.0 = Power Supply 0
jnxOperatingDescr.2.1.2.0 = Power Supply 1
jnxOperatingDescr.4.1.1.1 = FAN 0
jnxOperatingDescr.7.1.0.0 = FPC: EX3200-24T, 8 POE @ 0/*/*
jnxOperatingDescr.8.1.1.0 = PIC: 24x 10/100/1000 Base-T @ 0/0/*
jnxOperatingDescr.8.1.2.0 = PIC: 4x GE SFP @ 0/1/*
jnxOperatingDescr.9.1.0.0 = RE-EX3200-24-T
```

162

162

Hardver információk

163

163

QFX5120 Ethernet Switches

QFX5120-48Y	48x25GbE+8x100GbE
QFX5120-48YM	48x25GbE+8x100GbE MACsec AES256 switch
QFX5120-48T	48x10GbE+6x100GbE
QFX5120-32C	32x100GbE

164

164

QFX5120 Ethernet Switches – Performance Scale

- MAC addresses per system: 288,000
- VLAN IDs: 4093
- Number of LAGs:
  - QFX5120-48Y/YM, QFX5120-32C: 80
  - QFX5120-48T: 64
- Number of ports per LAG: 64
- IPv4 unicast routes: 351,000 prefixes; 208,000 host routes; 64 ECMP (Equal-cost multipath) paths
- IPv4 multicast routes: 104,000
- IPv6 unicast routes: 168,000 prefixes; 104,000 host routes
- IPv6 multicast routes: 52,000
- ARP entries: 64,000
- Jumbo frame: 9216 bytes

165

165

QFX5130 Switch

QFX5130-32CD	128x10/25GbE 32x40/100/400GbE

166

166

QFX5130 Switch – Specification

- MAC addresses per system: 160,000
- VLAN IDs: 4091
- Number of LAGs: 128
- Number of ports per LAG: 64
- IPv4 unicast/multicast routes: 1,24M prefixes; 160,000 host routes
- IPv6 unicast/multicast routes: 610,000; 80,000 host routes
- ARP entries: 32,000
- Jumbo frame: 9216 bytes

167

167

ACX2100 Universal Metro Router

ACX2100	16xT1/E1 interfaces 4xGbE Copper 4xGbE Combo (Copper/Fiber) 2xGbE (SFP) 2x10GbE (SFP+)

168

168

ACX2100 – Specification

- MAC addresses per system: 160,000
- VLAN IDs: 4091
- Number of LAGs: 128
- Number of ports per LAG: 64
- IPv4 unicast/multicast routes: 1,24M prefixes; 160,000 host routes
- IPv6 unicast/multicast routes: 610,000; 80,000 host routes
- ARP entries: 32,000
- Jumbo frame: 9216 bytes

169

169

SRX380 Services Gateway

SRX380

SRX380	16x1GbE 4x10GbE SFP+

170

170

SRX380 – Specification

- MAC table size: 16,000
- Maximum number of VLANs: 3000
- Route table size IPv4: 1 million
- Route table size IPv6: 600,000

171

171

MX204 Universal Routing Platform

MX204	8x10GbE slots 4x100GbE slots Maximum 24x1GbE; 24x10GbE; 4x40GbE; 4x100GbE

172

172

MX204 Universal Routing Platform - Specification

- IPv4 RIB scaling with NSR: 30M
- IPv6 RIB scaling with NSR: 30M
- IPv4 FIB scaling with NSR: 8M
- IPv6 FIB scaling with NSR: 8M
- MAC scale 512K/PFE

173

173

MX150 Universal Routing Platform

MX150	10x10/100/1000BASE-T ports 2x100/1000BASE-X ports 2x10GBASE-X SFP+ ports

174

174

MX150 Universal Routing Platform - Specification

- Total IPv4 RIB entries: 20M
- Total IPv6 RIB entries: 20M

175

175

Liszenszelési információk
Juniper FLEX Program

176

176

Why Customers Move to Subscription

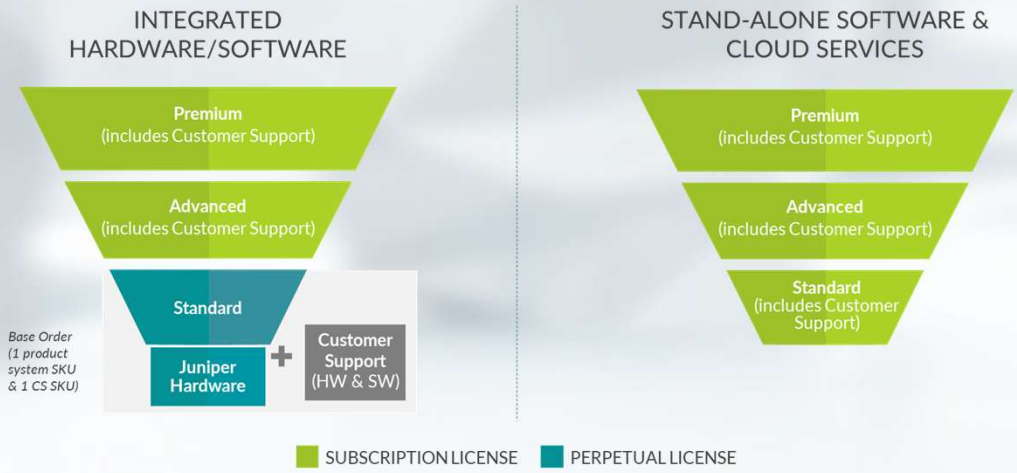
- Flexibility and ability to manage costs
  - Lower upfront investments
    - Recurring Operating Expenses (OpEx) versus upfront Capital Expenses (CapEx)
  - Alignment of costs with recurring revenue
- Lower Total Cost of Ownership (TCO)
- Lower financial barriers to new technology adoption
- Ability to add features as needed

These are Some of the Benefits of Flex Software Licenses

177

177

Juniper FLEX | Common Tiered Licensing/Pricing Model



178

178

MX Series

179

Steps For Choosing License SKUs

CLICO

1 ›

Choose HW Bundle

For Modular Systems
- Choose Chassis Bundle
- Choose Line Card SKU

For Fixed Systems
- Choose HW SKU

PayGo
- Not applicable on base HW

2a ›

Choose SW Tier

For Transport Applications
- Advanced SKU

For Services Applications
- Premium SKU

2b ›

Choose Consumption Option for SW SKU

Subscription
- Choose Appropriate term

Perpetual
- Choose Perpetual SKU
- Add separate Support SKU

PayGo (Optional)
- Full capacity
- Fractional capacity
- Min purchase limits

180

180

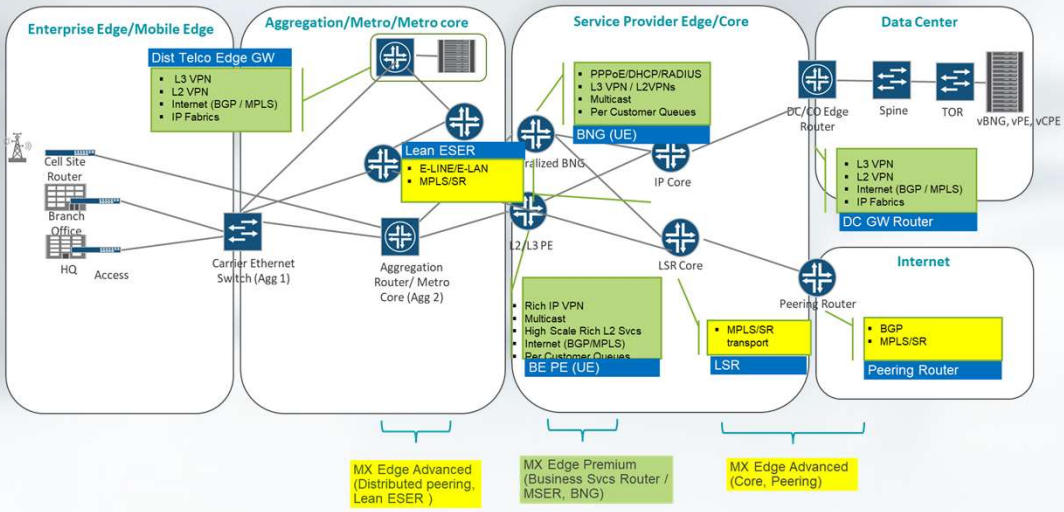
Services Offered In Each Tier

	MX
Base	Simple bridging
Advanced	All L2 services (E-LINE, E-LAN, E-TREE) (including EVPN) Core transport Internet Peering (IPv4, IPv6 (6PE)) IP multicast (PIM, IGMP) IP VPN (32) Multicast VPN w/ NG-MVPN (8) Timing (All appropriate transport, MPLS, SR, IP-Fabric and OAM, telemetry, policers, VLAN shapers, service mapping)
Premium	High scale IP VPNs (more than 32) High scale Multicast VPN (Rosen, NG-MVPN) BNG/CUPS (Premium SKU reqd for BNG RTU) PWHT for L3VPN/BNG SRv6, SRm6 TRIO features – (1:1 jflow, Static NAT, MDI)

181

181

Which MX Flex License Tiers To Use In Which Use Case?



182

182

Detailed Feature List For All Tiers

Base

1. Bridging with port and single level VLAN (dot1Q), LAG

Advanced

1. IP routing, IGP (OSFP, ISIS), IP-FRR, PIM variants, IGMP
2. Internet eBGP Peering, BGP multihoming (Add path, multi-path), EPE, BGP PIC
3. BGP Flow Spec
4. All L2 Services - **E-LINE** (L2VPNs, L2ckt, EVPN VPWS, EVPN FXC), **E-LAN** (Bridging, H-VPLS, EVPN, IRB), **E-TREE** (H-VPLS, EVPN, IRB) , L2 multicast (snooping included)
5. Limited scale IP VPNs (32) – all address families w/ BGP PIC
6. Limited scale NG-MVPN (8)
7. All MPLS transport: LDP, RSVP-TE, SR, SR-TE, MPLS-FRR (including TI-LFA)
8. IP Fabrics (GRE, MPLSoUDP, VxLAN, IPinIP)
9. Streaming Telemetry, SNMP
10. Policers, jflow (sampled) Port mirroring, sFLow; per VLAN Queuing
11. Timing (All variants)
12. OAM: BFD, Eth CFM/LFM, MPLS/SR (ping, traceroute), Services OAM, RPM, TWAMP

Premium (in addition to advanced)

1. High Scale IP-VPNs (> 32)
2. High Scale Multicast VPNs (All Rosen, NG-MVPN) (> 8)
3. IP Fabrics (SRv6, SRm6)
4. PWHT for L3VPNs/BNG
5. Inline NAT, Inline MDI
6. 1:1 inline jflow
7. Subscriber Management (BNG/CUPS) – Additional RTU SKU on top of premium SKU

183

183

What Is Different In Software SKUs?

Old X, IR, R Model

1. All license tiers sold as hardware SKUs
2. Most features were available across all tiers
   - Unique TRIO specific features like Static NAT, in-line MDI were part of R license
3. Scale differentiated between tiers
4. Implemented as perpetual SKUs as well as subscription SKUs

New Flex 2020 License Model

1. Base SKU for hardware (very basic use case)
2. Software licenses for use-case tiers
3. Implemented as subscription license
4. Perpetual SKUs available as last resort option
5. Functionality difference between tiers
6. Advanced Tier for Transport
   - Support for management VPN
7. Premium tier for IP VPN and other premium services
   - Unique TRIO specific features like Static NAT, in-line MDI were part of R license

MX10003-LC2103
MX10003-LC2103-IR

New Model

MX10K3-L2103-BASE

+

S-MX-12C-A1-C1-3
S-MX-12C-A1-C1-5
S-MX-12C-A1-C1-P
S-MX-12C-A1-C1-1*

184

184

Product To Base HW & SW SKUs Mapping

Field (Y) :
1 = 1 year
3 = 3 years
5 = 5 years
P = perpetual

Field (XX) :
Advanced (A1)
or Premium(P1)

Product	Base HW SKU	Applicable SW SKUs
MX10K-LC2101	MX10K-LC2101-BASE	S-MX-24C-XX-Y
MX2K-MPC11E	MX2K-MPC11E-BASE	S-MX-40C-XX-Y
MPC10E-10C	MPC10E-10C-P-BASE	S-MX-10C-XX-Y
MPC10E-15C	MPC10E-15C-P-BASE	S-MX-15C-XX-Y
MX204	MX204-HW-BASE	S-MX-4C-XX-C1-Y
MX10K3-L2103	MX10K3-L2103-BASE	S-MX-12C-XX-C1-Y

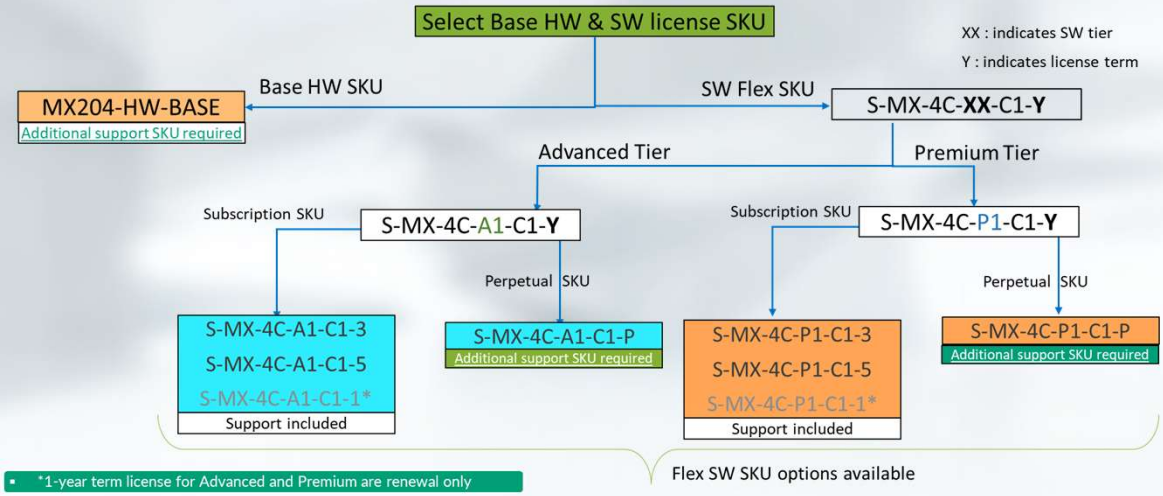
Modular Systems
S-MX-1C-XX-Y

Compact Systems
S-MX-1C-XX-C1-Y

185

185

MX204: New Flex 2020 SKUs Selection



186

186

MX204 How To Order Flex SKUs – Subscription Example

Example: MX204 with 3 Year Advanced Tier License

HW SKU bundle		Flex SW SKU
MX204-HW-BASE This SKU includes the JNP204 HW + S-MX204-S-P	+	**S-MX-4C-A1-C1-3** - Software term license for a 3-year term for the Advanced Tier for MX 0.4T

Support SKU requirements:
Additional support SKU required for base Hardware SKU – details click MX204-HW-BASE
No additional support SKU required for flex subscription SW SKU [SW support is included in the subscription SKU]

187

187

MX204 How To Order Flex SKUs - Perpetual

		Appropriate Advanced or Premium Perpetual SKU
MX204-HW-BASE This SKU includes the JNP204 HW + S-MX204-S-P	+	**S-MX-4C-A1-C1-P** -Software perpetual license for Advanced Edge Tier for MX 0.4T **S-MX-4C-P1-C1-P** - Software perpetual license for Premium Edge Tier for MX 0.4T

Support SKU requirements:
Additional support SKU required for base hardware SKU – details click MX204-HW-BASE
Additional support SKU required for perpetual SW SKUs – details click S-MX-4C-XX-C1-P
Partners also can use the above perpetual SKUs as support is not included

188

188

Support SKUs For MX204-HW-Base

SKU	Long Description	CAT1 ($)	CAT2 ($)	CAT3 ($)	CAT4($)
SVC-COR-MX204-B	Juniper Care Core Support for MX204-HW-BASE	1200	1200	1200	1200
SVC-CP-MX204-B	Juniper Care Core Plus Support for MX204-HW-BASE	1350	1458	1575	1701
SVC-NDS-MX204-B	Juniper Care Next Day Ship Support for MX204-HW-BASE	NA	1620	1750	1890
SVC-ND-MX204-B	Juniper Care Next Day Support for MX204-HW-BASE	1500	1620	1750	1890
SVC-NDCE-MX204-B	Juniper Care Next Day Onsite Support for MX204-HW-BASE	1875	2025	2187	2362
SVC-SD-MX204-B	Juniper Care Same Day Support for MX204-HW-BASE	2400	2592	2799	3023
SVC-SDCE-MX204-B	Juniper Care Same Day Onsite Support for MX204-HW-BASE	3000	3240	3499	3779
PAR-SUP-MX204-B	PSS Basic Support for MX204-HW-BASE	1200	1200	1200	1200
PAR-RTF-MX204-B	PSS RTF Support for MX204-HW-BASE	1350	1458	1575	1701
PAR-AR5-MX204-B	PSS AR5 Support for MX204-HW-BASE	1425	1539	1662	1795
PAR-NDS-MX204-B	PSS Next Day Ship Support for MX204-HW-BASE	NA	1620	1750	1890
PAR-ND-MX204-B	PSS Next Day Support for MX204-HW-BASE	1500	1620	1750	1890
PAR-NDCE-MX204-B	PSS Next Day Onsite Support for MX204-HW-BASE	1875	2025	2187	2362
PAR-SD-MX204-B	PSS Same Day Support for MX204-HW-BASE	2400	2592	2799	3023
PAR-SDCE-MX204-B	PSS Same Day Onsite Support for MX204-HW-BASE	3000	3240	3499	3779

189

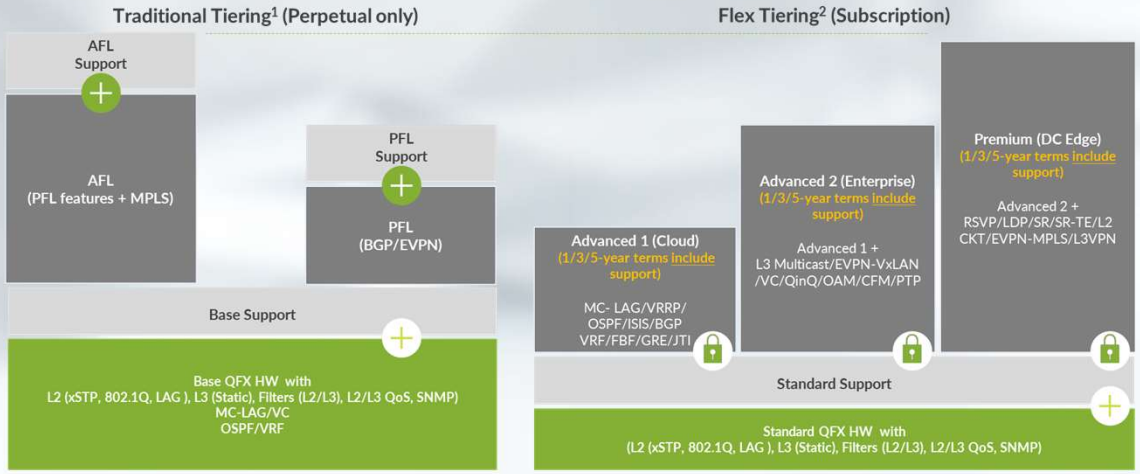
189

QFX 5k Series

190

QFX5K Software Tiering – FLEX Program

FLEX TERM LICENSES



(1) Except new products, which only support the Flex Program, and QFX5200-32C

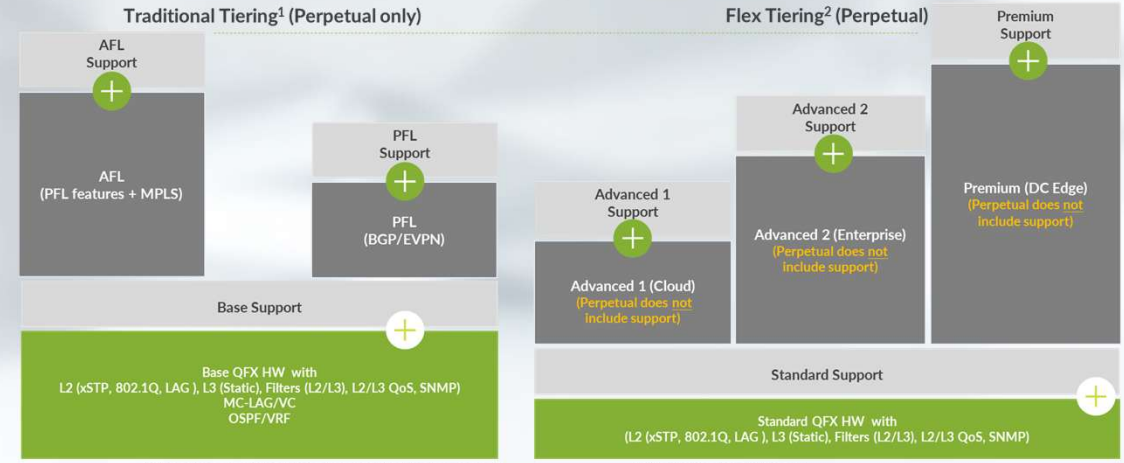
(2) Qualitative view. Not to scale.

191

191

QFX5K Software Tiering – FLEX Program

FLEX PERPETUAL LICENSES



(1) Except new products, which only support the Flex Program, and QFX5200-32C

(2) Qualitative view. Not to scale.

192

192

Use Case Mapping To FLEX Software Licenses

- Flex Software Licenses offer better alignment to common customer use cases

Use Cases	Flex License	Software Features
Layer 2	Standard (Included with HW)	L2 (xSTP, 802.1Q, LAG), L3 (Static), Filters (L2/L3), L2/L3 QoS, SNMP
IP Fabric	Advanced 1 (Cloud Features)	OSFP, RIP, ISIS, BGP, VRF, FBF, GRE, JTI, MC-LAG, VRRP, sFlow
Overlay	Advanced 2 (Enterprise Features)	Advanced 1 + L3 Multicast, EVPN-VxLAN, Virtual Chassis, QinQ, OAM, CFM, PTP
PTP Timing		
DC Edge	Premium (DC Edge Features)	Advanced 2 + RSVP, LDP, Segment Routing(SR), SR-TE, L2 Circuit, EVPN-MPLS, L3VPN
Other	Select Flex License according to required SW features	

193

193

QFX5K In The FLEX Program

S-QFX5K-C2-A1-1

Software | Product Family | Product Class (C1,C2,C3) | Software Tier (A1,A2,P1) | Term Length (1,3,5,P)

QFX5K product class	Models
C1: Class 1 (<3.2T)	QFX5120-48Y*, QFX5120-48T*, QFX5110-32Q*, QFX5110-48S*
C2: Class 2 (<6.4T)	QFX5200-32C-L*, QFX5200-32C*, QFX5120-32C*
C3: Class 3 (<12.8T)	QFX5220-32CD*, QFX5210-64C*, QFX5220-128C*

Term Lengths (1,3,5,P): 1-year, 3-year, 5-year, Perpetual

Flex Tier	Software Features	License SKU
Standard (included with HW)	L2 (xSTP, 802.1Q, LAG), L3 (Static), Filters (L2/L3), L2/L3 QoS, SNMP	Included with HW
Advanced 1 (Cloud)	OSFP, RIP, ISIS, BGP, VRF, FBF, GRE, JTI, MC-LAG, VRRP, sFlow	S-QFX5K-Cx-A1-1 S-QFX5K-Cx-A1-3 S-QFX5K-Cx-A1-5 S-QFX5K-Cx-A1-P
Advanced 2 (Enterprise)	Advanced 1 + L3 Multicast, EVPN-VxLAN, Virtual Chassis, QinQ, OAM, CFM, PTP	S-QFX5K-Cx-A2-1 S-QFX5K-Cx-A2-3 S-QFX5K-Cx-A2-5 S-QFX5K-Cx-A2-P
Premium (DC Edge)	Advanced 2 + RSVP, LDP, Segment Routing(SR), SR-TE, L2 Circuit, EVPN-MPLS, L3VPN	S-QFX5K-Cx-P1-1 S-QFX5K-Cx-P1-3 S-QFX5K-Cx-P1-5 S-QFX5K-Cx-P1-P

(*) New product going straight to Flex Model: * available SKU, * upcoming SKU (*) Existing product going to Flex Model: * available SKU, * upcoming SKU

194

194

SRX Series

195

CLICO

Why Is The SRX Moving To Flex?

Map customers purchase decisions to use cases for DC Security, SD-WAN and Next-Generation Firewall

Simplify the BOM creation process for the SRX hardware and virtual offerings

Provide an easier license upgrade path for customers and partners

Move the vSRX from throughput based to CPU core-based licensing

196

196

197