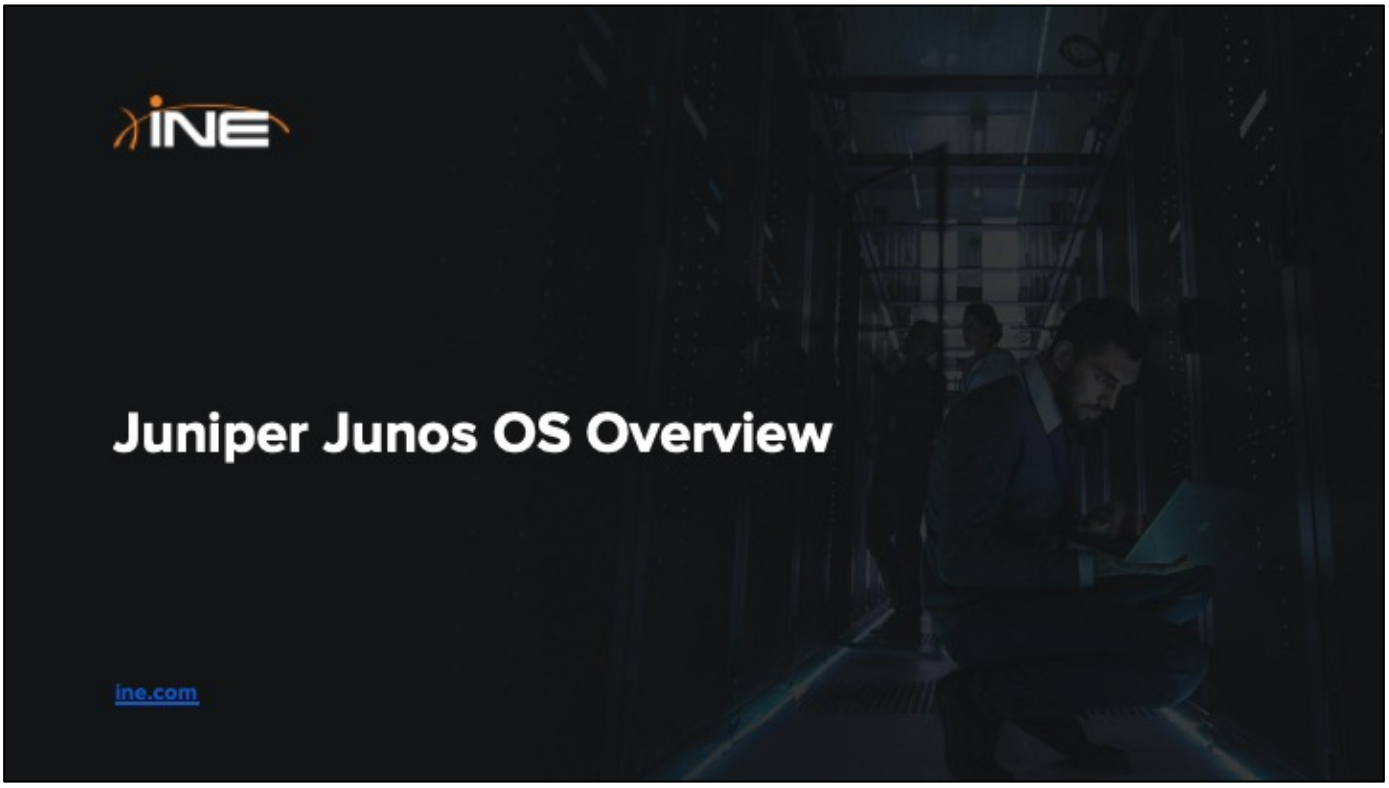
+ None

Course Prerequisites

Course Objectives

- Understand The Benefits Of The Junos OS Software
- Explain The Differences Between The Control, Forwarding, & Management Planes
- Identify The Functionality Of The Juniper RE & PFE Hardware
- Summarize The Purpose Of Junos Daemons & How To Restart Them



Topic Overview

- General Characteristics Of Junos OS
- Junos OS Evolved Overview
- Junos Release Schedule

Characteristics Of Junos OS

- Junos OS is a network operating system
- Runs on top of Free BSD
- Implements Daemons for process isolation
- Operated in a consistent manner across platforms
  - Called “One Junos”

Junos OS can actually be thought of as an application that runs on top of FreeBSD
- Wanted an underlying Operating System that was stable, easy to secure and didn’t require a lot of system resources.

FreeBSD = Open Source OS based on Unix
----This shell can be used for maintenance purposes such as backing up files, viewing log files making some tasks related to software updates easier.

Daemons have dedicated memory spaces

Characteristics Of Junos OS

- Junos OS has a hierarchical command line
- Configuration changes are not applied until they are committed
- During the commit stage Junos OS will check your pending configuration for any errors or conflicting statements
- Junos OS provides easy methods to rollback to previously saved configuration files

Junos OS Evolved

- Junos OS Evolved is an upgrade to Junos OS
- Runs on top of a Linux kernel (not FreeBSD)
- More distributed in nature than Junos OS
- Supports the installation of many, separate versions of Junos OS on a single box
- Most of the CLI is the same as Junos OS
  - Name change to management interfaces
  - From: ***em0, fxp0, me0***
  - To: ***device-name:type-port***
    - re0:mgmt-0, re0:mgmt-1
    - re1:mgmt-0, re1:mgmt-1

- More industry support for Linux-based applications than FreeBSD
- Linux is supported on more types of hardware than FreeBSD
- Allows for more integration of third-party apps into Juniper devices than when Junos OS was running atop freeBSD

The DDS

- Junos OS Evolved applications store their state information in a centralized database
- The “Distributed Data Store”
- State:
  - “The retained information or status about physical or logical entities that the system preserves and shares across the system and supplies during restarts. ”
  - “State includes both operational and configuration state, including committed configuration, interface state, routes, and hardware state. ”
  - “In Junos OS Evolved, state can be held in a central database called the Distributed Data Store (DDS).”

The DDS

- Subscribers and Consumers
  - With regards to state information, applications are either subscribers (aka "publishers" or "originators"), consumers or both
  - Allows for application independence

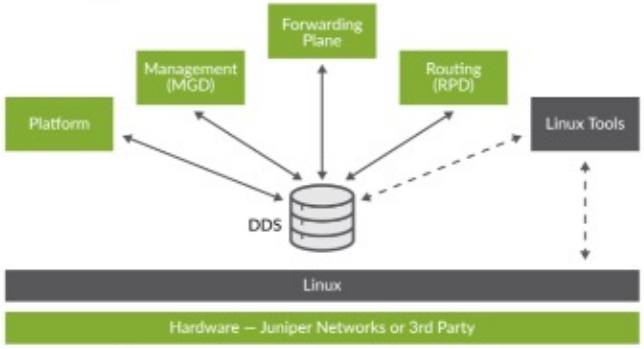


image courtesy of juniper.net/documentation

Release Schedule & Naming Conventions

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

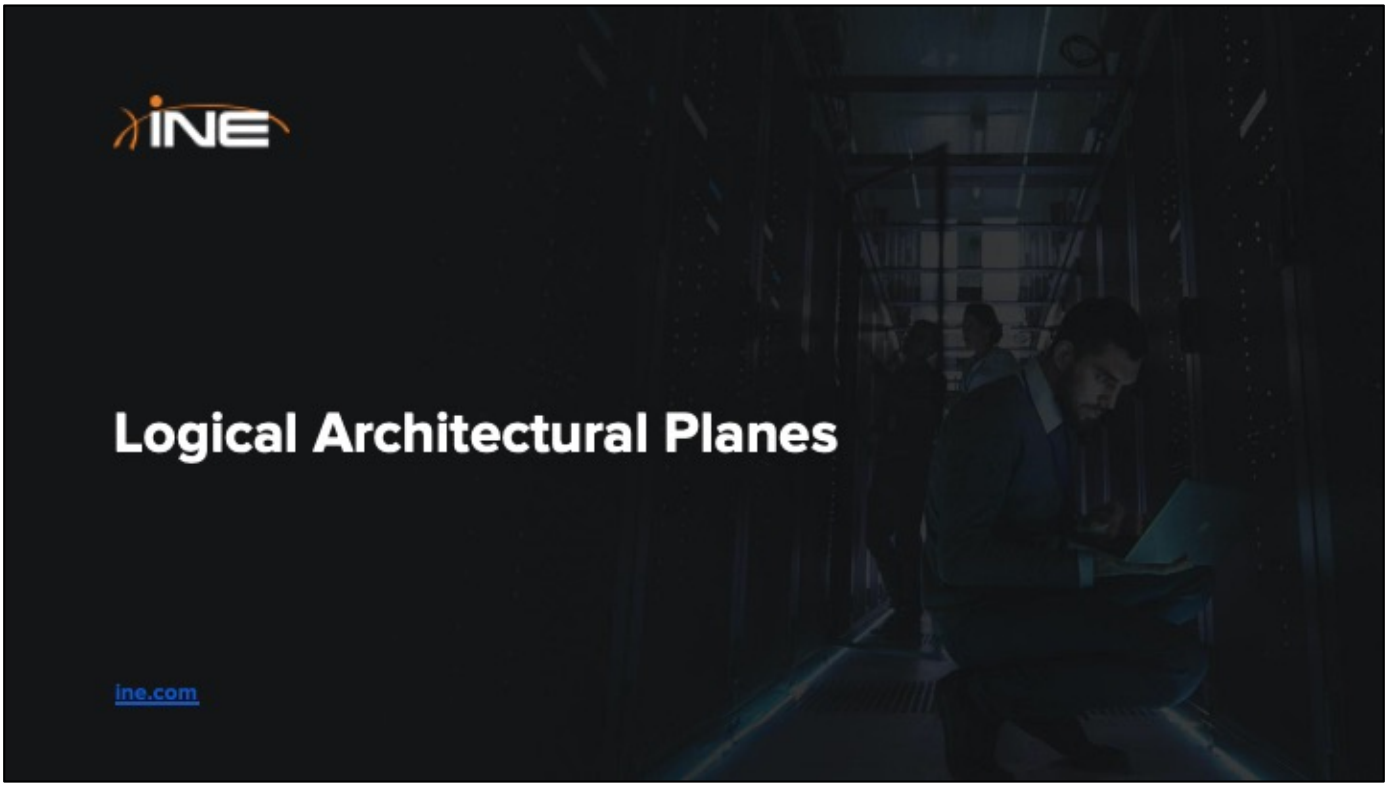
Bug Fixes

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

Historical Software Releases

- https://support.juniper.net/support/eol/software/junos/

Thanks for Watching!



Topic Overview

- Understanding The Functional Planes (Control, Data & Management)
- Transit & Exception Traffic

Introduction To Functional Planes

- Juniper infrastructure devices (routers, switches and firewalls) are internally split into multiple parts known as "planes" which are joined together by fast internal links
  - Control Plane
  - Forwarding Plane
  - Management Plane
- Provides a logical separation of device responsibilities

Control Plane

- CPU resides in this plane
- Responsible for:
  - All processes that learn of network paths (L2 and L3)
  - Populating software forwarding tables
  - Routing Protocol Maintenance
  - Runs the Junos Operating System (OS) and associated daemons

Forwarding Plane

- Also called the "Data Plane" and resides on the PFE (Packet Forwarding Engine)
- Composed of ASIC-based hardware and software microcode
- Does not sit in one piece of hardware but rather **spans many aspects of the chassis and its modules**
- Responsible for quick traffic forwarding using forwarding tables and firewall filters

Management Plane

- A subset of the Control Plane
- Controls any process that is used to manage the box such as:
  - Local CLI via the console port
  - Telnet and SSH CLI sessions
  - SNMP/RESTCONF/NETCONF sessions
  - Incoming, locally-terminated ICMP packets

NOTE: Although ICMP Echo Requests and Replies to the local device are technically handled in the Management Plane...these packets are handled in the PFE (unless an ICMP Unreachable or TTL-Expired needs to be generated)

Transit & Exception

- Traffic that is received by a Juniper devices is classified as either transit or exception traffic
- Transit traffic
  - Traffic meant to be routed (or switched) through the box to a remote destination endpoint
  - Handled by the Data Plane (Forwarding Plane)
- Exception traffic
  - Traffic that is either destined for the local box, or traffic that (due to some unique characteristic) cannot be routed/switched in hardware
  - Handled by the Control or Management Planes

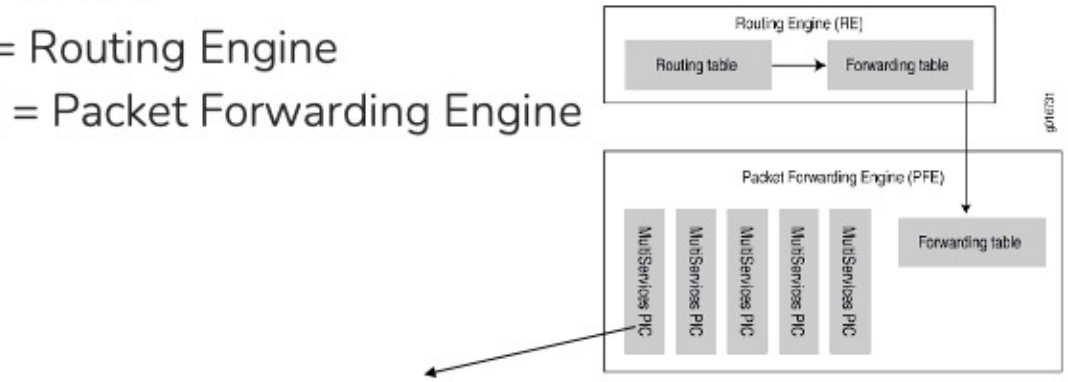
Thanks for Watching!

Topic Overview

+ Introducing The PFE & RE
+ Identifying Functional Differences Between The RE & PFE

Separate Components

- Juniper platform architecture is separated into two components:
  - RE = Routing Engine
  - PFE = Packet Forwarding Engine



PIC = "Physical Interface Concentrator"

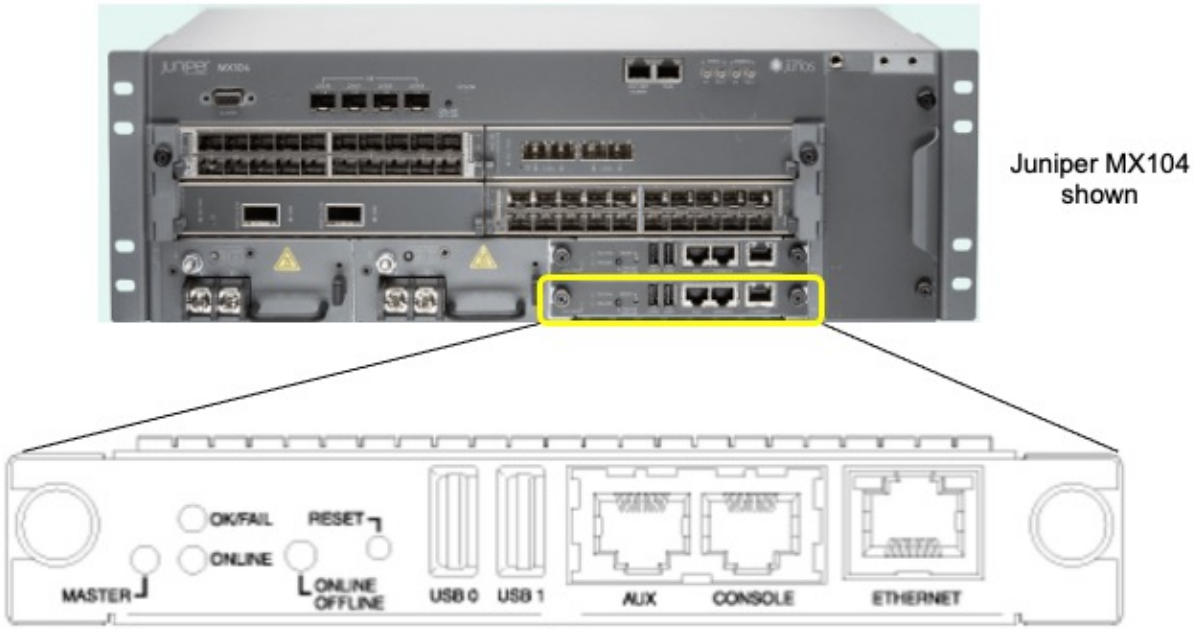
Juniper Router Architecture

Image courtesy of juniper.net/documentation

Juniper Routing Engines

- The RE is the “brains” of the unit.
  - Runs Junos OS and its various daemons
  - Performs various tasks involved with the learning of paths, routes, MAC addresses, controls some chassis components, etc
  - Stores this information in various CPU-accessible tables.
- The RE then disseminates forwarding information to the PFE for fast packet lookups
- Many platforms support redundant REs

Juniper Routing Engines



- The AUX port is present but not supported on this platform.
- The USB ports support external USB Flash Drives for installation of new Junos OS software or as a recovery drive.

Packet Forwarding Engine

- The PFE utilizes ASICS to move traffic between ingress and egress interfaces
- PFEs are a component of linecards, so a single Juniper chassis may have several PFEs
- PFE responsibilities include:
  - Process ingress packets
  - Apply filters, routing policies, and other features
  - Forwards packets to the next hop along the route to their destination

Internal rate-limiting is automatically applied for traffic FROM PFE to the RE to prevent DOS. This is on by default and is called DDOS. It can be configured and modified.

RE & PFE Interaction

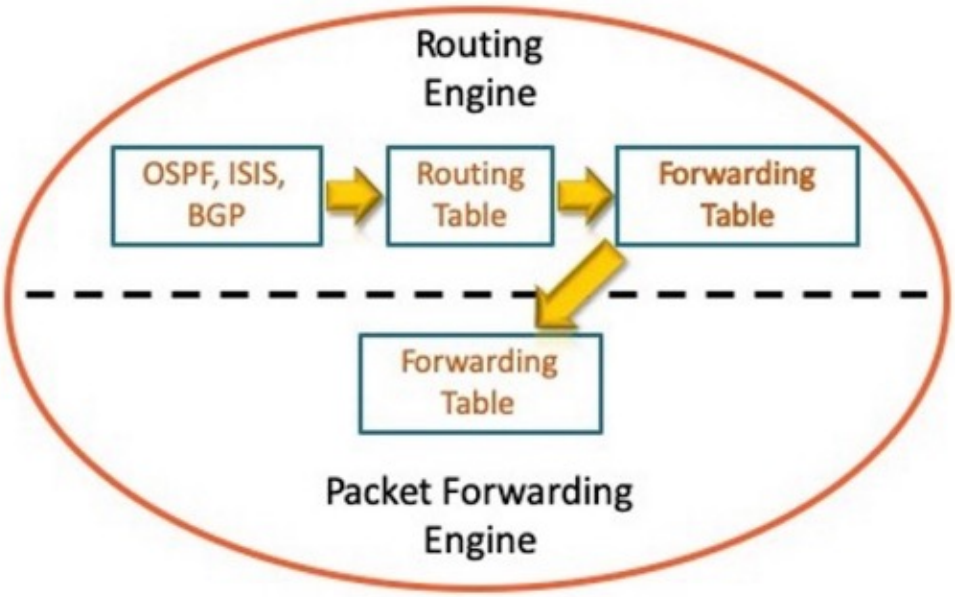
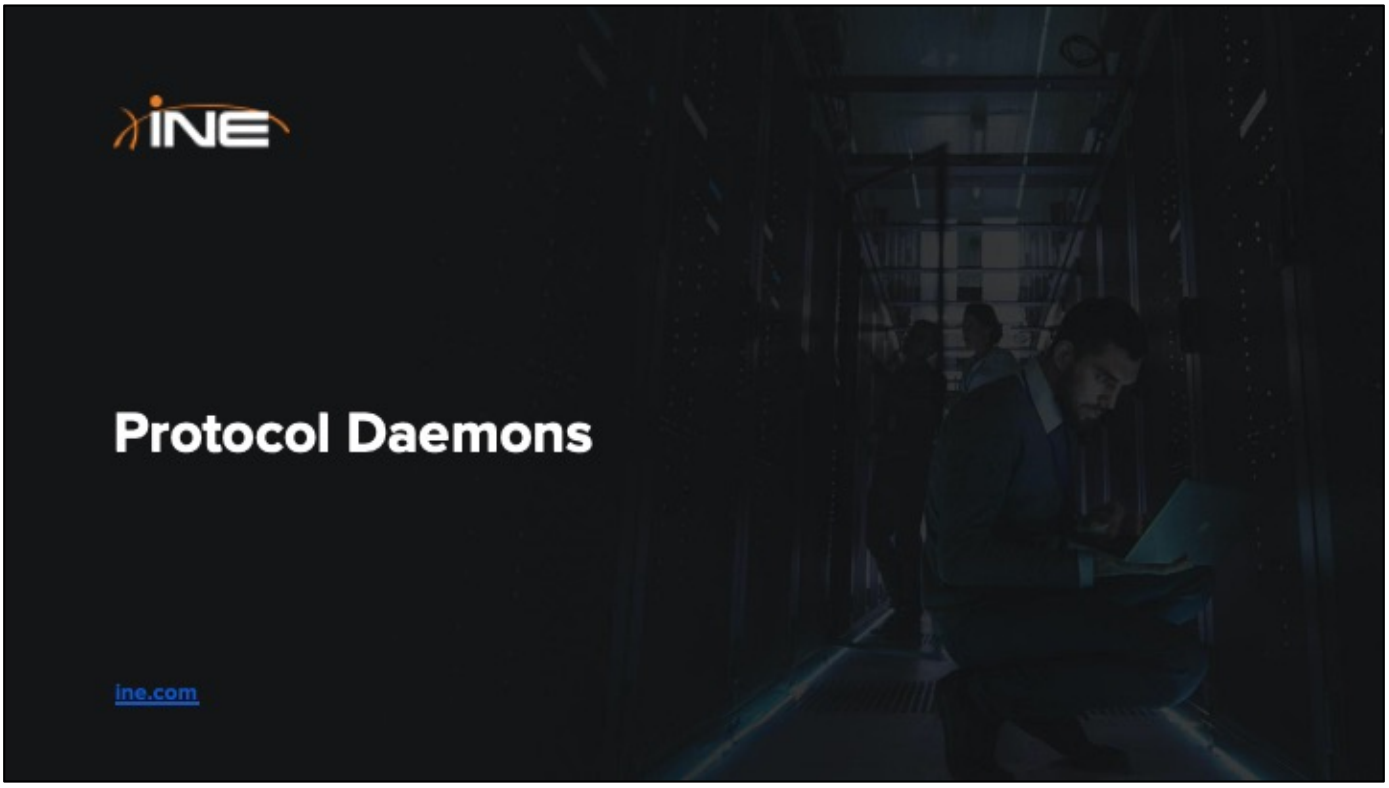


Image courtesy of: Day One: Beginner's Guide to Learning Junos

Linking RE To PFE

- The RE and PFE are connected by one-or-more high-speed internal links
  - These links go by different names depending on the platform
    - FXP1 (and FXP2)
    - EM1 (and EM2)
    - BCM0
- Default, internal rate-limiting (i.e. policers) automatically inspect traffic from PFE to RE to prevent DOS

Thanks for Watching!



Topic Overview

- Defining Daemons
- Junos OS Common Daemons
- Viewing Daemons
- Restarting Problematic Daemons

Definitions Of Daemons

- "A daemon is a ***service process that runs in the background*** and supervises the system or provides functionality to other processes." – man7.org
- "In multitasking computer operating systems, a daemon is a ***computer program that runs as a background process***, rather than being under the direct control of an interactive user." - wikipedia

Junos OS Daemons

- Because Junos OS runs on top of Free BSD (or Linux) it implements daemons to handle most tasks
  - A daemon is also called a “process”
- Daemons run in their own memory space
- Ensure fault isolation
- Some daemons run all the time; others are started as needed

Important Daemons

- If a daemon becomes problematic (i.e. stuck or is hogging the CPU) it can be stopped and restarted
- Important Daemons:

Daemon	Name	Description
rpd	Routing Protocol Daemon	Maintains routing protocol functionality covering protocols such as BGP, OSPF, and RIP
chassisd	Chassis Daemon	Controls the chassis and environment processes
Snmpd	SNMP Daemon	Allows SNMP servers to interact with the device and allows reporting from the SNMP agent to server
mgd	Management Daemon	Controls management connections to the network device via SSH, telnet, or HTTPS
dcd	Device Control Daemon	Junos needs to communicate with the hardware and interface cards and the dcd permits this
inetd	Internet Service Daemon	Provides internetwork connectivity to Junos
dhcpd	DHCP Daemon	Allows Junos to offer dynamic IP addresses and perform DHCP discoveries

- Routing protocol daemon (rpd)
- ---handles all routing protocol-related tasks
- ---ensure each routing protocol is runs as a separate process
- ---exchanges info with Junos Kernel to receive interface modifications, send route information and send interface changes

- Chassis daemon (chassisd)
- --- Supports all chassis, alarm and environmental processes (health monitoring and inventory)

- Management daemon (mgd)
- ---holds the entire User Interface (UI) together
- ---provides the Junos CLI
- ---provides an XML remote procedure call (RPC) interface which allows API calls through Junoscript and NETCONF
- Device control daemon (dcd)
- --- responsible for setting up interfaces based on the current configuration and available hardware
- --- allows you to configure non-existant hardware so the config will be ready-and-waiting when the hardware is installed later (like interface configuration for future PICs)
-

Analytics daemon (analyticsd) [Not shown in chart]

- --- provides detailed data and reporting on the network's behavior and performance such as:
- ---------queue statistics (queue latency and queue depth)
- ---------traffic statistics (measuring traffic per interface in PPS, packets dropped, port utilization, etc)

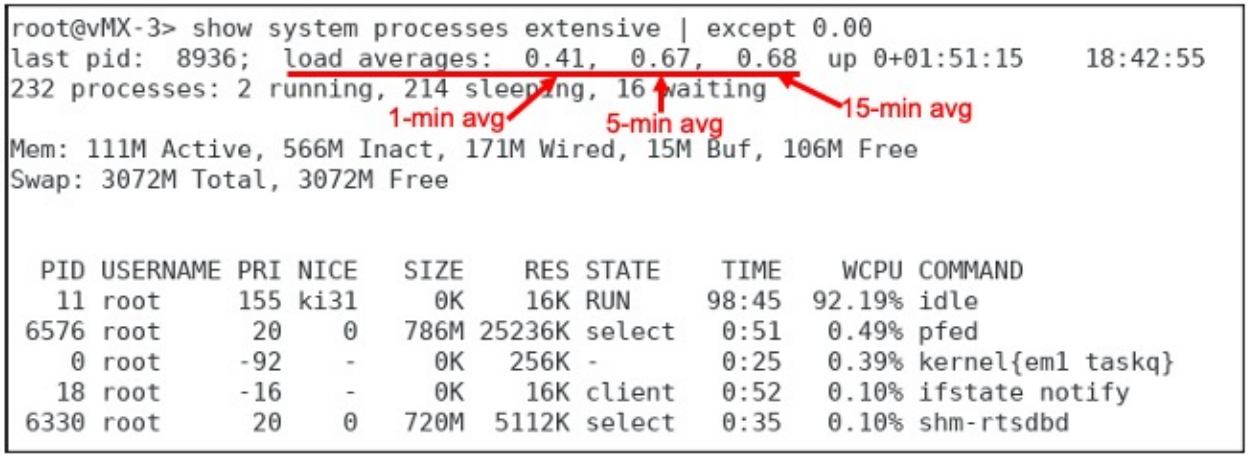
Viewing Daemons

- All processes can be viewed with:

```
root@vMX-3> show system processes
 PID TT  STAT     TIME COMMAND
   0  -  DLs   0:25.16 [kernel]
   1  -  ILs   0:00.02 /sbin/init --
   2  -  DL    0:31.27 [jfe_job_0_0]
   3  -  DL    0:00.00 [jfe_job_1_0]
   4  -  DL    0:00.00 [jfe_job_2_0]
   5  -  DL    0:00.00 [jfe_job_3_0]
   6  -  DL    0:00.24 [jfe_job_4_0]
   7  -  DL    0:00.00 [kpf0]
   8  -  DL    0:00.00 [kpf1]
   9  -  DL    0:00.00 [kpf2]
  10  -  DL    0:00.00 [audit]
```

Identifying Problematic Daemons

+ If “load averages” shows very high values (i.e. high CPU) then look for Daemons with high WCPU values:



- In this example, the CPU has been pretty busy. The CPU load over the past 5-minutes has been 67%!

- show system processes [summary]
- ----------"sleeping" = a configured process that is currently inactive
- ----------Zombie processes are processes which get stuck while performing a task; they do not release the CPU or memory used. They do not show up as a top process because they are performed at the root level.

Restarting Problematic Daemons

- Problematic daemons can be restarted
- The “restart” command reference a descriptive name of the daemon

```
root@ACME-HQ-SRX-01> restart ?
Possible completions:
  802.1x-protocol-daemon  Port based Network Access Control
  application-identification  Application-identification process
  application-security  Application security daemon
  audit-process        Audit process
  autoinstallation     Autoinstallation process
  chassis-control      Chassis control process
  class-of-service     Class-of-service process
!
output omitted for brevity
!
  remote-operations    Remote operations process
  routing              Routing protocol process
  sampling             Traffic sampling control process
  sdk-service          SDK Service Daemon
  secure-neighbor-discovery  Secure Neighbor Discovery Protocol process
  security-intelligence  IPF daemon
  security-log         Security Log Daemon
  service-deployment   Service Deployment Client
  services             Restart a service
  simple-mail-client-service  Simple Mail Transfer Protocol Client process
  snmp                 Simple Network Management Protocol process
```

Restarting Problematic Daemons

- Confirmation of daemon restart will be displayed by a syslog entry, "*terminated by signal number 9*"

```
root@ACME-HQ-SRX-01> restart routing ?
Possible completions:
  <[Enter]>            Execute this command
  gracefully           Gracefully restart the process
  immediately          Immediately restart (SIGKILL) the process
  soft                 Soft reset (SIGHUP) the process
  |                    Pipe through a command
```

```
Apr 12 18:13:46  ACME-HQ-SRX-01 init: routing (PID 2216) terminated by signal number 9!
Apr 12 18:13:46  ACME-HQ-SRX-01 init: routing (PID 2823) started
```

Gracefully – Attempts to restart the daemon gently without affecting other processes. (default)
Immediately – Kills and restarts the daemon without delay. --> **This is the one that should be used to restart stuck Daemons**
Softly – Rereads and reactivates the configuration without completely restarting the software processes.

Another way to verify daemon restart is to take note of what the PID was of the daemon before...and after...the restart (the number should have changed)

- Daemons/processes can also be started from Shell mode
- https://kb.juniper.net/InfoCenter/index?page=content&id=KB10956&actp=METADATA

Killing Daemon's From The Shell

- Daemons/Processes can also be killed from the FreeBSD/Linux Shell
- Should only be done by advanced users
- More info:

https://kb.juniper.net/InfoCenter/index?page=content&id=KB10956

Thanks for Watching!