Wprowadzenie do systemu Junos

Przemysław Krzywiec

Sales Support Engineer

Przemyslaw.Krzywiec@ingrammicro.com

tel. 601-536-379

Agenda

- Wprowadzenie do kursu i platformy warsztatów

- **Moduł 0:** Introducing Junos OS and the Hardware
- **Moduł 1:** Junos Fundamentals
- **Moduł 2:** Junos User Interface
- **Moduł 3:** Junos Basic Configuration
- **Moduł 4:** Junos Logging and Monitoring
- **Moduł 5:** Junos Routing Policy and Firewall Filters
- **Moduł 6:** Junos Routing Fundamentals
- **Moduł 7:** Junos OS Switches
- **Moduł 8:** Junos OS Architecture
- **Moduł 9:** Junos J-Web
- **Moduł 10:** Jak przygotować się do JNCIA-Junos

Przemysław Krzywiec

Sales Support Engineer

Przemyslaw.Krzywiec@ingrammicro.com

tel. 601-536-379

1. Środowisko wirtualizacji vJunos-Router, vJunos-Switch, vJunos-SRX

1. Hypervisor typu 1 (bare-metal)

- Działa **bezpośrednio na sprzęcie**, bez potrzeby systemu operacyjnego.
- Cechuje się lepszą wydajnością i stabilnością – stosowany głównie w centrach danych.

Przykłady:

- **VMware ESXi**
- **Microsoft Hyper-V (w wersji serwerowej)**
- **Xen**
- **KVM (Kernel-based Virtual Machine)** – technicznie działa w jądrze Linuksa
- **Oracle VM Server**

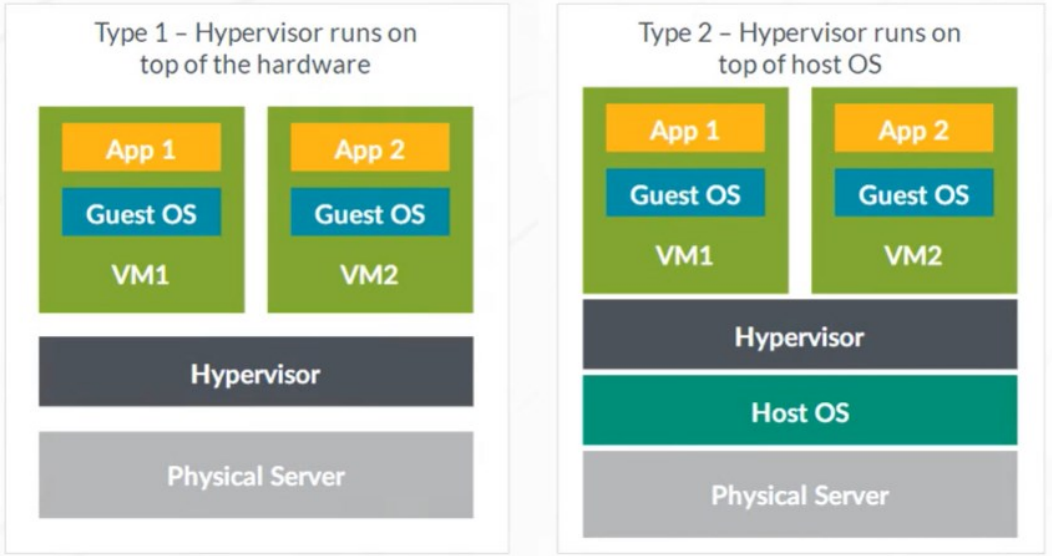
2. Hypervisor typu 2 (hostowany)

- Działa **w systemie operacyjnym hosta** (np. Windows, Linux, macOS).
- Mniej wydajny niż typ 1, ale łatwiejszy w użyciu – popularny w środowiskach developerskich.

Przykłady:

- **VMware Workstation / VMware Player**
- **Oracle VirtualBox**
- **Microsoft Hyper-V (na Windows 10/11 Pro)**
- **Parallels Desktop (macOS)**
- **QEMU (często używany z KVM)**

- Virtualization is the process of running multiple virtual instances of a device on a single physical hardware resource

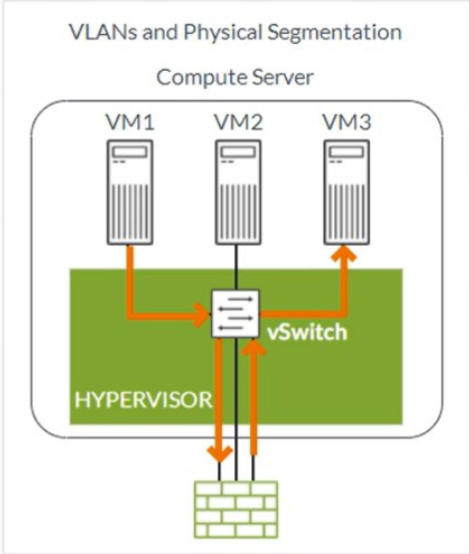


Cecha	Hypervisor typu 1	Hypervisor typu 2
Wydajność	Wyższa	Niższa
Instalacja	Na gołym sprzęcie	W systemie operacyjnym
Zastosowanie	Serwery, produkcja	Testowanie, deweloperka
Przykładowy interfejs	Zdalny, konsolowy	Graficzny, GUI

1. Środowisko wirtualizacji vJunos-Router, vJunos-Switch, vJunos-SRX

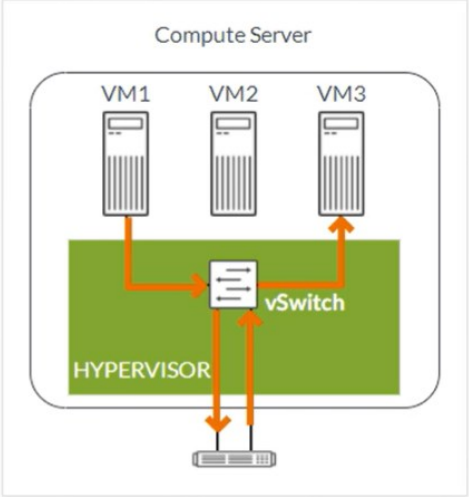
Option 1: Passing traffic through a physical network security device:

- Benefits:
  - Single network device to configure
  - Leverage existing IT knowledge
- Drawbacks:
  - High network traffic overhead
  - Complex VLAN configuration
  - Limited resource pooling



Option 2: Security software based on guest OS:

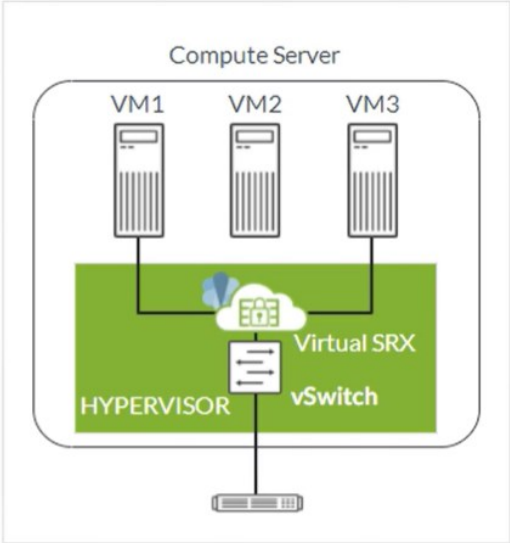
- Benefits:
  - No networking device to configure
  - Leverages existing IT knowledge
- Drawbacks:
  - Duplication of effort
  - Management overhead
  - Guest OS permissions



1. Środowisko wirtualizacji vJunos-Router, vJunos-Switch, vJunos-SRX

Option 3: Virtual network security device:

- Benefits:
  - Software duplication reduced
  - More efficient—traffic does not leave the server
  - Security processing resources are taken from the same pool as compute resources
- Drawbacks:
  - Increased physical resource usage



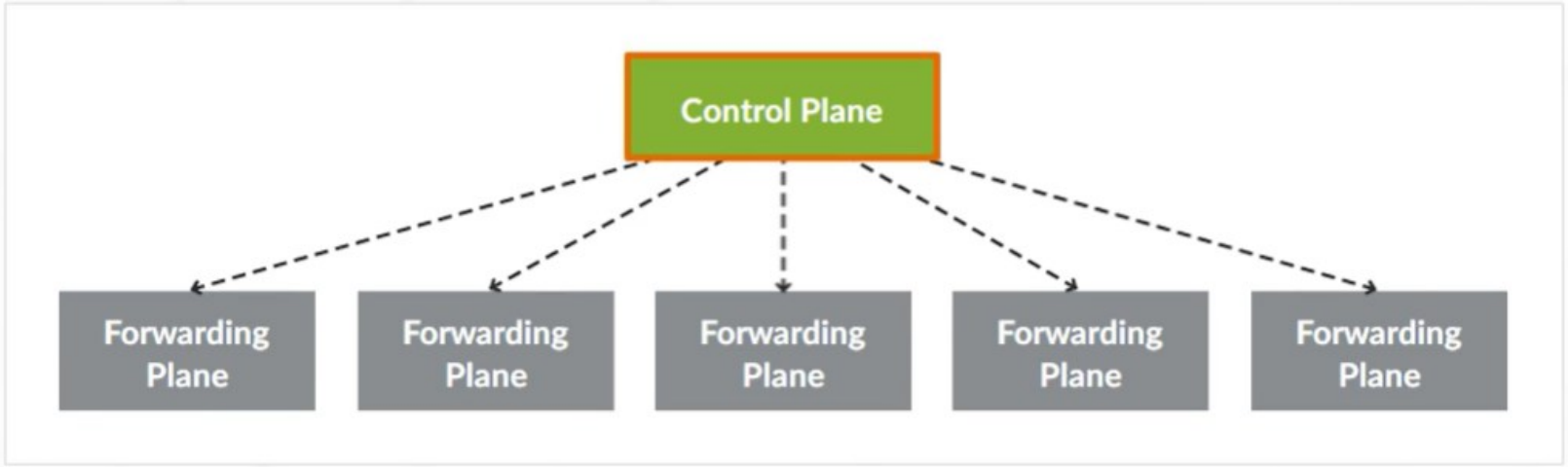
Network Functions Virtualization

- Network Functions Virtualization (NFV) is a conceptual framework wherein the network functions traditionally provided by dedicated hardware are provided by virtual machines (VMs)
  - Virtualized network functions (VNFs) are the actual functions provided (routing, switching, security)
- Brings the benefits of server virtualization to the functions of the network
  - Scalability, faster provisioning, greater efficiency
- Juniper Networks offers virtualized versions of the SRX and MX

1. Środowisko wirtualizacji vJunos-Router, vJunos-Switch, vJunos-SRX

Centralization

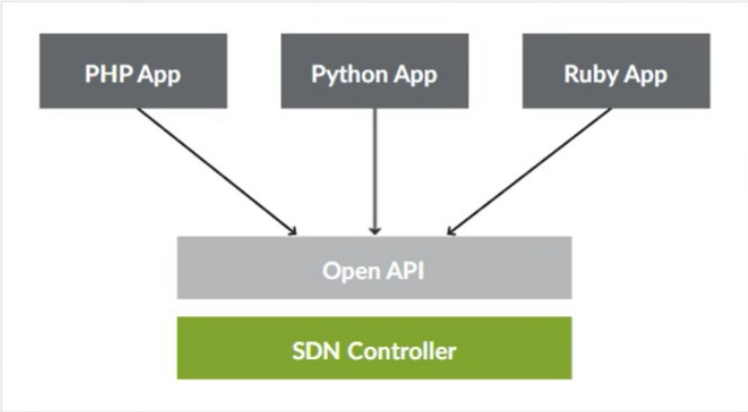
- Centralization is a key aspect of SDN
  - The SDN controller has a top-down view of the entire network
  - Separation of control and forwarding planes enables control to be centralized
  - Configuration is also centralized



1. Środowisko wirtualizacji vJunos-Router, vJunos-Switch, vJunos-SRX

Programmability

- Open northbound APIs enable network applications to be written in any language
- Network applications can interact directly with the SDN controller
  - Supports reactive network behavior based on changing business needs and capacity



SDN as an Overlay: Part 1

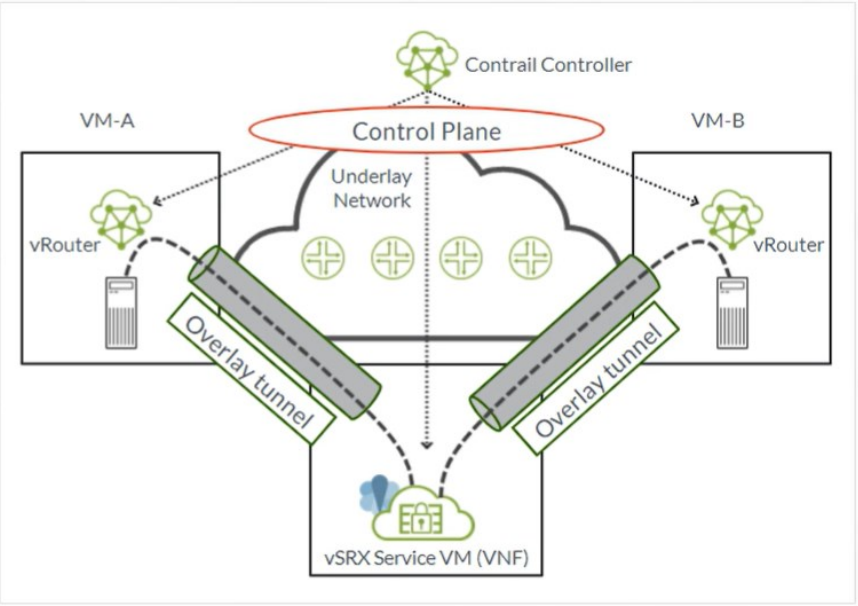
- Uses various tunneling technologies to enable communication between hypervisors
  - VXLAN
  - NVGRE
  - MPLS over GRE
- Separates control and forwarding planes
- Overlays can use a number of tunneling protocols:
  - Virtual Extensible LAN (VXLAN) is a MAC in the IP tunnel format standardized in IETF RFC 7438
  - Network virtualization using GRE (NVGRE) is also a MAC in IP tunnel format based on the standard generic routing encapsulation (GRE) protocol
  - MPLS over GRE is the tunneling format that Juniper uses in its Contrail solution, by default (Contrail can also use VXLAN)

Technologia	Przeznaczenie	Tuneluje co?	Encapsulation	Warstwa OSI	Przykład zastosowania
MPLS over GRE	Sieci operatorów (WAN)	Pakiety MPLS	GRE/IP	Warstwa 2.5 / 3	Ruch MPLS przez IP (np. przez Internet)
VXLAN	Sieci data center / chmury	Ramek Ethernet	UDP/IP	Warstwa 2/3	Rozszerzenie L2 przez L3 (sieć overlay) Wirtualizacja
NVGRE	Sieci data center (Microsoft)	Ramek Ethernet	GRE/IP	Warstwa 2/3	Sieci overlay w środowiskach Hyper-V

1. Środowisko wirtualizacji vJunos-Router, vJunos-Switch, vJunos-SRX

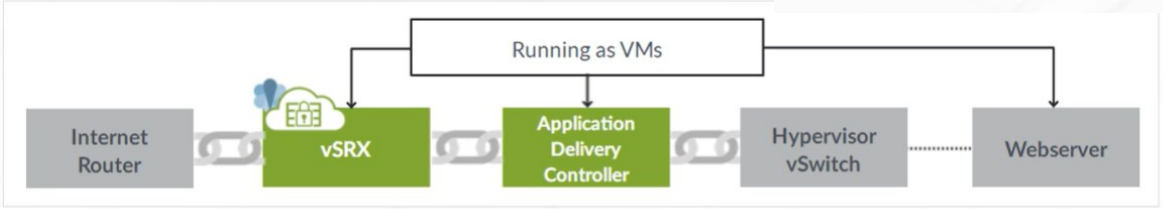
Service Chaining with Contrail

- Juniper's Contrail is a simple, open, and agile SDN solution that automates and orchestrates the creation of highly scalable virtual networks
- Contrail can increase business agility by enabling the migration of applications and IT resources to more flexible private or hybrid cloud environments
- Service VM is launched
- Overlay tunnels to the hypervisor running service VMs are created
- Routes are created as needed
- Traffic between VMs now passes through a service instance

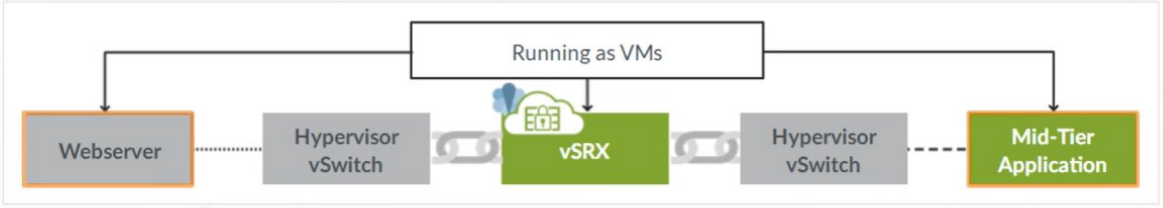


Virtual SRX with NFV Examples

- Service chain examples
  - Example 1

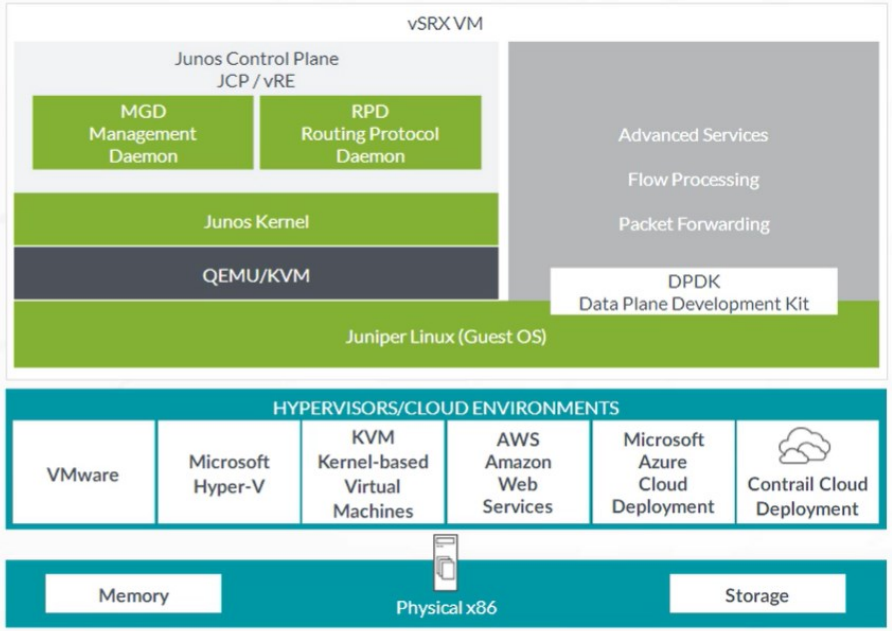


  - Example 2



Virtual SRX

- vSRX characteristics
  - Linux as Base OS
  - Data Plane Development Kit (DPDK) for I/O operations offers higher throughput
  - Single-root I/O virtualization (SR-IOV)
  - Small computer system interface (SCSI) virtual disk support
  - vNIC and VMXNET3 vNIC
  - 4GB vRAM and 16GB vDisk
  - vCPUs can be independently assigned to Routing Engine (RE)/Packet Forwarding Engine (PFE)
  - Junos version—D70 introduces multicore support for PFE



Virtual SRX Installation

- vSRX installation:
  - Download the vSRX package
  - Import the VM
    - Configure hardware resources
  - Configure interfaces
  - Start the VM and configure Junos

Virtual SRX System Requirements

- vSRX system requirements:
  - Supported hypervisors
    - VMware ESXi
    - Linux KVM (CentOS, Ubuntu, Red Hat)
    - Hyper-V
  - Memory and Disk Space
    - 4 GB RAM
    - 16 GB (IDE or SCSI drives)
  - 2 vCPUs
  - vNICs
    - Up to 10 for ESXi
    - Up to 8 for KVM/QEMU

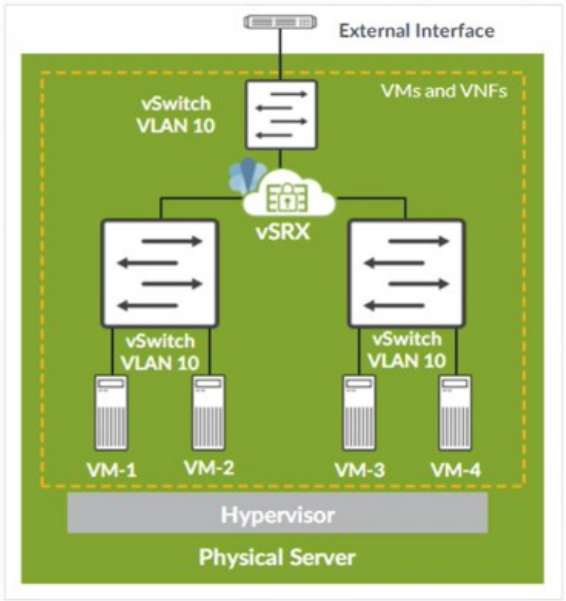
For VMware, you can check for CPU and other hardware compatibility at the **VMware Compatibility Guide** webpage. You can access the website from the **Reference Links** document in the Resources section.

For KVM, it is recommended that you enable hardware-based virtualization on the host machine. You can verify CPU compatibility at the **Linux-KVM** website. You can access the website from the **Reference Links** document in the Resources section.

1. Środowisko wirtualizacji vJunos-Router, vJunos-Switch, vJunos-SRX

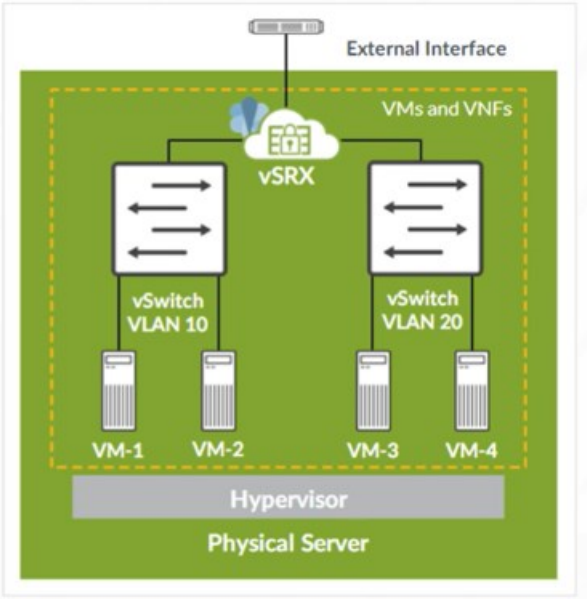
Segmentation with vSRX: Layer 2

- Layer 2 case:
  - Provides virtualized server segmentation
  - vSRX operates in transparent mode
  - Virtual machines (VMs) are tied to different vSwitches
  - VMs are mapped to the same VLAN



Segmentation with vSRX: Layer 3

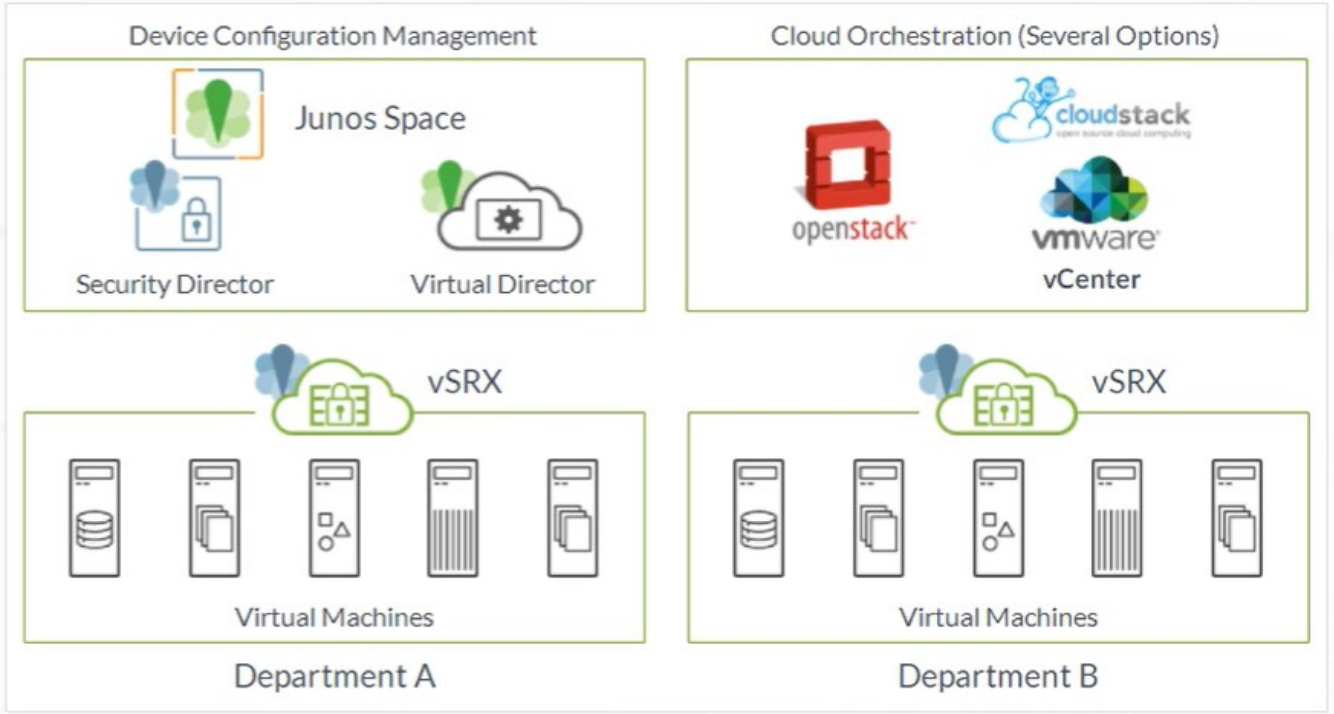
- Layer 3 case:
  - vSRX operates in routed mode
  - Inter-VLAN traffic is inspected by the vSRX
  - VMs are tied to different vSwitches



1. Środowisko wirtualizacji vJunos-Router, vJunos-Switch, vJunos-SRX

Use Case: Private Cloud: Part 1

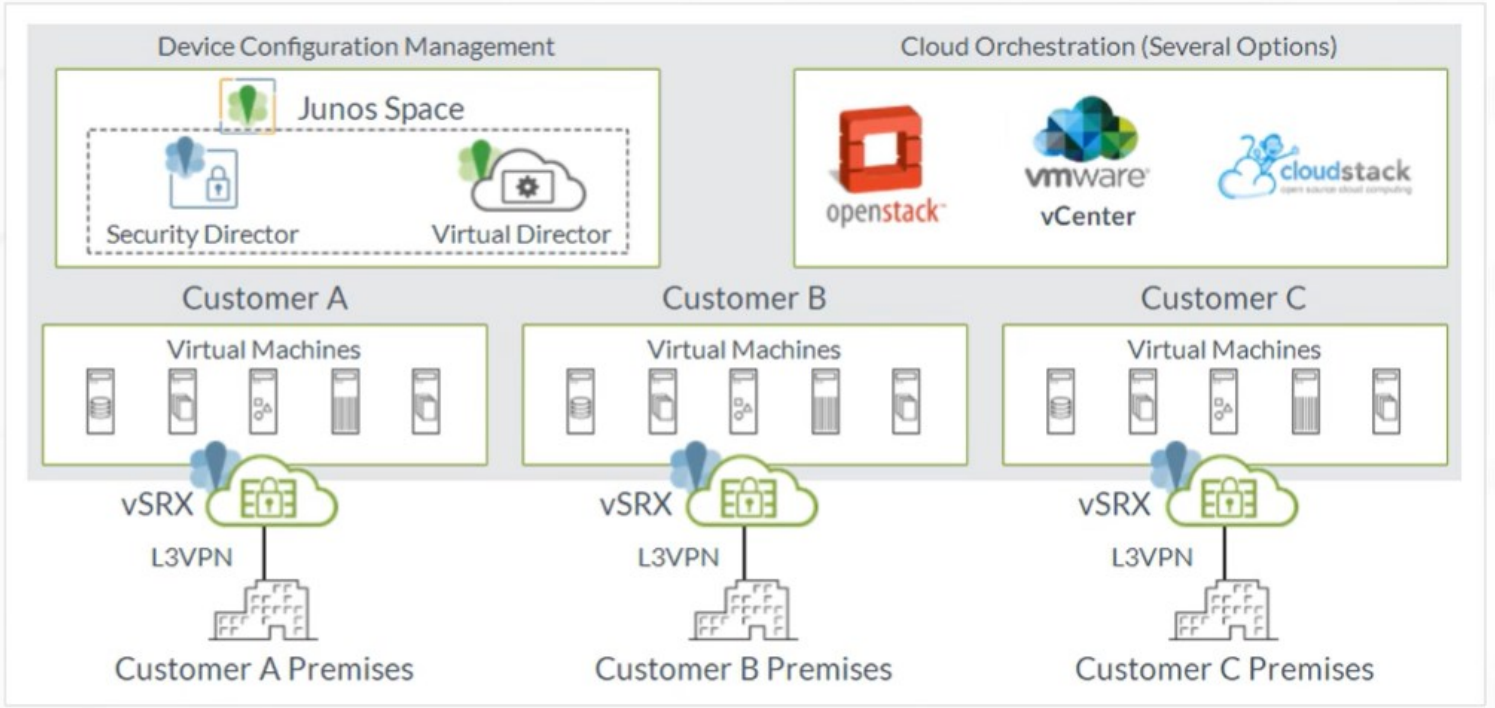
- Many large enterprises, financial institutions, universities, and other organizations have virtualized their data centers to implement private clouds that they control and manage
- The vSRX can provide all the necessary security benefits:
  - Securely communicate through routing, NAT, and VPNs
  - Provide edge security
  - Keep and provide functional separation
  - Support compliance and regulatory needs



1. Środowisko wirtualizacji vJunos-Router, vJunos-Switch, vJunos-SRX

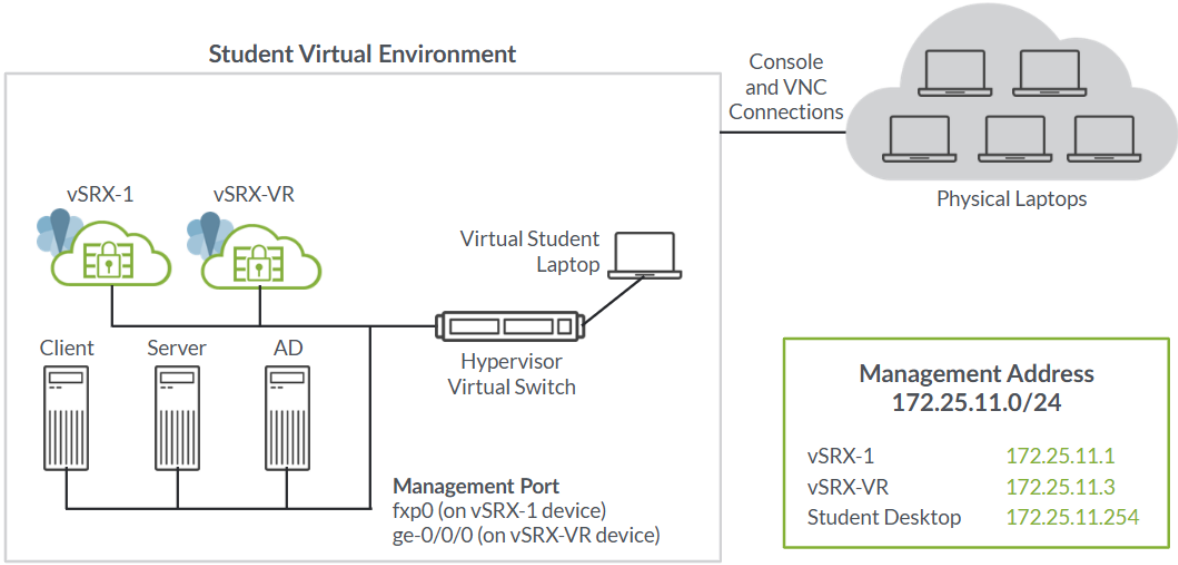
Public and Hybrid Cloud

- Customer segmentation
- Edge security
- Controlled access to the VMs
- Granular control over physical and virtual assets

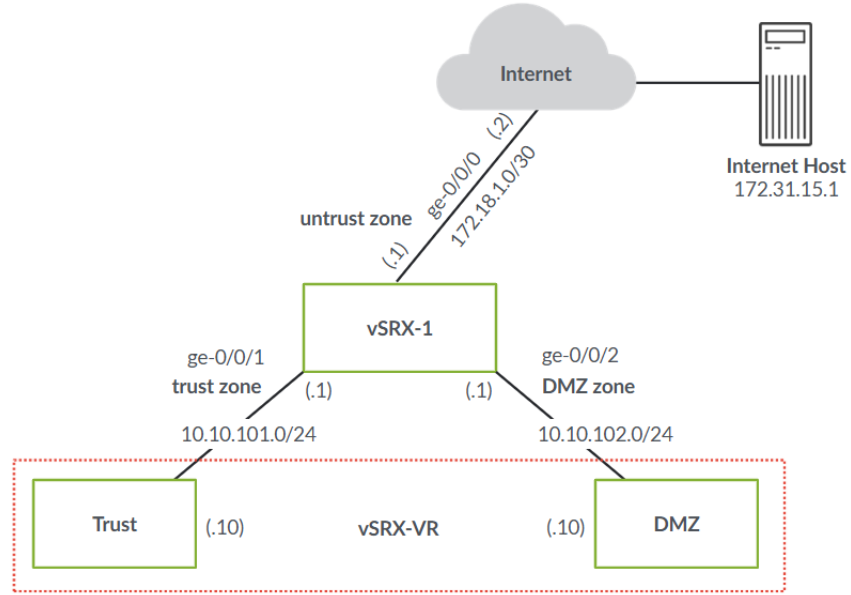


1. Środowisko wirtualizacji vJunos-Router, vJunos-Switch, vJunos-SRX

Management Network Diagram



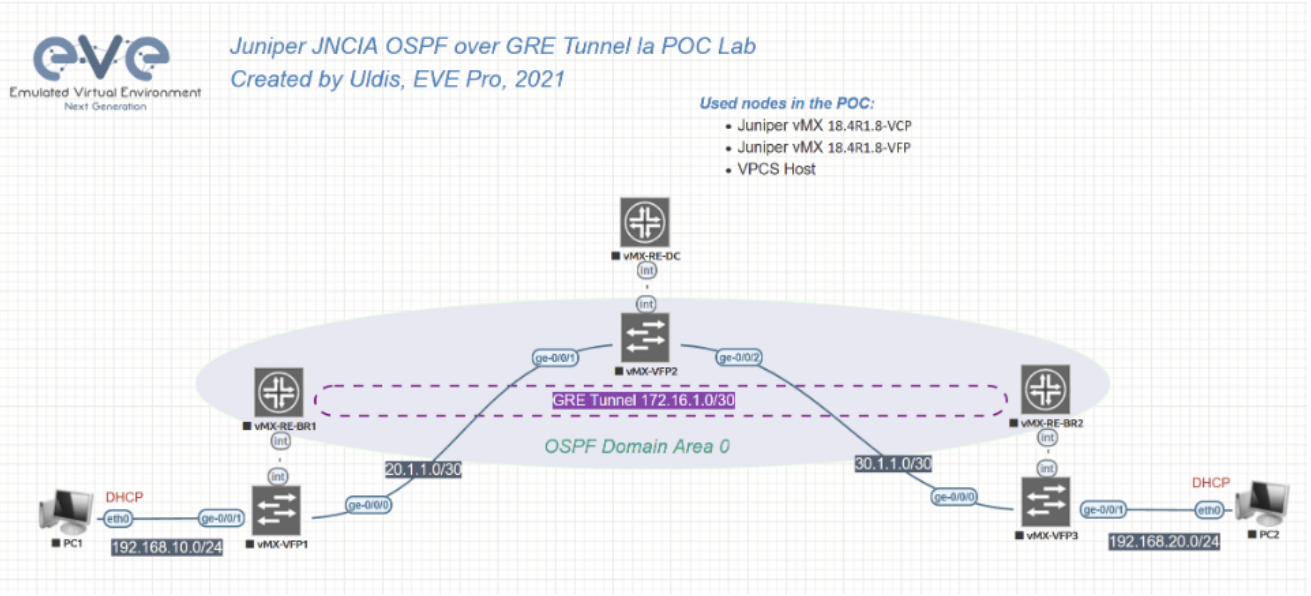
Lab Network Diagram

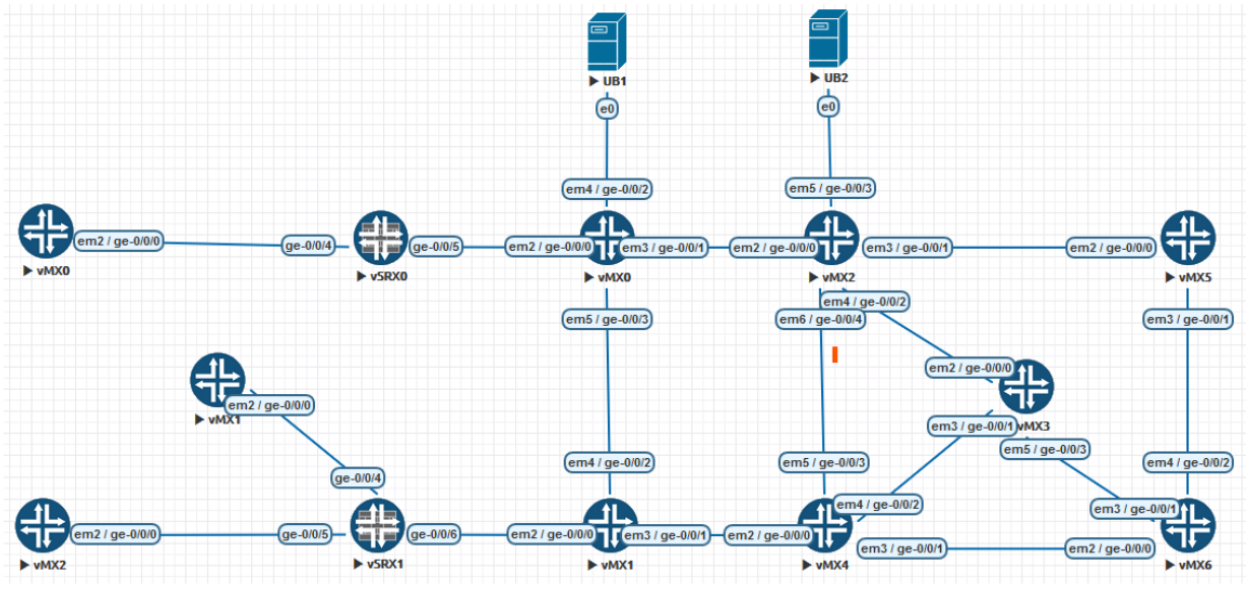


2. Platforma Lab Junipera na bazie usługi Google Cloud oraz EVE-NG

Emulated Virtual Environment
Next Generation

Free EVE Community Edition Version 6.2.0-4





https://www.eve-ng.net/index.php/documentation/community-cookbook/

2. Platforma Lab Junipera na bazie usługi Google Cloud oraz EVE-NG

Dlaczego wirtualizacja na EVE-NG

- Szybkie testy laboratoryjne, np. nowego projektu
- Brak dodatkowych kosztów sprzętu
- ŚWIETNE do certyfikacji
- Połącz vLabs, Mist oraz rzeczywiste urządzenia
- Rozpocznij bezpłatnie z Community Edition
- Niezależność od dostawcy
- Świetne do testów migracji
- Niezwykle elastyczny (Bare-Metal, VM | On-Prem | Kolokacja | Chmura)

Bare-Metal – Najlepsza wydajność

Vmware – Najlepsza elastyczność

Google Cloud

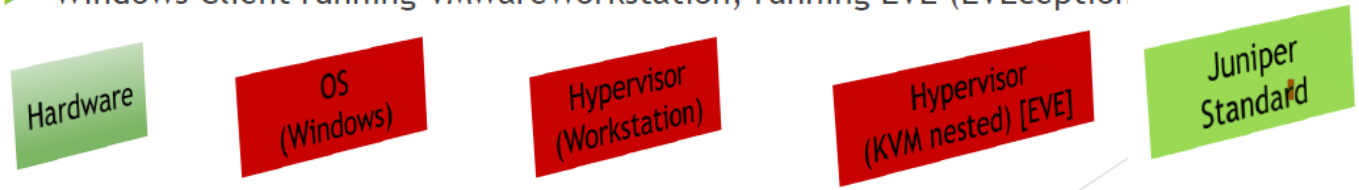
Cloud - najlepsza opcja „większej mocy”

2. Platforma Lab Junipera na bazie usługi Google Cloud oraz EVE-NG

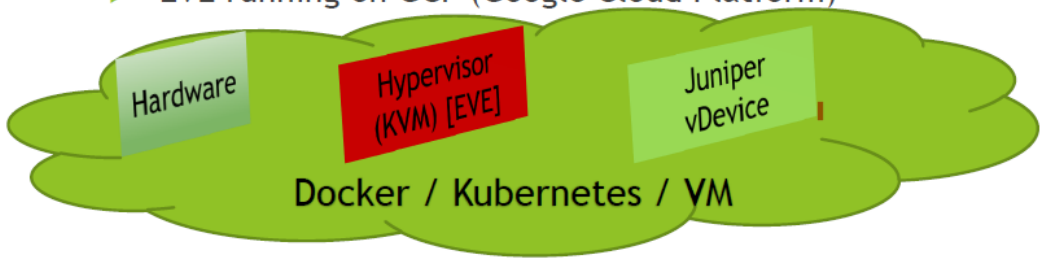
- Ubuntu installed on BMS

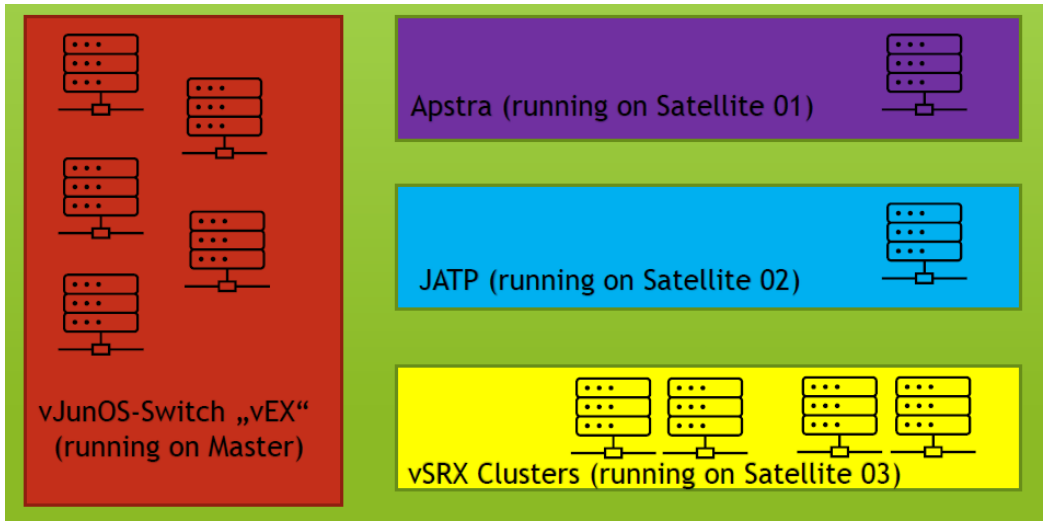
- ESXi installed on BMS and EVE as VM inside ESXi

- Windows Client running VMwareWorkstation, running EVE (EVEception)



- EVE running on GCP (Google Cloud Platform)

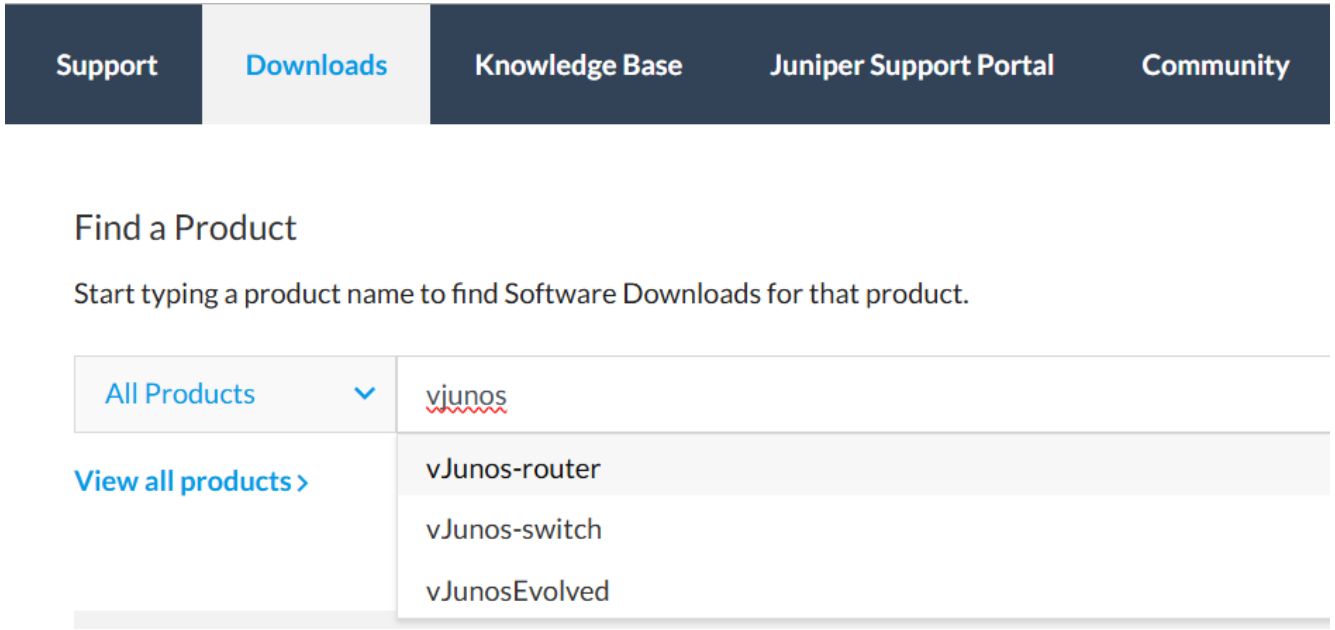




2. Platforma Lab Junipera na bazie usługi Google Cloud oraz EVE-NG

Zagadnienia dotyczące środowiska laboratoryjnego:

- Potrzebny DHCP?
- Potrzebny DNS?
- AD? Komputery zarządzające? Klienci? Serwery testowe? (funkcja commit w EVE-NG)
- Sieć zewnętrzna? Potrzebne laboratorium testowe
- Karty interfejsów na serwerze?
- Zapory sieciowe / NAT?
- VPN / MIST-Edge do laboratorium?
- Generatory ruchu? np. https://ostinato.org/
- Jedna duża topologia lub wiele mniejszych?



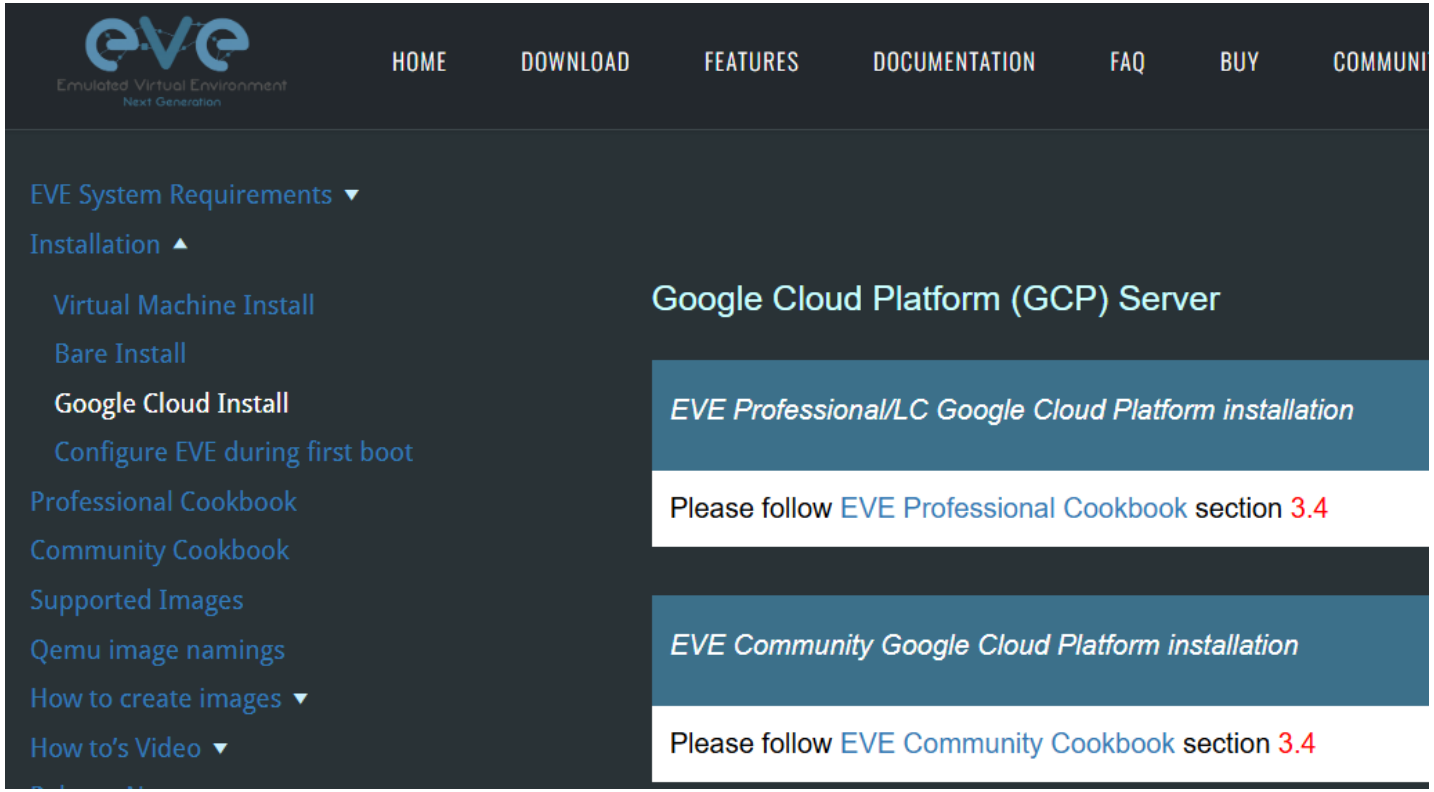
Virtual Route Reflector (vRR)
https://support.juniper.net/support/downloads/?p=virtual-route-reflector

Apstra Fabric Conductor
https://support.juniper.net/support/downloads/?p=apstra

JSA VM
https://support.juniper.net/support/downloads/?p=juniper-secure-analytics

2. Platforma Lab Junipera na bazie usługi Google Cloud oraz EVE-NG

Instalacja EVE-NG na GCP
https://www.eve-ng.net/index.php/documentation/installation/google-cloud-install/



2. Platforma Lab Junipera na bazie usługi Google Cloud oraz EVE-NG

Step 1: Connect to Google Cloud Platform (GCP)
https://console.cloud.google.com/getting-started

Step 2: Sign into GCP. Create a new GCP account if you do not already have one

Step 3: On the left side navigate to Compute Engine and press “Activate Cloud Shell”

Step 4: Create a nested Ubuntu 22.04 image.

```
gcloud compute images create nested-ubuntu-jammy --source-image-family=ubuntu-2204-lts --source-image-project=ubuntu-os-cloud --licenses
https://www.googleapis.com/compute/v1/projects/vm-options/global/licenses/enable-vmx
```

Step 5: Navigate: Navigation Menu/Compute Engine/VM Instances and press “CREATE INSTANCE”
- Assign the name for your VM
- Set your own region and zone
- Edit your Machine Configuration. General-Purpose. Choose the series of CPU platform, Preferred are Intel CPU Cascade Lake. Series N2 CPU
- Choose Machine Type your desirable CPU and RAM settings
- Select Custom images, select OS nested-ubuntu-jammy you created previously. Choose Boot Disk type: HDD disk type and size. HDD size can vary depends of your needs.
- Edit your Data Protection, select No backups.
- Edit your Networking Allow http traffic.
- Edit your Networking Allow http traffic.
- Edit Security and Disable Secure Boot and vTPM
- Create VM

Lab 1

Tworzenie środowiska wirtualizacji dla vJunos

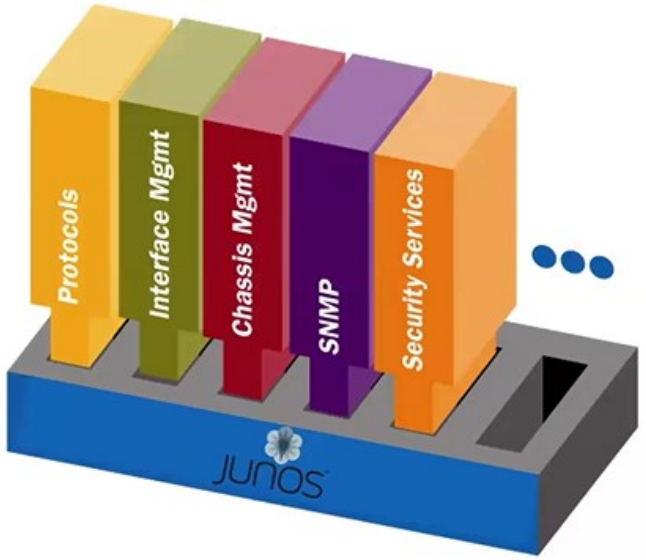
Connect to Google Cloud Platform

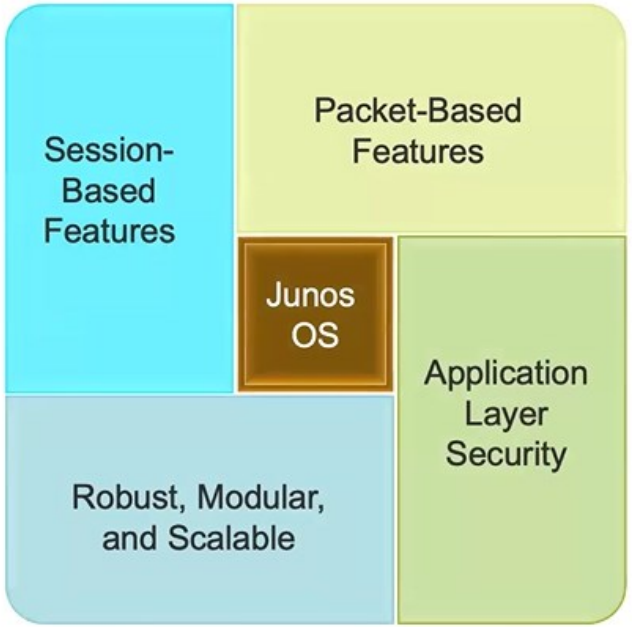
Introducing Junos OS and the Hardware

- Wprowadzenie do kursu i platformy warsztatów

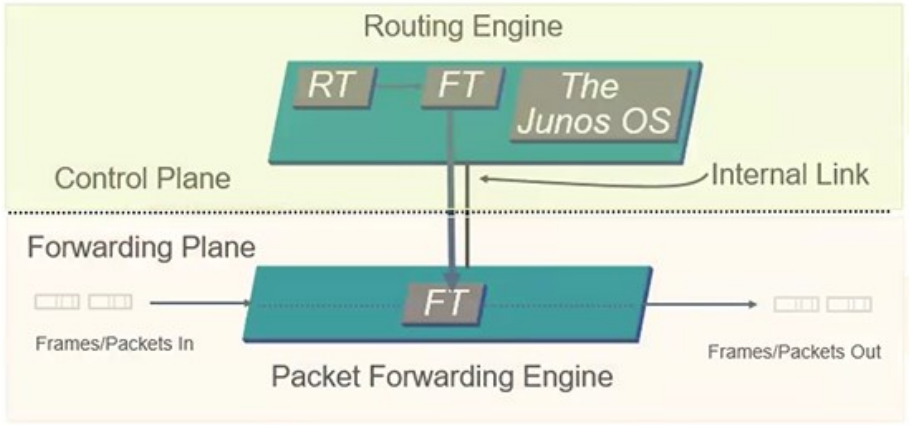
- **Moduł 0:** Introducing Junos OS and the Hardware
- **Moduł 1:** Junos Fundamentals
- **Moduł 2:** Junos User Interface
- **Moduł 3:** Junos Basic Configuration
- **Moduł 4:** Junos Logging and Monitoring
- **Moduł 5:** Junos Routing Policy and Firewall Filters
- **Moduł 6:** Junos Routing Fundamentals
- **Moduł 7:** Junos OS Switches
- **Moduł 8:** Junos OS Architecture
- **Moduł 9:** Junos J-Web
- **Moduł 10:** Jak przygotować się do JNCIA-Junos

Junos OS Architecture (1 of 3)



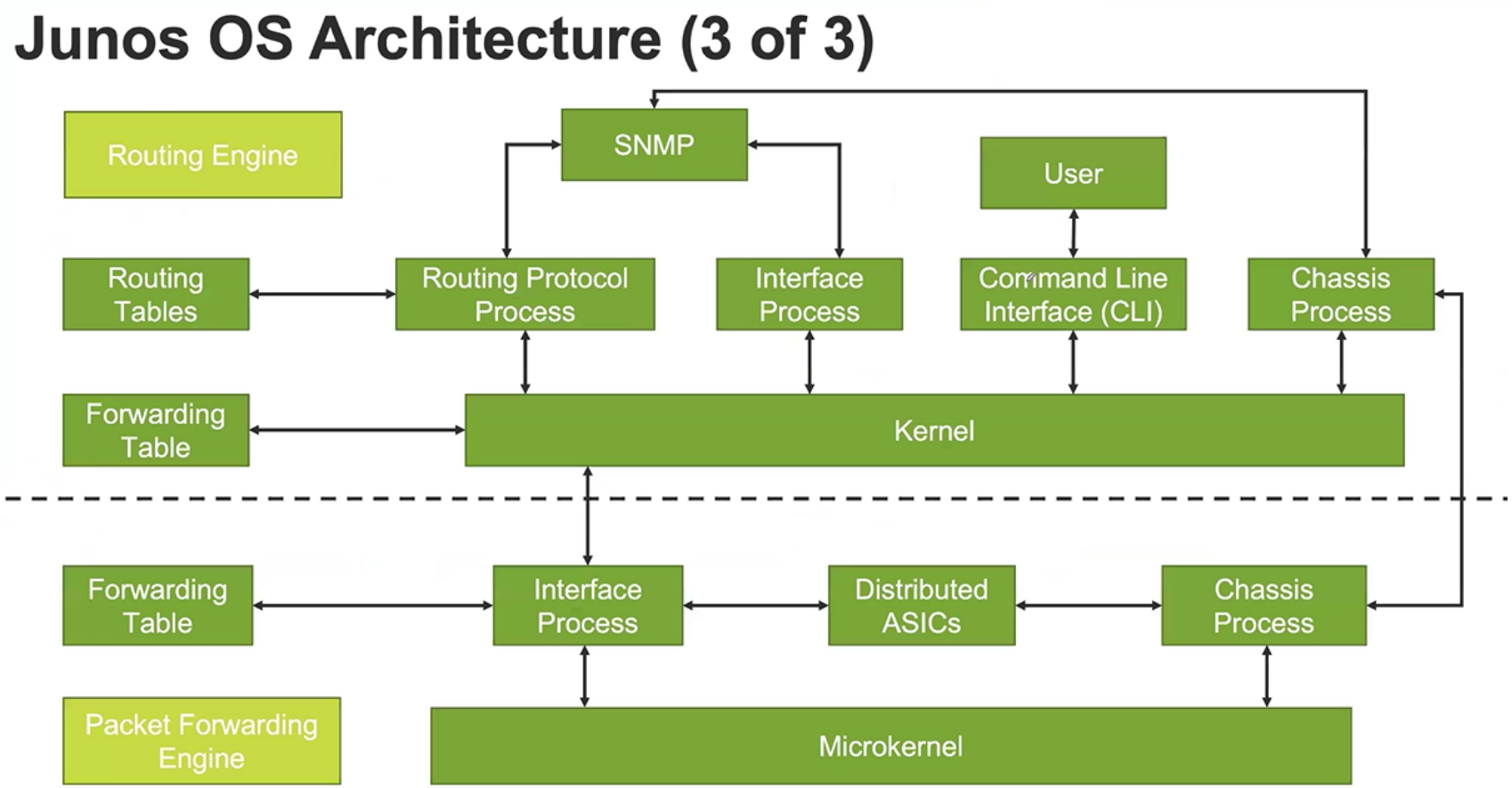


Junos OS Architecture (2 of 3)



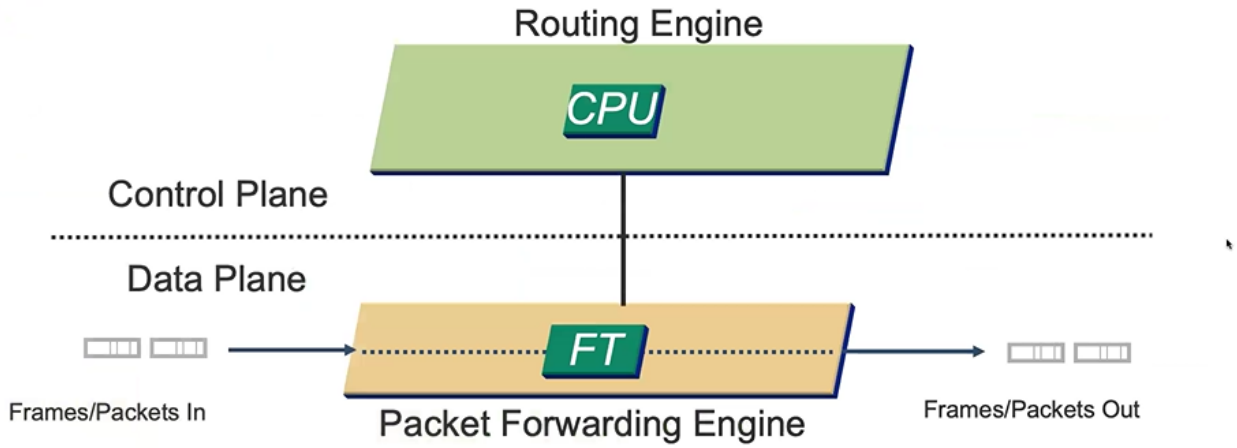
- Implemented on the RE
- Comprises Junos kernel, processes, chassis management, UI, routing protocols, system monitoring, and clustering control

- Implemented on the PFE
- Handles packet forwarding, session setup and maintenance, load balancing, security policy, screen options, VPN, and IDP



Transit Traffic Processing

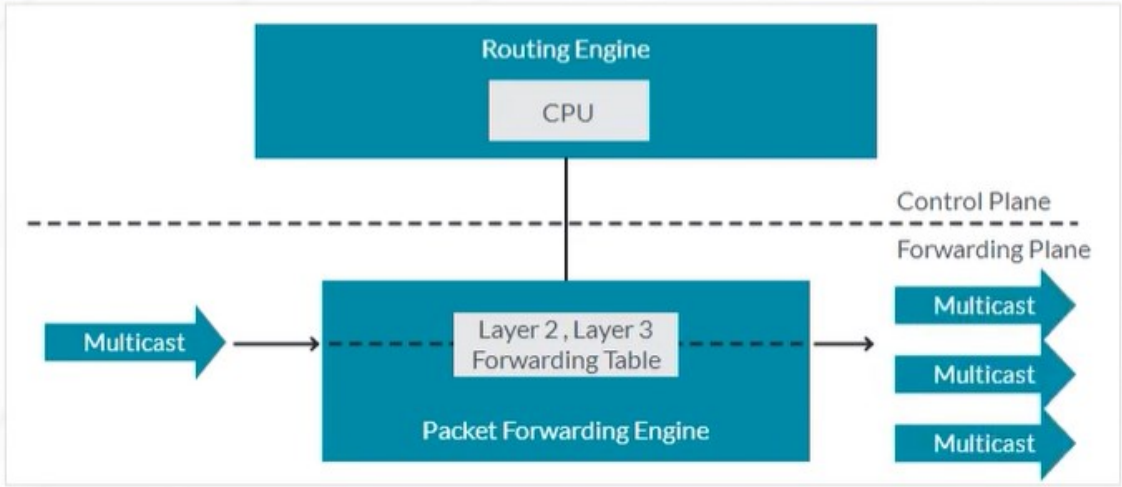
- Is forwarded through the local system
- RE provides forwarding table to PFE
- Examples of transit traffic include unicast and multicast traffic



Transit Traffic Processing

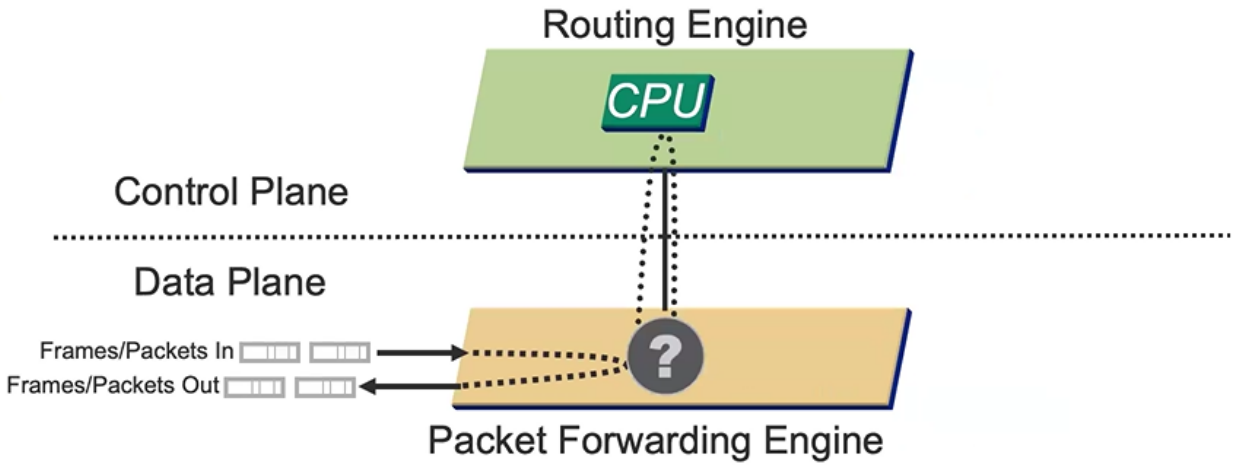
- Transit traffic is forwarded through the local system
  - Examples of transit traffic include unicast and multicast traffic

Unicast	Enters one ingress port and is transmitted out exactly one egress port toward its destination.
Multicast	Enters the transit device through a single ingress port and sent out through multiple egress ports after replication.



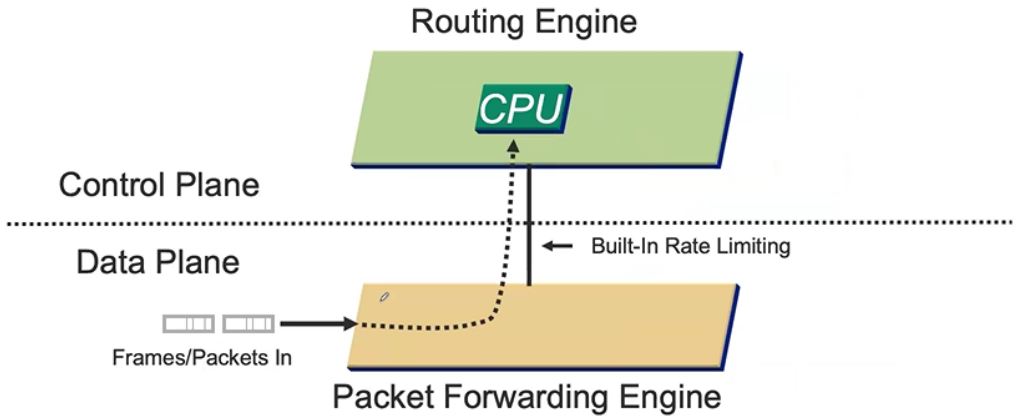
Exception Traffic Processing (1 of 2)

- Traffic destined for the local system is processed by RE CPU
- PFE processes special traffic including ICMP reply messages and TTL expired messages

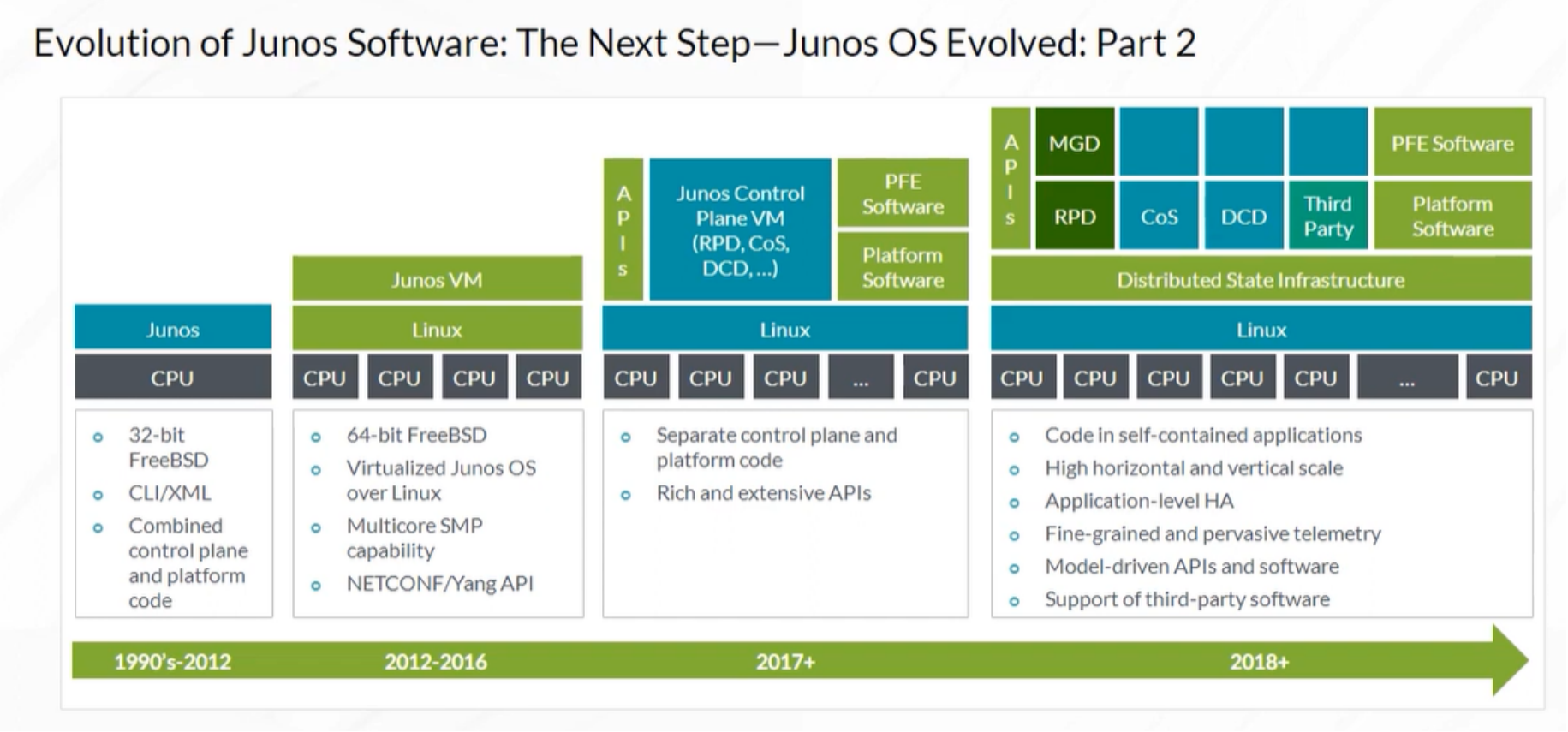


Exceptional Traffic Processing (2 of 2)

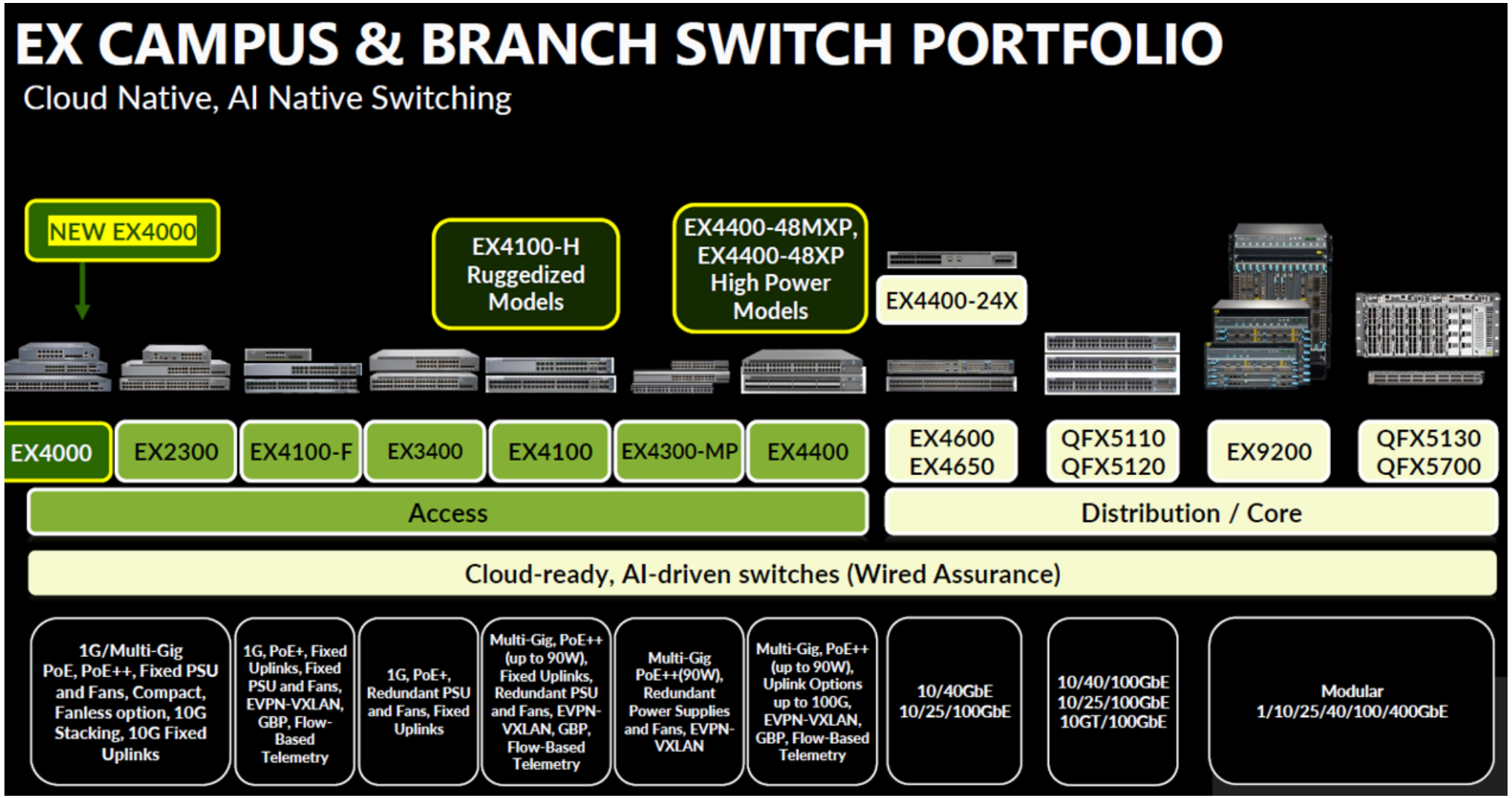
- Rate-limited on the internal link to protect the RE from potential DoS attacks
- Control traffic is given preference when congestion exists



Introducing Junos OS and the Hardware



Introducing Junos OS and the Hardware



Juniper SRX Series Portfolio

Branch


SRX300


SRX320


SRX340


SRX345


SRX380

Campus & Scale-Out DC/Cloud/SP


SRX1500


SRX4100


SRX4200


SRX4600

SRX1600

SRX2300

SRX4300

SRX4700

Scale-Up DC/Cloud/SP


SRX5800


SRX5600


SRX5400

Cloud Firewalls


vSRX


cSRX


Secure Edge

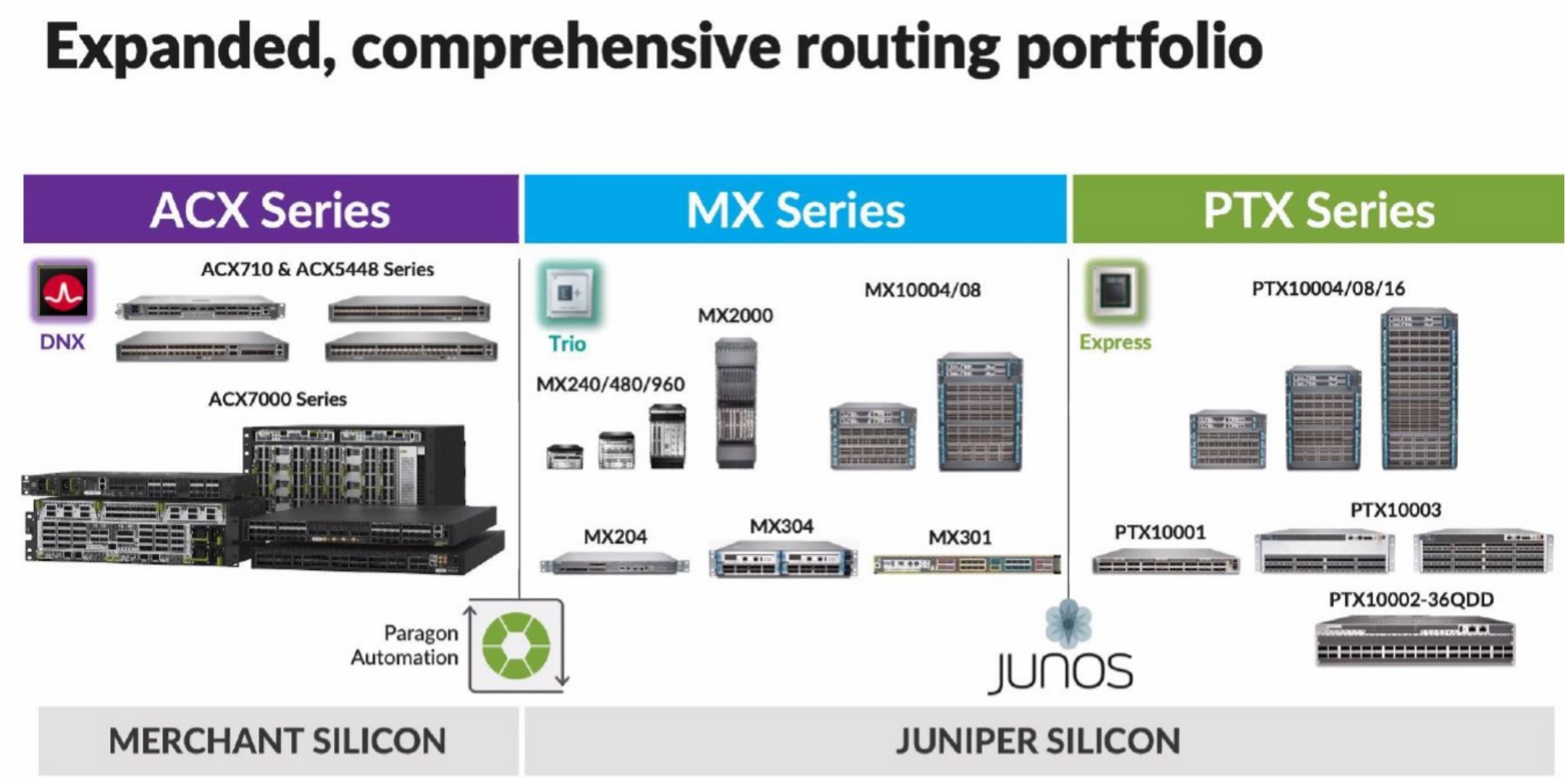
Cyberratings "AAA" Rating

Secure Edge "2022 Product of the Year" by CRN

Gartner

Enterprise Firewall Magic Quadrant **CHALLENGERS**

3-time Cybersecurity **Channel Champions** by Canalys



Junos Fundamentals

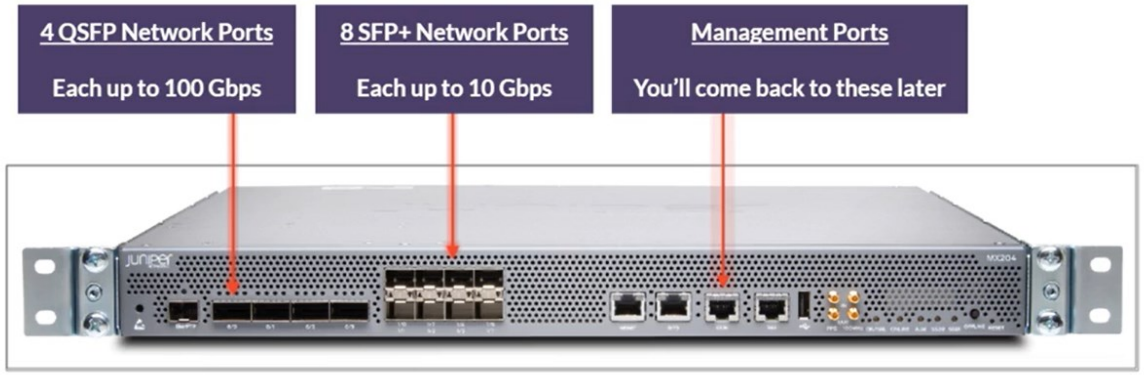
- Wprowadzenie do kursu i platformy warsztatów

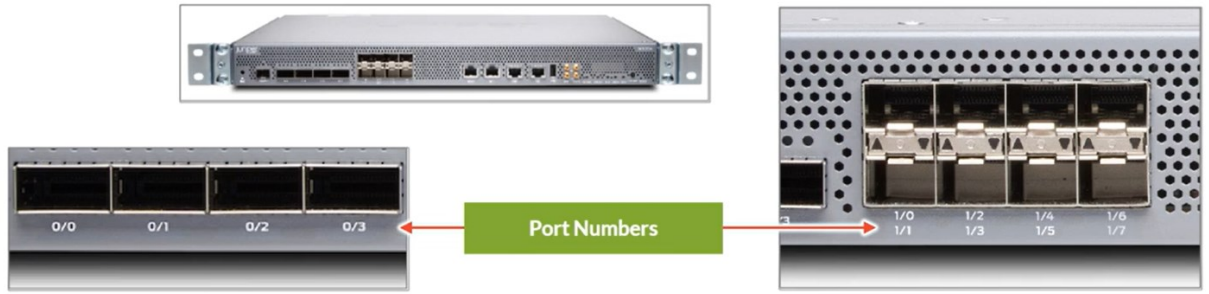
- **Moduł 0:** Introducing Junos OS and the Hardware
- **Moduł 1:** Junos Fundamentals
- **Moduł 2:** Junos User Interface
- **Moduł 3:** Junos Basic Configuration
- **Moduł 4:** Junos Logging and Monitoring
- **Moduł 5:** Junos Routing Policy and Firewall Filters
- **Moduł 6:** Junos Routing Fundamentals
- **Moduł 7:** Junos OS Switches
- **Moduł 8:** Junos OS Architecture
- **Moduł 9:** Junos J-Web
- **Moduł 10:** Jak przygotować się do JNCIA-Junos

Junos Fundamentals

- Networking devices can be routers, switches, firewalls, or a combination of these
- There are two broad categories of devices that run Junos OS

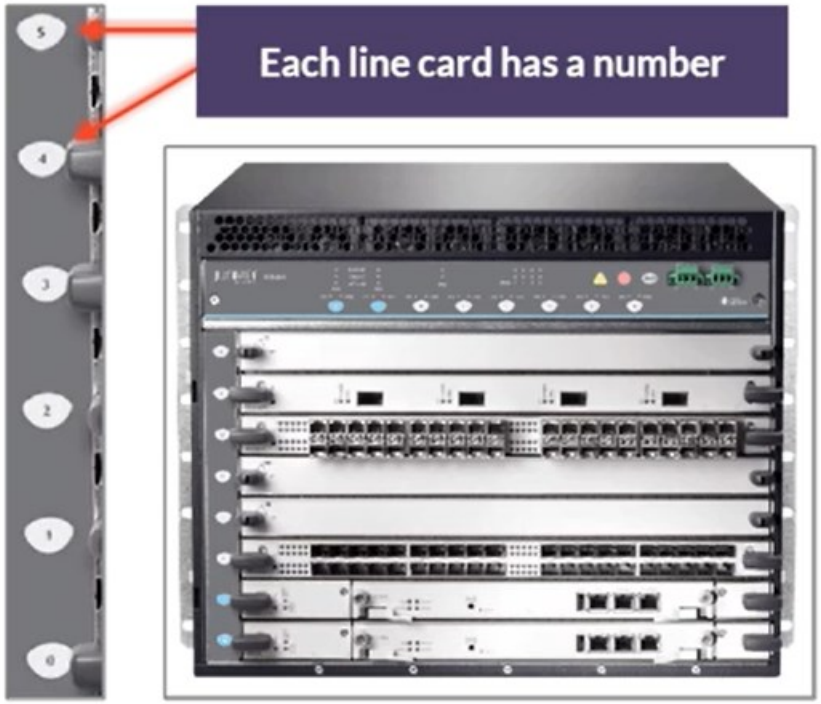
Fixed-Port Devices	Modular Devices
	





Juniper Hardware Components

- Major Router Hardware Components
  - Routing Engines
  - Control Board
  - Switch Interface Board (SIB)
  - Power Supply
  - Cooling System
  - Switch Control Board (SCB)
  - Dense Port Concentrators (DPC)
  - Flexible PIC Concentrators (FPC)
  - Physical Interface Card (PIC)



The Routing Engine is built into the device.

One primary RE, one backup RE, both swappable

These slots can take:

- Line cards with network interfaces
  - For transit traffic
- Line cards with extra features
  - Such as security services

Management Interfaces	Internal Interfaces	Network Interfaces	Loopback Interfaces	Services Interfaces
Connect to management network	Connect control and forwarding planes	Connect media-specific network	Logical interface–lo0, that is always up	○ User-configurable services such as encryption and tunneling • st0, gr, and ip

Interface Naming

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

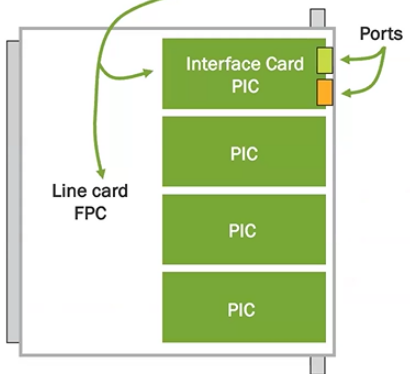
Note: While different platforms use different names for line cards and interface cards, the CLI almost always uses FPC and PIC.

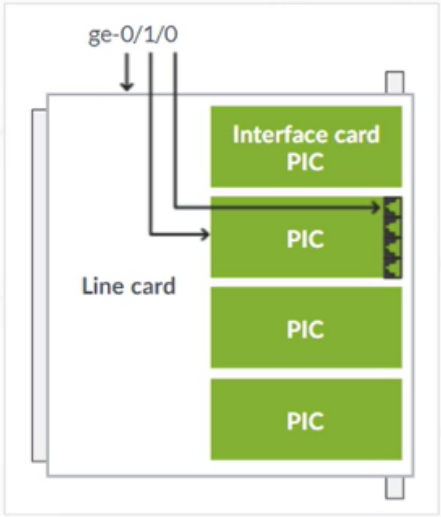
Interface naming example:

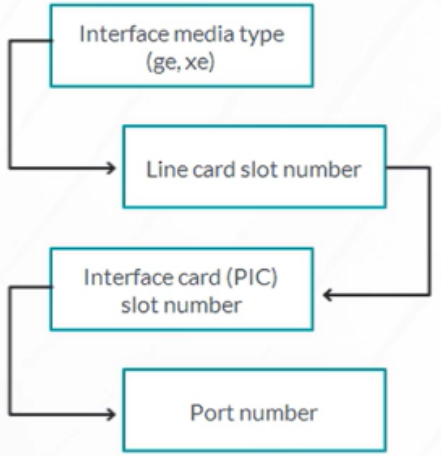
`ge-0/2/3` = port 3 of a Gigabit Ethernet PIC in slot 2 on FPC 0

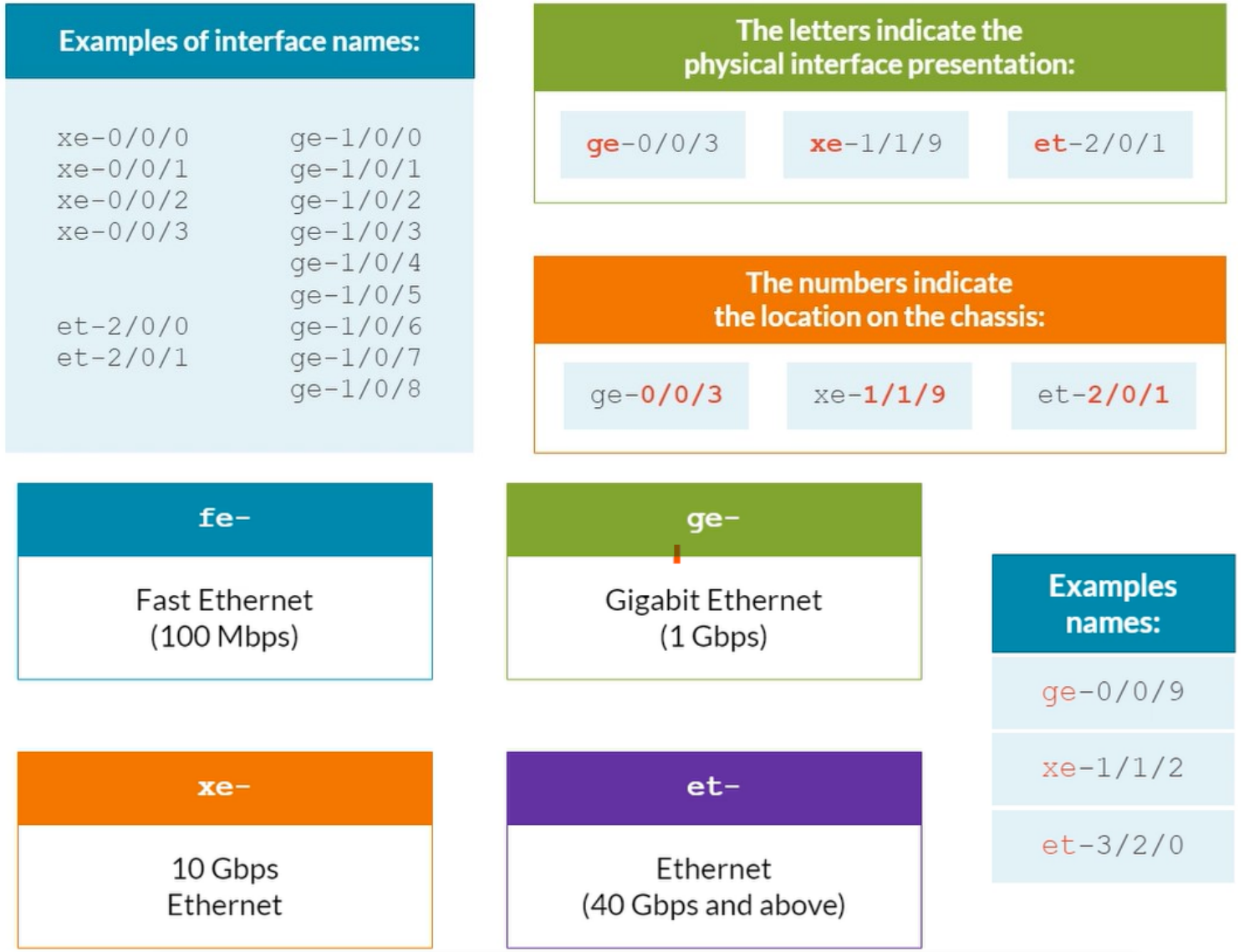
Note: Slot and port numbering begins with zero (0) rather than one (1).

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





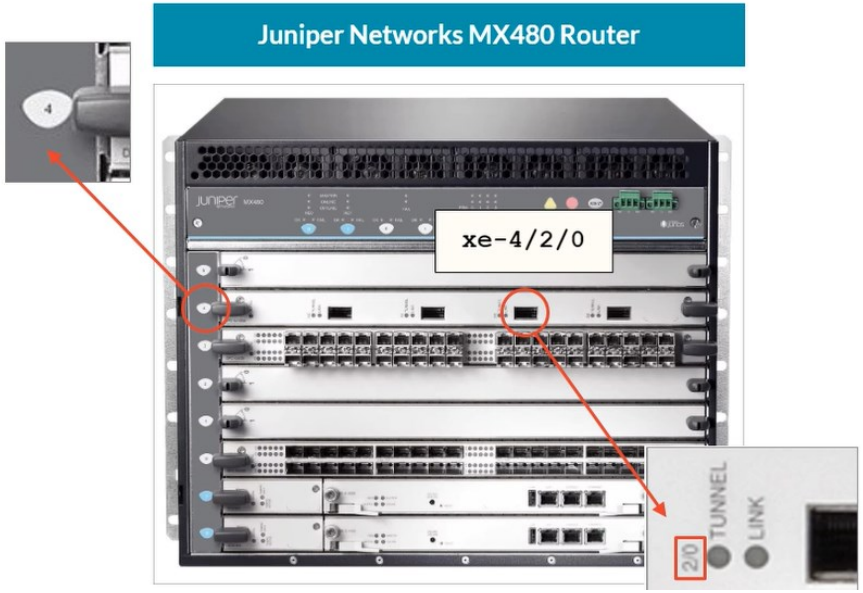


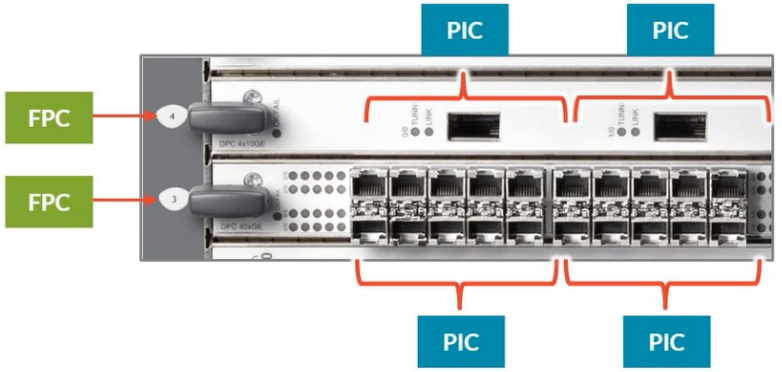


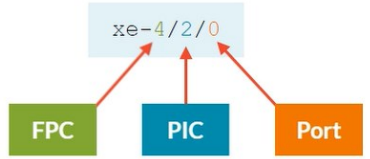
Junos Fundamentals

- The numbers indicate the location on the chassis
  - For example:
  - 4 = Line card
  - 2 = One of two things:
    - An interface card within a line card
    - A specific section of a line card
  - 0 = The port number itself

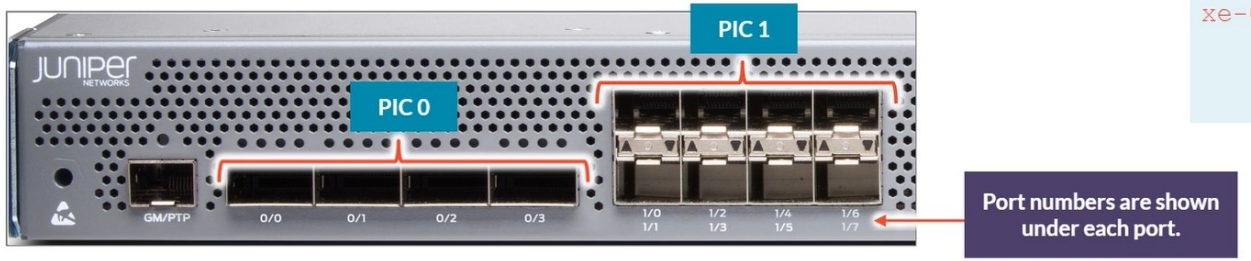
```
xe-4/2/0
```







- This interface naming scheme is used on all Junos devices
  - There is one consistent naming convention, on every single device
    - Even on devices without line cards
  - Your customer's MX204 has 12 network ports across two PICs
    - Four 100-Gbps QSFP ports, `0/0/0` to `0/0/3`
    - Eight 10-Gbps SFP+ ports, `0/1/0` to `0/1/7`




```
employee@R1> show interfaces terse xe-0/1/1
Interface               Admin Link Proto    Local                 Remote
xe-0/1/1                up    up
xe-0/1/1.0              up    up   inet     172.16.10.1/24
                                   inet6    2001:db8:0:10::1/64
                                            fe80::3a4f:49ff:fe80:f408/64
                                   multiservice
```

Logical Units

- Logical unit and encapsulation
  - Logical unit is always required in Junos OS
  - Some encapsulations support only one logical unit
  - Logical unit numbers and circuit identifiers do not need to match
  - Support multiple protocol addresses

For example—ge-0/0/14.51

Junos OS strives for consistency

- Physical properties always go on the physical interface.
  - Physical MTU
  - Speed
  - Duplex
- Logical properties always go on the logical interface.
  - Layer 3 MTU
  - IP addresses
  - VLAN tags

```
employee@R1> show interfaces terse xe-0/1/1
Interface               Admin Link Proto    Local                 Remote
xe-0/1/1                up    up
xe-0/1/1.0              up    up   inet     172.16.10.1/24
                                  inet6    2001:db8:0:10::1/64
                                           fe80::3a4f:49ff:fe80:f408/64
                                  multiservice
```

xe-0/1/1.0 is the logical interface.

Notice that IP protocols are on the logical interface.

```
employee@R1> show interfaces descriptions
Interface       Admin Link Description
xe-0/1/1        up    up   Corporate LAN
xe-0/1/2        up    up   Server LAN
xe-0/1/3        up    up   Guest LAN
xe-0/1/6        up    up   WAN - Connection to the Internet
```

Speed and MTU info

Can you find the Layer 2 MTU?

```
employee@R1> show interfaces xe-0/1/2
Physical interface: xe-0/1/2, Enabled, Physical link is Up
  Interface index: 167, SNMP ifIndex: 519
  Description: Server LAN
  Link-level type: Ethernet, MTU: 1514, MRU: 1522, LAN-PHY mode, Speed: 10Gbps,
  BPDU Error: None, Loop Detect PDU Error: None, MAC-REWRITE Error: None,
  Loopback: None, Source filtering: Disabled, Flow control: Enabled,
  Speed Configuration: Auto
  Pad to minimum frame size: Disabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Schedulers     : 0
  Current address: 38:4f:49:80:f4:09, Hardware address: 38:4f:49:80:f4:09
  Last flapped   : 2023-09-21 11:45:06 UTC (1w3d 09:01 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
```

```
employee@R1> show interfaces xe-0/1/2
Physical interface: xe-0/1/2, Enabled, Physical link is Up
  Interface index: 167, SNMP ifIndex: 519
  Description: Server LAN
  Link-level type: Ethernet, MTU: 1514, MRU: 1522, LAN-PHY mode, Speed: 10Gbps,
  BPDU Error: None, Loop Detect PDU Error: None, MAC-REWRITE Error: None,
  Loopback: None, Source filtering: Disabled, Flow control: Enabled,
  Speed Configuration: Auto
  Pad to minimum frame size: Disabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Schedulers     : 0
  Current address: 38:4f:49:80:f4:09, Hardware address: 38:4f:49:80:f4:09
  Last flapped   : 2023-09-21 11:45:06 UTC (1w3d 09:01 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
{snip - we've cut 10 lines of statistics that you'll return to later}
{The output continues on the next slide}
```

```
employee@R1> show interfaces xe-0/1/2
{snip - output below is continued from the previous slide}

 Logical interface xe-0/1/2.0 (Index 350) (SNMP ifIndex 545)
    Flags: Up SNMP-Traps 0x4004000 Encapsulation: ENET2
    Input packets : 502467
    Output packets: 32319
    Protocol inet, MTU: 1500
    Max nh cache: 100000, New hold nh limit: 100000, Curr nh cnt: 0,
    Curr new hold cnt: 0, NH drop cnt: 0
      Flags: Sendbcast-pkt-to-re
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 172.16.200/24, Local: 172.16.200.1, Broadcast:
172.16.200.255
    Protocol inet6, MTU: 1500
    Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 0,
    Curr new hold cnt: 0, NH drop cnt: 0
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 2001:db8:0:20::/64, Local: 2001:db8:0:20::1
      Addresses, Flags: Is-Preferred
        Destination: fe80::/64, Local: fe80::3a4f:49ff:fe80:f409
```

IPv6 (inet6) info starts here.

On many keyboards:
Shift + Backslash = |

In this module, we will show you:

- **count**
- **except**
- **find**
- **match**

```
employee@R1> show interfaces terse | ?
Possible completions:
  append               Append output text to file
  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
  refresh              Refresh a continuous display of the command
  request              Make system-level requests
  resolve              Resolve IP addresses
  save                 Save output text to file
  tee                  Write to standard output and file
  trim                 Trim specified number of columns from start of line
```

- Use `match` to only show lines of output that contain certain text
  - Very helpful with long output—you can display only the pieces you want to see
  - This output shows all `xe` interfaces on PIC 1 that are `down`

Very helpful command during troubleshooting!

```
employee@R1> show interfaces terse xe-0/1/* | match down
xe-0/1/0                up    down
xe-0/1/0.16386          up    down
xe-0/1/4                up    down
xe-0/1/4.16386          up    down
xe-0/1/5                up    down
xe-0/1/5.16386          up    down
xe-0/1/7                up    down
xe-0/1/7.16386          up    down
```

Junos adds dummy units to unconfigured interfaces.

- You can even search for multiple strings of text at once
  - The syntax is `match "physical|flapped"`
    - Put a pipe between each string of text and wrap the whole thing in quotes
  - The pipe in quotes is a logical OR
    - A line matches if it contains either word

Quick way to see if any interfaces have flapped recently.

We've color-coded some interfaces to help them stand out.

```
employee@R1> show interfaces xe-0/1/* | match "physical|flapped"
Physical interface: xe-0/1/0, Enabled, Physical link is Up
  Last flapped   : 2023-09-21 11:45:06 UTC (1w3d 09:06 ago)
Physical interface: xe-0/1/1, Enabled, Physical link is Up
  Last flapped   : 2023-09-21 11:45:06 UTC (1w3d 09:06 ago)
Physical interface: xe-0/1/2, Enabled, Physical link is Up
  Last flapped   : 2023-09-21 11:45:06 UTC (1w3d 09:06 ago)
Physical interface: xe-0/1/3, Enabled, Physical link is Up
  Last flapped   : 2023-09-21 11:45:08 UTC (1w3d 09:06 ago)
Physical interface: xe-0/1/4, Enabled, Physical link is Down
  Last flapped   : 2023-09-21 11:45:02 UTC (1w3d 09:06 ago)
Physical interface: xe-0/1/5, Enabled, Physical link is Down
  Last flapped   : 2023-09-21 11:45:02 UTC (1w3d 09:06 ago)
```

Multiple IPs on Interfaces

The reasons for adding multiple IPs to one interface, and the consequences for the source IP of traffic from that device

Virtual Interfaces

You'll learn about IRB interfaces on Layer 3 switches, and loopback interfaces for protocols and management

Management and Console Ports

A variety of ports dedicated to different management methods

Interfaces with Multiple Units

The solution to your customer's poorly designed LAN network

Hierarchy View

Displays the configuration as a series of indented tabs.

```
interfaces {
    xe-0/1/1 {
        description "Corporate LAN";
        mtu 1800;
        unit 0 {
            family inet {
                address 172.16.10.1/24;
            }
            family inet6 {
                address 2001:db8:0:10::1/64;
```

Notice how elements of this interface are indented below it.

The same is true of this interface.

```
interfaces {
    xe-0/1/1 {
        description "Corporate LAN";
        mtu 1800;
        unit 0 {
            family inet {
                address 172.16.10.1/24;
            }
            family inet6 {
                address 2001:db8:0:10::1/64;
            }
        }
    }
    xe-0/1/2 {
        description "Server LAN";
        unit 0 {
            family inet {
```

Set View

Displays the actual commands that you type to configure the device.

```
set interfaces xe-0/1/1 description "Corporate LAN"
set interfaces xe-0/1/1 mtu 1800
set interfaces xe-0/1/1 unit 0 family inet address 172.16.10.1/24
set interfaces xe-0/1/1 unit 0 family inet6 address 2001:db8:0:10::1/64
set interfaces xe-0/1/2 description "Server LAN"
set interfaces xe-0/1/2 unit 0 family inet address 172.16.200.1/24
set interfaces xe-0/1/2 unit 0 family inet6 address 2001:db8:0:20::1/64
set interfaces xe-0/1/3 description "Guest LAN”
set interfaces xe-0/1/3 unit 0 family inet mtu 1000
```

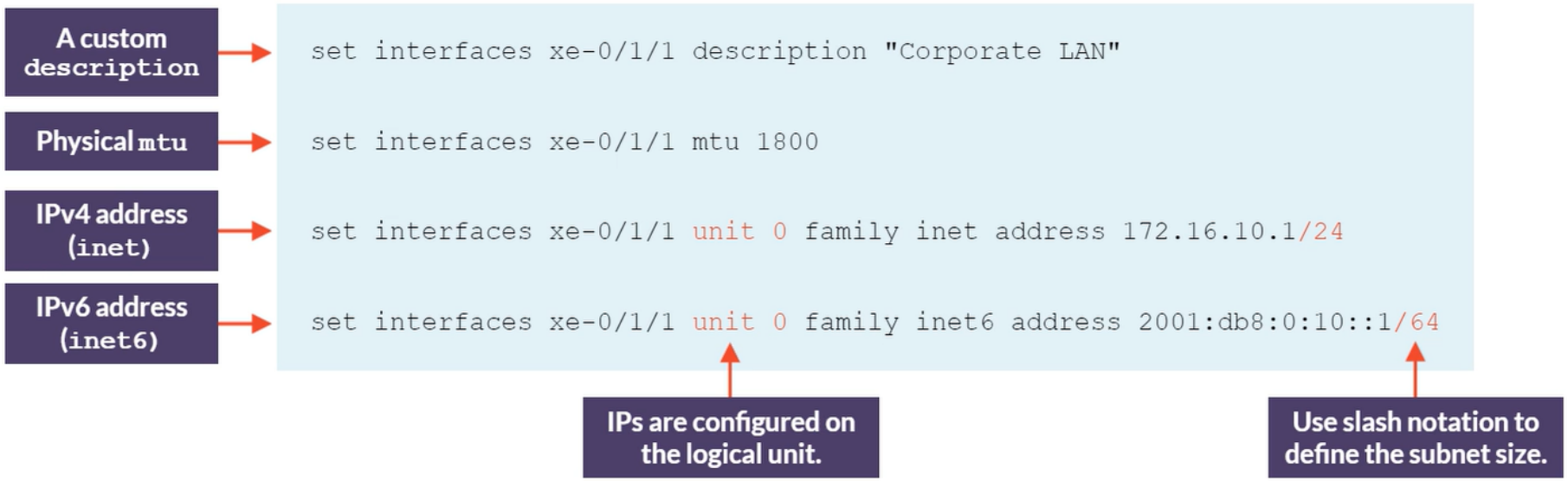
```
employee@R1> show configuration | display set
set version 22.4R1.10
set system host-name R1
set system root-authentication encrypted-password
"$6$YS4u.arp$bebdbutypFVSpqJo33nXvYlMEEnUh9yRHD4Vq8ka03lR7CI.xKdb0pf8
set system login user employee uid 2000
set system login user employee class super-user
set system login user employee authentication encrypted-password
"$6$gjAdQdwK$LTqWOAjOLYA8BQjSNGmQYG9iO.VXoyIV8dVrkfPuE87fNczS6d8sGyra
set system services ssh
set system services telnet
set system syslog file interactive-commands interactive-commands any
```

```
employee@R1> show configuration system | display set
set system host-name R1
set system root-authentication encrypted-password "$6$YS4u.arp
$bebdbutypFVSpqJo33nXvYlMEEnUh9yRHDka03lR7CI.xKdb0pf8cExd7EZslsMBIbFCO00"
{snip}
```

```
employee@R1> show configuration interfaces | display set
set interfaces xe-0/1/1 description "Corporate LAN"
set interfaces xe-0/1/1 mtu 1800
{snip}
```

```
employee@R1> show configuration protocols | display set
set protocols lldp interface xe-0/1/1
set protocols lldp interface xe-0/1/2
{snip}
```

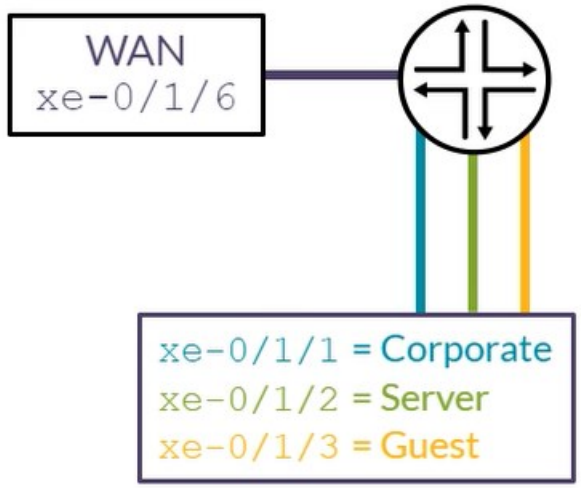
- These four lines define the interface settings on `xe-0/1/1`
  - Read them slowly—they are quite easy to understand
  - Notice that the IP settings are on `unit 0`
    - Logical settings are set on the logical interface




```
employee@R1> show configuration interfaces | display set | match mtu

set interfaces xe-0/1/1 mtu 1800
set interfaces xe-0/1/3 unit 0 family inet mtu 900
```

Show the config in set format and search for the text `mtu`.

```
employee@R1> show configuration interfaces | display set

set interfaces xe-0/1/1 description "Corporate LAN"
set interfaces xe-0/1/1 mtu 1800
set interfaces xe-0/1/1 unit 0 family inet address 172.16.10.1/24
set interfaces xe-0/1/1 unit 0 family inet6 address 2001:db8:0:10::1/64

set interfaces xe-0/1/2 description "Server LAN"
set interfaces xe-0/1/2 unit 0 family inet address 172.16.200.1/24
set interfaces xe-0/1/2 unit 0 family inet6 address 2001:db8:0:20::1/64

set interfaces xe-0/1/3 description "Guest LAN"
set interfaces xe-0/1/3 unit 0 family inet mtu 900
set interfaces xe-0/1/3 unit 0 family inet address 172.16.33.1/24
set interfaces xe-0/1/3 unit 0 family inet6

set interfaces xe-0/1/6 description "WAN - Connection to the Internet"
set interfaces xe-0/1/6 unit 0 family inet address 10.1.254.1/24
set interfaces xe-0/1/6 unit 0 family inet6 address 2001:db8:1:254::1/64
```



Lab 2

Tworzenie środowiska wirtualizacji dla vJunos

EVE-NG jak ustawić dostęp do Internetu

Junos User Interface

- Wprowadzenie do kursu i platformy warsztatów

- **Moduł 0:** Introducing Junos OS and the Hardware
- **Moduł 1:** Junos Fundamentals
- **Moduł 2:** Junos User Interface
- **Moduł 3:** Junos Basic Configuration
- **Moduł 4:** Junos Logging and Monitoring
- **Moduł 5:** Junos Routing Policy and Firewall Filters
- **Moduł 6:** Junos Routing Fundamentals
- **Moduł 7:** Junos OS Switches
- **Moduł 8:** Junos OS Architecture
- **Moduł 9:** Junos J-Web
- **Moduł 10:** Jak przygotować się do JNCIA-Junos

Junos User Interface

Junos CLI Basics

Junos CLI	• Text-based command shell
	• Available through the virtual console on virtualized devices
	• Accessible through the console port using a terminal emulation program • Uses RJ-45 RS232 @ 9600 bps, 8/1/N (non-configurable)

	• Accessible through network ports using management protocol such as Telnet or SSH program
	  • Requires network interface and related configuration
	  • Many Junos devices include a dedicated management Ethernet interface used for out-of-band access

Junos User Interface

Logging In

- 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 21.2R1.10 Kernel 64-bit XEN JNPR-12.1-20210529.2f59a40_buil
root@router%
root@router% cli
root@router>
```

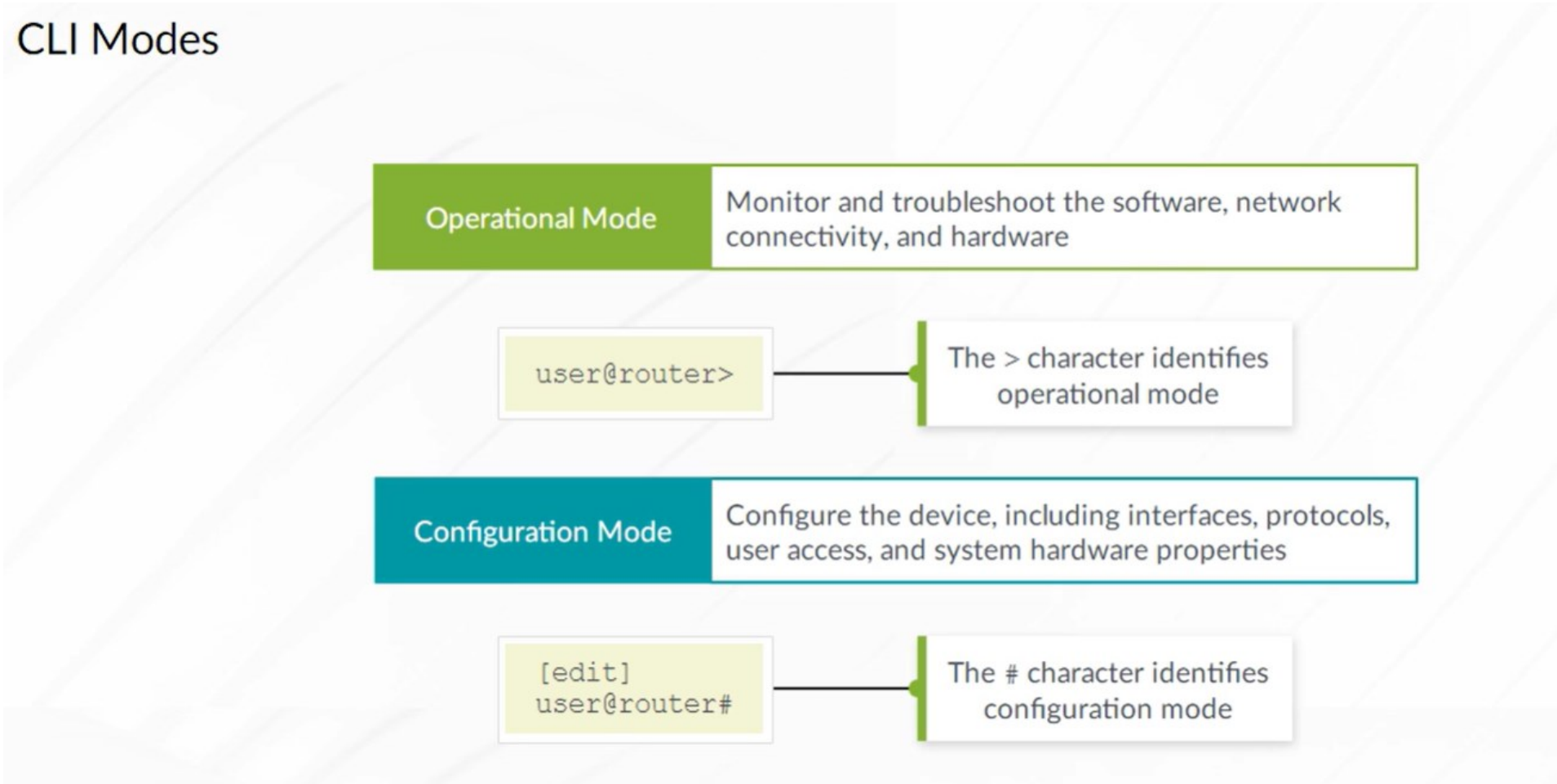
Shell Prompt

CLI Prompt

```
root@router:~ # cli
```

In vSRX, the root login prompt from the shell shows # instead of %.

Junos User Interface



Context-Sensitive Help

- CLI provides context-sensitive help at any point in a command line
- 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
  . . .
```

Junos User Interface

Topical Help

The **help topic** command 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.
   ...
```

Help With Configuration Syntax

The **help reference** command 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]

...
```

Commands and Variable Completion

- Use the Tab key to complete commands and variables:

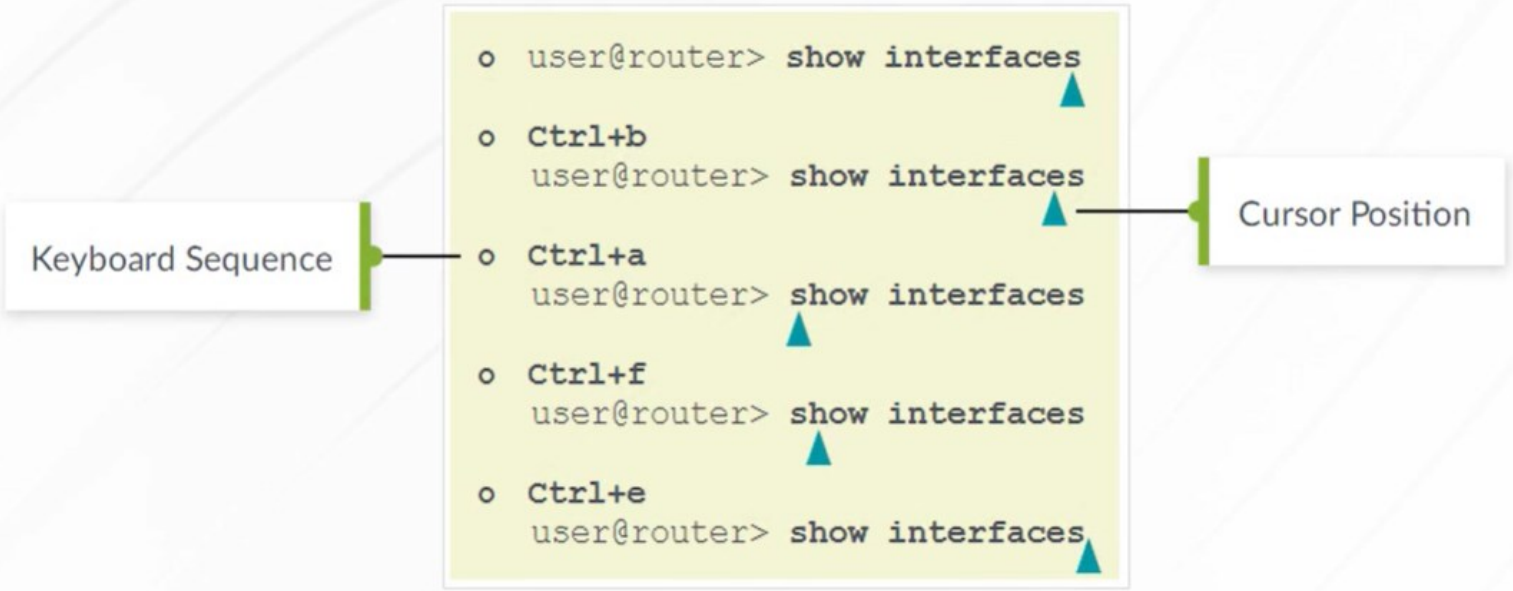
```
[edit policy-options]
user@router# show policy-statement t<tab>his-is-my-policy
then accept;

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

Press **Tab** to complete assigned variables.

Junos User Interface

Editing Command Lines

- Emacs-style editing sequences are supported:



Junos User Interface

Editing Command Lines

Supported keyboard sequences are:

Keyboard Sequence	Function
`Ctrl+b`	Moves the cursor left one character
`Ctrl+a`	Moves the cursor to the beginning of the command line
`Ctrl+f`	Moves the cursor right one character
`Ctrl+e`	Moves the cursor to the end of the command line
`Delete` and `Backspace`	Deletes the character before the cursor
`Ctrl+d`	Deletes the character over the cursor
`Ctrl+k`	Deletes from the cursor to the end of the line
`Ctrl+u`	Deletes all characters and negates the current command
`Ctrl+w`	Deletes the entire word to the left of the cursor
`Ctrl+l`	Redraws the current line
`Ctrl+p`, `Ctrl+n`	Repeats the previous and next command in the command history, respectively
`Esc+d`	Deletes the word to the right
`Esc+b`	Moves the cursor back one word with no delete
`Esc+f`	Moves the cursor forward one word with no delete
`Esc` key	Release the key and press it again for each occurrence
`Ctrl` key	Hold down for multiple occurrences

Junos User Interface

Using | (Pipe)

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

```
user@router> show route | ?
Possible completions:
  append               Append output text to file
  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
  tee                  Write to standard output and file
  trim                 Trim specified number of columns from start of line
```

Junos User Interface

Using | (Pipe)

List of pipe options available are:

Pipe Option	Description
`append <file-name>`	Append the output to a specified `<file-name>`
`compare <filename> | rollback <n>`	Available in the configuration mode using only the `show` command. Compares configuration changes with another configuration file.
`count`	Displays the number of lines in the output
`display changed`	Available only in the configuration mode. Tags changes with `junos:changed` attribute only for XML use.
`display commit-scripts`	Shows data after Junos OS applies commit scripts
`display detail`	Available only in the configuration mode. Displays additional information about the contents of the configuration.
`display inheritance`	Available only in the configuration mode. Displays inherited configuration data and source group.
`display omit`	Available only in the configuration mode. Omits configuration statements with the `omit` option.
`display set`	Available only in the configuration mode. Shows `set` commands that created configuration statements.
`display xml`	Displays the output in Network Configuration Protocol/XML format

Junos User Interface

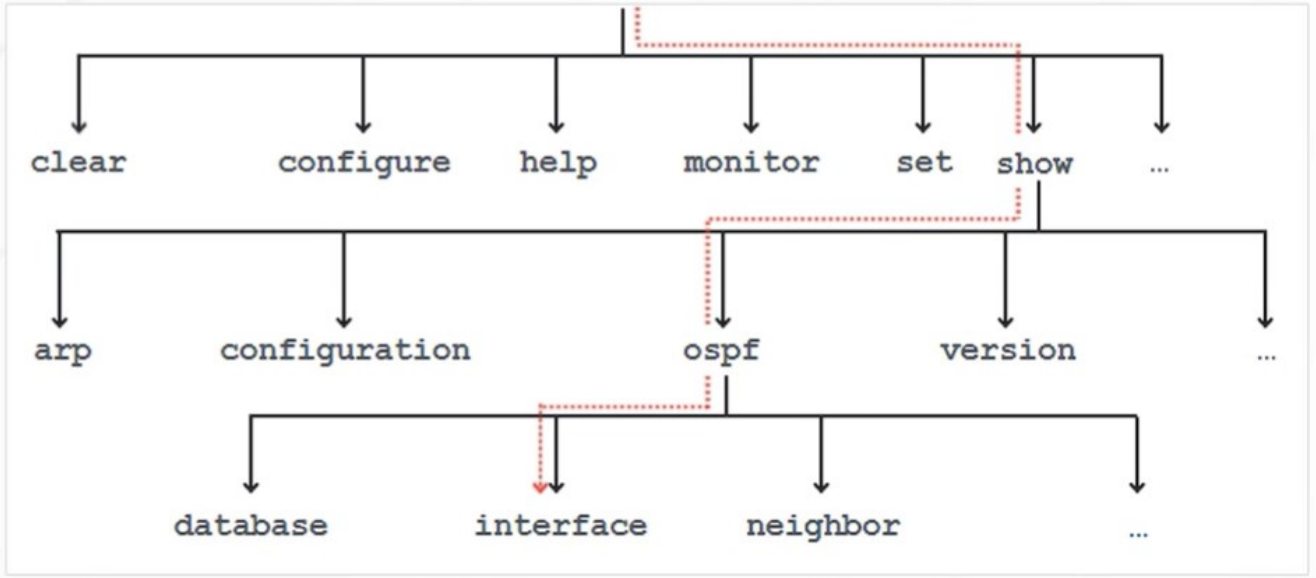
CLI Operational Mode

- Execute operational mode commands to monitor and control the operation of devices running Junos OS
  - Hierarchy of commands
- Junos OS also adds additional flexibility through the **run** command

```
user@router> show ospf interface
```

Less Specific

More Specific



Operational Mode Capabilities

- Key operational mode capabilities include:
  - Entering the configuration mode
  - Controlling the CLI environment
  - Exiting the CLI
  - Monitoring and troubleshooting
  - Copying files
  - Restarting software processes

```
user@router> ?
Possible completions:
  clear                Clear information in the system
  configure            Manipulate software configuration information
  file                 Perform file operations
  help                 Provide help information
  load                 Load information from file
  monitor              Show real-time debugging information
  mtrace               Trace multicast path from source to receiver
  op                   Invoke an operation script
  ping                 Ping remote target
  quit                 Exit the management session
  request              Make system-level requests
  restart              Restart software process
  save                 Save information to file
  set                  Set CLI properties, date/time, craft interface message
  show                 Show system information
  ssh                  Start secure shell on another host
  start                Start shell
  telnet               Telnet to another host
  test                 Perform diagnostic debugging
  traceroute           Trace route to remote host
```

Entering Configuration Mode: `configure` and `configure exclusive`

- Multiple users can enter the configuration mode and commit changes
- When any of the users editing the configuration issues a **`commit`** command, all changes made by all users are checked and activated
- Use the **`configure exclusive`** command 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#
```

Entering Configuration Mode: `configure private`

- Use the **`configure private`**, **`configure batch`**, and **`configure dynamic`** commands to customize the way configuration changes are handled by Junos OS
- Use the **`configure private`** command to enable 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
  - When a user is in the private mode, other users must enter the private mode or use **`configure exclusive`**
  - 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 pts/0 (pid 73413) on since 2021-04-07 10:22:42 UTC, idle 00:00:30
      private [edit]
[edit]
walter@router#
```

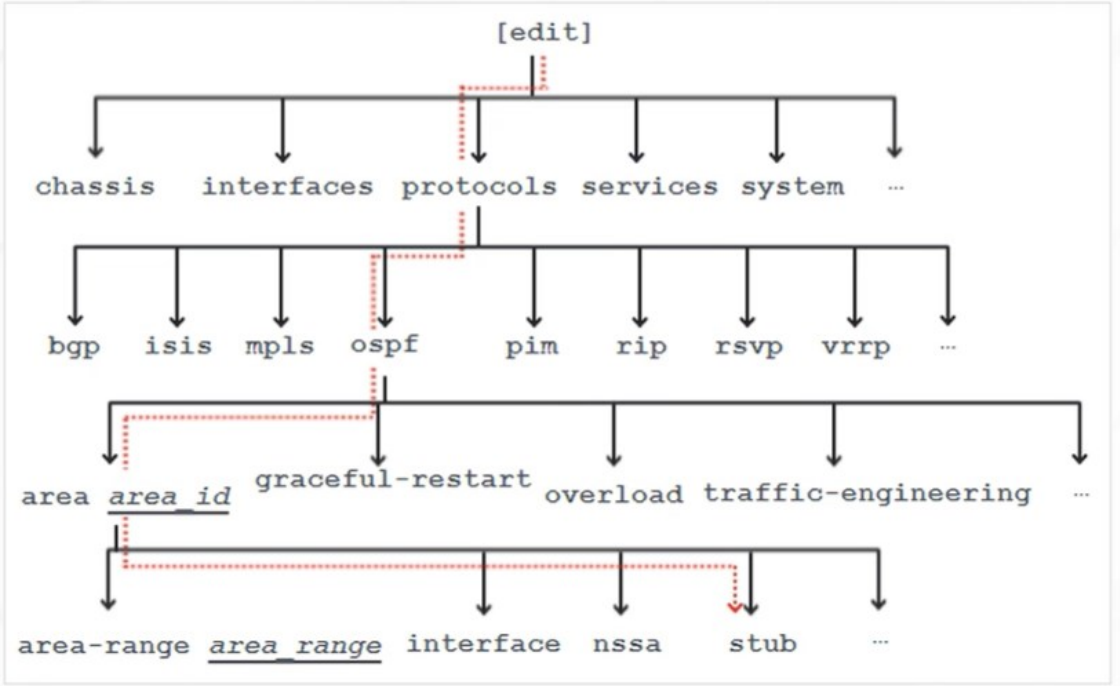
Enables other users to edit private copies of the candidate configuration.

Junos User Interface

Configuration Statement Hierarchy

- CLI operational mode: **show** command: To display specific operational information
- CLI configuration mode: **show** command: To display the statement hierarchy

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

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



Less Specific

More Specific

Configuration File is Hierarchical

- Enter CLI commands without curly brackets:

```
[edit system]
user@router# set services rest traceoptions level brief
```

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

```
[edit system]
user@router# show services
ssl {
    traceoptions {
        level brief;
    }
}
```

Junos User Interface

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 is added.

Removing Configuration Statements

- Use the **delete** command to remove statements:

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

[edit system services]
user@router# delete telnet

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

Junos User Interface

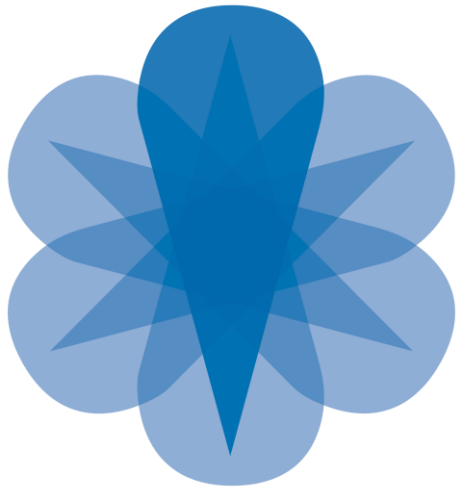
- Wprowadzenie do kursu i platformy warsztatów

- **Moduł 0:** Introducing Junos OS and the Hardware
- **Moduł 1:** Junos Fundamentals
- **Moduł 2:** Junos User Interface
- **Moduł 3:** Junos Basic Configuration
- **Moduł 4:** Junos Logging and Monitoring
- **Moduł 5:** Junos Routing Policy and Firewall Filters
- **Moduł 6:** Junos Routing Fundamentals
- **Moduł 7:** Junos OS Switches
- **Moduł 8:** Junos OS Architecture
- **Moduł 9:** Junos J-Web
- **Moduł 10:** Jak przygotować się do JNCIA-Junos

Junos Basic Configuration

Initial Configuration Checklist

- When the initial configuration is performed, the root authentication must be included
- Typically also includes:
  - Hostname
  - Domain name
  - Domain Name System (DNS) servers
  - Time and date
  - Users
- Must include root password (restrictions exist):
  - Minimum password length is six characters
  - Password requires change of case, digits, or punctuation



Junos Basic Configuration

Ustawienie hostname w Junos:
> configure
set system host-name NAZWA_HOSTA

Ustawienie domain-name w Junos:
set system domain-name example.local

Ustawienie DNS w Junos:
set system name-server 8.8.8.8
set system name-server 1.1.1.1

Ręczne ustawienie daty i czasu:
run set date YYYYMMDDHHMM (set date 202505271030)
run show system uptime

Ustawienie automatyczne – przez NTP:
set system ntp server 0.pool.ntp.org
set system ntp server 1.pool.ntp.org

Ustawienie hasła root przez CLI:
set system root-authentication plain-text-password
New password:
Retype new password:
#commit

Tworzenie nowego użytkownika:
set system login user admin1 class super-user
set system login user admin1 authentication plain-text-password
#commit

Klasa	Uprawnienia
`super-user`	🔧 **Pełne uprawnienia** – może wykonywać wszystkie operacje, w tym `commit`
`operator`	👁️ **Tylko operacyjne** – może wykonywać operacje, ale nie zmieniać konfiguracji
`read-only`	📖 **Tylko do odczytu** – może przeglądać konfigurację i status, ale nic nie może zmieniać
`unauthorized`	❌ **Brak dostępu** – użytkownik może się zalogować, ale nie ma uprawnień do żadnych komend
`maintenance`	⚙️ **Dostęp serwisowy** – używany głównie przez Juniper do celów serwisowych
`device-control`	🔌 Dostęp do funkcji zarządzania fizycznym urządzeniem (np. zasilanie, hardware)

Lab 3

Podstawy konfiguracji Junos OS (vJunos)

Wstępna konfiguracja routera SRX

Junos Basic Configuration

Wyłączenie zasilania (shutdown). Aby bezpiecznie wyłączyć urządzenie, najpierw należy zapisać konfigurację (jeśli zmieniałeś coś):

> request system power-off

Zatrzymanie systemu (shutdown/reboot w trybie awaryjnym). Jeśli chcesz zatrzymać system, ale nie wyłączać zasilania, można użyć:

> request system halt

Restart (reboot) systemu. Aby zrestartować urządzenie:

> request system reboot

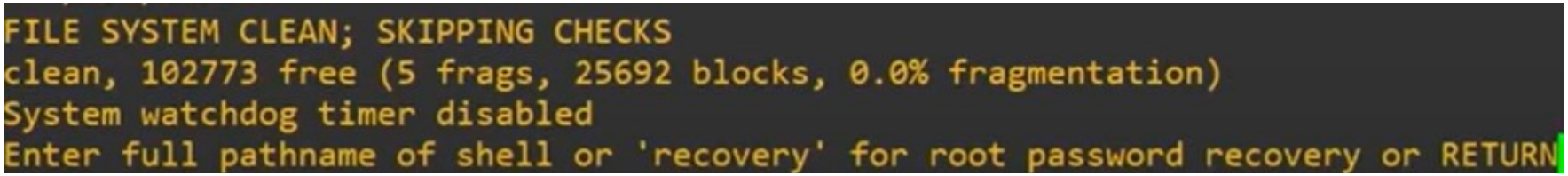
System zapyta o potwierdzenie — możesz potwierdzić lub dodać no-confirm aby pominąć pytanie:

> request system reboot no-confirm

Jak usunąć bieżącą konfigurację:
> request system zeroize

Co robi request system zeroize? Usuwa wszystkie konfiguracje zapisane na urządzeniu.
Usuwa pliki dziennika i inne dane systemowe. Resetuje hasło użytkownika do fabrycznego (zwykle root bez hasła lub z domyślnym hasłem).Urządzenie zostanie automatycznie zrestartowane po zakończeniu procesu. Po restarcie urządzenie będzie miało ustawienia fabryczne i będzie wymagało ponownej konfiguracji. Operacja ta jest odpowiednia, gdy chcesz całkowicie wyczyścić urządzenie przed np. przekazaniem dalej.

Odzyskiwanie hasła roota podczas bootowania. Dostęp tylko z konsoli



Junos Basic Configuration

W Junos OS sekcja system services w konfiguracji służy do włączania usług sieciowych takich jak SSH, Telnet, HTTP, Netconf, SNMP itp. Usługi te umożliwiają zdalny dostęp i zarządzanie urządzeniem.

```
> configure
set system services ssh
set system services netconf ssh
set system services web-management http interface vlan.0
set system services web-management https system-generated-certificate
commit
```

Co to oznacza:
ssh – włącza dostęp SSH do urządzenia.
netconf ssh – włącza protokół NETCONF przez SSH (do automatyzacji zarządzania).
web-management – umożliwia dostęp przez HTTP/HTTPS do GUI (jeśli obsługiwane).
interface vlan.0 – wskazuje interfejs, na którym ma działać GUI

Ograniczenie usług do konkretnego interfejsu lub adresu:

```
set system services ssh interface ge-0/0/0.0 ( lub np fx0)
```

Sprawdzenie konfiguracji:

```
show configuration system services
```

Usługa	Polecenie do włączenia
SSH	`set system services ssh`
Telnet	`set system services telnet`
FTP (dostęp do plików)	`set system services ftp`
Web GUI (HTTP/HTTPS)	`set system services web-management`
NETCONF (do automatyzacji)	`set system services netconf ssh`
SNMP	`set snmp community public authorization read-only`

Junos Basic Configuration

https://support.juniper.net/support/downloads/?p=srx300

Ustawienie konfiguracji ratunkowej:
set system rescue-configuration

Wyświetlenie konfiguracji ratunkowej:
show system rescue-configuration

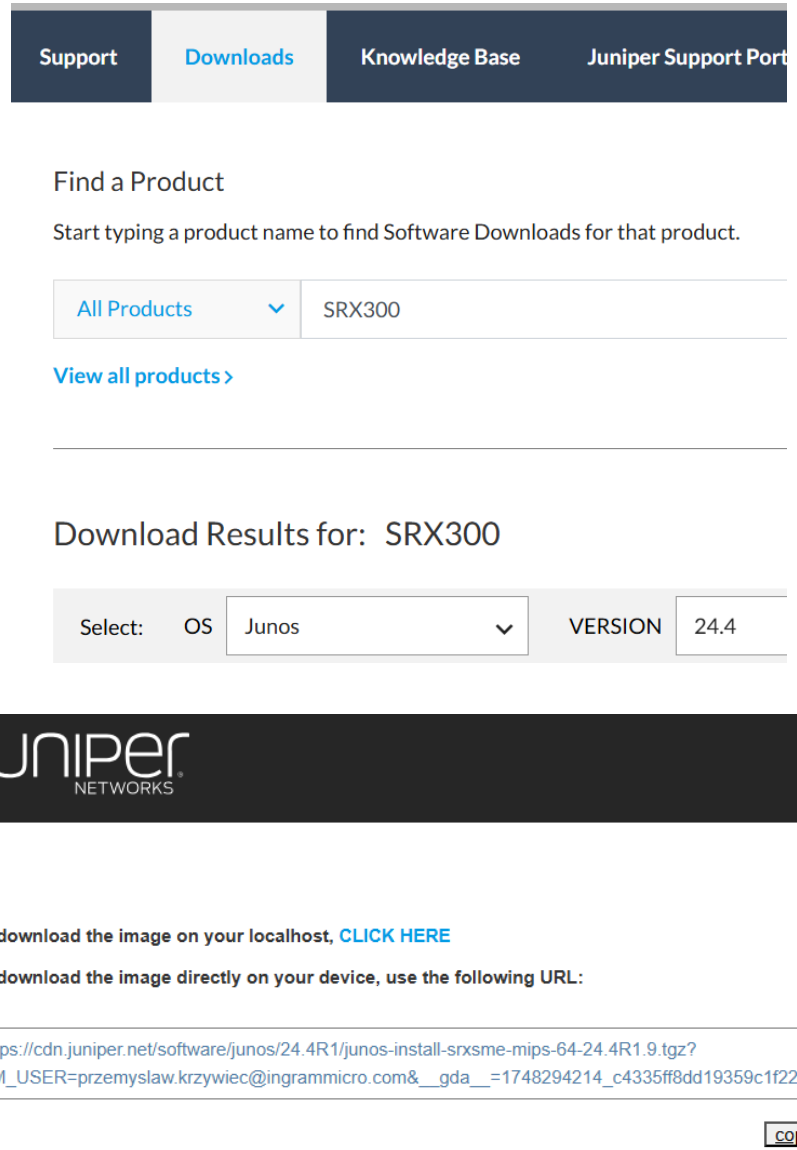
Przywracanie:
rollback rescue

Jak zrobić upgrade Junos z CLI:
- Możesz przesłać obraz na urządzenie przez SCP, FTP lub USB:
> scp junos-install-package.tgz user@device:/var/tmp/

Pobranie obrazu ze strony Junipera z tymczasowego linku:
> start shell
> fetch http://adres-do-obrazu/junos-install-package.tgz /var/tmp/junos-install-package.tgz

Instalacja obrazu:
> save /var/tmp/config-backup.conf
> request system software add /var/tmp/junos-install-package.tgz no-validate
> request system reboot

Po restarcie sprawdź wersję:
>show version



Junos Basic Configuration

Committing a Configuration: Part 1

- 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
```

Committing a Configuration: Part 1

- Use **commit confirmed** to temporarily activate:
  - Issue a **commit confirmed <time-out>** command

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

Committing a Configuration: Part 2

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

```
[edit]
user@router# commit at 10:59
configuration check succeeds
commit at will be executed at 2021-04-07 10:59:00 UTC
The configuration has been changed but not committed
Exiting configuration mode
```

```
user@router> show system commit
commit requested by user via cli at 2021-04-07 10:59:00 UTC
0 2021-04-07 10:57:01 UTC by user via cli
...
user@router> clear system commit
Pending commit cleared
```

Committing a Configuration: Part 2

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

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

user@router> show system commit
0   2021-04-17 10:43:48 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>
```

Committing a Configuration: Part 3


```
[edit]
user@router# commit prepare
commit prepare successful
```

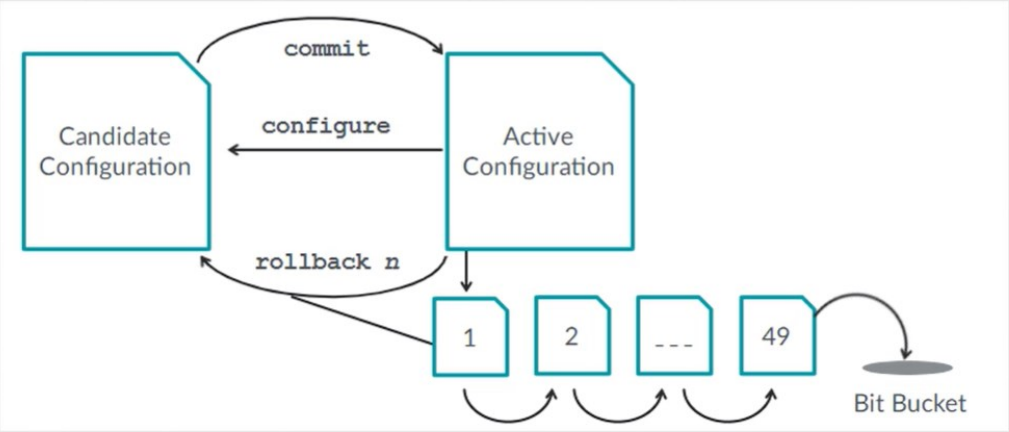
```
[edit]
user@router# commit activate
commit complete
```

Active Versus Candidate Configuration

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

Overview: The Life of a Configuration File

- Junos OS:
  - Maintains a configuration history by storing previously active configurations
  - Saves a maximum of 50 configurations
  - Rollback operation: Related configuration does not become active until you issue a **commit**



Junos Basic Configuration

```
[edit]
employee@R1# set system host-name ROUTER_1

[edit]
employee@R1#
```

The hostname did not change when you entered this configuration statement.

A change has been proposed within the system hierarchy.

```
[edit]
employee@R1# show | compare
[edit system]
-   host-name R1;
+   host-name ROUTER_1;

[edit]
employee@R1#
```

show | compare now reveals nothing.

The active and candidate configs are now identical again.

```
[edit]
employee@R1# rollback
load complete

[edit]
employee@R1# show | compare

[edit]
employee@R1#
```

Backing out of Configuration Changes: Part 1

- 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
  - Default setting can be changed to increase the number of rollback files

```
[edit]
user@router# rollback ?
Possible completions:
 <[Enter]>            Execute this command
  0                    2021-04-07 10:43:48 UTC by user via cli
  1                    2021-04-07 10:29:25 UTC by user via cli
…TRIMMED…
  49                   2021-04-05 06:29:07 UTC by lab via cli
  rescue               2021-04-05 11:43:07 UTC by lab via junoscript
  revision             Rollback to given configuration revision
```

```
[edit system]
user@router# set max-configurations-on-flash ?
Possible completions:
<max-configurations-on-flash> Number of configuration files stored on flash
```

Backing out of Configuration Changes: Part 2

- To return to the rescue configuration, use the **rollback rescue** command

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

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

Backing out of Configuration Changes: Part 2

```
user@router> show system commit include-configuration-revision
0   2021-04-07 11:04:09 UTC by lab via cli commit activate re0-1617793448-92
1   2021-04-07 10:59:09 UTC by lab via cli commit at re0-1617793148-91
2   2021-04-07 10:57:01 UTC by lab via cli re0-1617793021-90
3   2021-04-07 10:43:48 UTC by lab via cli re0-1617792226-89
    Changed OSPF Configuration

...TRIMMED...
48  2021-04-05 06:34:52 UTC by lab via cli re0-1617604490-44
49  2021-04-05 06:34:19 UTC by lab via cli re0-1617604457-43
rescue  2021-04-05 11:43:07 UTC by lab via junoscript

user@router>
```

Saving Configuration Files

- Use the **save** command 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'
```

Saving Configuration Files

- Specify a filename by:
  - <filename> or <path/filename> (FTP and SCP)
  - ftp://user:password@router/<path/filename>
  - scp://user@router/<path/filename>

```
[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
```

Loading Configuration Files: Part 1

- 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
```

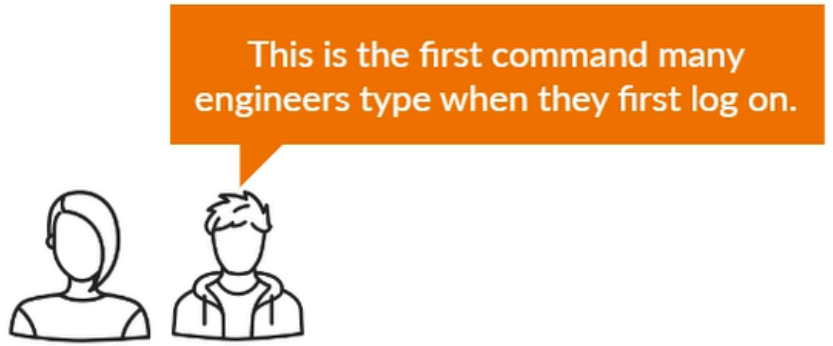
Loading Configuration Files: Part 2

- Some of the arguments to the **load** command:

Argument	Description
factory-default	Replaces the full current configuration with the factory-default configuration
merge	Combines the current configuration with the configuration you load
override	Completely overwrites the current configuration with the configuration you load
patch	Adds or deletes variables from the configuration based on the contents of a specified patch file
replace	Searches for a replace tag in the configuration you load
set	Enables users to load **set** commands from the terminal or from a saved file
update	Updates the existing configuration with the configuration you load
terminal	Uses the text you type at the terminal as input to the configuration
relative	Instructs the device to add the data you load relative to the current configuration hierarchy

Junos Basic Configuration

- To fix this, change the `cli screen-width`
  - You use this `set` command in operational mode
    - Default is 80 characters
    - Max is 1024 characters



Remember, this is an operational mode command.

```
employee@ROUTER_1> set cli screen-width 1000
Screen width set to 1000

employee@ROUTER_1>
```

This command is not saved, and is only live for the duration of your CLI session.

Back in config mode, now the CLI shows your full typed command.

```
[edit]
employee@ROUTER_1# set interfaces xe-0/1/5 unit 0 family inet6 address 2001:db8:1:2::1/64
```

Junos Basic Configuration

Interfejs fizyczny lub logiczny. Konfiguracja interfejsu ge-0/0/1 z adresem IP:

```
# set interfaces ge-0/0/1 unit 0 family inet address 192.168.1.1/24
# set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:1::1/64
```

ge-0/0/1 – nazwa interfejsu (GigabitEthernet)
unit 0 – jednostka logiczna w obrębie fizycznego portu (wymagana w Junos)
family inet – oznacza IPv4 (dla IPv6 użyj family inet6)
address – przypisany adres IP z maską

Każdy interfejs w Junos musi mieć przynajmniej jedną jednostkę logiczną, nawet jeśli to tylko jedna konfiguracja na jednym porcie.

Zastosowanie	Przykład
Podział interfejsu na wiele podsieci	`unit 0` dla VLAN 10, `unit 1` dla VLAN 20
Obsługa wielu VLANów (np. trunking)	interfejs typu `unit X` z `vlan-id` i `family`
Routing między podsieciami	każda jednostka ma osobny adres IP
PPPoE, GRE, IPSec, itd.	jednostki logiczne używane do tuneli

Przykład – jeden interfejs z wieloma jednostkami (np. trunk):

```
set interfaces ge-0/0/1 unit 10 vlan-id 10 family inet address 192.168.10.1/24
set interfaces ge-0/0/1 unit 20 vlan-id 20 family inet address 192.168.20.1/24
```

Ustawienie adresu IP na interfejsie zarządzającym:

```
set interfaces me0 unit 0 family inet address 192.168.100.10/24
set interfaces fx0 unit 0 family inet address 192.168.100.10/24
```

Junos J-Web

- Wprowadzenie do kursu i platformy warsztatów

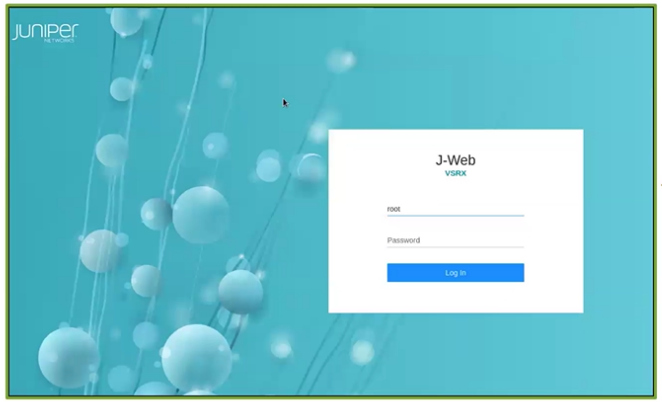
- **Moduł 0:** Introducing Junos OS and the Hardware
- **Moduł 1:** Junos Fundamentals
- **Moduł 2:** Junos User Interface
- **Moduł 3:** Junos Basic Configuration
- **Moduł 9:** Junos J-Web
- **Moduł 4:** Junos Logging and Monitoring
- **Moduł 5:** Junos Routing Policy and Firewall Filters
- **Moduł 6:** Junos Routing Fundamentals
- **Moduł 7:** Junos OS Switches
- **Moduł 8:** Junos OS Architecture
- **Moduł 10:** Jak przygotować się do JNCIA-Junos

J-Web User Interface

- GUI-based configuration of security features
  - Security policies, IPSec VPNs, NAT, and so on
- Allow for easy setup and maintenance
  - Fast deployment with minimal configuration
  - HTTP-based—no user software required
- Facilitates resource monitoring
  - Provides real-time graphs, statistics, and output
- J-Web provides sections for viewing and editing the CLI
- Commits are required, matching the CLI

Initial Configuration

- The first login:



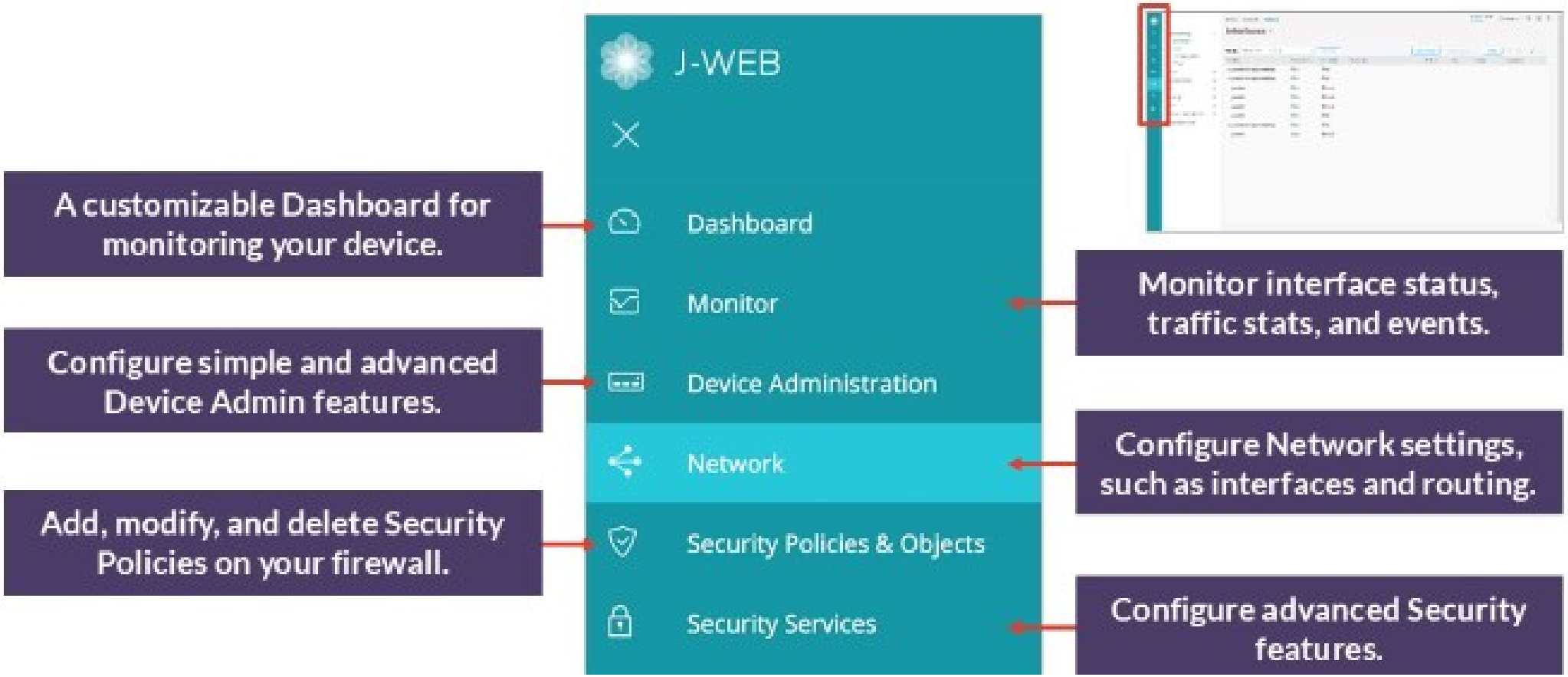
- J-Web Login
  - Browse the IP address of fxp0.0
  - Username— root
  - No password

J-Web and the CLI: Choosing the Best Tool for the Job

- The J-Web gives you a great alternative to the CLI
  - But how do you choose which one to use?

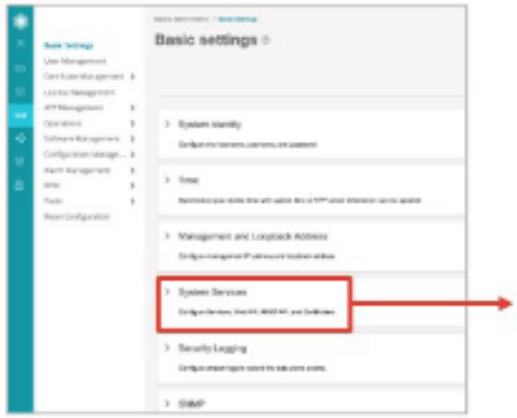
J-Web is not a replacement for the CLI	J-Web is mostly used on security devices
• It's a great helper for Junos beginners • Some engineers always prefer the CLI to any graphical interface • Some tasks might seem easier to do in J-Web—it's a personal choice	• J-Web is commonly used on SRX firewalls • It's much easier to manage hundreds of firewall policies using a graphical interface • J-Web also enables easier management of advanced firewall settings

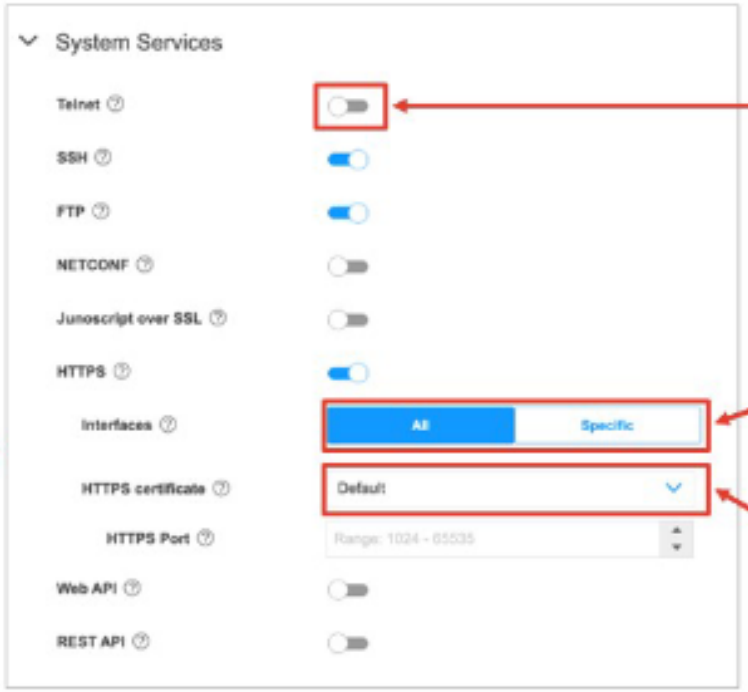
Junos J-Web



Basic Settings: System Services: Part 1

- Use this section to enable different management services
  - Telnet, SSH, HTTPS, and so on





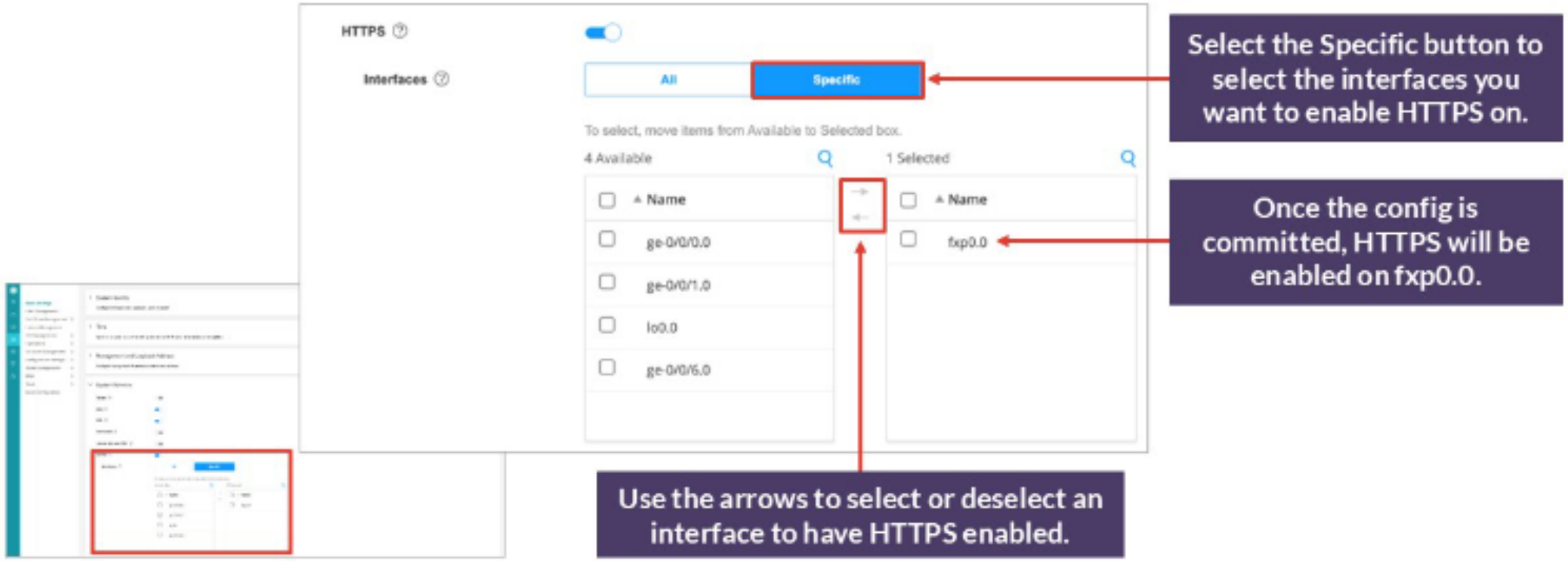
Enable and disable services using a toggle switch.

Choose to enable HTTP on All interfaces, or on Specific ones.

Select the type of HTTPS certificate from the drop-down.

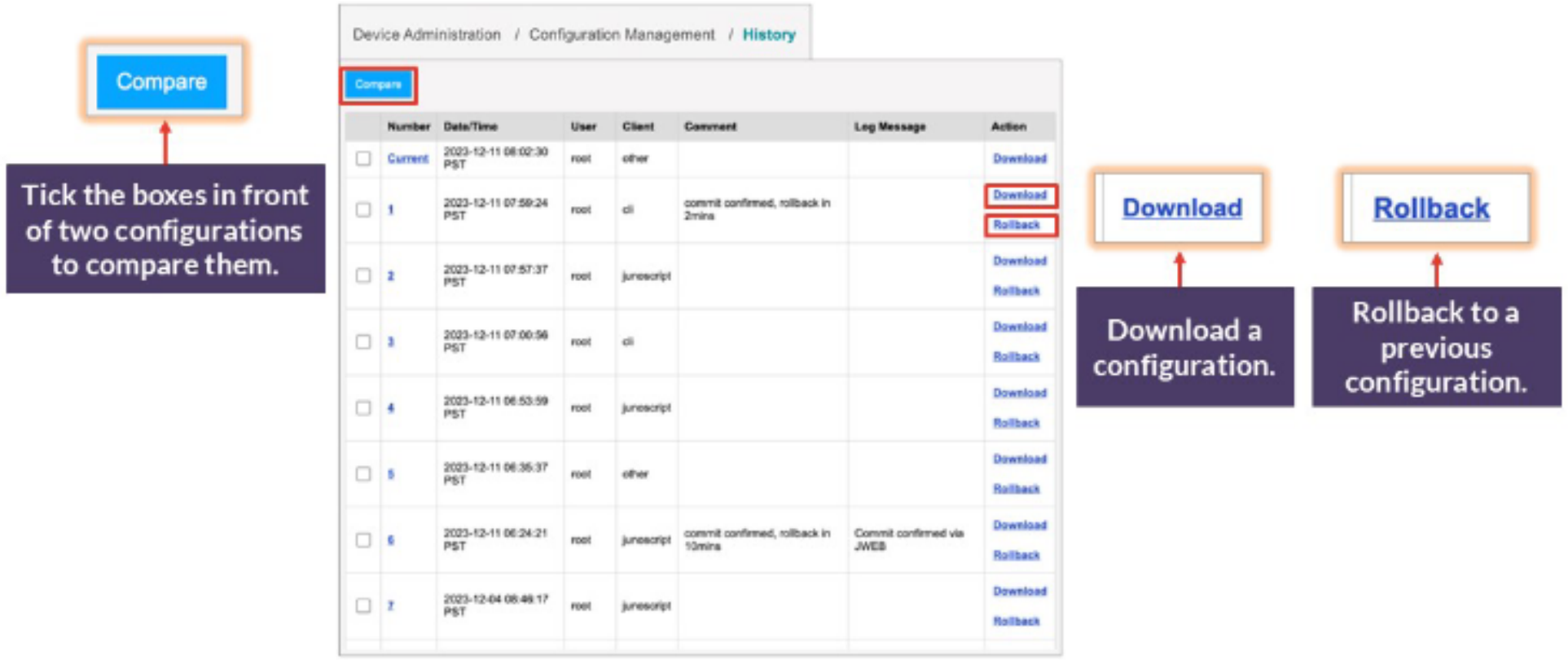
Basic Settings: System Services: Part 2

- In production, you might restrict J-Web to the management interface
  - HTTPS-enabled interfaces are vulnerable to potential attacks



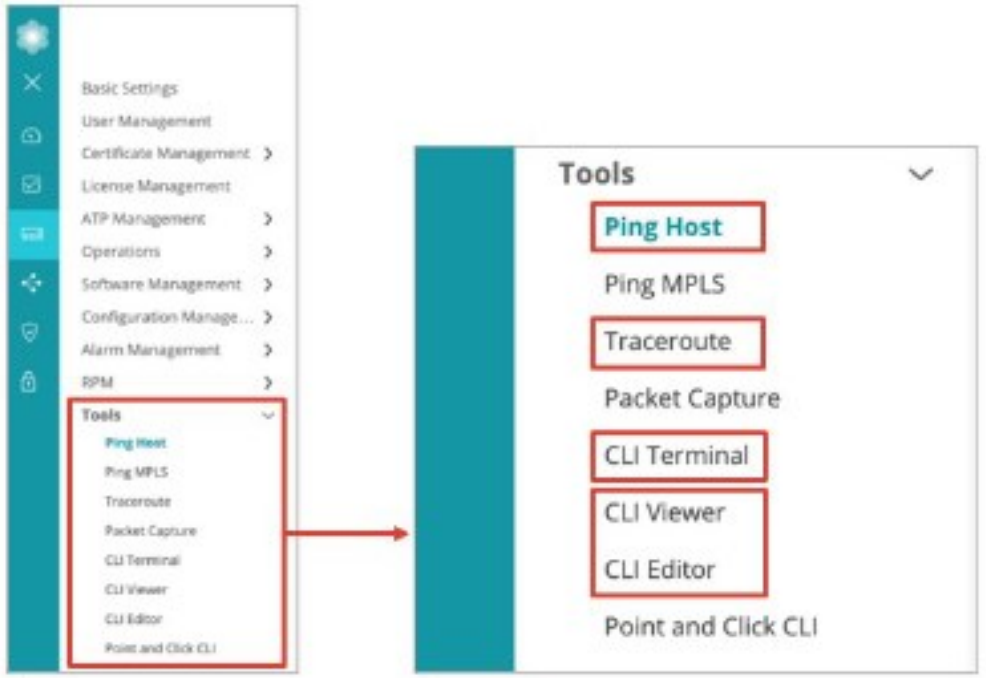
Device Administration: Configuration Management

- You will find a list of previous configurations under the **History** tab:



Device Administration: Tools

- You can use some of the CLI troubleshooting tools on J-Web
  - The next page shows the `Ping Host` option



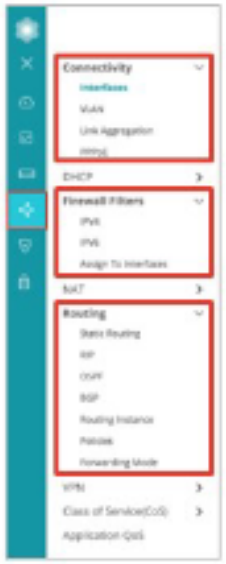
Use Ping and Traceroute to test that you can reach a remote IP address.

Connect to the Junos CLI Terminal.

View and edit the configuration in hierarchy mode—you can't do this on the CLI!

The J-Web Interface: Network

- The next tab in the list is all about Network configuration
  - This page highlights the sections you've learned about in this course



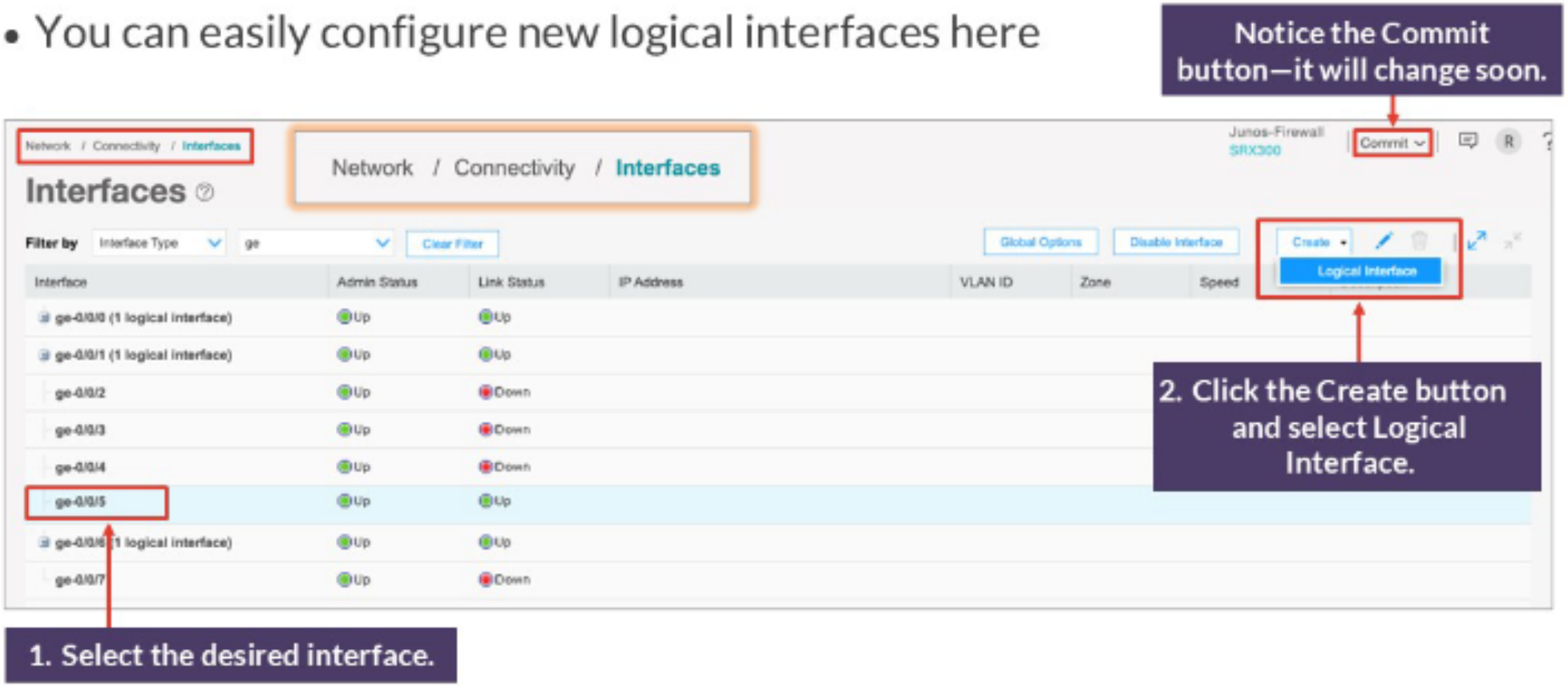
Configure Interfaces and VLANs, under Connectivity.

Configure and apply Firewall Filters.

Configure Static Routing, dynamic routing, and more.

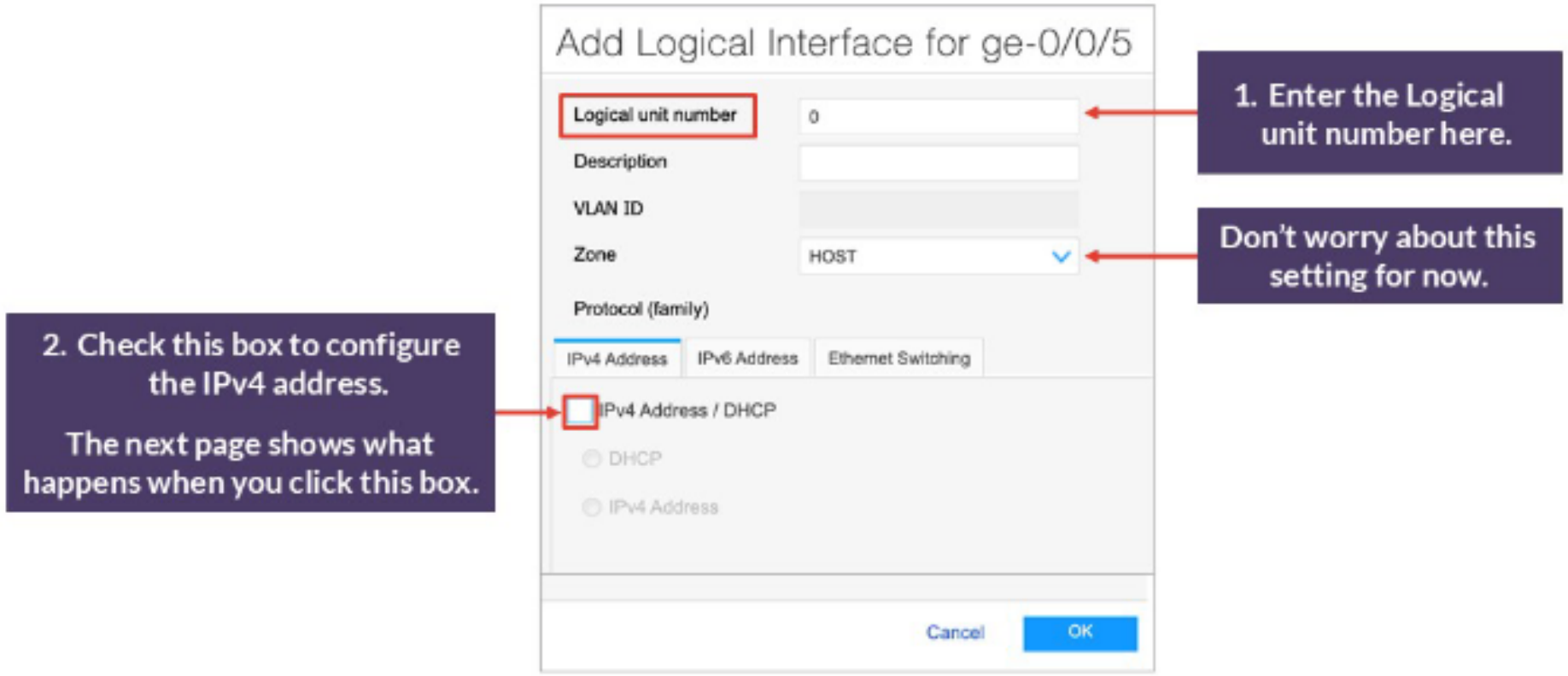
Network: Configuring an IPv4 Address: Part 1

- This page explores the Connectivity section of the Network tab
  - You can easily configure new logical interfaces here



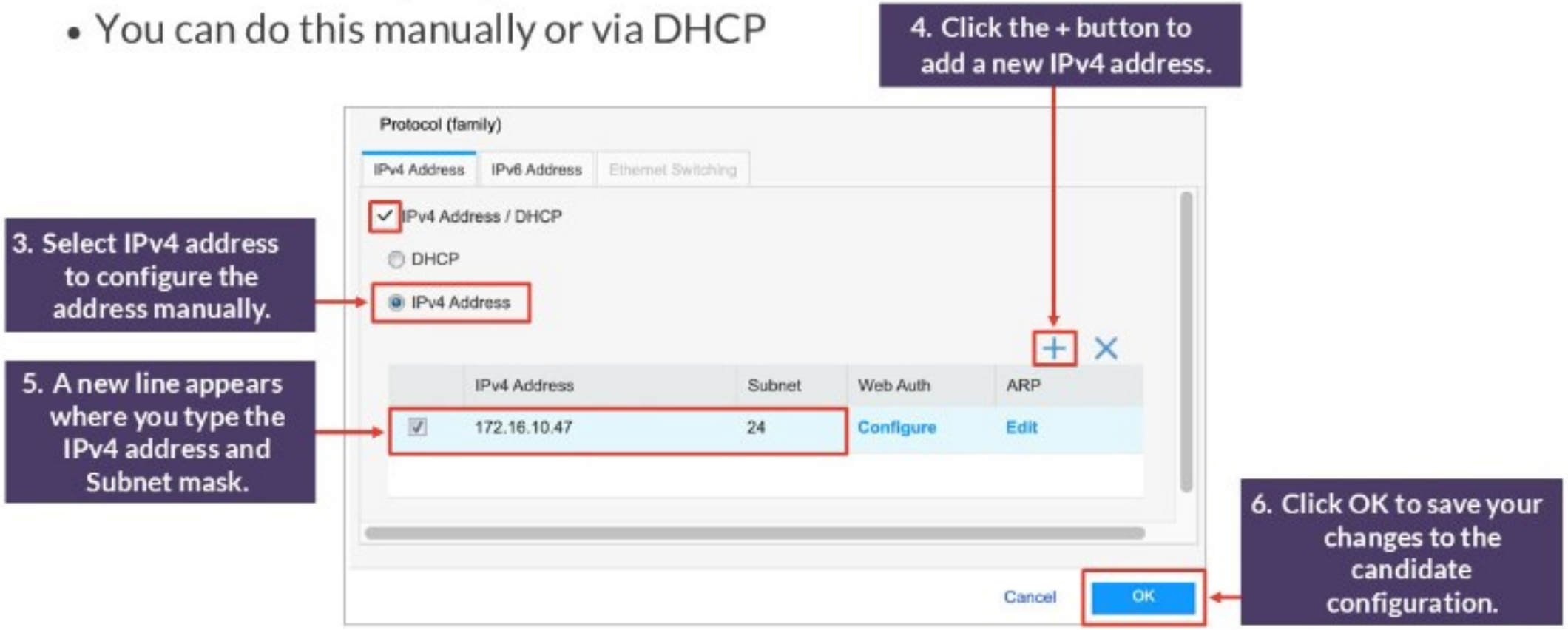
Network: Configuring an IPv4 Address: Part 2

- A new window pops up—you configure the interface details here



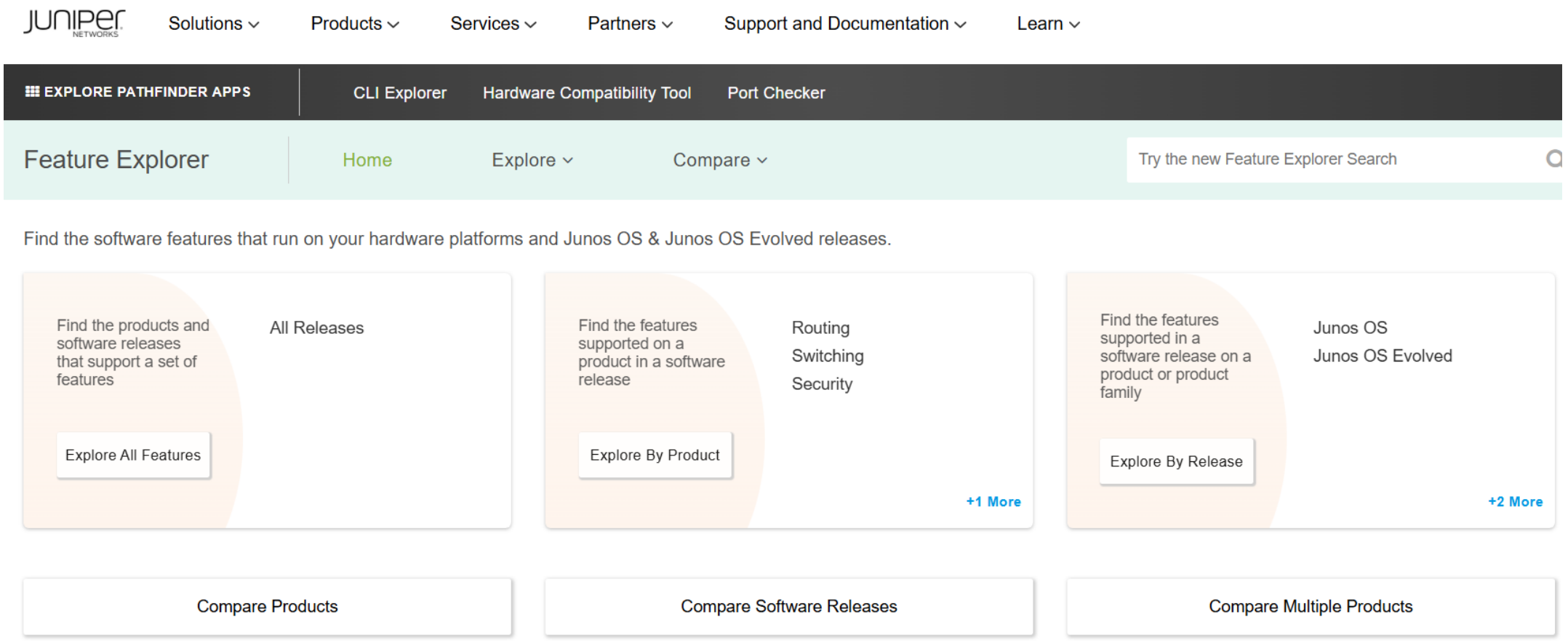
Network: Configuring an IPv4 Address: Part 3

- You're now ready to give the interface an IP address
  - You can do this manually or via DHCP



Junos J-Web - Feature Explorer Tool

https://apps.juniper.net/feature-explorer/



Junos Logging and Monitoring

- Wprowadzenie do kursu i platformy warsztatów

- **Moduł 0:** Introducing Junos OS and the Hardware
- **Moduł 1:** Junos Fundamentals
- **Moduł 2:** Junos User Interface
- **Moduł 3:** Junos Basic Configuration
- **Moduł 9:** Junos J-Web
- **Moduł 4:** Junos Logging and Monitoring
- **Moduł 5:** Junos Routing Policy and Firewall Filters
- **Moduł 6:** Junos Routing Fundamentals
- **Moduł 7:** Junos OS Switches
- **Moduł 8:** Junos OS Architecture
- **Moduł 10:** Jak przygotować się do JNCIA-Junos

Junos Logging and Monitoring

- To view log files, type **show log FILENAME**
  - The two default files are `messages` and `interactive-commands`
  - Use **show log messages** to view general system events

Notice the timestamp, the hostname, daemon names, and events.

```
r.chen@ROUTER_1> show log messages
Sep 11 17:14:11  ROUTER_1 newsyslog[32618]: logfile turned over due to size>500K
Sep 12 11:19:34  ROUTER_1 mgd[53356]: UI_COMMIT_COMPLETED:  : commit complete
Sep 12 11:19:34  ROUTER_1 mib2d[12507]: SNMP_TRAP_LINK_DOWN: ifIndex 13, ifAdminStatus down(2),
ifOperStatus down(2), ifName fxp0.0
Sep 12 11:31:13  ROUTER_1 mgd[53356]: UI_DBASE_LOGOUT_EVENT: User 'r.chen' exiting configuration mode
Sep 12 16:12:03  ROUTER_1 mib2d[12507]: SNMP_TRAP_LINK_DOWN: ifIndex 534, ifAdminStatus up(1),
ifOperStatus down(2), ifName xe-0/1/0
Sep 12 16:12:09  ROUTER_1 fpc0 SMIC(0/1) link 0 SFP receive power low  alarm set
(snip)
```

Pipe this command to match for `error`, `alert`, `warning`, `down`, or protocol and interface names.

Junos Logging and Monitoring

- You can also view new log entries in real time
  - Type `monitor start FILENAME` to view logs as they are written to a file
  - Type `monitor stop` to end the process

```
r.chen@ROUTER_1> monitor start messages

r.chen@ROUTER_1>
*** messages ***
Oct 28 14:43:03  ROUTER_1 rpd[20622]: RPD_OSPF_NBRDOWN: OSPF neighbor 10.1.2.2 (realm ospf-v2 xe-0/1/5.0 area
0.0.0.0) state changed from Full to Init due to 1WayRcvd (event reason: neighbor is in one-way mode)
monitor stop

r.chen@ROUTER_1>
```

Remember the text `RPD_OSPF_NBRDOWN`.
You are going to use it next.

Junos Logging and Monitoring

- Some syslog entries are very technical and difficult to understand
  - If the log has a message code, you can use `help syslog` to decode it

```
r.chen@ROUTER_1> help syslog RPD_OSPF_NBRDOWN
Name:          RPD_OSPF_NBRDOWN
Message:       OSPF neighbor <neighbor-address> (realm <realm-name> <interface-name> area <area-id>i)
state changed from <old-state> to <new-state> due to <event-name> (event reason: <reason>)
Help:          OSPF neighbor adjacency was terminated
Description:   An OSPF adjacency with the indicated neighboring router was terminated. The local router no
longer exchanges routing information with, or directs traffic to, the neighboring router.
Type:          Event: This message reports an event, not an error
Severity:      notice
Facility:      LOG_DAEMON
Action:        For more information, see KB19074.
```

Junos Logging and Monitoring

- Here's two more helpful logging options
  - You can stream your syslogs to an external server
    - Useful if you want to keep logs long term
    - Rogue actors can delete logs (evidence) from the Junos device—but it's harder to delete logs from a remote server

```
[edit]
r.chen@ROUTER_1# set system syslog host 172.16.20.5 any notice
```

  - You can send logs directly to the CLI
    - Use * for all users, or specify users by name

```
[edit]
r.chen@ROUTER_1# set system syslog user * any emergency
```

Junos Logging and Monitoring

- Troubleshooting connectivity issues is a big part of your job!
  - You've already seen some helpful commands:
    - **show interfaces** (terse)
    - **show route**
    - **show ospf neighbors**
    - **ping** (count) (source)

- You've already used **ping** with the count and source options
  - Here's some other useful ping options to keep in your back pocket:

```
r.chen@ROUTER_1> ping 10.1.2.2 rapid
PING 10.1.2.2 (10.1.2.2): 56 data bytes
!!!!!
--- 10.1.2.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.428/0.570/0.735/0.099 ms
```

Send the next ping immediately after the previous one receives a reply.

```
r.chen@ROUTER_1> ping 10.1.2.2 rapid count 100
PING 10.1.2.2 (10.1.2.2): 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
--- 10.1.2.2 ping statistics ---
100 packets transmitted, 100 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.292/0.562/0.926/0.091 ms
```

Send many rapid pings to get potential visibility of packet loss on the link.

(Some devices do not prioritize ICMP replies, and some rate-limit ICMP traffic.)

Junos Logging and Monitoring

```
r.chen@ROUTER_1> traceroute 10.5.6.6
traceroute to 10.5.6.6 (10.5.6.6), 30 hops max, 40 byte packets
 1  10.1.2.2 (10.1.2.2)  2.705 ms  2.630 ms  2.543 ms
 2  * * *
 3  10.3.4.4 (10.3.4.4)  4.097 ms  4.635 ms  3.490 ms
 4  10.4.5.5 (10.4.5.5)  525.358 ms  924.224 ms  565.332 ms
 5  10.5.6.6 (10.5.6.6)  6.485 ms  5.659 ms  6.503 ms
```

```
r.chen@ROUTER_1> show arp
MAC Address         Address          Name             Interface          Flags
00:05:86:71:7d:00   10.1.2.2         10.1.2.2         xe-0/1/5.0         none
00:05:86:71:f0:01   10.1.254.254     10.1.254.254     xe-0/1/6.0         none
38:4f:49:80:f4:08   172.16.10.4      172.16.10.4      xe-0/1/0.10        none
74:8f:3c:ba:e0:dd   172.16.10.8      172.16.10.8      xe-0/1/0.10        none
94:bf:94:b4:67:03   172.16.10.55     172.16.10.55     xe-0/1/0.10        none
74:8f:3c:ba:e0:da   172.16.10.69     172.16.10.69     xe-0/1/0.10        none
38:4f:49:80:f4:09   172.16.20.15     172.16.20.15     xe-0/1/0.20        none
36:72:dd:d5:e6:44   172.16.30.7      172.16.30.7      xe-0/1/0.30        none
Total entries: 8
```

Junos Logging and Monitoring

Real Time

- **show** gives you a snapshot of information in the moment that you type the command
- **monitor** tends to update in real time, giving you up-to-the-second information until you cancel the command

```
r.chen@ROUTER_1> monitor interface traffic
```

```
ROUTER_1                                Seconds: 61                 Time: 19:42:31

Interface     Link   Input packets          (pps)     Output packets          (pps)
 xe-0/0/2:0   Down               0            (0)                  0            (0)
 xe-0/0/2:1   Down               0            (0)                  0            (0)
 xe-0/0/2:2   Down               0            (0)                  0            (0)
 xe-0/0/2:3   Down               0            (0)                  0            (0)
 xe-0/0/3:0   Down               0            (0)                  0            (0)
 xe-0/0/3:1   Down               0            (0)                  0            (0)
 xe-0/0/3:2   Down               0            (0)                  0            (0)
 xe-0/0/3:3   Down               0            (0)                  0            (0)
 xe-0/1/0       Up           12655            (0)              12677            (0)
 xe-0/1/1       Up           12641            (0)                  0            (0)
 xe-0/1/2       Up           12642            (0)                  0            (0)
 xe-0/1/3       Up           12628            (0)                  0            (0)
 xe-0/1/4     Down               0            (0)                  0            (0)
 xe-0/1/5       Up           35265           (89)              86029           (88)
 xe-0/1/6       Up             135            (0)                140            (0)
 xe-0/1/7     Down               0            (0)                  0            (0)
 demux0         Up               0                                 0
 dsc            Up               0                                 0

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

Junos Logging and Monitoring

Real Time

```
r.chen@ROUTER_1> monitor interface xe-0/1/5

ROUTER_1                          Seconds: 19                  Time: 19:44:12
                                                              Delay: 1/0/2
Interface: xe-0/1/5, Enabled, Link is Up
Encapsulation: Ethernet, Speed: 10000mbps
Traffic statistics:                                              Current delta
  Input bytes:                21539132 (803864 bps)                  [2226160]
  Output bytes:               26908856 (894856 bps)                  [2313760]
  Input packets:                 43684 (67 pps)                        [1486]
  Output packets:                94461 (74 pps)                        [1532]
Error statistics:
  Input errors:                      0                                 [0]
  Input drops:                       0                                 [0]
  Input framing errors:              0                                 [0]
  Policed discards:                  0                                 [0]
  L3 incompletes:                    0                                 [0]
  L2 channel errors:                 0                                 [0]
  L2 mismatch timeouts:              0  Carrier transitions:            [0]


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

Again, use these keyboard shortcuts.

Junos Logging and Monitoring

- This example shows **monitor traffic interface**
  - It shows the source and destination IP, and the kind of traffic being sent
  - Use `no-resolve` to speed things up
    - It avoids translating IPs to domain names

```
r.chen@ROUTER_1> monitor traffic interface xe-0/1/5 no-resolve
verbose output suppressed, use <detail> or <extensive> for full protocol decode
Address resolution is OFF.
Listening on xe-0/1/5, capture size 96 bytes

19:45:00.007855 Out IP truncated-ip - 20 bytes missing! 10.1.2.1 > 224.0.0.5: OSPFv2, Hello, length 60
19:45:04.568491 Out IP6 truncated-ip6 - 16 bytes missing!fe80::3a4f:49ff:fe80:f40c > ff02::5: OSPFv3, Hello, length 36
19:45:06.326315  In IP 10.1.2.2 > 224.0.0.5: OSPFv2, Hello, length 60
^C
3 packets received by filter
0 packets dropped by kernel
```

Don't be alarmed when you see `20 bytes missing!` or `16 bytes missing!`. This is standard behavior. These messages can be ignored.

You can see exactly what message is being sent.

Junos Store log files in /var/log

See a log file	> show log FILENAME
See the most recent entries	> show log FILENAME | last 5
See the system boot log	> show system boot-messages
Monitor a log file	> monitor start FILENAME
Pause monitor output - ESC & q	
Show logs we ware monitoring	> monitor list
Stop monitoring logs	> monitor stop
Configure a syslog server	> set system syslog host IP any warning
View syslog setup in configuration	> show configuration system syslog
Write output to file	> monitor traffic interface INTERFACE write-file FILENAME
Collect inf. for JTAC	>request support information | save /var/log/FILENAME

> show chassis alarm	
> show chassis environment	
> show chassis hardware	(show PICs and Serial numbers)
> show chassis fpc	
> show chassis routing-engine	(RE stats, RE cpu temp)
> show system core-dumps	
> show system alarms	
> show system license	
> show system storage	
> show system uptime	

Junos Routing Policy and Firewall Filters

- Wprowadzenie do kursu i platformy warsztatów

- **Moduł 0:** Introducing Junos OS and the Hardware
- **Moduł 1:** Junos Fundamentals
- **Moduł 2:** Junos User Interface
- **Moduł 3:** Junos Basic Configuration
- **Moduł 9:** Junos J-Web
- **Moduł 4:** Junos Logging and Monitoring
- **Moduł 5:** Junos Routing Policy and Firewall Filters
- **Moduł 6:** Junos Routing Fundamentals
- **Moduł 7:** Junos OS Switches
- **Moduł 8:** Junos OS Architecture
- **Moduł 10:** Jak przygotować się do JNCIA-Junos

Junos Policy and Firewall Filters

- Junos has unique names for stateful and stateless rules
  - You may wonder why a firewall is not used for stateful security policies
    - After all, you use these on hardware firewalls
  - The answer is historical
    - Stateless rules came first, in the very late 1990s—before Juniper Networks sold hardware firewalls!
- Stateless and stateful firewall filters use similar logic and philosophy

Stateful rules are called "security policies"	Stateless rules are called "firewall filters"
Configured within the `security` hierarchy	Configured within the `firewall` hierarchy

Other vendors call their stateless (and sometimes their stateful) filtering "access lists" or "access control lists (ACLs)".

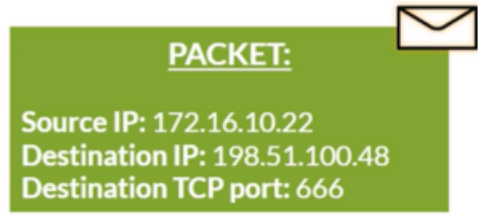
Junos Routing Policy and Firewall Filters

Stateful firewall (security policies) śledzi stan połączeń sieciowych (TCP, UDP itp.). Analizuje ruch **w kontekście całej sesji**, a nie tylko pojedynczych pakietów.
Cechy:
- Pamięta, które połączenia są dozwolone (np. jeśli klient zainicjował połączenie, to ruch powrotny z serwera jest dozwolony).
- Może automatycznie przepuszczać ruch powrotny (return traffic).
- jest bezpieczniejszy i inteligentniejszy niż stateless, bo zna kontekst połączenia.
- Typowo używany w **security policies** w Juniper SRX.
Przykład:
Jeśli klient wewnętrzny otwiera połączenie HTTP na zewnątrz, stateful firewall:
- Przepuszcza ruch wychodzący,
- Zapamiętuje to połączenie,
- Automatycznie przepuszcza odpowiedź z serwera (ruch przychodzący).

Stateless firewall nie śledzi stanu połączeń - każdy pakiet jest oceniany osobno, niezależnie od wcześniejszych.
Cechy:
- Każdy pakiet musi spełniać reguły bez względu na to, czy należy do istniejącej sesji.
- Nie rozróżnia, czy ruch jest odpowiedzią na wcześniej nawiązane połączenie.
- Szybszy i prostszy, ale mniej bezpieczny.
- Zazwyczaj używany w **firewall filters** (filtry stateless na interfejsach w Junos OS).
Przykład:
Aby umożliwić połączenie HTTP:
- Musisz napisać regułę zarówno na ruch wychodzący, jak i na ruch przychodzący,
- Bo firewall nie wie, czy dany pakiet należy do istniejącej sesji.

Junos Routing Policy and Firewall Filters

term	from	then
BLOCK_GAMING	• **Source IP:** 172.16.10.22 or 172.16.10.44 • **Destination IP:** 198.51.100.48 • **Destination TCP port:** 666 (video game)	Drop it Make a log entry for the attempt
GUEST_COUNT_WEB	• **Source IP:** Guest LAN (172.16.30.0/24) • **Destination TCP ports:** 80 or 443	Allow it Count the packets
BLOCK_GUEST_PING	• **Source IP:** Guest LAN (172.16.30.0/24) • **Protocol:** Ping (Echo Request / Echo Reply)	Drop it

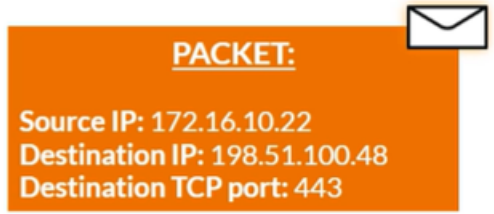


Przykład działania **stateless firewall filter** (czyli filtrowania pakietów na podstawie pojedynczych reguł - bez śledzenia stanu sesji). Zasady działania są następujące: filtr ma trzy term (reguły), które są oceniane po kolei od góry do dołu. Gdy pakiet pasuje do jakiegoś terma, wykonuje się akcja z sekcji "then", a dalsze przetwarzanie jest zatrzymywane (terminating action).

Analiza pakietu (zielone pole)
Source IP: 172.16.10.22
Destination IP: 198.51.100.48
Destination TCP port: 666
Ten pakiet **pasuje do pierwszego term** (BLOCK_GAMING), ponieważ:
IP źródłowe to 172.16.10.22
IP docelowe to 198.51.100.48
port TCP to 666
Efekt:
Pakiet zostaje **odrzucony**
Tworzony jest **wpis w logu**
Dalsze reguły nie są już sprawdzane.

Junos Routing Policy and Firewall Filters

term	from	then
BLOCK_GAMING	• **Source IP:** 172.16.10.22 or 172.16.10.44 • **Destination IP:** 198.51.100.48 • **Destination TCP port:** 666 (video game)	Drop it Make a log entry for the attempt
GUEST_COUNT_WEB	• **Source IP:** Guest LAN (172.16.30.0/24) • **Destination TCP ports:** 80 or 443	Allow it Count the packets
BLOCK_GUEST_PING	• **Source IP:** Guest LAN (172.16.30.0/24) • **Protocol:** Ping (Echo Request / Echo Reply)	Drop it



Analizowany pakiet jest porównywany z regułami filtrowania. Filtr zawiera trzy terms, a pakiet trafia tylko do pierwszego dopasowanego termu - dalsze są ignorowane.

Pakiet nie pasuje do żadnego z termów w filtrze.
Domyślna akcja (jeśli nie zdefiniowano „default-action”) zależy od konfiguracji firewalla:
W Junos firewall filters typowo brak dopasowania oznacza: przepuść pakiet (accept), chyba że zdefiniowano inaczej.

Sprawdzenie zgodności z regułami:
1. BLOCK_GAMING
Source IP: 172.16.10.22 ✅ (pasuje)
Destination IP: 198.51.100.48 ✅ (pasuje)
Destination TCP port: 666 ❌ (pakiet ma 443, więc NIE pasuje)
➡️ Nie dopasowuje się, przetwarzanie idzie dalej.
2. GUEST_COUNT_WEB
Source IP: 172.16.30.0/24 ❌
(Pakiet pochodzi z 172.16.10.22, czyli NIE z Guest LAN – NIE pasuje)
➡️ Nie pasuje, przechodzimy dalej.
3. BLOCK_GUEST_PING
Dotyczy tylko protokołu Ping (ICMP) ❌
(Nasz pakiet to TCP port 443, więc nie pasuje)
➡️ Nie pasuje.

Junos Routing Policy and Firewall Filters

```
firewall {
    family inet {
        filter BLOCK_FILTER {
            term BLOCK_GAMING {
                from {
                    source-address {
                        172.16.10.22/32;
                        172.16.10.44/32;
                    }
                    destination-address {
                        198.51.100.48/32;
                    }
                    protocol tcp;
                    destination-port 666;
                }
                then {
                    discard;
                    log;
                }
            }
```

```
            term GUEST_COUNT_WEB {
                from {
                    source-address {
                        172.16.30.0/24;
                    }
                    protocol tcp;
                    destination-port [ 80 443 ];
                }
                then {
                    accept;
                    count WEB_ACCESS;
                }
            }
```

```
            term BLOCK_GUEST_PING {
                from {
                    source-address {
                        172.16.30.0/24;
                    }
                    protocol icmp;
                    icmp-type [ echo-request echo-reply ];
                }
                then {
                    discard;
                }
            }

            term DEFAULT_ACCEPT {
                then accept;
            }
        }
    }
}
```

Junos Routing Policy and Firewall Filters

```
r.chen@ROUTER_1> show configuration firewall family inet filter LAN_TO_WAN | display set relative

set term BLOCK_GAMING from source-address 172.16.10.22/32
set term BLOCK_GAMING from source-address 172.16.10.44/32
set term BLOCK_GAMING from destination-address 198.51.100.48/32
set term BLOCK_GAMING from destination-port 666
set term BLOCK_GAMING then log
set term BLOCK_GAMING then discard

set term GUEST_COUNT_WEB from source-address 172.16.30.0/24
set term GUEST_COUNT_WEB from protocol tcp
set term GUEST_COUNT_WEB from destination-port http
set term GUEST_COUNT_WEB from destination-port https
set term GUEST_COUNT_WEB then count COUNTER_GUEST_WEB

set term BLOCK_GUEST_PING from source-address 172.16.30.0/24
set term BLOCK_GUEST_PING from icmp-type echo-request
set term BLOCK_GUEST_PING from icmp-type echo-reply
set term BLOCK_GUEST_PING then discard
```

Junos Routing Policy and Firewall Filters

term BLOCK_GAMING – blokuje ruch TCP na port 666 od wybranych IP i loguje zdarzenie.
term GUEST_COUNT_WEB – pozwala na ruch HTTP/HTTPS z sieci Guest LAN i liczy pakiety.
term BLOCK_GUEST_PING – blokuje ICMP Echo (ping) z sieci Guest LAN.
term DEFAULT_ACCEPT – **domyślna reguła przepuszczająca**, gdy żaden wcześniejszy warunek nie został spełniony (zabezpieczenie przed przypadkowym zablokowaniem całego ruchu).

Filtry **stateless** w Junos przypisuje się **do interfejsów**, np.

set interfaces ge-0/0/1 unit 0 family inet filter input BLOCK_FILTER

To przypisuje filtr BLOCK_FILTER do interfejsu ge-0/0/1 w kierunku **przychodzącym** (input).

Stateful firewall filter (czyli „security policy”) przypisuje się do stref (zones - strefy bezpieczeństwa), a nie do interfejsów, jak w przypadku stateless filtrów.

Strefa to logiczna grupa interfejsów o wspólnym poziomie zaufania (np. LAN, WAN, DMZ). Przykładowe strefy:
trust → sieć wewnętrzna (LAN)
untrust → sieć zewnętrzna (Internet)
dmz → strefa z ograniczonym dostępem (serwery)

Junos Routing Policy and Firewall Filters

Stateful zasady tworzy się jako policies między strefami, np.:

set security policies from-zone trust to-zone untrust policy ALLOW_WEB match source-address any destination-address any application [junos-http junos-https] set security policies from-zone trust to-zone untrust policy ALLOW_WEB then permit

Ta polityka:
- dotyczy ruchu z trust do untrust
- pozwala na HTTP/HTTPS
- śledzi sesję, więc odpowiedź z serwera wraca automatycznie

Żeby to działało, trzeba:
1.Przypisać interfejsy do stref:
set security zones security-zone trust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces ge-0/0/0.0
2.Włączyć screeny (ochronę sesji – opcjonalnie):
set security zones security-zone untrust screen untrust-screen
3.Ustawić politykę (security policy) między strefami.

Stateful filtering NIE działa jeśli:
- interfejs nie jest przypisany do strefy (czyli działa jak router, nie firewall)

- próbujesz go użyć na EX lub MX – tam nie ma security policies, tylko stateless firewall filters

Element	Stateless Filter (`firewall filter`)	Stateful Filter (`security policy`)
Przypisanie	Do interfejsu (input/output)	Do ruchu między strefami (`from-zone → to-zone`)
Używany na	EX, MX (routing/switching)	SRX (firewall)
Śledzenie sesji	❌ Nie	✅ Tak

Junos Routing Policy and Firewall Filters

Użyj **stateless**, gdy:
Potrzebujesz szybkiego filtrowania na poziomie pakietów.
Chcesz kontrolować konkretny port/IP bez potrzeby sesji.

Użyj **stateful**, gdy:
Budujesz zasady bezpieczeństwa między sieciami (np. LAN ↔ WAN).
Chcesz uprościć obsługę i zapewnić automatyczne przepuszczanie powrotnego ruchu.

Cecha	Stateless Firewall Filter	Stateful Firewall
Śledzenie stanu połączeń	❌ Nie	✅ Tak
Filtrowanie kierunkowe	Tak (trzeba osobno dla każdego kierunku)	Automatyczne obustronne
Wydajność	✅ Szybszy, mniejsze zużycie zasobów	Wolniejszy, ale dokładniejszy
Zastosowanie w Juniperze	`firewall filter` (routery, EX, MX)	`security policies` (SRX)
Obsługa aplikacji	Ograniczona	Wysoka (może rozpoznawać aplikacje)
Konfiguracja	Bardziej techniczna, precyzyjna	Bardziej intuicyjna dla bezpieczeństwa
Najlepsze do...	Proste filtrowanie, logowanie	Ochrony systemów, ruchu aplikacji

Junos Routing Policy and Firewall Filters

Zone	Interfejs	Adres IP	Opis
trust	ge-0/0/1	192.168.1.1/24	LAN
untrust	ge-0/0/0	DHCP lub statyczny	Internet
dmz	ge-0/0/2	172.16.0.1/24	Serwer WWW

```
# Trust zone - LAN
set interfaces ge-0/0/1 unit 0 family inet address 192.168.1.1/24
set security zones security-zone trust interfaces ge-0/0/1

# Untrust zone - Internet
set interfaces ge-0/0/0 unit 0 family inet dhcp
set security zones security-zone untrust interfaces ge-0/0/0
set security zones security-zone untrust host-inbound-traffic system-services ping

# DMZ zone - Serwery
set interfaces ge-0/0/2 unit 0 family inet address 172.16.0.1/24
set security zones security-zone dmz interfaces ge-0/0/2
```

Junos Routing Policy and Firewall Filters

W Junos OS składnia konfiguracji polityk bezpieczeństwa (security policies) zależy od kontekstu.

1. Składnia w trybie konfiguracyjnym:
set security policies from-zone <nazwa-strefy-źródłowej> to-zone <nazwa-strefy-docelowej> policy <nazwa-polityki> ...

2. Komponenty polityki:
match (warunki)
set security policies ... policy <nazwa-polityki> match source-address <adres/nazwa-obiektu>
set security policies ... policy <nazwa-polityki> match destination-address <adres/nazwa-obiektu>
set security policies ... policy <nazwa-polityki> match application <nazwa-aplikacji>

then (akcja):
set security policies ... policy <nazwa-polityki> then <permit|deny|reject>

Dodatkowe opcje permit:
set security policies ... policy <nazwa-polityki> then permit application-services idp
set security policies ... policy <nazwa-polityki> then permit application-services ssl-proxy

Junos Routing Policy and Firewall Filters

Kolejność polityk: Polityki są przetwarzane od góry do dołu.
Domyślna akcja: Jeśli ruch nie pasuje do żadnej polityki, jest blokowany.
Nazwy aplikacji: Używaj predefiniowanych nazw Junos (np. junos-http, junos-ssh) lub zdefiniuj własne aplikacje.

Podgląd konfiguracji:
show security policies
show security policies from-zone trust to-zone untrust

Polityka umożliwiająca ping z trust do untrust:
1. Policy - zezwól na ICMP z trust do untrust
set security policies from-zone trust to-zone untrust policy allow-ping match source-address any destination-address any application junos-ping
set security policies from-zone trust to-zone untrust policy allow-ping then permit

junos-ping to wbudowana aplikacja w Junos, która odpowiada za ICMP echo request/reply (czyli ping).

2. Opcjonalnie - upewnij się, że strefa untrust pozwala na odpowiedzi (host-inbound-traffic)
Jeśli chcesz, aby SRX sam był pingowany (np. z zewnątrz lub z LAN do jego interfejsu untrust), dodaj:
set security zones security-zone untrust host-inbound-traffic system-services ping

3. Efekt:
Użytkownicy w trust mogą pingować dowolne adresy w Internecie.
Odpowiedzi ICMP z untrust będą akceptowane automatycznie (bo polityki w SRX są stateful).

Junos Routing Policy and Firewall Filters

show configuration security

```
security
├── zones
│   └── security-zone untrust
│       └── host-inbound-traffic
│           └── system-services
│               └── ping
└── policies
    └── from-zone trust
        └── to-zone untrust
            └── policy allow-ping
                ├── match
                │   ├── source-address any
                │   ├── destination-address any
                │   └── application junos-ping
                └── then
                    └── permit
```

```
security {
    zones {
        security-zone untrust {
            host-inbound-traffic {
                system-services {
                    ping;
                }
            }
        }
    }
    policies {
        from-zone trust to-zone untrust {
            policy allow-ping {
                match {
                    source-address any;
                    destination-address any;
                    application junos-ping;
                }
                then {
                    permit;
                }
            }
        }
    }
}
```

Junos Routing Policy and Firewall Filters

```
# Zezwól LAN → Internet (HTTP/HTTPS)
set security policies from-zone trust to-zone untrust policy allow-web match source-address any destination-address any application [ junos-http junos-https ]
set security policies from-zone trust to-zone untrust policy allow-web then permit

# Zezwól LAN → DMZ (np. zarządzanie serwerem)
set security policies from-zone trust to-zone dmz policy allow-dmz-admin match source-address any destination-address any application junos-ssh
set security policies from-zone trust to-zone dmz policy allow-dmz-admin then permit

# Zezwól Internet → DMZ (HTTP tylko do serwera)
set security address-book global address web-server 172.16.0.10/32
set security policies from-zone untrust to-zone dmz policy allow-http-to-dmz match source-address any destination-address web-server application junos-http
set security policies from-zone untrust to-zone dmz policy allow-http-to-dmz then permit
```

Polityka domyślna - odrzucenie pozostałego ruchu:
Nie trzeba jej definiować, ponieważ **ruch między zonami bez jawnej polityki jest domyślnie blokowany**.

Efekt końcowy:
- Użytkownicy z LAN mogą przeglądać Internet.
- Administratorzy z LAN mają dostęp SSH do serwerów w DMZ.
- Użytkownicy z Internetu mogą po HTTP dotrzeć do serwera WWW w DMZ.
- Inny ruch między zonami jest zablokowany.

```
security {
    address-book {
        global {
            address web-server 172.16.0.10/32;
        }
    }
    policies {
        from-zone trust to-zone untrust {
            policy allow-web {
                match {
                    source-address any;
                    destination-address any;
                    application [ junos-http junos-https ];
                }
                then {
                    permit;
                }
            }
        }
        from-zone trust to-zone dmz {
            policy allow-dmz-admin {
                match {
                    source-address any;
                    destination-address any;
                    application junos-ssh;
                }
                then {
                    permit;
                }
```

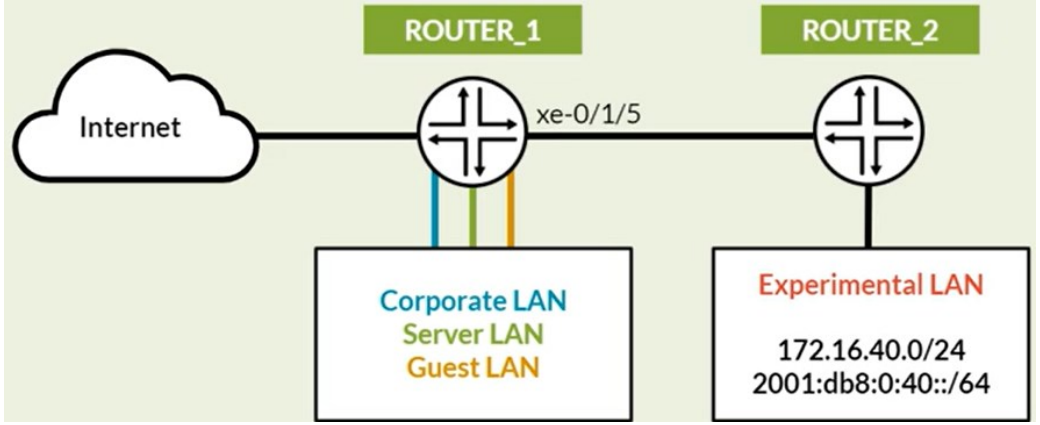
```
        from-zone trust to-zone dmz {
            policy allow-dmz-admin {
                match {
                    source-address any;
                    destination-address any;
                    application junos-ssh;
                }
                then {
                    permit;
                }
            }
        }
        from-zone untrust to-zone dmz {
            policy allow-http-to-dmz {
                match {
                    source-address any;
                    destination-address web-server;
                    application junos-http;
                }
                then {
                    permit;
                }
            }
        }
```

Junos Routing Fundamentals

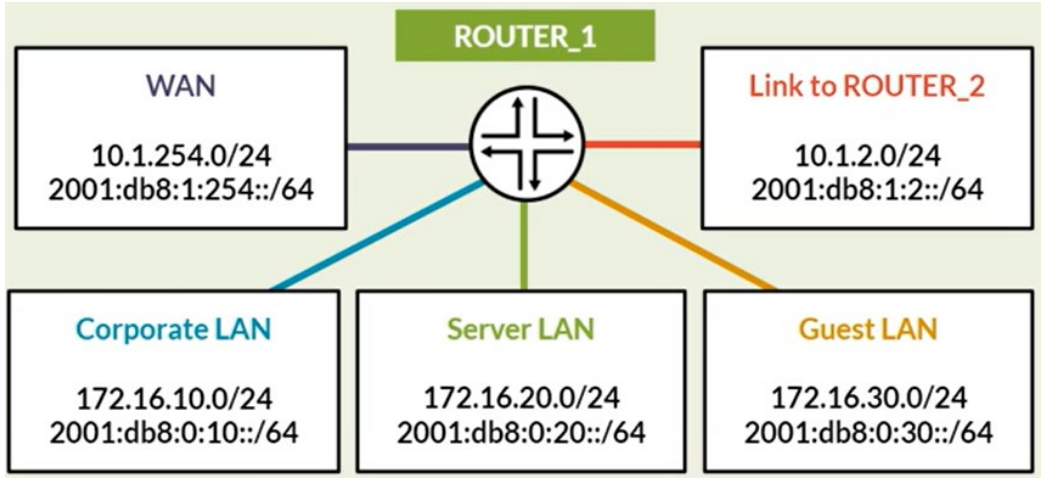
- Wprowadzenie do kursu i platformy warsztatów

- **Moduł 0:** Introducing Junos OS and the Hardware
- **Moduł 1:** Junos Fundamentals
- **Moduł 2:** Junos User Interface
- **Moduł 3:** Junos Basic Configuration
- **Moduł 9:** Junos J-Web
- **Moduł 4:** Junos Logging and Monitoring
- **Moduł 5:** Junos Routing Policy and Firewall Filters
- **Moduł 6:** Junos Routing Fundamentals
- **Moduł 7:** Junos OS Switches
- **Moduł 8:** Junos OS Architecture
- **Moduł 10:** Jak przygotować się do JNCIA-Junos

Junos Routing Fundamentals



- How could ROUTER_1 learn the route?
  - Out of the box, routers do not magically learn all subnets in a network
    - It is not enough to simply connect devices together
  - Instead, you need to use one of these methods to create a route:



Configure a subnet on a local interface

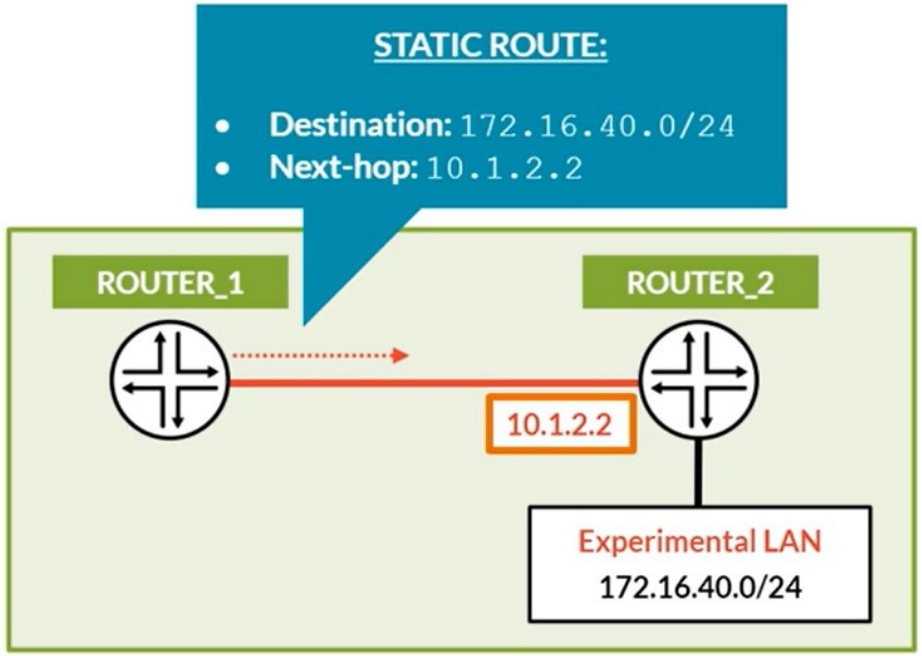
If the interface is up, this creates a directly connected route.

Configure a route to a remote destination

Configured routes pointing to a next-hop router are called static routes.

Enable a routing protocol

Routes learned automatically from a routing protocol are called dynamic routes.



Static routes have advantages:
• Simple to create
• Easy to understand
• Ideal in very small networks

There are many routing protocols to choose from

- Each has different features, advantages, and trade-offs
- You may have heard of these four protocols
  - What do these acronyms stand for?
  - How do they work?

Some Well-Known Routing Protocols:
OSPF
IS-IS
BGP
RIP

Some are open standards (anyone can use them).

Some are designed for specialist situations.

Routing Prefix
172.16.10

Host Identifier
0 to 255

Prefix Length
24

Routes are stored in a routing table

- Junos routing tables store all known prefixes, learned from all sources
  - There are separate tables for IPv4, IPv6, and many advanced features
- In the next section, we will introduce a brand new topology
  - Follow along, and see if you can spot the problems

This is an example of a simplified routing table.

Notice that each prefix lists how it was learned.

```
10.20.30.0/24
-- Direct, via ge-0/0/1.0

192.168.77.0/24
-- Static, via ge-0/0/2.0, to 10.1.2.1

192.168.123.0/24
-- OSPF, via ge-0/0/3.0, to 10.2.3.3
```

Junos Routing Fundamentals

- Static routes are simple to deploy and easy to understand
  - However, they have three important trade-offs
  - Each of these problems is explained next

Junos IPv4 RIB	Junos IPv6 RIB
inet.0	inet6.0

1. Static routes cannot react intelligently to changes in the topology.

```
[edit]
employee@ROUTER_1# set routing-options static route 172.16.40.0/24 next-hop 10.1.2.2
```

2. If a new subnet is added, you must manually update each router.

3. Routers do not communicate to verify that their static routing knowledge is accurate.

```
employee@ROUTER_1# show | compare
[edit]
+  routing-options {
+      static {
+          route 172.16.40.0/24 next-hop 10.1.2.2;
+      }
```

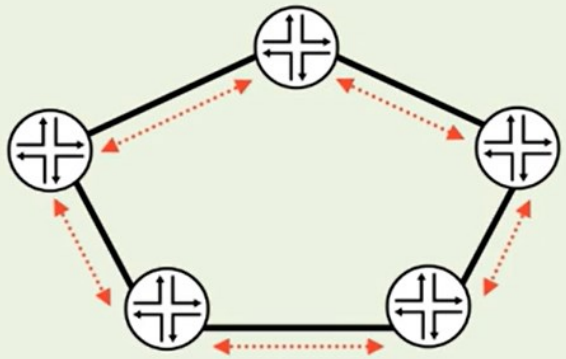
```
employee@ROUTER_1# set routing-options rib inet6.0 static route 2001:db8:0:40::/64 next-hop 2001:db8:1:2::2
```

```
employee@ROUTER_1> show route protocol static

inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 00:00:40
                    >  to 10.1.254.254 via xe-0/1/6.0
172.16.40.0/24     *[Static/5] 00:00:10
                    >  to 10.1.2.2 via xe-0/1/5.0

inet6.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

::/0               *[Static/5] 00:02:02
                    >  to 2001:db8:1:254::254 via xe-0/1/6.0
2001:db8:0:40::/64 *[Static/5] 00:00:07
```

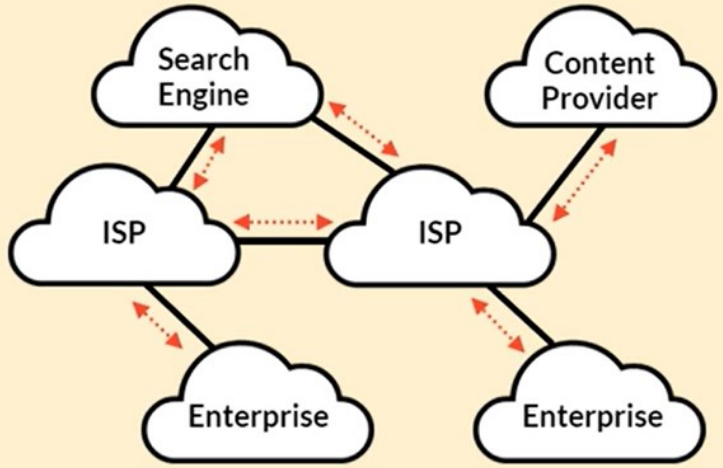
There are protocols for different situations, with different advantages

- All routing protocols can be categorized into one of two types:

Interior Gateway Protocols (IGP)
Runs inside the network of one organization



Exterior Gateway Protocols (EGP)
Runs between organizations



There are two link-state protocols

- They both have a great deal in common
- Both are popular and excellent choices in the 2020s

OSPF:
Open Shortest Path First

- "Open" means it's an open standard
- Popular in enterprises, service providers, and data centers

IS-IS:
Intermediate System to Intermediate System

- Intermediate System is an older word for router
- Pronounced "Eye Ess Eye Ess"
- Popular in service providers and data centers
  - Not often used in enterprises

You use EGPs (exterior gateway protocols) between organizations

- Nowadays, there is only one exterior protocol
  - BGP—Border Gateway Protocol

IGPs are built for use within one single organization

- You use them to learn your own prefixes, in your own network
  - These are called your "infrastructure prefixes"

Advantage:

They can react very fast to network changes.

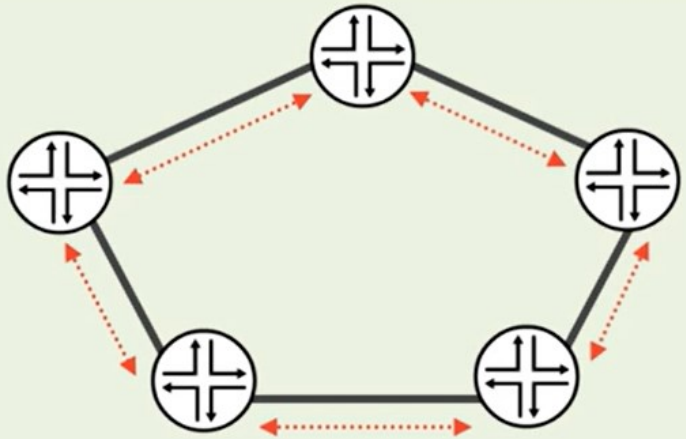
Advantage:

They can discover detailed network information.

Trade-Off:

Not designed to learn every prefix on the global Internet.

A Network within One Organization



Distance Vector Protocols

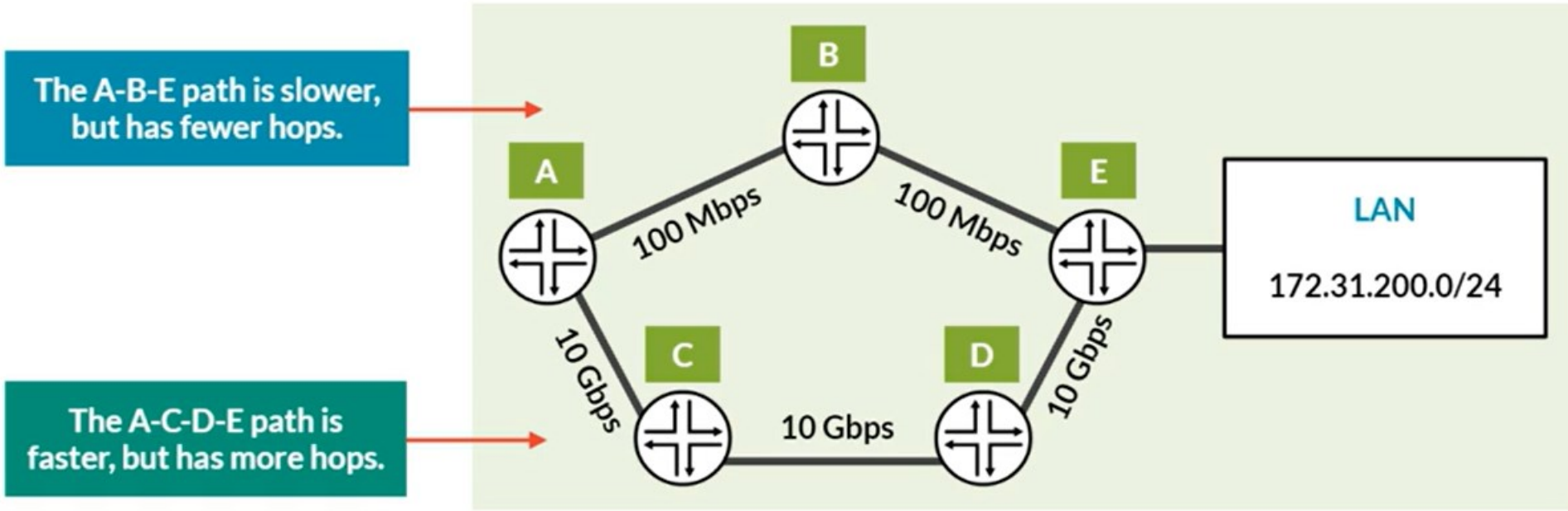
- Very basic
- The best path is the shortest number of hops away—regardless of link speed
- Legacy, slow, rarely used nowadays

Link-State Protocols

- More advanced
- The best path is calculated by learning the topology, including link speeds

Think of a "vector" as a connection between two routers

- Distance vector protocols only learn the hop count of the network
  - They do not learn the speed of the links!
- Consider the best path from A to E
  - Can you identify which path A chooses?



There are two link-state protocols

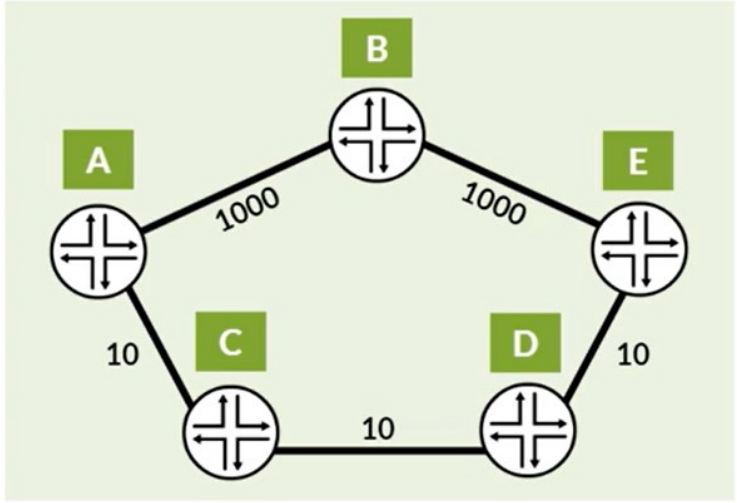
- They both have a great deal in common
- Both are popular and excellent choices in the 2020s

OSPF:
Open Shortest Path First

- "Open" means it's an open standard
- Popular in enterprises, service providers, and data centers

IS-IS:
Intermediate System to Intermediate System

- Intermediate System is an older word for router
- Pronounced "Eye Ess Eye Ess"
- Popular in service providers and data centers
  - Not often used in enterprises



Popular custom metrics you'll find in modern networks:

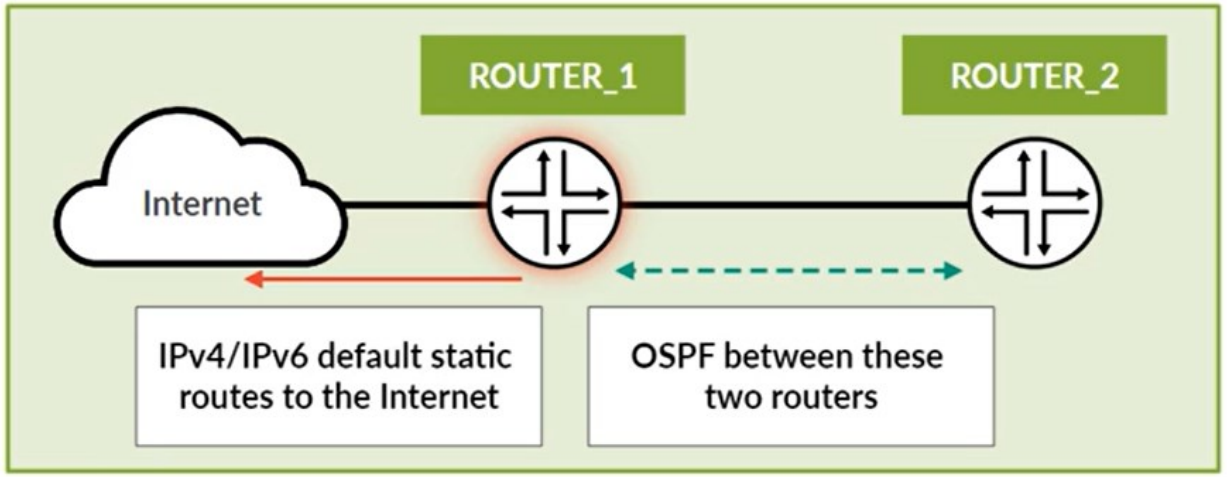
Speed	Metric
100 Gbps	1
10 Gbps	10
1 Gbps	100
100 Mbps	1000

- There is an advantage to using metric instead of link speed
  - It's your choice—the best path doesn't always have to be the fastest link

Junos Routing - OSPF

```
[edit]
employee@ROUTER_1# delete routing-options static route 172.16.40.0/24
```

```
[edit]
employee@ROUTER_1# delete routing-options rib inet6.0 static route 2001:db8:0:40::/64
```



The End Goal:

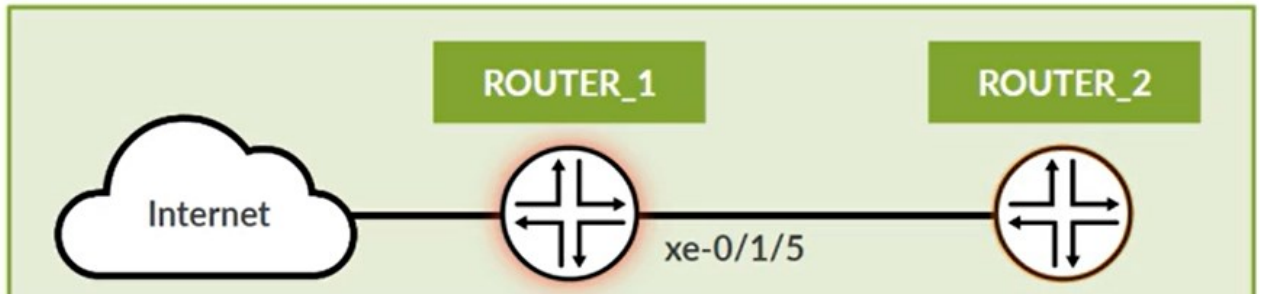
- Form an OSPF neighborship between the two routers
- Both routers advertise their LANs using OSPF
- ROUTER_1 advertises default routes to ROUTER_2 via OSPF

Only one line of config is needed to enable IPv4 OSPF

- Configured using `protocols ospf`
- Put the `interface` into `area 0`
  - You can just type `area 0`—Junos expands it to `area 0.0.0.0`
  - Areas are 32-bit numbers

```
[edit]
employee@ROUTER_1# set protocols ospf area 0 interface xe-0/1/5.0
```

```
[edit]
employee@ROUTER_1# show | compare
[edit protocols]
+    ospf {
+        area 0.0.0.0 {
+            interface xe-0/1/5.0;
+        }
```



When routers receive each other's hello, they try to become neighbors

- Type **`show ospf neighbor`** to see if the process was successful
  - Use detail to see the total uptime
  - `ID` uniquely identifies each OSPF device in the topology
  - `Pri` is short for priority

```
employee@ROUTER_1> show ospf neighbor
Address          Interface              State           ID               Pri  Dead
10.1.2.2         xe-0/1/5.0             Full            10.1.2.2         128    31
```

Your neighbor's interface IP.

Outgoing Interface

`Full` means the process is complete.

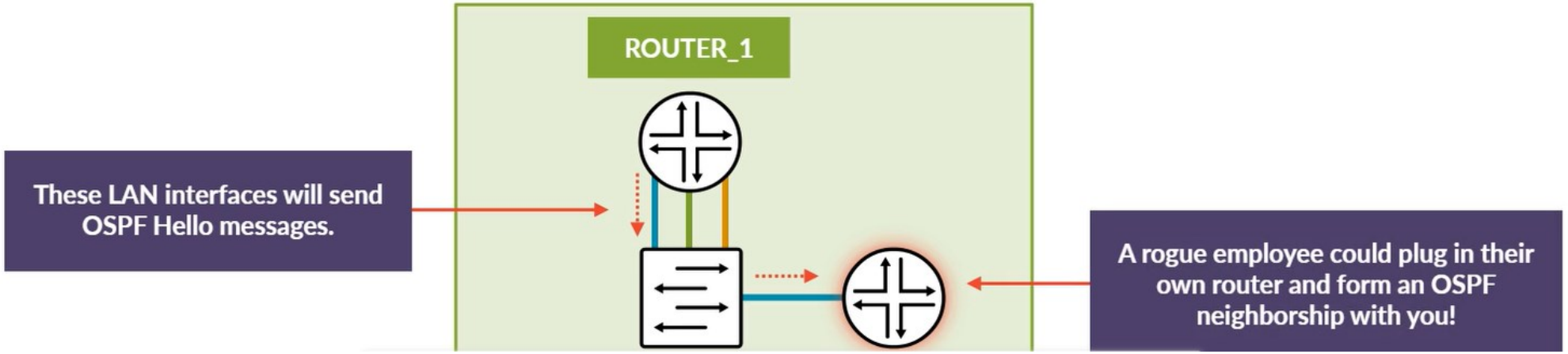
Countdown since the last Hello was received.

OSPF only advertises what you tell it to advertise

- It advertises two things by default:
  - Prefixes on interfaces configured for OSPF
  - Prefixes learned by OSPF from other routers
- You could configure the LAN interfaces for OSPF
  - But this presents a new problem…

Configure each LAN interface in `passive` mode

- This advertises the prefix in OSPF…
- …but it does not send hello messages out of the interface




```
employee@ROUTER_1# set protocols ospf area 0 interface xe-0/1/1.0 passive

[edit]
employee@ROUTER_1# set protocols ospf area 0 interface xe-0/1/2.0 passive
```

IS-IS can advertise both IPv4 and IPv6

- By contrast, OSPF was originally designed to only advertise IPv4
- A new version was required to advertise IPv6

OSPFv2
• Advertises IPv4 only
• When people say OSPF, they often mean v2
• Configured and verified using `ospf`

```
employee@ROUTER_1> show ospf neighbor
Address          Interface              State           ID               Pri  Dead
10.1.2.2         xe-0/1/5.0             Full            10.1.2.2         128    31
```

OSPFv3
• Can advertise IPv6 on its own
• Can advertise IPv4 and IPv6 at the same time

```
employee@ROUTER_1> show ospf3 neighbor
ID                   Interface              State     Pri   Dead
10.1.2.2             xe-0/1/5.0             Full      128     38
  Neighbor-address fe80::3a4f:49ff:fe71:f40c
```

- Review all your OSPF work in this module, in hierarchy format
- All interfaces are indented under `area 0.0.0.0`
- The `passive` statement is indented under some interfaces

```
employee@ROUTER_1> show configuration protocols ospf
area 0.0.0.0 {
    interface xe-0/1/5.0;
    interface xe-0/1/1.0 {
        passive;
    }
    interface xe-0/1/2.0 {
        passive;
    }
    interface xe-0/1/3.0 {
        passive;
    }
```

Protokół	Juniper Preference
Direct (interfejs lokalny)	0
Static	5
OSPF internal	10
IS-IS Level 1/2	15
RIP	100
BGP	170 (internal), 170 (external)

Lab 4

Podstawy konfiguracji Junos OS (vJunos)

Dynamic Routing Protocols - konfiguracja OSPF

Junos OS Switches

- Wprowadzenie do kursu i platformy warsztatów

- **Moduł 0:** Introducing Junos OS and the Hardware
- **Moduł 1:** Junos Fundamentals
- **Moduł 2:** Junos User Interface
- **Moduł 3:** Junos Basic Configuration
- **Moduł 9:** Junos J-Web
- **Moduł 4:** Junos Logging and Monitoring
- **Moduł 5:** Junos Routing Policy and Firewall Filters
- **Moduł 6:** Junos Routing Fundamentals
- **Moduł 7:** Junos OS Switches
- **Moduł 8:** Junos OS Architecture
- **Moduł 10:** Jak przygotować się do JNCIA-Junos

Junos OS Switches

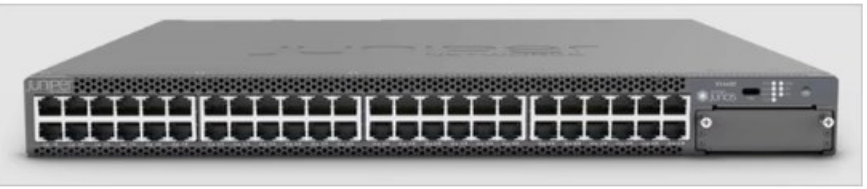
Some vendors make you learn a new CLI for switches—but not us!

- There is one CLI across all devices that run Junos OS

These devices all have the same CLI.

Routers

Switches



Firewalls

There are a few commands that are unique to certain platforms.

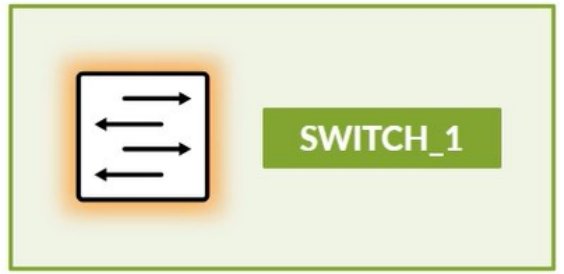
Example: Firewalls offer additional security options

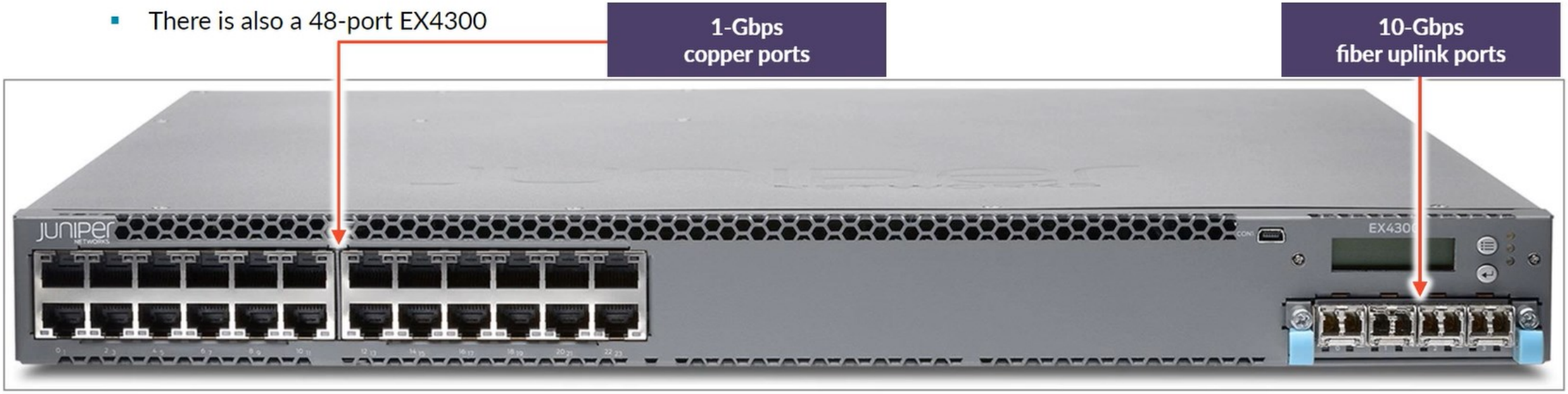
There are occasional CLI differences between platforms.

For example, to accommodate for an expanded feature set

Junos OS Switches

- Two ranges of switches—EX and QFX Series
- This is the hardware that powers your customer's LAN
  - Junos Switches usually have many ports
    - Their role is to connect LAN devices together
  - This particular switch has 24 ports for hosts
    - There is also a 48-port EX4300





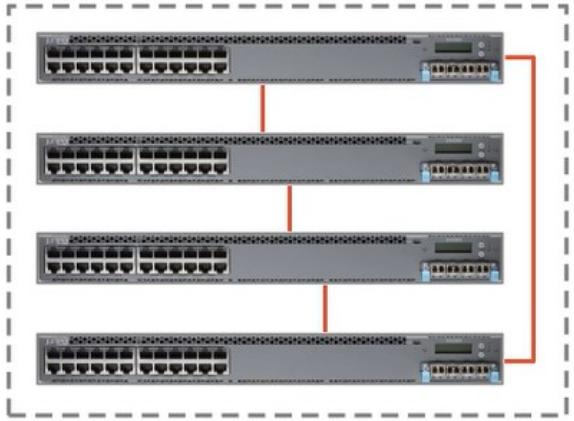
Junos OS Switches

- Take a look at the interface naming scheme on this switch
  - PIC 0 and PIC 2 are on the front of the device
    - PIC 0 is used exclusively as network ports
    - PIC 2 can be used as network ports—but also offers a second application…

Junos OS Switches

- There are ports at the back that offer additional functionality
  - On this EX4300, the back ports are preconfigured as VC ports
    - PIC 2 can also be configured as VC ports
  - Virtual Chassis (VC) comes as a preconfigured feature
    - You can connect switches together to create one logical switch

PIC 1
40 Gbps ports
Interface names:
et-0/1/0 to et-0/1/3



Four physical switches, but one logical switch for management

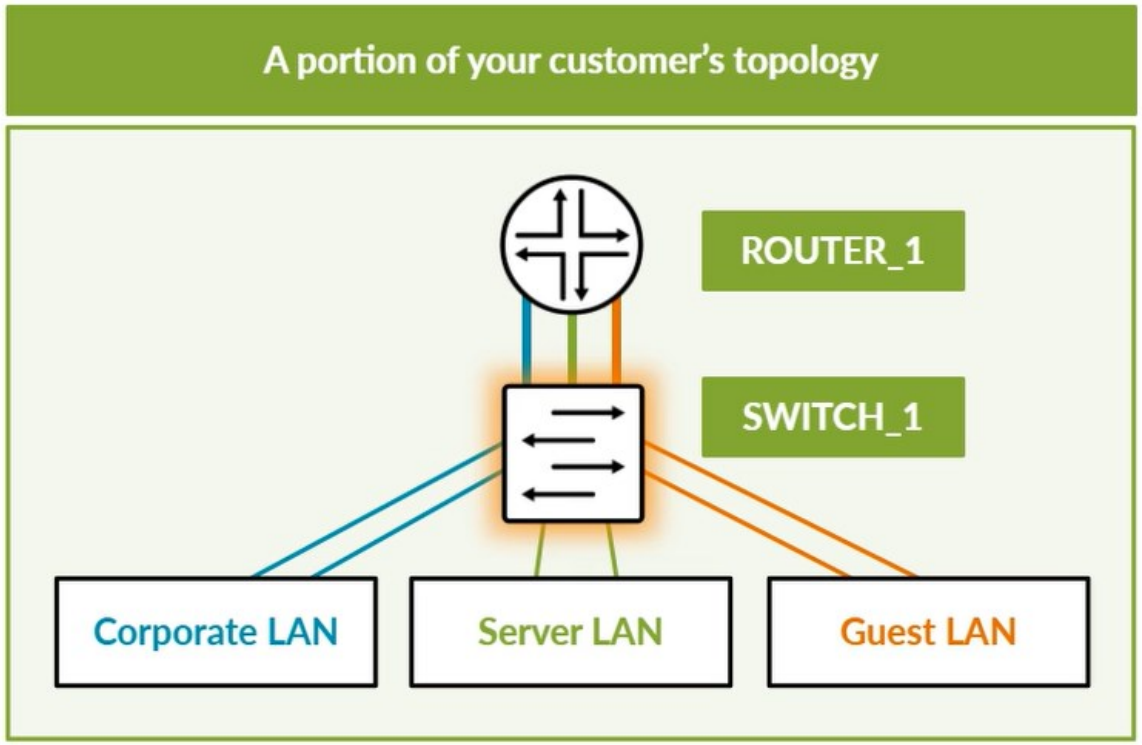
By the way, at this early stage in your studies, we don't recommend spending too much time memorizing which ports can be used as VC ports, because this can change radically between devices. It's definitely something that you can return to when you study VCs in more detail at a later date.

Junos OS Switches

The uplinks to the router are 10-Gbps interfaces

- This table lists all active switch network interfaces and the VLAN they should belong to
- Look out for mismatches between this table and the CLI config

Interface	Network	Description
ge-0/0/0	Corporate LAN	Host Port
ge-0/0/1	Corporate LAN	Host Port
ge-0/0/2	Server LAN	Host Port
ge-0/0/3	Server LAN	Host Port
ge-0/0/4	Guest LAN	Host Port
ge-0/0/5	Guest LAN	Host Port
xe-0/2/1	Corporate LAN	To ROUTER_1
xe-0/2/2	Server LAN	To ROUTER_1
xe-0/2/3	Guest LAN	To ROUTER_1



Junos OS Switches

- To view VLAN configuration, type **show configuration vlans**
  - Each VLAN is given:
    - A name
    - A VLAN ID number (the “tag”)
  - You then configure each interface with the appropriate VLAN

```
{master:0}
employee@SWITCH_1> show configuration vlans
Corporate {
    vlan-id 10;
}
Server {
    vlan-id 20;
}
```

```
{master:0}
employee@SWITCH_1> show configuration vlans | display set
set vlans Corporate vlan-id 10
set vlans Server vlan-id 20
```

Junos OS Switches

```
{master:0}[edit]
employee@SWITCH_1# set vlans Guest vlan-id 30
```

```
{master:0}[edit]
employee@SWITCH_1# show | compare
[edit vlans]
+    Guest {
+        vlan-id 30;
+    }

{master:0}[edit]
employee@SWITCH_1# commit and-quit
```

VLAN Name	Number
Corporate	10
Server	20
Guest	30

VLAN names are case-sensitive, so be consistent with your naming convention.

Junos OS Switches

```
{master:0}
employee@SWITCH_1> show vlans

Routing instance        VLAN name               Tag          Interfaces
default-switch          Corporate               10
                                                             ge-0/0/0.0*
                                                             ge-0/0/1.0*
                                                             xe-0/2/1.0*
default-switch          Guest                   30

default-switch          Server                  20
                                                             ge-0/0/2.0*
                                                             ge-0/0/3.0*
                                                             xe-0/2/2.0*
default-switch          default                 1
                                                             ge-0/0/10.0
                                                             ge-0/0/11.0
                                                             ge-0/0/12.0
```

Junos OS Switches

```
employee@SWITCH_1> show configuration interfaces ge-0/0/1 | display set
set interfaces ge-0/0/1 description "Corporate LAN"
set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members Corporate
```

It's easy to define the interface mode and the VLAN.

Look at the two pieces of config at the end of `family ethernet-switching`.

```
{master:0}
employee@SWITCH_1> show configuration interfaces ge-0/0/1
description "Corporate LAN";
unit 0 {
    family ethernet-switching {
        interface-mode access;
        vlan {
            members Corporate;
        }
    }
}
```

Using this knowledge, you can configure the three Guest VLAN ports

- The interfaces are:
  - ge-0/0/4, ge-0/0/5, and xe-0/2/3

The current settings on ge-0/0/4
(the other two ports have identical config)

```
{master:0}
employee@SWITCH_1> show configuration | display set | match ge-0/0/4
set interfaces ge-0/0/4 description "Guest LAN"
set interfaces ge-0/0/4 unit 0 family ethernet-switching interface-mode access
```

Simply add this line of configuration
(and the equivalent for the other two interfaces)

```
{master:0}[edit]
employee@SWITCH_1# set interfaces ge-0/0/4 unit 0 family ethernet-switching vlan members Guest
```

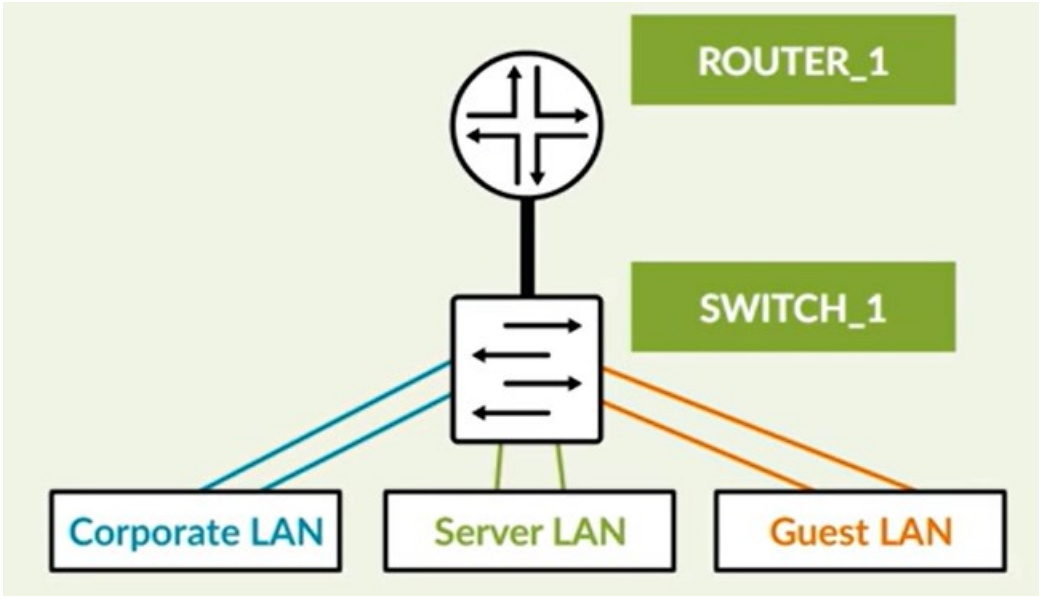
```
{master:0}
employee@SWITCH_1> show vlans Guest

Routing instance        VLAN name             Tag          Interfaces
default-switch          Guest                 30
                                                           ge-0/0/4.0*
                                                           ge-0/0/5.0*
                                                           xe-0/2/3*
```

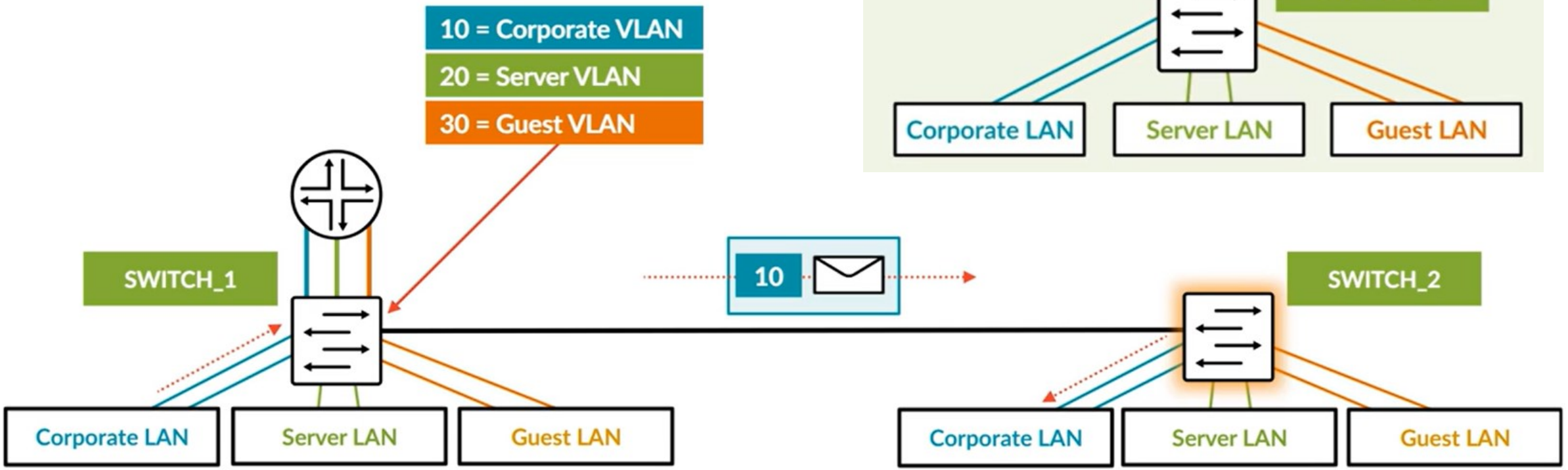
Junos OS Switches

```
employee@SWITCH_1> show ethernet-switching table

MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static, C - Control MAC
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC)


Ethernet switching table : 9 entries, 9 learned
Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical          NH        RTR
    name                address             flags              interface        Index     ID
    Corporate           38:4f:49:80:f4:08   D             -    xe-0/2/1.0       0         0
    Corporate           74:8f:3c:ba:e0:dd   D             -    ge-0/0/0.0       0         0
    Corporate           94:bf:94:b4:67:03   D             -    ge-0/0/1.0       0         0
    Guest               38:4f:49:80:f4:0a   D             -    xe-0/2/3.0       0         0
    Guest               36:72:dd:d5:e6:44   D             -    ge-0/0/4.0       0         0
    Guest               1e:00:a2:3e:d5:5a   D             -    ge-0/0/5.0       0         0
    Server              38:4f:49:80:f4:09   D             -    xe-0/2/2.0       0         0
    Server              78:8F:3C:00:00:00   D             -    ge-0/0/2.0       0         0
    Server              94:bf:94:b4:67:07   D             -    ge-0/0/3.0       0         0
```

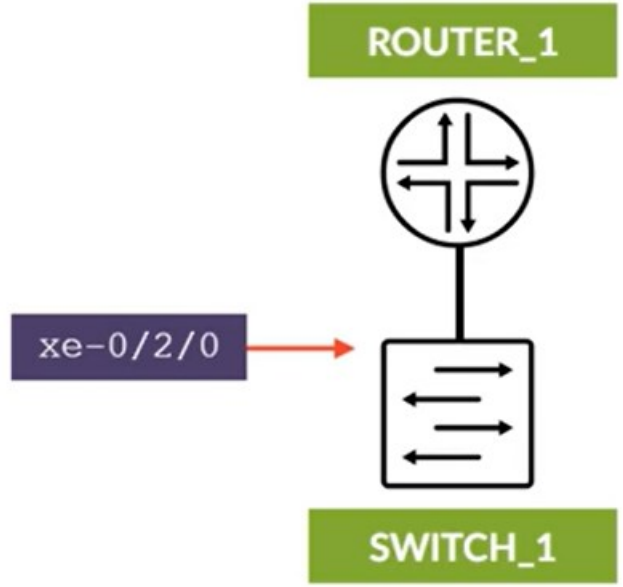
Junos OS Switches

- In practice, traffic in and out is tagged with the relevant VLAN number
  - Trunk ports are the solution—they belong to multiple VLANs at once
    - The tags map incoming frames to the relevant VLAN
    - And therefore, the relevant broadcast domain





```
{master:0}[edit]
employee@SWITCH_1# edit interfaces xe-0/2/0

{master:0}[edit interfaces xe-0/2/0]
employee@SWITCH_1# set description "Trunk to ROUTER_1"

{master:0}[edit interfaces xe-0/2/0]
employee@SWITCH_1# set unit 0 family ethernet-switching interface-mode trunk
```




```
{master:0}[edit interfaces xe-0/2/0]
employee@SWITCH_1# set unit 0 family ethernet-switching vlan members Corporate

{master:0}[edit interfaces xe-0/2/0]
employee@SWITCH_1# set unit 0 family ethernet-switching vlan members Server

{master:0}[edit interfaces xe-0/2/0]
employee@SWITCH_1# set unit 0 family ethernet-switching vlan members Guest
```

```
{master:0}[edit interfaces xe-0/2/0]
employee@SWITCH_1# set unit 0 family ethernet-switching vlan members [ Corporate Server Guest ]
```

```
{master:0}[edit interfaces xe-0/2/0 unit 0 family ethernet-switching]
employee@SWITCH_1# set vlan members all
```

Junos OS Switches

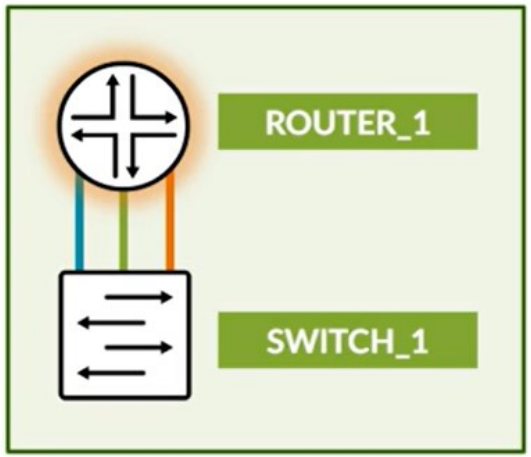
```
{master:0}
employee@SWITCH_1> show configuration interfaces xe-0/2/0
description "Trunk to ROUTER_1";
unit 0 {
    family ethernet-switching {
        interface-mode trunk;
        vlan {
            members [ Corporate Server Guest ];
        }
    }
}
```

```
employee@SWITCH_1> show vlans

Routing instance        VLAN name               Tag          Interfaces
default-switch          Corporate               10
                                                             ge-0/0/0.0*
                                                             ge-0/0/1.0*
                                                             xe-0/2/0.0*
default-switch          Guest                   30
                                                             ge-0/0/4.0*
                                                             ge-0/0/5.0*
                                                             xe-0/2/0.0*
default-switch          Server                  20
                                                             ge-0/0/2.0*
                                                             ge-0/0/3.0*
                                                             xe-0/2/0.0*
```

Junos OS Switches

- Take all this config for these three interfaces
- Put it all onto one single physical interface
- You can `match` on multiple strings of text at once

Check out this powerful search option.




```
employee@ROUTER_1> show configuration interfaces | display set | match "1/1|1/2|1/3"
set interfaces xe-0/1/1 description "Corporate LAN"
set interfaces xe-0/1/1 unit 0 family inet address 172.16.10.1/24
set interfaces xe-0/1/1 unit 0 family inet6 address 2001:db8:0:10::1/64

set interfaces xe-0/1/2 description "Server LAN"
set interfaces xe-0/1/2 unit 0 family inet address 172.16.20.1/24
set interfaces xe-0/1/2 unit 0 family inet6 address 2001:db8:0:20::1/64

set interfaces xe-0/1/3 description "Guest LAN"
set interfaces xe-0/1/3 unit 0 family inet address 172.16.30.1/24
set interfaces xe-0/1/3 unit 0 family inet6 address 2001:db8:0:30::1/64
```

Junos OS Switches

This statement enables IP interfaces to accept VLAN tags.

```
set interfaces xe-0/1/0 vlan-tagging
```

```
set interfaces xe-0/1/0 unit 10 description "Corporate LAN"
set interfaces xe-0/1/0 unit 10 vlan-id 10
set interfaces xe-0/1/0 unit 10 family inet address 172.16.10.1/24
set interfaces xe-0/1/0 unit 10 family inet6 address 2001:db8:0:10::1/64
```

The configuration for each unit is identical to a single-unit configuration.

```
set interfaces xe-0/1/0 unit 20 description "Server LAN"
set interfaces xe-0/1/0 unit 20 vlan-id 20
set interfaces xe-0/1/0 unit 20 family inet address 172.16.20.1/24
set interfaces xe-0/1/0 unit 20 family inet6 address 2001:db8:0:20::1/64
```

Each unit gets a unique `vlan-id`.

```
set interfaces xe-0/1/0 unit 30 description "Guest LAN"
set interfaces xe-0/1/0 unit 30 vlan-id 30
set interfaces xe-0/1/0 unit 30 family inet address 172.16.30.1/24
set interfaces xe-0/1/0 unit 30 family inet6 address 2001:db8:0:30::1/64
```

Junos OS Switches

```
employee@ROUTER_1> show configuration interfaces xe-0/1/0
vlan-tagging;
unit 10 {
    description "Corporate LAN";
    vlan-id 10;
    family inet {
        address 172.16.10.1/24;
    }
    family inet6 {
        address 2001:db8:0:10::1/64;
    }
}
unit 20 {
    description "Server LAN";
    vlan-id 20;
    family inet {
        address 172.16.20.1/24;
    }
    family inet6 {
        address 2001:db8:0:20::1/64;
    }
}
unit 30 {
    description "Guest LAN";
    vlan-id 30;
    family inet {
        address 172.16.30.1/24;
    }
    family inet6 {
        address 2001:db8:0:30::1/64;
    }
}
```

Lab 5

Podstawy konfiguracji Junos OS (vJunos)

Switches - konfiguracja VLAN

Junos OS Architecture

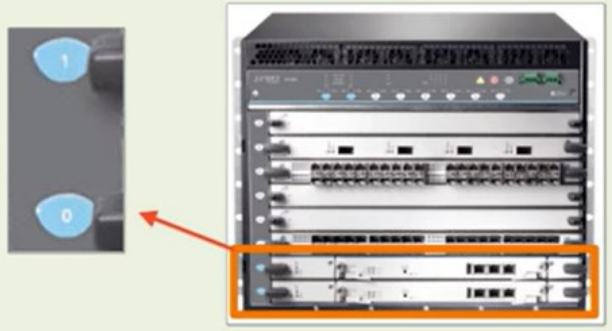
- Wprowadzenie do kursu i platformy warsztatów

- **Moduł 0:** Introducing Junos OS and the Hardware
- **Moduł 1:** Junos Fundamentals
- **Moduł 2:** Junos User Interface
- **Moduł 3:** Junos Basic Configuration
- **Moduł 9:** Junos J-Web
- **Moduł 4:** Junos Logging and Monitoring
- **Moduł 5:** Junos Routing Policy and Firewall Filters
- **Moduł 6:** Junos Routing Fundamentals
- **Moduł 7:** Junos OS Switches
- **Moduł 8:** Junos OS Architecture
- **Moduł 10:** Jak przygotować się do JNCIA-Junos

Junos OS Architecture

- A Routing Engine is just a regular computer, containing exactly the same kind of CPU and memory
- A Routing Engine performs many different tasks
  - Monitoring, management, system, chassis, protocols, and more
  - For this reason, it has a "general-purpose" CPU
    - Not specialized to any one task

Some devices have primary and backup REs, both of which can be replaced



Some REs are fixed inside the chassis, and cannot be removed

You've learned enough Junos to revisit the RE's responsibilities in more detail

- This table focuses on device, system, and process management

The Routing Engine runs and maintains:	
• Junos OS itself	
• The processes that make up the functionality of Junos OS	*Separate processes for routing, chassis management, DHCP, interfaces, and more*
• The chassis and system	*Electrical power, fans/cooling, maintaining line cards and interfaces*
• All management traffic	*CLI, J-Web, external platforms, interaction via user-created code/scripts*
• Responses to ping and traceroute	

The Routing Engine runs and maintains:	
• Running all protocols	*Routing, switching, monitoring, management*
• Maintaining routing tables	
• Choosing active routes	*By default, one active route per-prefix*
• Installing active routes in a separate forwarding table	
• Writing the forwarding table to a dedicated forwarding hardware	

Junos OS Architecture

The RE is not responsible for processing transit traffic

- Performed by a completely separate set of dedicated, specialized infrastructure in the device
- To understand how this works, let's explore the forwarding table

So far, you've been told that:

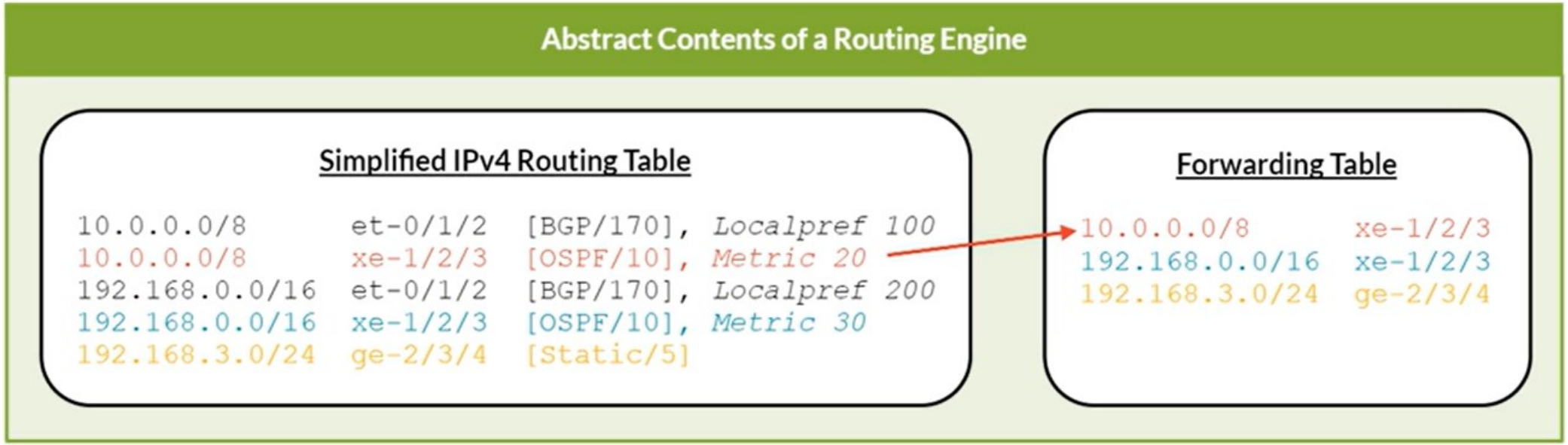
- Junos processes routes in the routing table
- Junos chooses the best routes for forwarding
- Junos tags them as "active routes"

Here's what happens next:

- Junos writes all active routes to a dedicated "forwarding table"
- It only copies info that is required for forwarding
- The RE stores and maintains this table

The forwarding table does broadly keep track of how a prefix was installed into the table

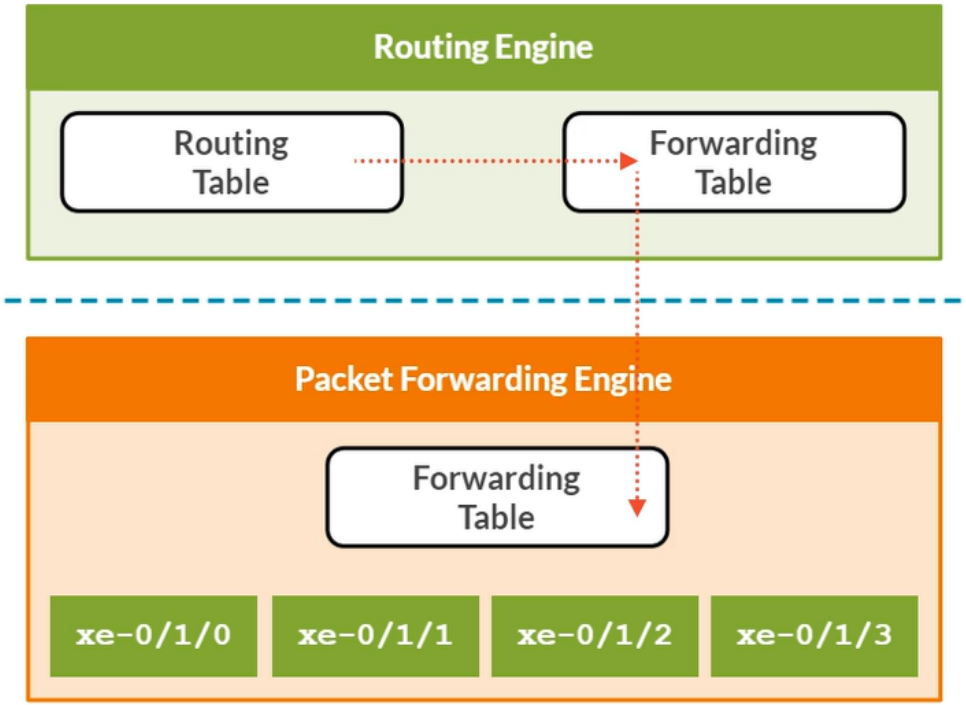
- No visibility in the table of the originating protocol



Routing tables: Routing information base (RIB)

Forwarding tables: Forwarding information base (FIB)

RIB:	FIB:
May contain many millions of prefixes	Only needs to contain the best routes
Contains all the protocol information for each route	Only needs to contain the information required for forwarding



The brains of the device

- Processes the best paths to all prefixes, as they are learned

The muscle of the device

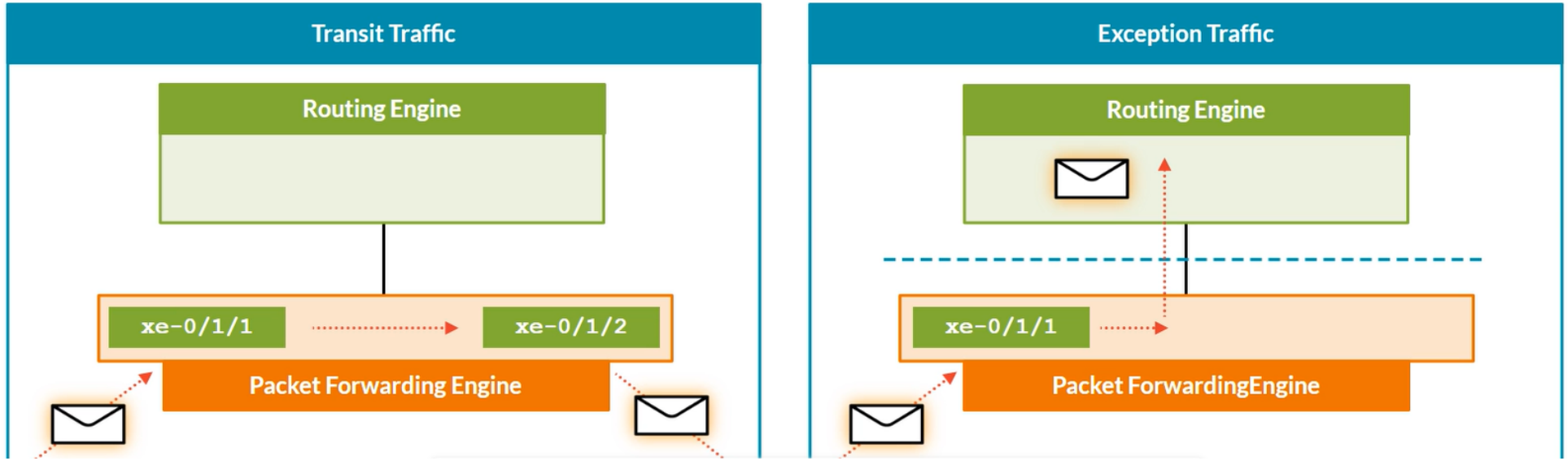
- Designed to forward traffic as fast and efficiently as possible
- It follows the instructions in the forwarding table

- On (almost) all platforms, the PFE is a set of dedicated hardware
- The PFE's engine is a special chip called an application-specific integrated circuit (ASIC)
  - Designed to perform a narrow range of tasks—and perform them extremely well
- ASICs are extremely good at performing a very limited set of tasks
  - ASICs do them in orders of magnitudes faster than a general-purpose CPU

The Main Tasks of the PFE:	
• Look up and forward incoming traffic	
• Manipulate packet info	*Ethernet headers, VLAN tags, decrementing time-to-live and Hop Limit*
• Control services that can impact traffic	*Rate-limiting (policing), firewall filters, prioritizing certain traffic*

Junos OS Architecture

BIG advantages to this separation:
• If the RE crashes, the PFE can continue to forward traffic
• Their hardware can be designed for optimal performance of their tasks
• Their hardware can be upgraded independently
• Junos adds rate-limiting and denial-of-service (DoS) protection on the internal link between the RE and PFE



Jak przygotować się do JNCIA-Junos

- Wprowadzenie do kursu i platformy warsztatów

- **Moduł 0:** Introducing Junos OS and the Hardware
- **Moduł 1:** Junos Fundamentals
- **Moduł 2:** Junos User Interface
- **Moduł 3:** Junos Basic Configuration
- **Moduł 9:** Junos J-Web
- **Moduł 4:** Junos Logging and Monitoring
- **Moduł 5:** Junos Routing Policy and Firewall Filters
- **Moduł 6:** Junos Routing Fundamentals
- **Moduł 7:** Junos OS Switches
- **Moduł 8:** Junos OS Architecture
- **Moduł 10:** Jak przygotować się do JNCIA-Junos

Jak przygotować się do JNCIA-Junos

https://community.juniper.net/discussion/how-to-prepare-for-a-certification-exam

Certification Tracks	Associate Level	Specialist Level	Professional Level	Expert Level
Automation and DevOps	JNCIA-DevOps	JNCIS-DevOps	n/a	n/a
Cloud	JNCIA-Cloud	n/a	n/a	n/a
Data Center	JNCIA-DC	JNCIS-DC	JNCIP-DC*	JNCIE-DC
Design	JNCIA-Design	n/a	n/a	n/a
Enterprise Routing and Switching	JNCIA-Junos	JNCIS-ENT	JNCIP-ENT	JNCIE-ENT
Mist AI	JNCIA-MistAI	JNCIS-MistAI-Wireless JNCIS-MistAI-Wired	JNCIP-MistAI	n/a
Security	JNCIA-SEC	JNCIS-SEC	JNCIP-SEC	JNCIE-SEC
Service Provider Routing and Switching	JNCIA-Junos	JNCIS-SP	JNCIP-SP	JNCIE-SP

* The JNCIS-ENT is an acceptable prerequisite for the JNCIP-DC.