JUNIPER NETWORKS | Engineering Simplicity

Junos® OS

Getting Started with Junos OS

Published
2022-09-09

JUNOS

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Junos® OS Getting Started with Junos OS
Copyright © 2022 Juniper Networks, Inc. All rights reserved.

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at https://support.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About This Guide | v

1 Understanding Junos OS

About the Getting Started Guide | 2

Operating System Infrastructure and Processes | 2

2 Access a Juniper Networks Device

Initial Configuration Overview for Juniper Networks Devices | 8

Console Port Overview | 9

How to Access a Juniper Networks Device the First Time | 9

3 Root Password

Root Password Overview | 12

Protect Network Security by Configuring the Root Password | 12

4 Device Hostname

Hostnames Overview | 16

Configure the Hostname of a Device in a Configuration Group | 16

5 DNS, Server Caching, and Device Identity

Understanding and Configuring DNS | 20

- DNS Overview | 20
- Configure a DNS Name Server for Resolving Hostnames into Addresses | 21

Example: Configure a Device's Unique Identity for the Network | 24

- Requirements | 25
- Overview | 25
- Configuration | 26
- Verification | 28

6 Management Ethernet and Loopback Interfaces

Management Ethernet Interfaces Overview | 31

Management Interface in a Non-Default Instance | 33

Why Use a Non-Default VRF Instance? | 33

Configure the mgmt_junos VRF Instance | 34

Before You Begin: Determine Static Routes | 35

Enable the mgmt_junos VRF Instance | 36

Configure Processes to Use mgmt_junos | 37

How to Disable the mgmt_junos VRF Instance | 41

Loopback Interface Overview | 41

Loopback Interface Configuration | 42

Configure the Loopback Interface | 43

Example: Configure Two Addresses on the Loopback Interface with Host Routes | 44

Example: Configure Two Addresses on the Loopback Interface with Subnetwork Routes | 45

Example: Configure an IPv4 and an IPv6 Address on the Loopback Interface with Subnetwork Routes | 46

7 Initial User Accounts

User Accounts Overview | 48

Configure User Accounts in a Configuration Group | 50

Enable Remote Access Services | 53

8 Backup Routers

Backup Routers Overview | 56

Configure a Backup Router | 56

Configure a Backup Router Running IPv4 for Routers | 57

Configure a Backup Router Running IPv6 for Routers | 59

Configure a Backup Router for SRX Series Devices | 61

About This Guide

Use this guide to configure common system management features on Juniper Networks devices using Junos OS.

1

CHAPTER

Understanding Junos OS

About the Getting Started Guide | 2

Operating System Infrastructure and Processes | 2

About the Getting Started Guide

Getting Started with Junos OS provides a high-level introduction to Junos OS and explains basic concepts and operational principles for working with Juniper Networks devices.

In this guide, we explain the basics of Junos OS, including:

- Understanding the network operating system software
- How to access Juniper Networks devices
- How to perform initial device configuration, including the configuration of the root password, hostname, Domain Name System (DNS), management and loopback interfaces, user accounts, and backup routers

For a more technical and detailed exploration of Junos OS, see the Overview for Junos OS. This overview further explains Junos OS features, including security, configuration, monitoring, and management of network devices.

To learn how to use the command-line interface (CLI) and understand even more advanced topics, see the CLI User Guide for Junos OS. This guide explains how to use configuration statements and manage configurations. It also explains how to use operational commands for monitoring Juniper Networks devices.

Operating System Infrastructure and Processes

IN THIS SECTION

- Routing Engine and Packet Forwarding Engine | 3
- Junos OS Processes | 3

Junos OS includes the processes that run the device, including IP routing, Ethernet switching, managing interfaces, and a variety of other functions.

Junos OS runs on the Routing Engine. The Routing Engine kernel coordinates communication among the software processes and provides a link to the Packet Forwarding Engine.

Using the CLI, you configure device features and set the properties of network interfaces. After activating a software configuration, use either the Junos Space or CLI user interface to monitor, manage operations, and diagnose protocol and network-connectivity problems.

Routing Engine and Packet Forwarding Engine

A Juniper Networks router or switch has two primary software processing components:

- Packet Forwarding Engine—Processes packets; applies filters, routing policies, and other features; and forwards packets to the next hop along the route to their final destination.
- Routing Engine—Provides three main functions:
  - Maintains the routing tables used by the network device and controls the routing protocols that run on the device.
  - Packet forwarding, which provides route lookup, filtering, and switching on incoming data packets, and then directs outbound packets to the appropriate interface for transmission to the network.
  - Provides control and monitoring functions for the device.

Junos OS Processes

The Junos OS software running on the device consists of multiple processes that are responsible for individual functions.

The separation of functions provides operational stability, because each process accesses its own protected memory space.

The following table describes the primary software processes.

Table 1: Junos OS Processes

Process	Name	Description
Chassis process	chassisd	Detects hardware on the system that is used to configure network interfaces. Monitors the physical status of hardware components and field-replaceable units (FRUs), detecting when environment sensors such as temperature sensors are triggered. Relays signals and interrupts—for example, when devices are taken offline, so that the system can close sessions and shut down gracefully.
DNS server process	named-service	Resolves hostnames into addresses.
Dynamic Host Configuration Protocol (DHCP) process	dhcp-service	Enables a DHCP server to allocate network IP addresses and deliver configuration settings to client hosts without user intervention.
Ethernet switching process	eswd	Handles Layer 2 switching functionality such as MAC address learning, spanning-tree protocols, and access port security. Manages Ethernet switching interfaces, VLANs, and VLAN interfaces. **NOTE**: This process is not applicable for EX2300 and EX3400 switches.
Firewall management process	firewalld	Manages the firewall configuration and helps accept or reject packets that are transiting an interface on a switch.
Forwarding process	pfem	Defines how routing protocols operate on the partition. The overall performance of the partition is largely determined by the effectiveness of the forwarding process.
Interface process	dcd	Configures and monitors network interfaces by defining physical characteristics such as link encapsulation, hold times, and keepalive timers.

Table 1: Junos OS Processes *(Continued)*

Process	Name	Description
Integrated Local Management Interface (ILMI) process	ilmi	Provides bidirectional exchange of management information between two ATM interfaces across a physical connection.
Link Management Protocol (LMP) process	link-management	Establishes and maintains LMP control channels.
Management process	mgd	Provides communication between the other processes and an interface to the configuration database. Populates the configuration database with configuration information and retrieves the information when queried by other processes to ensure that the system operates as configured. Interacts with the other processes when commands are issued through one of the user interfaces on the partition. If a process terminates or fails to start when called, the management process attempts to restart it a limited number of times to prevent thrashing and logs any failure information for further investigation.
Multicast snooping process	multicast-snooping	Makes Layer 2 devices, such as VLAN switches, aware of Layer 3 information, such as the media access control (MAC) addresses of members of a multicast group.
Routing protocol process	rpd	Defines how routing protocols such as RIP, OSPF, and BGP operate on the device, including selecting routes and maintaining forwarding tables.

Table 1: Junos OS Processes *(Continued)*

Process	Name	Description
Secure Neighbor Discovery (SEND) protocol process	send	Protects Neighbor Discovery Protocol (NDP) messages.
Simple Network Management Protocol (SNMP) process	snmpd	Enables the monitoring of network devices from a central location and provides the switch's SNMP primary agent.
Tunnel OAM process	tunnel-oamd	Enables the Operation, Administration, and Maintenance of Layer 2 tunneled networks. Layer 2 protocol tunneling (L2PT) allows service providers to send Layer 2 protocol data units (PDUs) across the provider's cloud and deliver them to devices that are not part of the local broadcast domain.
Virtual Router Redundancy Protocol (VRRP) process	vrrpd	Enables hosts on a LAN to make use of redundant routing platforms on that LAN without requiring more than the static configuration of a single default route on the hosts.

RELATED DOCUMENTATION

- Junos OS Administration Library for Routing Devices

2

CHAPTER

Access a Juniper Networks Device

Initial Configuration Overview for Juniper Networks Devices | 8

Console Port Overview | 9

How to Access a Juniper Networks Device the First Time | 9

Initial Configuration Overview for Juniper Networks Devices

After you install and power on the Juniper Networks device, you are ready to begin initial configuration. All devices have a version of Junos OS preinstalled. The procedures in this guide show you how to connect the device to the network but do not enable the device to forward traffic. For complete information about enabling the device to forward traffic, including examples, see the software configuration guides.

For information about how to upgrade or reinstall software, see the Junos OS Installation and Upgrade Guide.

NOTE: For an overview of Junos OSand for details regarding configuration statements and CLI commands, see the Overview for Junos OS and the CLI User Guide for Junos OS.

Console access to the device is enabled by default. Use a console port to connect to the device initially.

For an overview of Junos OS and for details regarding configuration statements and CLI commands, see Introducing Junos OS Evolved and the CLI User Guide for Junos OS Evolved.

Gather the following information before configuring the device:

- Name the device will use on the network
- Domain name the device will use
- IP address and prefix-length information for the Ethernet interface
- IP address of a default device
- IP address of a DNS server
- Password for the root user

The most common method of configuring the device is through the use of CLI commands.

Console Port Overview

The console port allows access to a device running Junos OS, regardless of the state of the device, unless it is completely powered off. By connecting to the console port, you can access the root level of the device without using the network to which the device might not be connected.

A console port connection provides persistent direct access to a device that can often be accessed even when the primary network has failed.

We recommend that you perform all Junos OS and software package upgrades using the console port connection because this connection remains up for the duration of the upgrade, enabling you to monitor status and progress. Other network-based connections such as SSH or telnet are often interrupted during software upgrades, which can cause status or error messages to be missed.

NOTE: See the hardware guide for your particular Juniper Networks device for instructions on how to connect to the console port.

How to Access a Juniper Networks Device the First Time

NOTE: Before you access any new Juniper Networks device, be sure to follow the quick start and initial setup instructions that came with the device.

When you power on a device running Junos OS, Junos OS automatically boots and starts.

To configure the device initially, you must connect a terminal or laptop computer to the device through the console port. Console port access to the device is **enabled** by default. Remote management access to the device and all management access protocols—such as Telnet, FTP, and SSH—are **disabled** by default.

To access a network device for the first time:

1. Connect a laptop or a desktop PC to the console port on the front panel of the device.
2. Power on the device and wait for it to boot.

The software boots automatically. When the boot process is complete, you'll see the `login:` prompt on the console.

3. Log in as the user `root`.

   Initially, you won't need a password for the root user account. The device prompt `root@%` indicates that you are the root user.

4. Type **cli** to start the Junos OS CLI.

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

5. Type **configure** to access CLI configuration mode.

```
root@> configure
[edit]
root@#
```

3

CHAPTER

Root Password

Root Password Overview | 12

Protect Network Security by Configuring the Root Password | 12

Root Password Overview

The root user has complete privileges to operate and to configure the Juniper Networks device, perform upgrades, and manage files in the file system. Initially, the root password is not defined on the device. To ensure basic security, you must define the root password during initial configuration. If your device does not have a defined root password, you cannot commit configuration settings on the device.

The root password must meet the following conditions:

- Be at least six characters long. You can include most character classes (alphabetic, numeric, and special characters) in a password, except control characters.
- Contain at least one change of case or of character class.

Protect Network Security by Configuring the Root Password

Configure the root password on your Juniper Networks device to help prevent unauthorized users from making changes to your network. The root user (also referred to as superuser) has unrestricted access and full permissions within the system, so it is crucial that you protect this account by setting a strong password when setting up a new device.

After you initially power on a new device, you log in as the user `root` with no password. The software requires you to configure the root password before it accepts a commit operation.

To set the root password, you have three options:

- Enter a plain-text password that the software encrypts.
- Enter a password that is already encrypted.
- Enter a Secure Shell (SSH) public key string.

Among these options, using a pre-encrypted password or an SSH public key string is the most secure. If you use one of these methods, then the plain-text version of your password will never be transferred over the Internet, protecting it from being intercepted by a man-in-the-middle attack.

BEST PRACTICE: Optionally, instead of configuring the root password at the `[edit system]` hierarchy level, you can use a configuration group to strengthen security.

To set the root password:

1. Use one of these methods to configure the root password:
   - To enter a plain-text password that the system encrypts for you:

     ```
     [edit groups global system]
     root@# set root-authentication plain-text-password
     New Password: type password here
     Retype new password: retype password here
     ```

     As you enter a plain-text password into the CLI, the device software hides it from view and encrypts it immediately. You don't have to configure the software to encrypt the password. In the resulting configuration, the encrypted password is marked as `## SECRET-DATA` so that it cannot be seen.

   - To enter a password that is already encrypted:

     CAUTION: Do not use the `encrypted-password` option unless the password is *already* encrypted and you are entering that encrypted password.

     If you accidentally configure the `encrypted-password` option with a plain-text password or with blank quotation marks (" "), you will not be able to log in to the device as the root user. You will then need to complete the root password recovery process.

     ```
     [edit groups global system]
     root@# set root-authentication encrypted-password password
     ```

   - To enter an SSH public key string:

     ```
     [edit groups global system]
     root@# set root-authentication (ssh-ecdsa | ssh-rsa key)
     ```

2. If you used a configuration group, replace the *group-name* variable with the configuration group's name.

```
[edit]
root@# set apply-groups group-name
```

3. Commit the changes.

```
root@# commit
```

Release History Table

Release	Description
18.3R1	Starting in Junos OS Release 18.3R1, the ssh-dss and ssh-dsa hostkey algorithms are deprecated—rather than immediately removed—to provide backward compatibility and a chance to bring your configuration into compliance with the new configuration.

4

CHAPTER

Device Hostname

Hostnames Overview | 16

Configure the Hostname of a Device in a Configuration Group | 16

Hostnames Overview

Almost all devices in your network have a hostname.

The hostname is the name that identifies the device on the network. A hostname is easier to remember than an IP address.

When you first power on a Juniper Networks router, switch, or security device, the default hostname is `Amnesiac`. The `Amnesiac` prompt indicates that the device is booting from a factory-fresh Junos OS software load. By definition, such a device doesn't have a hostname configured.

As an administrator, you follow conventions for naming devices. One such convention is to name the device based on its location—for example: germany-berlin-R1. Make the hostname unique within your local network so that users can connect to the device by using the hostname. You don't need to make the local hostname globally unique.

A device's hostname usually has a corresponding entry in the Domain Name System (DNS) so that you (the administrator) can connect to the device using the hostname. The fully qualified domain name (FQDN), which is used in the DNS, includes the hostname and the entire domain name. A period (or a dot) separates the hostname and the domain name labels, so the FQDN format is *hostname.domain*. For example, if the hostname is `germany-berlin-R1` and the domain name is `example`, the FQDN is `germany-berlin-R1.example`. If the `example.net` domain is registered and can be reached as `example.net` on the Internet, the FQDN for the device is `germany-berlin-R1.example.net`. The FQDN is globally unique.

In Junos OS, the hostname can contain any combination of alphabetic characters, numbers, dashes, and underscores. No other special characters are allowed.

The software allows hostnames to contain up to 255 characters. Keep in mind that the total length of the hostname as an FQDN cannot exceed 255 characters (including the delimiting dots), with each domain name label having a maximum length of 63 characters. As a best practice, use short and meaningful hostnames, as long hostnames are difficult to type and to remember.

Configure the Hostname of a Device in a Configuration Group

The hostname of a device running Junos OS device is its identification. A network device must have its identity established to be accessible on the network. That is perhaps the most important reason to have a hostname, but a hostname also has other purposes.

Junos OS uses the configured hostname as part of the command prompt and to prepend log files and other accounting information. The hostname is also used anywhere else where knowing the device identity is important. For these reasons, we recommend that you provide hostnames that are descriptive and memorable.

You can configure the hostname at the `[edit system]` hierarchy level. Optionally, instead of configuring the hostname at the `[edit system]` hierarchy level, you can use a configuration group, as shown in this procedure. This is a recommended best practice for configuring the hostname, especially if the device has dual Routing Engines. This procedure uses groups called `re0` and `re1` as an example.

NOTE: If you configure hostnames that are longer than the CLI screen width, regardless of the terminal screen width setting, the commit operation occurs successfully. Even if the terminal screen width is less than the hostname length, the commit is successful.

To set the hostname by using a configuration group:

1. Include the `host-name` statement at the `[edit groups` *`group-name`* `system]` hierarchy level.

   The *hostname* value must be less than 256 characters.

   ```
   [edit groups group-name system]
   host-name hostname;
   ```

   For example:

   ```
   [edit groups re0 system]
   root@# set host-name san-jose-router0
   ```

   ```
   [edit groups re1 system]
   root@# set host-name san-jose-router1
   ```

2. If you used one or more configuration groups, apply the configuration groups, substituting the appropriate group names.

   For example:

   ```
   [edit]
   user@host# set apply-groups [re0 re1]
   ```

3. Commit the changes.

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

The hostname appears in the device CLI prompt.

```
san-jose-router0#
```

5
CHAPTER

DNS, Server Caching, and Device Identity

Understanding and Configuring DNS | 20

Example: Configure a Device's Unique Identity for the Network | 24

Understanding and Configuring DNS

IN THIS SECTION

- DNS Overview | 20
- Configure a DNS Name Server for Resolving Hostnames into Addresses | 21

DNS Overview

IN THIS SECTION

- DNS Components | 20
- DNS Server Caching | 21

A Domain Name System (DNS) is a distributed hierarchical system that converts hostnames to IP addresses. The DNS is divided into sections called zones. Each zone has name servers that respond to the queries belonging to their zones.

DNS Components

DNS includes three main components:

- DNS resolver: Resides on the client side of the DNS. When a user sends a hostname request, the resolver sends a DNS query request to the name servers to request the hostname's IP address.
- Name servers: Processes the DNS query requests received from the DNS resolver and returns the IP address to the resolver.
- Resource records: Data elements that define the basic structure and content of the DNS.

DNS Server Caching

DNS name servers provide a hostname's IP address to users. The TTL field in the resource record defines the period for which DNS query results are cached. When the TTL value expires, the name server sends a fresh DNS query and updates the cache.

Configure a DNS Name Server for Resolving Hostnames into Addresses

You use Domain Name System (DNS) name servers to resolve hostnames to IP addresses.

Before you begin, configure your name servers with the hostname and an IP address for your Juniper Networks device. It does not matter which IP address you assign as the address of your device in the name server, as long it is an address that reaches your device. Normally, you would use the management interface IP address, but you can choose the loopback interface IP address or a network interface IP address. You can even configure multiple addresses on the name server.

For redundancy, as a best practice, configure access to multiple name servers. You can configure a maximum of three name servers. The approach is similar to the way Web browsers resolve the names of a website to its network address.

You can use Junos OS to configure one or more domain names. The software uses these domain names to resolve hostnames that are not fully qualified (that is, hostnames for which the domain names are missing). Being able to configure domain names is convenient because you can use a hostname in configuring and operating the software without the need to reference the full domain name. After adding name server addresses and domain names to your configuration, you can use DNS resolvable hostnames in your configurations and commands instead of IP addresses.

Optionally, instead of configuring the name server at the `[edit system]` hierarchy level, you can use a configuration group, as shown in this procedure. This is a recommended best practice for configuring the name server.

You can route traffic between a management routing instance and a DNS name server. After you configure a routing instance at the `[edit system name-server` *`server-ip-address`*`]` hierarchy level, the name server becomes reachable through this routing instance.

NOTE: This management routing instance option is not supported for SRX Series devices.

To enable a management routing instance for DNS, use the following configuration:

```
user@host# set system management-instance
user@host# set routing-instances mgmt_junos description description
user@host# set system name-server server-ip-address routing-instance mgmt_junos
```

If you've configured the name server using a configuration group, use the `[edit groups group-name system name-server]` hierarchy level, which is a recommended best practice for configuring the name server.

To configure the device to resolve hostnames into addresses:

1. Reference the IP addresses of your name servers.

   ```
   [edit groups group-name system]
   name-server {
       address;
   }
   ```

   The following example shows how to reference two name servers:

   ```
   [edit groups global system]
   user@host# set name-server 192.168.1.253
   user@host# set name-server 192.168.1.254
   user@host# show
   name server {
       192.168.1.253;
       192.168.1.254;
   }
   ```

2. (Optional) Configure the routing instance for DNS.

   The following example shows how to configure the routing instance for one of the name servers:

   ```
   [edit groups global system]
   user@host# set name-server 192.168.1.253 routing-instance mgmt_junos
   ```

   Remember to also configure the following:

   - `management-instance` statement at the `[edit system]` hierarchy level
   - `routing-instance` statement at the `[edit routing-instances]` hierarchy level

3. (Optional) Configure the name of the domain in which the device itself is located.

This is a good practice. The software then uses this configured domain name as the default domain name to append to hostnames that are not fully qualified.

```
[edit system]
domain-name domain-name;
```

The following example shows how to configure the domain name:

```
[edit groups global system]
user@host# set domain-name company.net
user@host# show
domain-name company.net;
```

4. (Optional) Configure a list of domains to be searched.

   If your device can reach several different domains, you can configure a list of domains to be searched. Junos OS then uses this list to set an order in which it appends domain names when searching for the IP address of a host.

   ```
   [edit groups global system]
   domain-search [ domain-list ];
   ```

   The domain list can contain up to six domain names, with a total of up to 256 characters.

   The following example shows how to configure three domains to be searched. This example configures the software to search the company.net domain, next the domainone.net domain, and finally the domainonealternate.com domain when attempting to resolve unqualified hosts.

   ```
   [edit groups global system]
   domain-search [ company.net domainone.net domainonealternate.com ]
   ```

5. If you used a configuration group, apply the configuration group, replacing `global` with the appropriate group name.

   ```
   [edit]
   user@host# set apply-groups global
   ```

6. Commit the configuration.

```
user@host# commit
```

7. Verify the configuration.

   If you've configured your name server with the hostname and an IP address for your device, you can issue the following commands to confirm that DNS is working and reachable. You can either use the configured hostname to confirm resolution to the IP address or use the IP address of your device to confirm resolution to the configured hostname.

```
user@host> show host host-name
user@host> show host host-ip-address
```

   For example:

```
user@host> show host device.example.net
device.example.net
device.example.net has address 192.168.187.1
```

```
user@host> show host 192.168.187.1
10.187.168.192.in-addr.arpa domain name pointer device.example.net.
```

Example: Configure a Device's Unique Identity for the Network

IN THIS SECTION

- Requirements | 25
- Overview | 25
- Configuration | 26
- Verification | 28

To use a device in a network, you must configure the device’s identity. Configuring the device's identity makes the device accessible on the network and allows other users to log in to it. You can refer to any Internet-connected device in either of two ways:

- By its IP address
- By its hostname

Once you have a hostname, you can:

- Find the IP address
- Use the Domain Name System (DNS) to resolve an IP address from a hostname
- Manually map the hostname to a static IP address

Using DNS is an easy and scalable way to resolve IP addresses from hostnames. However, you might not have a DNS entry for the device. You might not want the computer to contact the DNS server to resolve a particular IP address. Perhaps you use this particular IP address frequently. Maybe you use it only for testing or development purposes and do not want to give it a DNS entry.

To configure a device's unique identity, you might need to include some or all of the following details: The hostname of the device, its IP address, the domain name, and IP addresses for two or three domain name servers.

Requirements

You don't need to do any special configuration beyond device initialization.

Overview

In this example, the hostname is the device’s name. Most people find it easier to remember a hostname than an IP address. The software uses the configured hostname as part of the command prompt, to prepend log files and other accounting information, and in other places where knowing the device identity is useful. You can also use the hostname to telnet to a device.

You append a domain name to hostnames that are not fully qualified. The domain name is the name of a network associated with an organization. For sites in the United States, domain names typically take the form of *org-name.org-type*—for example, “Juniper.net.”

If your hostname and IP address do not have a DNS entry in a name server, configure a static mapping. See Step 4 in the following procedure for an example.

This example uses the values given in the following table to configure each of these variables. You need to substitute data specific to your device and network for these values.

Table 2: Values to Use in Example

Name of Variable	Value Used in Example	Value You Substitute
`domain-name` *`domain-name`*	`domain-name device.example.net`	*Provide your value.*
`host-name` *`host-name`*	`host-name example-re0`	*Provide your value.*
`inet` *`ip-address`*	`inet 172.22.147.39`	*Provide your value.*
`name-server` *`ip-address`*	`name-server 172.24.16.115` `name-server 192.0.2.10`	*Provide your value.*

Configuration

IN THIS SECTION

- CLI Quick Configuration | 26
- Configure the Device's Identity | 27
- Results | 28

CLI Quick Configuration

To quickly configure a device using this example:

1. Copy the following commands and paste the commands in a text file.
2. Remove any line breaks.
3. Change the values listed here to match your network configuration.

4. Copy and paste the commands into the CLI at the `[edit]` hierarchy level.
5. Enter `commit` in configuration mode.

```
set system domain-name device.example.net
set system host-name example-re0
set system name-server 172.24.16.115
set system name-server 192.0.2.10
set system static-host-mapping example-re0 inet 172.22.147.39
```

Configure the Device's Identity

Step-by-Step Procedure

To configure the identity settings of a device:

1. Configure the domain name of your network.

```
[edit]
user@host# set system domain-name device.example.net
```

2. Configure the hostname of the device.

```
[edit]
user@host# set system host-name example-re0
```

3. Configure from one to three name servers.

```
[edit]
user@host# set system name-server 172.24.16.115
user@host# set system name-server 192.0.2.10
```

4. Map the device hostname to its IP address.

```
[edit]
user@host# set system static-host-mapping example-re0 inet 172.22.147.39
```

Results

To check the configuration, use the configuration mode `show system` command.

```
[edit]
user@host# show system
domain-name device.example.net;
host-name example-re0;
name-server {
    172.24.16.115;
    192.0.2.10;
}
static-host-mapping {
    example-re0 {
        inet 172.22.147.39;
    }
}
```

When you have the correct configuration, enter `commit`.

Verification

IN THIS SECTION

- Verify the Hostname and the IP Address of the Device | 28

Verify the Hostname and the IP Address of the Device

Purpose

Verify that the hostname and the IP address of a device are as expected.

Action

Issue the `show host` *host-name* operational command.

```
user@example-re0> show host newton
newton.device.example.net is an alias for example-re0.device.example.net.
example-re0.device.example.net has address 172.22.147.39
```

CHAPTER 6

Management Ethernet and Loopback Interfaces

Management Ethernet Interfaces Overview | 31

Management Interface in a Non-Default Instance | 33

Loopback Interface Overview | 41

Loopback Interface Configuration | 42

Management Ethernet Interfaces Overview

Management interfaces are the primary interfaces for accessing the device remotely. Typically, a management interface is not connected to the in-band network but is connected instead to the device's internal network. You (the system administrator) can use the management interface to access the device over the network using utilities such as `ssh` and `telnet`. You can configure the device from anywhere, regardless of its physical location. SNMP can use the management interface to gather statistics from the device.

Authorized users and management systems use a management interface to connect to the device over the network. Some Juniper Networks devices have a dedicated management port on the front panel. For other types of platforms, you can configure a management interface on one of the network interfaces. You can dedicate this interface to management or share it with other traffic.

You must configure the management interface before users can access it. To set up the management interface, you need information such as its IP address, prefix, and next hop. We recommend configuring your devices so that traffic is not routed between the management interface and the other ports. In many devices running Junos OS, it is not possible to route traffic between the management interface and the other ports. Therefore, you should select an IP address in a separate (logical) network with a separate prefix (netmask).

The name of the management interface depends on the platform. For devices running Junos OS, the management Ethernet interface is usually named fxp0, em0, or me0. For Junos OS Evolved, use re0:mgmt-* for Routing Engine 0 and re1:mgmt-* for Routing Engine 1 management interfaces, where the * is the index of the management interface. If there is only one management interface, the index is 0.

Table 3 on page 31 summarizes the management interfaces typically used on Junos and Junos Evolved platforms. It's always a good idea to refer to the specific documentation for your platform to confirm details about its management interface.

Refer to the Product Documentation page for details on your platform.

Alternatively, refer to the Day One + quick start guide for your platform at: Day One + Guides.

Table 3: Typical Management Interfaces on Junos and Junos Evolved Platforms

Platform	Interface Name	Description
MX Series routers	`fxp0`	The `fxp0` interface is typically an RJ-45 port on the Routing Engine.

Table 3: Typical Management Interfaces on Junos and Junos Evolved Platforms *(Continued)*

Platform	Interface Name	Description
EX Series switches	`me0`, `vme`	The `me0` interface is typically an RJ-45 port on the Routing Engine. The `vme` interface is used when the device is part of a Virtual Chassis (VC), and is accessed via the `me0` port. For consistency you can configure and use the `vme` interface on a stand alone switch.
QFX Series switches	`em0`, `vme`	The `em0` interface is typically an RJ-45 port on the Routing Engine. The `vme` interface is used when the device is part of a Virtual Chassis (VC), and is accessed via the `em0` port. For consistency you can configure and use the `vme` interface on a stand alone switch.
SRX Security Gateways	`fxp0`, `ge-0/0/0`	The `fxp0` interface is typically an RJ-45 port on the Routing Engine. On some SRX platforms the ge-0/0/0 interface is used as the management interface.

Table 3: Typical Management Interfaces on Junos and Junos Evolved Platforms *(Continued)*

Platform	Interface Name	Description
Platforms running Junos Evolved, for example, PTX10001-36MR, PTX10003, PTX10004, PTX10008, QFX5130, QFX5220, etc.	`re0:mgmt-*` and `re1:mgmt-*`	Junos Evolved platforms typically support two management interfaces per Routing Engine. The two Routing Engines are identified as `re0` and `re1`. The RJ-45 copper port on each Routing Engine is indexed as 0, while the SFP fiber port is indexed as 1. For example, the RJ-45 management Ethernet port on Routing Engine 0 is typically named `re0:mgmt-0`.

Management Interface in a Non-Default Instance

IN THIS SECTION

- Why Use a Non-Default VRF Instance? | **33**
- Configure the mgmt_junos VRF Instance | **34**
- Configure Processes to Use mgmt_junos | **37**
- How to Disable the mgmt_junos VRF Instance | **41**

Why Use a Non-Default VRF Instance?

By default, the management Ethernet interface (usually named fxp0 or em0 for Junos OS, or re0:mgmt-* or re1:mgmt-* for Junos OS Evolved) provides the out-of-band management network for the device.

Out-of-band management traffic is not clearly separated from in-band protocol control traffic. Instead, all traffic passes through the default routing instance and shares the default inet.0 routing table. This system of traffic handling gives rise to concerns over security, performance, and troubleshooting.

You (the network administrator) can confine the management interface to a non-default virtual routing and forwarding (VRF) instances. After you configure the non-default management VRF instance, management traffic no longer has to share a routing table with other control traffic or protocol traffic. This configuration improves security and makes it easier to use the management interface to troubleshoot.

NOTE:

- For Junos OS, the non-default management VRF instance supports only the em0, fxp0, and vme interfaces. The non-default management VRF instance doesn't support other management interfaces such as em1.
- SRX chassis clusters do not support the non-default management VRF instance.

Configure the mgmt_junos VRF Instance

IN THIS SECTION

- Before You Begin: Determine Static Routes | 35
- Enable the mgmt_junos VRF Instance | 36

The name of the dedicated management VRF instance is reserved and hardcoded as `mgmt_junos`; you cannot configure any other routing instance by the name `mgmt_junos`. Because some applications assume that the management interface is always present in the default inet.0 routing table, the dedicated management VRF instance is not instantiated by default.

You must add any static routes that have a next hop over the management interface to the `mgmt_junos` VRF instance. If needed, you must also configure the appropriate processes or applications to use `mgmt_junos`. All of these changes must be done in a single commit. Otherwise, the existing sessions might be lost and need to be renegotiated.

Once you deploy the `mgmt_junos` VRF instance, management traffic no longer shares a routing table (that is, the default routing table) with other control traffic or protocol traffic in the system. Traffic in the

mgmt_junos VRF instance uses private IPv4 and IPv6 routing tables. After you configure mgmt_junos, you cannot configure dynamic protocols on the management interface.

Before You Begin: Determine Static Routes

Some static routes have a next hop through the management interface. As part of configuring the mgmt_junos VRF instance, you must add all these static routes to mgmt_junos so they can reach the management interface. Each setup is different. First, you need to identify the static routes that have a next hop through the management interface.

1. Use the show interfaces *interface-name* terse command to find the IP address of the default management interface. The default management interface is fxp0 or em0 for Junos OS, or re0:mgmt-0 or re1:mgmt-0 for Junos OS Evolved.

```
user@host> show interfaces fxp0 terse

Interface               Admin Link Proto    Local                 Remote
fxp0                    up    up
fxp0.0                  up    up   inet     10.102.183.152/20
```

2. Use the show route forwarding-table command to look at the forwarding table for next-hop information for static routes. Static routes show up as type user. The next hop for any static route that is affected has an IP address that falls under the subnet of the IP address configured for the management interface.

```
user@host> show route forwarding-table

Routing table: default.inet
Internet:
Enabled protocols: Bridging,
Destination        Type RtRef Next hop           Type Index    NhRef Netif
default            perm     0                    rjct       36     1
0.0.0.0/32         perm     0                    dscd       34     1
10.0.0.0/8         user     0 0:0:5e:0:1:d0      ucst      341     6 fxp0.0
10.0.1.0/24        intf     0                    rslv      584     1 ge-0/0/0.0
10.0.1.0/32        dest     0 10.0.1.0           recv      582     1 ge-0/0/0.0
10.0.1.1/32        intf     0 10.0.1.1           locl      583     2
10.0.1.1/32        dest     0 10.0.1.1           locl      583     2
10.0.1.255/32      dest     0 10.0.1.255         bcst      581     1 ge-0/0/0.0
10.102.176.0/20    intf     0                    rslv      340     1 fxp0.0
10.102.176.0/32    dest     0 10.102.176.0       recv      338     1 fxp0.0
10.102.176.3/32    dest     1 0:50:56:9f:1b:2e   ucst      350     2 fxp0.0
```

```
10.102.183.152/32  intf     0 10.102.183.152     locl      339     2
10.102.183.152/32  dest     0 10.102.183.152     locl      339     2
10.102.191.253/32  dest     0 10:e:7e:b1:b0:80   ucst      348     1 fxp0.0
10.102.191.254/32  dest     0 0:0:5e:0:1:d0      ucst      341     6 fxp0.0
10.102.191.255/32  dest     0 10.102.191.255     bcst      337     1 fxp0.0
172.16.0.0/12      user     0 10.102.191.254     ucst      341     6 fxp0.0
192.168.0.0/16     user     0 10.102.191.254     ucst      341     6 fxp0.0
224.0.0.0/4        perm     0                    mdsc       35     1
224.0.0.1/32       perm     0 224.0.0.1          mcst       31     1
255.255.255.255/32 perm     0                    bcst       32     1
```

3. Another way to find your static routes is to use the `show route protocol static` command.

```
user@host> show route protocol static

inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.0/8          *[Static/5] 2d 21:48:36
                     > to 10.102.191.254 via fxp0.0
172.16.0.0/12       *[Static/5] 2d 21:48:36
                     > to 10.102.191.254 via fxp0.0
192.168.0.0/16      *[Static/5] 2d 21:48:36
                     > to 10.102.191.254 via fxp0.0
```

Enable the mgmt_junos VRF Instance

NOTE: We recommend using the device console port for these operations. If you use SSH or Telnet, the connection to the device will be dropped when you commit the configuration, and you will have to reestablish it. If you do use SSH or Telnet, use `commit confirm`.

To enable the dedicated management VRF instance:

1. Configure the `mgmt_junos` VRF instance.

```
[edit]
user@host# set routing-instances mgmt_junos description description
```

2. Configure the `management-instance` statement.

```
[edit]
user@host# set system management-instance
```

3. Add the appropriate static routes to the `mgmt_junos` VRF instance.

   For how to determine the static routes to change, see "Before You Begin: Determine Static Routes" on page 35.

```
[edit routing-instances mgmt_junos routing-option static route]
user@host# set 10.0.0.0/8 next-hop 10.102.191.254
user@host# set 172.16.0.0/12 next-hop 10.102.191.254
user@host# set 192.168.0.0/16 next-hop 10.102.191.254
```

   If you are using configuration groups, you can set these changes as part of a group:

```
[edit groups global routing-instances mgmt_junos routing-options static route]
user@host# set 10.0.0.0/8 next-hop 10.102.191.254
user@host# set 172.16.0.0/12 next-hop 10.102.191.254
user@host# set 192.168.0.0/16 next-hop 10.102.191.254
```

4. Commit the configuration.

   If you are using SSH or Telnet, use `commit confirm`. Expect to lose, and then have to reestablish, the SSH or Telnet session.

Configure Processes to Use mgmt_junos

Many processes communicate through the management interface. A process must support a management VRF instance to be able to use `mgmt_junos`. Not all of these processes use `mgmt_junos` by default. You must configure these processes to use `mgmt_junos`.

The following processes require this additional configuration:

Table 4: Processes You Can Configure to Use the Management VRF Instance

Process	First Release to Support Managment VRF	For More Information
Automation scripts	Junos OS Release 18.1R1	*Using an Alternate Source Location for a Script* *Configuring and Using a Master Source Location for a Script*
BGP Monitoring Protocol (BMP)	Junos OS Release 18.3R1	*Configuring BGP Monitoring Protocol to Run Over a Different Routing Instance*
NTP	Junos OS Release 18.1R1	*ntp*
Outbound SSH	Junos OS Release 19.3R1	*Configure Outbound SSH Service*
RADIUS	Junos OS Release 18.1R1	*Configuring RADIUS Server Authentication* *Configuring RADIUS System Accounting*
REST API	Junos OS Release 20.3R1	*rest*
System Logging	Junos OS Release 18.1R1	*syslog (System)*
	Junos OS Release 18.4R1	*routing-instance (Syslog)*
TACACS+	Junos OS Release 17.4R1	*Configuring TACACS+ Authentication*
	Junos OS Release 18.2R1	*Configuring TACACS+ System Accounting*

Configuring these processes to use the `mgmt_junos` VRF instance is optional. If you skip this step, these processes continue to send packets using the default routing instance only.

1. To update automation scripts from a source using `mgmt_junos`, configure the following:

a. Commit, op, or SNMP scripts:

```
[edit]
user@host# set system scripts (commit | op | snmp) file filename routing-instance
mgmt_junos
```

b. Event scripts:

```
[edit]
user@host# set event-options event-script file filename routing-instance mgmt_junos
```

c. Juniper Extension Toolkit (JET) scripts:

```
[edit]
user@host# set system extensions extension-service application file filename routing-
instance mgmt_junos
```

2. BMP:

a. BMP in passive connection mode:

```
[edit]
user@host# set routing-options bmp station station-name routing-instance mgmt_junos
user@host# set routing-options bmp station station-name connection-mode passive
user@host# set routing-options bmp station station-name local-address ip-address
user@host# set routing-options bmp station station-name local-port port-number
user@host# set routing-options bmp station station-name station-address ip-address
```

b. BMP in active connection mode:

```
[edit]
user@host# set routing-options bmp station station-name routing-instance mgmt_junos
user@host# set routing-options bmp station station-name connection-mode active
user@host# set routing-options bmp station station-name station-address ip-address
user@host# set routing-options bmp station station-name station-port port-number
```

3. NTP service:

```
[edit]
user@host# set system ntp server server-address routing-instance mgmt_junos
```

   You must also configure at least one IP address on a physical or logical interface within the default routing instance. Ensure that this interface is up so that the NTP service can work with the `mgmt_junos` VRF instance.

4. RADIUS:

```
[edit]
user@host# set system radius-server server-address routing-instance mgmt_junos
user@host# set system accounting destination radius server server-address routing-instance
mgmt_junos
```

5. TACACS+:

```
[edit]
user@host# set system tacplus-server server-address routing-instance mgmt_junos
user@host# set system accounting destination tacplus server server-address routing-instance
mgmt_junos
```

6. The REST API:

```
[edit]
user@host# set system services rest routing-instance mgmt_junos
```

7. System logging:

```
[edit]
user@host# set system syslog host ip-address routing-instance mgmt_junos
```

8. Outbound SSH:

```
[edit]
user@host# set system services outbound-ssh routing-instance mgmt_junos
```

How to Disable the mgmt_junos VRF Instance

When you disable the mgmt_junos VRF instance, you must also remove the other configuration changes you made.

1. Remove the management-instance statement to disable the dedicated management VRF instance.

```
[edit]
user@host# delete system management-instance
```

2. (Optional) Remove the static routes from the mgmt_junos VRF instance.

```
[edit routing-instances mgmt_junos routing-option static route]
user@host# delete 10.0.0.0/8 next-hop 10.102.191.254
user@host# delete 172.16.0.0/12 next-hop 10.102.191.254
user@host# delete 192.168.0.0/16 next-hop 10.102.191.254
```

3. (Optional) Remove the configurations for processes that use mgmt_junos. These processes will return to sending packets using the default routing instance. For example, to remove the mgmt_junos configuration for TACACS+ :

```
[edit]
user@host# delete system tacplus-server server-address routing-instance mgmt_junos
```

Release History Table

Release	Description
17.3R1	Starting with Junos OS Release 17.3R1, you can confine the em0 and fxp0 management interfaces to a non-default virtual routing and forwarding (VRF) instance, the mgmt_junos VRF instance.

Loopback Interface Overview

The Internet Protocol (IP) specifies a loopback network with the (IPv4) address 127.0.0.0/8. Most IP implementations support a loopback interface (lo0) to represent the loopback facility. Any traffic that a computer program sends on the loopback network is addressed to the same computer. The most

commonly used IP address on the loopback network is `127.0.0.1` for IPv4 and `::1` for IPv6. The standard domain name for the address is `localhost`.

A network device also includes an internal loopback interface (`lo0.16384`). The internal loopback interface is a particular instance of the loopback interface with the logical unit number 16384.

You use the loopback interface to identify the device. While you can use any interface address to determine if the device is online, the loopback address is the preferred method. Whereas interfaces might be removed or addresses changed based on network topology changes, the loopback address never changes.

When you ping an individual interface address, the results do not always indicate the health of the device. For example, a subnet mismatch in the configuration of two endpoints on a point-to-point link makes the link appear to be inoperable. Pinging the interface to determine whether the device is online provides a misleading result. An interface might be unavailable because of a problem unrelated to the device configuration or operation. You can use the loopback interface to address these issues.

Benefits

- As the loopback address never changes, it is the best way to identify a device in the network.
- The loopback interface is always up and reachable as long as the route to that IP address is available in the IP routing table. Hence, you can use the loopback interface for diagnostics and troubleshooting purposes.
- Protocols such as OSPF use the loopback address to determine protocol-specific properties for the device or network. Further, some commands such as `ping mpls` require a loopback address to function correctly.
- You can apply stateless firewall filters to the loopback logical unit to filter packets originating from, or destined for, the Routing Engine.
- Junos OS creates a separate loopback interface for the internal routing instance, which prevents any filter on `lo0.0` from disrupting internal traffic.

Loopback Interface Configuration

IN THIS SECTION

Configure the Loopback Interface | 43

- Example: Configure Two Addresses on the Loopback Interface with Host Routes | 44
- Example: Configure Two Addresses on the Loopback Interface with Subnetwork Routes | 45
- Example: Configure an IPv4 and an IPv6 Address on the Loopback Interface with Subnetwork Routes | 46

You (a system administrator, network administrator, or end user) can use this procedure to configure the loopback interface on your device.

Configure the Loopback Interface

When specifying the loopback address on a device, do not include a destination prefix. Also, in most cases, specify a loopback address only on unit 0 and no others.

NOTE: For Layer 3 virtual private networks (VPNs), you can configure multiple logical units for the loopback interface. This allows you to configure a logical loopback interface for each virtual routing and forwarding (VRF) routing instance. For more information, see the Junos OS VPNs Library for Routing Devices.

For some applications, such as SSL for Junos XML protocol, at least one address for the interface `lo0.0` must be `127.0.0.1`.

You can configure loopback interfaces using a host (recommended), a subnetwork address for both `inet` and `inet6` address families, or an ISO network entity title (NET) address for the `iso` address family. Many protocols require a loopback address as their source address. Configuring a loopback address as a donor interface for unnumbered interfaces enables these protocols to run on unnumbered interfaces.

In some cases, the loopback interface can also be the router identifier (router ID). If the router ID is not explicitly configured, the device determines its router ID as shown in the following table:

Table 5: Default Router ID

If the loopback interface is:	Then the default router ID is:
Configured	The loopback interface
Not configured	The lowest IP address of any interface in operational state `up`

In both cases, the router ID changes when the operational state of the interface changes. Therefore, we recommend configuring the address on a stable loopback interface.

If you configure more than one address on the loopback interface, we recommend that you configure one to be the primary address. The device selects the primary address as the router ID when the router ID is not configured. The device also uses the primary address as the default source address for traffic sourced from the loopback interface by the Routing Engine.

To configure the physical loopback interface (`lo0`), include the following statements at the `[edit interfaces]` hierarchy level:

```
[edit interfaces]
lo0 {
    unit 0 {
        family inet {
            address loopback-address;
            address <loopback-address2>;
            ...
        }
        family inet6 {
            address loopback-address;
        }
    }
}
```

You can configure one or more addresses on the loopback interface. You can configure more than just `unit 0` for `lo0`, but you must place each additional unit in a separate routing instance.

Example: Configure Two Addresses on the Loopback Interface with Host Routes

In the following example, the user configures two addresses on the loopback interface with host routes:

```
[edit]
user@host# edit interfaces lo0 unit 0 family inet
[edit interfaces lo0 unit 0 family inet]
user@host# set address 10.0.0.1
[edit interfaces lo0 unit 0 family inet]
user@host# set address 172.16.0.1
[edit interfaces lo0 unit 0 family inet]
```

```
user@host# top
[edit]
user@host# show interfaces
lo0 {
    unit 0 {
        family inet {
            10.0.0.1/32;
            172.16.0.1/32;
        }
    }
}
```

Example: Configure Two Addresses on the Loopback Interface with Subnetwork Routes

In some instances, you may need to advertise a subnetwork route as internal rather than a Type 5 route for an redistributed static route using OSPF. In this scenario, you may want to configure subnetwork routes on the loopback interface, as shown in the following example:

```
[edit]
user@host# edit interfaces lo0 unit 0 family inet
[edit interfaces lo0 unit 0 family inet]
user@host# set address 10.2.0.1/16
[edit interfaces lo0 unit 0 family inet]
user@host# set address 192.16.0.1/24
[edit interfaces lo0 unit 0 family inet]
user@host# top
[edit]
user@host# show
interfaces {
    lo0 {
        unit 0 {
            family inet {
                10.2.0.1/16;
                192.16.0.1/24;
            }
        }
```

```
    }
}
```

Example: Configure an IPv4 and an IPv6 Address on the Loopback Interface with Subnetwork Routes

In the following example, the user configures an IPv4 and an IPv6 address on the loopback interface with subnetwork routes:

```
[edit]
user@host# edit interfaces lo0 unit 0 family inet
[edit interfaces lo0 unit 0 family inet]
user@host# set address 192.16.0.1/24
[edit interfaces lo0 unit 0 family inet]
user@host# up
[edit interfaces lo0 unit 0 family]
user@host# edit interfaces lo0 unit 0 family inet6
[edit interfaces lo0 unit 0 family inet6]
user@host# set address 2001:db8::200:f8ff:fe75:50df/64
[edit interfaces lo0 unit 0 family inet6]
user@host# top
[edit]
user@host# show
interfaces {
    lo0 {
        unit 0 {
            family inet {
                192.16.0.1/24;
            }
            family inet6 {
                2001:db8::200:f8ff:fe75:50df/64;
            }
        }
    }
}
```

CHAPTER 7

Initial User Accounts

User Accounts Overview | 48

Configure User Accounts in a Configuration Group | 50

Enable Remote Access Services | 53

User Accounts Overview

User accounts provide one way for users to access a device. For each account, you define the user's login name, password, and any additional user information. After you have created an account, the software creates a home directory for the user.

An account for the user `root` is always present in the configuration. You can configure the password for `root` using the `root-authentication` statement.

While it is common to use remote authentication servers to centrally store information about users, it is also good practice to configure at least one non-root user on each device. This way, you can still access the device if its connection to the remote authentication server is disrupted. This non-root user usually has a generic name such as `admin`.

For each user account, you can define the following:

- Username (Required): Name that identifies the user. It must be unique. Avoid using spaces, colons, or commas in the username. The username can include up to 64 characters.
- User's full name: (Optional) If the full name contains spaces, enclose it in quotation marks. Avoid the use of colons or commas.
- User identifier (UID): (Optional) Numeric identifier that is associated with the user account name. The UID is assigned automatically when you commit the configuration, so you do not need to set it manually. However, if you choose to configure the UID manually, use a unique value in the range from 100 through 64,000.
- User's access privilege: (Required) One of the login classes you defined in the `class` statement at the `[edit system login]` hierarchy or one of the default login classes.
- Authentication method or methods and passwords for device access (Required): You can use a SSH key, a Message Digest 5 (MD5) password, or a plain-text password that Junos OS encrypts using MD5-style encryption before entering it in the password database. For each method, you can specify the user's password. If you configure the `plain-text-password` option, you receive a prompt to enter and confirm the password:

```
[edit system login user username]
user@host# set authentication plain-text-password
New password: type password here
Retype new password: retype password here
```

To create valid plain-text passwords, make sure that they:

- Contain between 6 and 128 characters.

- Include most character classes (uppercase letters, lowercase letters, numbers, punctuation marks, and other special characters) but do not include control characters.
- Contain at least one change of case or character class.

Junos-FIPS and Common Criteria have the following special password requirements. They must:

- Be between 10 and 20 characters long.
- Use at least three of the five defined character sets (uppercase letters, lowercase letters, digits, punctuation marks, and other special characters).

If Junos-FIPS is installed on the device, you must adhere to the special password requirements, or the passwords are not configured.

For SSH authentication, you can copy the contents of an SSH key file into the configuration. You can also configure SSH key information directly. Use the `load-key-file` statement to load an SSH key file that was generated previously, (for example, by using `ssh-keygen`). The `load-key-file` argument is the path to the file location and name. The `load-key-file` statement loads RSA (SSH version 1 and SSH version 2) public keys. The contents of the SSH key file are copied into the configuration immediately after you configure the `load-key-file` statement.

Avoid using the following Transport Layer Security (TLS) version and cipher suite (RSA host key) combinations, which will fail:

With RSA host keys:

- TLS_1.0@DHE-RSA-AES128-SHA
- TLS_1.0@DHE-RSA-AES256-SHA

For each user account and for root logins, you can configure more than one public RSA key for user authentication. When a user logs in using a user account or as root, the configured public keys are referenced to determine whether the private key matches any of the user accounts.

To view the SSH key entries, use the configuration mode `show` command. For example:

```
[edit system login user boojum]
user@host# set authentication load-key-file my-host:.ssh/id_rsa.pub
.file.19692               |          0 KB |   0.3 kB/s | ETA: 00:00:00 | 100%
[edit system login user boojum]
user@host# show
authentication {
    ssh-rsa "$ABC123"; # SECRET-DATA
}
```

Configure User Accounts in a Configuration Group

To make it easier to configure the same user accounts on multiple devices, configure the accounts inside of a configuration group. The examples shown here are in a configuration group called `global`. Using a configuration group for your user accounts is optional.

To create a user account:

1. Add a new user, using the user's assigned account login name.

```
[edit groups global]
user@host# edit system login user username
```

2. (Optional) Configure a descriptive name for the account.

   If the name includes spaces, enclose the entire name in quotation marks.

```
[edit groups global system login user user-name]
user@host# set full-name complete-name
```

   For example:

```
user@host# show groups
global {
    system {
        login {
            user admin {
                full-name "general administrator";
            }
        }
    }
}
```

3. (Optional) Set the user identifier (UID) for the account.

   As with UNIX systems, the UID enforces user permissions and file access. If you do not set the UID, the software assigns one for you. The format of the UID is a number between 100 and 64,000.

```
[edit groups global system login user user-name]
user@host# set uid uid-value
```

For example:

```
user@host# show groups
global {
    system {
        login {
            user admin {
                uid 9999;
            }
        }
    }
}
```

4. Assign the user to a login class.

   You can define your own login classes or assign one of the predefined login classes.

   The predefined login classes are as follows:

   - super-user—all permissions
   - operator—clear, network, reset, trace, and view permissions
   - read-only—view permissions
   - unauthorized—no permissions

```
[edit groups global system login user user-name]
user@host# set class class-name
```

For example:

```
user@host# show groups
global {
    system {
        login {
            user admin {
                class super-user;
            }
        }
    }
}
```

5. Use one of the following methods to configure the user password:

- To enter a clear-text password that the system encrypts for you, use the following command to set the user password:

```
[edit groups global system login user user-name]
user@host# set authentication plain-text-password
New Password: type password here
Retype new password: retype password here
```

  As you enter the password in plain text, the software encrypts it. You do not need to configure the software to encrypt the password. Plain-text passwords are hidden and marked as ## SECRET-DATA in the configuration.

- To enter a password that is encrypted, use the following command to set the user password:

  CAUTION: Do not use the `encrypted-password` option unless the password is *already* encrypted and you are entering the encrypted version of the password.

  If you accidentally configure the `encrypted-password` option with a plain-text password or with blank quotation marks (" "), you will not be able to log in to the device as this user.

```
[edit groups global system login user user-name]
user@host# set authentication encrypted-password "password"
```

- To load previously generated public keys from a named file at a specified URL location, use the following command:

```
[edit groups global system login user user-name]
user@host# set authentication load-key-file URL filename
```

- To enter an SSH public string, use the following command:

```
[edit groups global system login user user-name]
user@host# set authentication (ssh-ecdsa | ssh-ed25519 | ssh-rsa) authorized-key
```

6. At the top level of the configuration, apply the configuration group.

If you use a configuration group, you must apply it for it to take effect.

```
[edit]
user@host# set apply-groups global
```

7. Commit the configuration.

```
user@host# commit
```

8. To verify the configuration, log out and log back in as the new user.

Enable Remote Access Services

You must configure one or more enabling services such as SSH, Telnet, or FTP before authorized users can access your device. You must also configure at least one of these services before your device can exchange data with other systems. SSH, Telnet, and FTP are widely used standards for remotely logging in to network devices and exchanging files between systems. These services are all disabled by default in Junos OS.

The SSH protocol uses strong authentication and encryption for remote access across a network that is not secure. SSH provides remote login, remote program execution, file copy, and other functions. SSH succeeds Telnet and is the recommended method for remote access. SSH encrypts all traffic, including passwords, to effectively eliminate eavesdropping, connection hijacking, and other attacks. The SSH utility includes Secure Copy Protocol (SCP), a file-transfer program that uses SSH and is the recommended method for secure file exchange.

Because both Telnet and FTP are legacy applications that use cleartext passwords, we recommend that you use SSH (and SCP). Cleartext passwords create a potential security vulnerability. If you do not intend to use FTP or Telnet, you do not need to configure them on your device. However, consider that some users might use FTP to store configuration templates, retrieve software, or perform other administrative tasks.

To make it easier to configure these services on multiple devices, configure them inside of a configuration group. To set up remote access and file-transfer services:

1. Enable SSH access.

```
[edit groups global]
user@host# set system services ssh
```

2. Enable Telnet access.

```
[edit groups global]
user@host# set system services telnet
```

3. Enable FTP.

```
[edit groups global]
user@host# set system services ftp
```

4. (Optional) Apply the configuration group. If you use a configuration group, you must apply it at the top level of the configuration for it to take effect.

```
[edit]
user@host# set apply-groups global
```

5. Commit the configuration.

```
user@host# commit
```

RELATED DOCUMENTATION

- *Configuring SSH Service for Remote Access to the Router or Switch*
- *Configuring Telnet Service for Remote Access to a Router or Switch*
- *Configuring FTP Service for Remote Access to the Router or Switch*

8
CHAPTER

Backup Routers

Backup Routers Overview | 56

Configure a Backup Router | 56

Backup Routers Overview

The purpose of the backup router is not to forward transit traffic. You (the system administrator) can use the backup router for local management of the routing device by way of the out-of-band management interface (fxp0, em0, or me0).

The routing protocol process (rpd) is the software process responsible for establishing routes. The rpd does not run while a routing device is booting. Therefore, the device has no routes. The backup router allows the routing device to install a route to a management network before the rpd is up and running.

You can use a backup router during the initial boot process, before any routing protocols have converged. The backup router allows the device to establish a Layer 3 connection quickly, which keeps management unavailability to a minimum. You can also use the backup router to manage the device if the routing protocol process fails to start properly.

The backup router connects directly to the local routing device (that is, the device connected on the same subnet) through its private management interface. To select a backup router, you commonly choose the default gateway of the management network that is directly connected to your routing device. Make sure that the specified backup router address is reachable and directly connected to the management interface.

NOTE: Router A can be the backup router for Router B, and Router B can be the backup router for Router A, if the management interface of each router is connected to an interface on the other router, thus providing the necessary reachability.

When the rpd starts, the backup route (the route created by the backup router) is removed, and any default, static, or protocol-learned routes are installed.

Backup-router configurations can alter the backup Routing Engine kernel forwarding table even with nonstop active (NSR) enabled.

Configure a Backup Router

IN THIS SECTION

Configure a Backup Router Running IPv4 for Routers | 57

Configure a Backup Router Running IPv6 for Routers | 59
Configure a Backup Router for SRX Series Devices | 61

To add a backup router to your device, configure the `backup-router` or the `inet6-backup-router` statement at the `[edit system]` hierarchy level.

You (the network administrator) can use the backup router to reach the network while loading, configuring, and recovering the router or switch without installing a default route in the forwarding table. Include the optional `destination` option and specify an address that is reachable through the backup router. Use the address format *network/mask-length*. This configuration supports both IPv4 and IPv6 addresses. The prefix of the destination address cannot overlap with the destination prefix learned from the routing protocol process (rpd).

If the backup router configuration has multiple static routes that point to a gateway from the management Ethernet interface, you must configure prefixes that are more specific than the static routes.

For example, if you configure the static route 172.16.0.0/12 from the management Ethernet interface for management purposes, you must specify the backup router configuration as follows:

```
set system backup-router 172.29.201.62 destination [172.16.0.0/13 172.16.128.0/13]
```

Any destinations defined by the backup router are not visible in the routing table. They are only visible in the local forwarding table when the rpd is not running.

On systems with dual redundant Routing Engines, the backup Routing Engine's reachability through the private management interface is based only on the functionality of the `backup-router` configuration. It is not based on whether the rpd is running. On both Routing Engines, the `backup-router` statement adds the destination prefix upon bootup. On the primary Routing Engine, a static route requires the rpd to be running before the static route installs the destination prefix to the routing and forwarding tables.

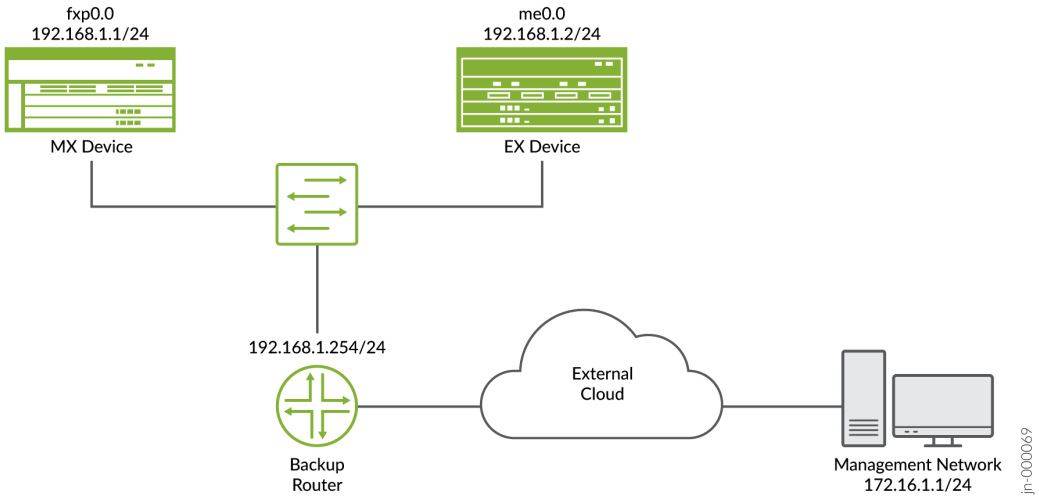
Active routes and more specific routes take precedence over destination prefixes defined with the `backup-router` statement.

Configure a Backup Router Running IPv4 for Routers

In the example shown in Figure 1 on page 58, the backup router is the default gateway of the management network.

As required, the backup router address is reachable and directly connected to the management interfaces on the two routing devices (fxp0 and me0).

Figure 1: Backup Router Sample Topology



Optionally, instead of configuring the backup router at the `[edit system]` hierarchy level, you can use a configuration group, as shown in this procedure. This is a recommended best practice for configuring the backup router, especially if the device has dual Routing Engines. This procedure uses groups called `re0` and `re1` as an example.

To configure a backup router running IPv4:

1. Include the `backup-router` statement at the `[edit system]` hierarchy level.

```
[edit groups group-name system]
backup-router address <destination destination-address>;
```

For example:

```
[edit groups re0 system]
backup-router 192.168.1.254 destination 172.16.1.0/24;
```

```
[edit groups re1 system]
backup-router 192.168.1.254 destination 172.16.1.0/24;
```

2. (Optional) Configure a static route to the management network.

   The software uses the backup router only during the boot sequence. If you want to configure a backup router for use after startup, you can set up a static route. The static route goes into effect when the rpd is running.

```
routing-options {
    static {
        route 172.16.1.0/24 {
            next-hop 192.168.1.254;
            retain;
        }
    }
}
```

3. If you used one or more configuration groups, apply the configuration groups, substituting the appropriate group names.

   For example:

```
[edit]
user@host# set apply-groups [re0 re1]
```

4. Commit the changes:

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

Configure a Backup Router Running IPv6 for Routers

To configure a backup router running IPv6:

1. Include the `inet6-backup-router` statement at the `[edit system]` hierarchy level.

```
[edit groups group-name system]
inet6-backup-router address <destination destination-address>;
```

For example:

```
[edit groups re0 system]
inet6-backup-router 2001:db8:10::1 destination 2001:db8::/48;
```

```
[edit groups re1 system]
inet6-backup-router 2001:db8:10::1 destination 2001:db8::/48;
```

2. (Optional) Configure a static route to the management network.

   The software uses the backup router only during the boot sequence. If you want to configure a backup router for use after startup, you can set up a static route. The static route goes into effect when the rpd is running.

```
routing-options {
    rib inet6.0 {
        static {
            route 2001:db8::/48 {
                next-hop 2001:db8:10::1;
                retain;
            }
        }
    }
}
```

3. If you used one or more configuration groups, apply the configuration groups, substituting the appropriate group names.

   For example:

```
[edit]
user@host# set apply-groups [re0 re1]
```

4. Commit the changes:

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

Configure a Backup Router for SRX Series Devices

This procedure describes how to manage two SRX Series devices in a chassis cluster mode using a backup router configuration. The backup router address is reachable and directly connected to the management interfaces on the SRX chassis cluster (fxp0).

When you configure the backup router for SRX Series devices in chassis cluster mode, the backup router configuration facilitates the management access on the backup node only. You enable access to the primary node through the routing on the primary node. When you configure the backup router, Junos OS injects a route into the forwarding table on the secondary node. You cannot view the routing table on the secondary node since the routing subsystem does not run on the secondary node. This example uses groups `node0` and `node1`.

1. Include the `backup-router` statement at the `[edit system]` hierarchy level.

```
[edit groups group-name system]
backup-router address <destination destination-address>;
```

To configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set groups node0 system host-name Corp-FW0
set groups node0 system backup-router 192.168.1.254
set groups node0 system backup-router destination 172.16.1.1/24
set groups node0 interfaces fxp0 unit 0 family inet address 192.0.2.0/24
set groups node1 system host-name Corp-FW1
set groups node1 system backup-router 192.168.1.254
set groups node1 system backup-router destination 172.16.1.1/24
set groups node1 interfaces fxp0 unit 0 family inet address 192.0.2.1/24
set apply-groups "${node}"
```

2. Commit the changes:

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