Engineering
Simplicity

Junos® OS

Getting Started with Junos OS and Junos OS Evolved

Published
2021-04-18

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

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 and Junos OS Evolved

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

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

RELATED DOCUMENTATION

Day One: Exploring the Junos CLI, Second Edition

1

CHAPTER

Understanding Junos OS and Junos OS Evolved

About the Getting Started Guide

Getting Started with Junos OS and Junos OS Evolved is intended to provide a high-level introduction to Junos OS and Junos OS Evolved, and explain basic concepts and operational principles for working with Juniper Networks devices.

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

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

NOTE: References to Junos OS here, unless stated otherwise, also include and apply to Junos OS Evolved. Any differences are noted.

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

For introductory and overview information specific to Junos OS Evolved, see Introducing Junos OS Evolved. This guide will acquaint you with Junos OS Evolved, the next generation Junos OS, and explain its strengths, similarities to, and differences from Junos OS.

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

RELATED DOCUMENTATION

CLI User Guide

Introducing Junos OS Evolved

Overview for Junos OS

Understanding Junos OS and Junos OS Evolved Infrastructure and Processes

IN THIS SECTION

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

Junos OS and Junos OS Evolved run 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 command-line interface (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 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 and Junos OS Evolved 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.

Table 1: Junos OS and Junos OS Evolved Processes *(Continued)*

Process	Name	Description
Ethernet switching process	eswd	Handles Layer 2 switching functionality such as MAC address learning, Spanning Tree Protocol, 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	firewall	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.
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.

Table 1: Junos OS and Junos OS Evolved Processes *(Continued)*

Process	Name	Description
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.
Secure Neighbor Discovery (SEND) protocol process	send	Protects Neighbor Discovery Protocol (NDP) messages.
Simple Network Management Protocol (SNMP) process	snmp	Enables the monitoring of network devices from a central location and provides the switch’s SNMP master agent.

Table 1: Junos OS and Junos OS Evolved Processes *(Continued)*

Process	Name	Description
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	vrrp	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

Accessing a Juniper Networks Device

Initial Configuration Overview for Juniper Networks Devices

After you install and power on the Juniper Networks device, you are ready to begin initial configuration. A version of Junos OS or Junos OS Evolved is preinstalled on all devices. The procedures in this guide show you how to connect the device to the network but do not enable it to forward traffic. For complete information about enabling the router 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 OS and Junos OS Evolved and for details regarding Configuration Statements and command-line interfaces (CLIs), see the Overview for Junos OS, Introducing Junos OS Evolved, and the CLI User Guide.

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

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.

RELATED DOCUMENTATION

Console Port Overview | 10

Accessing a Juniper Networks Device the First Time | 10

CLI User Guide

Introducing Junos OS Evolved

Overview for Junos OS

Console Port Overview

Console ports allow access to Junos OS and Junos OS Evolved devices, 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 or might not be connected.

A console 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, Junos OS Evolved and software package upgrades using the console because this connection will always remain 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.

RELATED DOCUMENTATION

Initial Configuration Overview for Juniper Networks Devices | 9

Accessing a Juniper Networks Device the First Time | 10

Accessing a Juniper Networks Device the First Time

NOTE: Before proceeding with any new Juniper Networks device, be sure to follow the quick start and initial set-up instructions which came with the device.

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

To configure the device initially, you must connect a terminal or laptop computer through the console port, which can vary in position and type, depending on the specific device model. Only console access to the device is enabled by default. Remote management access to the router and all management access protocols, including Telnet, FTP, and SSH, are disabled by default.

To access a network device for the first time:

1. Connect a terminal or laptop computer to the device through the console port.
2. Power on the device and wait for it to boot.

   The software boots automatically. The boot process is complete when you see the **login:** prompt on the console user interface.
3. Log in as the user **root**.

   Initially, the root user account requires no password. You can see that you are the **root** user, because the prompt on the device shows the username **root@%**.
4. Start the command-line interface (CLI).

   ```
   root@% cli
   root@>
   ```
5. Enter configuration mode.

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

RELATED DOCUMENTATION

Initial Configuration Overview for Juniper Networks Devices | 9

Understanding the Root Password | 13

Protecting Network Security by Configuring the Root Password

Root Password

3

CHAPTER

Changing the Device Default Root Password

Understanding the Root Password

The root user has complete privileges to operate and 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 a root password is not defined, you cannot commit configuration settings on the device.

The root password must meet the following conditions:

- Be at least six characters long. Most character classes can be included in a password (alphabetic, numeric, and special characters), except control characters.

- Contain at least one change of case or character class.

RELATED DOCUMENTATION

Protecting Network Security by Configuring the Root Password

Protecting Network Security by Configuring the Root Password

Configuring the root password on your Juniper Networks device helps 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 to protect this account by setting a strong password when setting up a new device.

After a new device is initially powered on, you log in as the user **root** with no password. The software requires configuration of the root password before it accepts a commit operation.

To set the root password, you have a few options as shown in the following procedure.

- Enter a plain-text password the software encrypts.

- Enter a password that is already encrypted.

- Enter a secure shell (ssh) public key string.

The most secure options of these three are using an already encrypted password or an ssh public key string. Pre-encrypting your password or using a ssh public key string means 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 do not have to configure the software to encrypt the password as in some other systems. 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 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 root, and you will 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-dsa | ssh-ecdsa | ssh-rsa key)
     ```

2. If you used a configuration group, apply it with the command **set apply-groups**, replacing *<group name>* with the configuration group name.

   For example:

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

3. Commit the changes.

   ```
   root@# commit
   ```

RELATED DOCUMENTATION

Accessing a Junos OS Device the First Time

4

CHAPTER

Setting the Device Hostname

Understanding Hostnames

Almost all devices in your network have a hostname.

The hostname is the name that identifies the device on the network and 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 is indicative of a device that is booting from a factory-fresh Junos OS software load, which, by definition, does not have a hostname configured.

Administrators often follow conventions for naming devices. One such convention is to name the device based on its location, for example: germany-berlin-R1. The hostname should be unique within your network infrastructure, but there is no need for the local hostname to be globally unique.

A device's hostname usually has a corresponding entry in the domain name system (DNS) so that administrators can connect to the device using the hostname. The fully qualified domain name (FQDN), which is used in DNS, includes the hostname and the entire domain name. The hostname and the domain name labels are separated by periods or dots, as follows: *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 and Junos OS Evolved, the hostname can contain any combination of alphabetic characters, numbers, dashes, and underscores. No other special characters are allowed.

Although 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. In any case, an overly long hostname is difficult to type and to remember, so short and meaningful hostnames are a best practice.

Configuring the Hostname of a Device by Using a Configuration Group

The hostname of a Junos OS or Junos OS Evolved 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 has other purposes.

The software 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 when knowing the device identity is important. For these reasons, we recommend hostnames be descriptive and memorable.

You can configure the hostname at the **[edit system]** hierarchy level, a procedure shown in "Configuring a Device's Unique Identity for the Network" on page 28. 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**: Starting with Junos OS Release 13.2R3, 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, commit is successful.
>
> In Junos OS releases earlier than Release 13.2R3, if you configured such hostnames by using the **host-name *hostname*** statement at the **[edit system]** hierarchy level and the the terminal screen width was less than the length of the hostname by using the **set cli screen-width** statement, a foreign file propagation (ffp) failure error message is displayed when you attempt to commit the configuration. In such a case, because of the ffp failure, the commit operation does not complete and you cannot recover the router unless you make the modification in the backend in the **juniper.conf.gz** file and commit the change from the shell prompt.

To set the hostname using a configuration group:

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

   The name 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 subsequently appears in the device CLI prompt.

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

5

CHAPTER

Configuring DNS, Server Caching, and Device Identity

Understanding and Configuring DNS

IN THIS SECTION

DNS Overview

IN THIS SECTION

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.

This topic includes the following sections:

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 are responsible for providing the hostname 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.

SEE ALSO

> *Configuring the TTL Value for DNS Server Caching*

Configuring a DNS Name Server for Resolving Hostnames into Addresses

Domain Name System (DNS) name servers are used for resolving 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, or even configure multiple addresses on the name server.

For redundancy, it is a best practice to 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 Web site to its network address. Additionally, the software enables you to configure one or more domain names, which it uses to resolve hostnames that are not fully qualified (in other words, the domain name is missing). This 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.

Starting in Junos OS Release 19.2R1, you can route traffic between a management routing instance and DNS name server. Configure a routing instance at the **[edit system name-server *server-ip-address*]** hierarchy level and 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, configure the following:

```
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 have 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 {
       192.168.1.253/32;
       192.168.1.254/32;
   }
   ```

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 these as a list of domains to be searched. The software 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 two domains to be searched. This example configures the software to search the company.net domain and then the domainone.net domain and then 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, substituting **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 have 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.
```

SEE ALSO

name-server (System Services)

domain-search

RELATED DOCUMENTATION

Understanding Hostnames | 17

DNSSEC Overview

Configuring the TTL Value for DNS Server Caching

IN THIS SECTION

This section describes how to configure the TTL value for a DNS server cache to define the period for which DNS query results are cached.

Requirements

No special configuration beyond device initialization is required before performing this task.

Overview

IN THIS SECTION

The DNS name server stores DNS query responses in its cache for the TTL period specified in the TTL field of the resource record. When the TTL value expires, the name server sends a fresh DNS query and updates the cache. You can configure the TTL value from 0 to 604,800 seconds. You can also configure the TTL value for cached negative responses. Negative caching is the storing of the record that a value does not exist. In this example, you set the maximum TTL value for cached (and negative cached) responses to 86,400 seconds.

Topology

Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

To configure the TTL value for a DNS server cache:

1. Specify the maximum TTL value for cached responses, in seconds. (In this example, 86400 seconds equals 24 hours.)

   ```
   [edit]
   user@host# set system services dns max-cache-ttl 86400
   ```

2. Specify the maximum TTL value for negative cached responses, in seconds.

   ```
   [edit]
   user@host# set system services dns max-ncache-ttl 86400
   ```

3. If you are done configuring the device, commit the configuration.

   ```
   [edit]
   user@host# commit
   ```

Verification

To verify the configuration is working properly, enter the **show system services** command.

RELATED DOCUMENTATION

DNS Overview | **21**

Configuring a Device’s Unique Identity for the Network

IN THIS SECTION

To use a device in a network, you must configure the device’s identity. Doing this makes the device accessible on the network and so that other users can log in to it. You can refer to any Internet-connected machine 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

Although using DNS is an easier and more scalable way to resolve IP addresses from hostnames, you might not have a DNS entry for the router, or you might not want the computer to contact the DNS server to resolve a particular IP address. In this latter instance, perhaps you use this particular IP address frequently, or you might be using it only for testing or development purposes and do not want to give it a DNS entry.

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

Requirements

No special configuration beyond device initialization is required.

Overview

In this context, the hostname is the device's name. It is easier for most people 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, as well as in other places where knowing the device identity is useful. You can also use the hostname to telnet to a device.

The domain name is the string appended 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."

In case your hostname and IP address do not have a DNS entry in a name server, configure a static mapping.

The values given in the following table are used 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**	
host-name *host-name*	**host-name example-re0**	

Table 2: Values to Use in Example *(Continued)*

Name of Variable	Value Used in Example	Value You Substitute
inet *ip-address*	**inet 172.22.147.39**	
name-server *ip-address*	**name-server 172.24.16.115** **name-server 192.0.2.0**	

Configuration

IN THIS SECTION

CLI Quick Configuration

To quickly configure a device using this example

- Copy the following commands and paste them in a text file
- Remove any line breaks
- Change the values listed here to match your network configuration
- Copy and paste the commands into the CLI at the **[edit]** hierarchy level
- Finally, enter **commit** from 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 192.0.2.0
set system static-host-mapping example-re0 inet 172.22.147.39
```

Configuring the Router’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, using the **set system host-name** command.

   ```
   [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 192.0.2.0
   ```

4. Map from the hostname to the IP address, using the **set system static-host-mapping** command.

   ```
   [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;
    192.0.2.0;
}
static-host-mapping {
    example-re0 {
        inet 172.22.147.39;
    }
}
```

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

Verification

IN THIS SECTION

Confirm the Device Hostname and IP Address

Purpose

Confirm the hostname and 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
```

RELATED DOCUMENTATION

Understanding Hostnames | 17

Configuring a DNS Name Server for Resolving Hostnames into Addresses | 22

6

CHAPTER

Understanding and Configuring Management Ethernet and Loopback Interfaces

Understanding Management Ethernet Interfaces

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. Through a management interface you can access the device over the network using utilities such as **ssh** and **telnet** and configure the device from anywhere, regardless of its physical location. SNMP can use the management interface to gather statistics from the device.

A management interface lets authorized users and management systems 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. This interface can be dedicated to management or shared with other traffic. Before users can access the management interface, you must configure it. Information required to set up the management interface includes its IP address and prefix. In many types of Junos OS and Junos OS Evolved devices (or recommended configurations), 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).

For Junos OS Evolved, use re0:mgmt-* for Routing Engine 0 and re1:mgmt-* for Routing Engine 1 management interfaces.

Management Interface in a Non-Default Instance

IN THIS SECTION

Why Use a Non-Default Management Interface?

By default, the management Ethernet interface (usually named fxp0 or em0 for Junos OS, or RE-0 or RE-1 for Junos OS Evolved) provides the out-of-band management network for the device. There is no clear separation between either out-of-band management traffic and in-band protocol control traffic, that is, user traffic at the routing-instance or routing-table level. Instead, all traffic is handled through the default routing instance, giving rise to concerns over security, performance, and how to troubleshoot.

Starting with Junos OS Release 17.3R1, you can confine the em0 and fxp0 management interfaces in a non-default virtual routing and forwarding (VRF) instance, the mgmt_junos routing instance. After you configure this management routing instance, management traffic no longer has to share a routing table (that is, the default inet.0 table) with other control or protocol traffic in the system. This improves security and makes it easier to use the management interface to troubleshoot.

NOTE: Only the em0 and fxp0 interfaces are supported in the non-default management VRF. Other management interfaces such as em1 are not supported in the non-default management VRF.

Applications and Processes That Are VRF Aware

Many processes communicate through the management interface. In order for the non-default management instance to support these processes, they must support a management VRF. To make many of these processes work with the non-default VRF instance, you must configure the name of the new management routing instance (mgmt_junos) for these processes. These processes have been enhanced to be able to use the management routing instance.

For the processes that require this additional configuration and where to find more information for those processes, see Table 3 on page 36.

Table 3: Junos Processes You Can Configure to Use the Management VRF

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*

Table 3: Junos Processes You Can Configure to Use the Management VRF *(Continued)*

Process	First Release to Support Managment VRF	For More Information
NTP	Junos OS Release 18.1R1	*ntp*
RADIUS	Junos OS Release 18.1R1	*Configuring RADIUS Server Authentication* *Configuring RADIUS System Accounting*
REST API	Junos OS Release 20.3R1	*rest*
syslog	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 the mgmt_junos Routing Instance

IN THIS SECTION

You can confine the management interface in a dedicated management instance by configuring the **management-instance** configuration statement at the **[edit system]** hierarchy level. The name of the dedicated management instance is reserved and hardcoded as **mgmt_junos**; you are prevented from configuring any other routing instance by the name **mgmt_junos**. Once the mgmt_junos routing instance is deployed, management traffic no longer shares a routing table (that is, the default inet.0 table) with other control or protocol traffic in the system, nor is configuring dynamic protocols on the management interface supported.

Because there are FreeBSD and Junos OS applications that assume that the management interface is always present in the default inet.0 routing table, the mgmt_junos routing instance is not instantiated by default.

As part of configuring the mgmt_junos routing instance, you must also move static routes that have a next hop over the default management interface to the mgmt_junos routing instance. If needed, you must also configure the appropriate daemons or applications to use the mgmt_junos routing instance. All of these changes must be done in a single commit. Otherwise, the transition to mgmt_junos will not be smooth and you will have to repair the system later by logging in from the console.

After you commit the configuration, expect to lose, and then have to reestablish, the Telnet session.

For an example of using this feature, see the following sections.

Determining Static Routes

As part of configuring the mgmt_junos routing instance, you must move all the static routes that have a next hop through the default management interface from the default routing instance to mgmt_junos. Each setup is different. In these examples, you need to identify the static routes that have a next hop through the fxp0 interface. The next hop for any static route that is affected will have an IP address that falls under the subnet of the IP address configured for fxp0.

You can use the following commands to determine static routes that need to be changed.

- Use the **show interfaces** command to find the IP address of the default management interface:

```
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
```

  In this case the default management interface is fxp0, But it could be em0 or re0:mgmt-*.

- 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**):

```
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
```

- 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
```

Enabling the mgmt_junos Routing Instance

NOTE: We recommend using the device console port for these operations, because at the point where you commit the configuration, if you are using SSH or telnet, the connection to the device will be dropped and you will have to reestablish it. If using SSH or telnet anyway, use **commit confirm**.

To enable the mgmt_junos routing instance:

1. Configure the mgmt_junos routing instance at the **[edit routing-instances** hierarchy level:

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

2. Configure the **management-instance** statement.

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

3. Move the appropriate static routes to the mgmt_junos routing instance.

   For how to determine static routes to change, see "Determining Static Routes" on page 38.

   ```
   [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 might want to 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 s172.16.0.0/12 next-hop 10.102.191.254
user@host# set 192.168.0.0/16 next-hop 10.102.191.254
```

4. Commit the configuration.
5. At this point you have configured the **management-instance** statement. Tables for the mgmt_junos table are set up for inet and inet6 and marked as private tables. The management interface is moved to the mgmt_junos routing table. Static routes with a next hop to the management interface are moved from the default routing table and added to the mgmt_junos routing instance.

   However, if you have not configured the **management routing-instance** option in the **tacplus server** statement, the TACACS+ packets continue to be sent using the default routing instance only.

Removing the mgmt_junos Routing Instance

When you remove the mgmt_junos routing instance, you must also move the static routes back to the default routing instance and delete the TACACS+ settings for mgmt_junos.

To remove the dedicated management interface:

1. Delete or deactivate the management routing-instance statement.

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

2. (Optional) Delete the TACACS+ settings for mgmt_junos.
3. Move the static routes back to the default routing 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
```

RELATED DOCUMENTATION

management-instance

Understanding the Loopback Interface

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 address (**lo0.16384**). The internal loopback address is a particular instance of the loopback address with the logical unit number 16384.

The loopback interface is used to identify the device. While any interface address can be used 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's configuration or operation. You can use the loopback interface to address these issues.

Benefits of Loopback Interface

- 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 it is 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 address to filter packets originating from, or destined for, the Routing Engine.
- Junos OS creates the loopback interface for the internal routing instance, which prevents any filter on **lo0.0** from disrupting internal traffic.

RELATED DOCUMENTATION

Understanding Interfaces

Loopback Interface Configuration

IN THIS SECTION

Configuring the Loopback Interface

When specifying the loopback address, do not include a destination prefix. Also, in most cases, do not specify a loopback address on any unit other than unit 0.

> **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, the address for the interface **lo0.0** must be **127.0.0.1**.

You can configure loopback interfaces using a subnetwork address for both inet and inet6 address families. Many protocols require a subnetwork address as their source address. Configuring a subnetwork loopback address as a donor interface enables these protocols to run on unnumbered interfaces.

If you configure the loopback interface, it is automatically used for unnumbered interfaces. If you do not configure the loopback interface, the router chooses the first interface to come online as the default. If you configure more than one address on the loopback interface, we recommend that you configure one to be the primary address to ensure that it is selected for use with unnumbered interfaces. By default, the primary address is used as the source address when packets originate from the interface.

On the router, you can configure the physical loopback interface, **lo0**, and one or more addresses on the interface. You can configure more than just **unit 0** for **lo0**, but each additional unit needs to be applied somewhere other than the main instance.

To configure the physical loopback interface, 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;
        }
    }
}
```

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

To configure 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 172.16.0.1
[edit interfaces lo0 unit 0 family inet]
user@host# set address 10.0.0.1
[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;
                127.0.0.1/32;
                172.16.0.1/32;
            }
        }
    }
}
```

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

To configure two addresses 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# set address 10.2.0.1/16
[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;
                127.0.0.1/32;
                192.16.0.1/24;
            }
        }
    }
}
```

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

To configure 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 3ffe::1:200:f8ff:fe75:50df/64
[edit interfaces lo0 unit 0 family inet6]
user@host# top
[edit]
user@host# show
interfaces {
    lo0 {
        unit 0 {
            family inet {
                127.0.0.1/32;
                192.16.0.1/24;
            }
            family inet6 {
                3ffe::1:200:f8ff:fe75:50df/64;
            }
        }
    }
}
```

RELATED DOCUMENTATION

Junos OS VPNs Library for Routing Devices

7

CHAPTER

Setting Up Initial User Accounts

User Accounts Overview

Junos OS and Junos OS Evolved user accounts provide one way for users to access the device. (Users can access the device without accounts if you configured RADIUS or TACACS+ servers, as described in *Junos OS User Authentication Methods*.) For each account, you define the login name and password for the user and, optionally, additional parameters and metadata for the user. 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 configure the password for **root** using the *root-authentication* statement, as described in *Configuring the Root Password*.

It is a common practice to use remote authentication servers to centrally store information about users. Even so, it is also a good practice to configure at least one non-root user directly on each device, in case access to the remote authentication server is disrupted. This one non-root user commonly has a generic name, such as **admin**.

For each user account, you can define the following:

- Username: Name that identifies the user. It must be unique within the device. Do not include spaces, colons, or commas in the username. The username can be up to 64 characters long.

- User’s full name: (Optional) If the full name contains spaces, enclose it in quotation marks. Do not include colons or commas.

- User identifier (UID): (Optional) Numeric identifier that is associated with the user account name. Typically there is no need to set the UID because the software automatically assigns it when you commit the configuration. However, if you manually configure the UID, it must be in the range from 100 through 64,000 and must be unique within the device.

  You must ensure that the UID is unique. However, it is possible to assign the same UID to different users. If you do this, the CLI displays a warning when you commit the configuration and then assigns the duplicate UID.

- User’s access privilege: (Required) One of the login classes you defined in the **class** statement at the **[edit system login]** hierarchy level, or one of the default classes listed in *Junos OS User Access Privileges*.

- Authentication method or methods and passwords that the user can use to access the device—You can use SSH or a Message Digest 5 (MD5) password, or you can enter a plain-text password that the 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 are prompted 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
```

The default requirements for plain-text passwords are:

- The password must be between 6 and 128 characters long.
- You can include most character classes in a password (uppercase letters, lowercase letters, numbers, punctuation marks, and other special characters). Control characters are not recommended.
- Valid passwords must contain at least one change of case or character class.

Junos-FIPS and Common Criteria have special password requirements. FIPS and Common Criteria passwords must be between 10 and 20 characters in length. Passwords must 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 cannot configure passwords unless they meet this standard.

For SSH authentication, you can copy the contents of an SSH key file into the configuration or directly configure SSH key information. Use the **load-key-file** *URL filename* command to load an SSH key file that was previously generated, e.g. by using **ssh-keygen**. The *URL filename* is the path to the file’s location and name. This command loads RSA (SSH version 1 and SSH version 2) and DSA (SSH version 2) public keys. The contents of the SSH key file are copied into the configuration immediately after you enter the **load-key-file** statement. Optionally, you can use the **ssh-dsa** *public key* **<from** *hostname*> and the **ssh-rsa** *public key* **<from** *hostname*> statements to directly configure SSH keys.

The following TLS version and cipher suite combinations will fail when you use the specified type of host key.

With RSA host keys:

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

With DSA host keys:

- TLS 1.0 (default ciphers)
- TLS 1.1 (default ciphers)

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

For each user account and for root logins, you can configure more than one public RSA or DSA 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 them.

To view the SSH keys 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_dsa.pub
.file.19692                |          0 KB |   0.3 kB/s | ETA: 00:00:00 | 100%
[edit system]
user@host# show
root-authentication {
    ssh-rsa "$ABC123"; # SECRET-DATA
}
```

Configuring User Accounts by Using a Configuration Group

Because Junos OS and Junos OS Evolved user accounts are configured on multiple devices, they are commonly configured inside of a configuration group. As such, 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 full descriptive name for the account.

If the full 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, as the software assigns one for you. The format of the UID is a number in the range of 100 to 64000.

```
[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 password
     New Password: type password here
     Retype new password: retype password here
     ```

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

   - To enter a password that is already 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 to set the user password:

```
[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 to set the user password:

```
[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.

Enabling Remote Access and File Access Services

SSH, telnet, and FTP are widely used standards for remotely logging into network devices and exchanging files between systems. Before authorized users can access your device, or your device can exchange data with other systems, you must configure one or more of these enabling services. They are all disabled by default in Junos OS and Junos OS Evolved.

SSH is a protocol that uses strong authentication and encryption for remote access across a nonsecure network. SSH provides remote login, remote program execution, file copy, and other functions. SSH is telnet's successor 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 SCP (secure copy), 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 clear text passwords (therefore creating a potential security vulnerability), we recommend that you use SSH (and SCP). If you do not intend to use FTP or telnet, you do not need to configure them on your device. However, do not forget to consider that some users might use FTP to store configuration templates, retrieve software, or perform other administrative tasks.

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. 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
```

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

Configuring Backup Routers

Understanding Backup Routers

The purpose of the backup router is not to forward transit traffic. It is for local management of the routing device, by way of the out-of-band management interface (fxp0 or me0 for Junos OS, or RE-0 or RE-1 for Junos OS Evolved, for example).

The software process responsible for establishing routes is known as the routing protocol process (rpd). The backup router feature allows the routing device to install a route to a management network, before the routing protocol process is up and running. A backup router can be used during the initial boot process, before any routing protocols have converged. It allows the device to establish a Layer 3 connection quickly, thus keeping management unavailability to a minimum. In selecting a backup router, it is common practice to choose the default gateway of the management network that is directly connected to your routing device.

When a routing device is booting, the routing protocol process is not running. Therefore, the router or switch has no routes. To ensure that the router or switch is reachable for management purposes while it boots or if the routing protocol process fails to start properly, configure a backup router, which is a router that is directly connected to the local router or switch (that is, on the same subnet) through its private management interface (for example, fxp0, me0, RE-0, or RE-1).

It is important to make sure that the specified backup router address is reachable and directly connected. The backup router address should be an address that is 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 routing protocol process 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.

RELATED DOCUMENTATION

Configuring a Backup Router | 58

Configuring a Backup Router

IN THIS SECTION

To achieve network reachability while loading, configuring, and recovering the router or switch, but without installing a default route in the forwarding table, include the **destination** option, specifying an address that is reachable through the backup router. Specify the address in the format ***network/ mask-length***. Both IPv4 and IPv6 addresses are supported.

If you have a backup router configuration in which multiple static routes point to a gateway from the management Ethernet interface, you must configure prefixes that are more specific than the static routes or include the **retain** option at the **[edit routing-options static route]** hierarchy level.

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:

```
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 routing protocol process is not running. Therefore, a recommended best practice is to include the destinations of the backup router configured as static routes with the **retain** option. The **retain** option is necessary to allow the static route to remain in the forwarding table when the routing protocol process stops running, because the routing table does not exist if the routing protocol process is not running.

Also, the **destination prefix** under backup-router configuration cannot overlap with the **destination prefix** learned from routing protocol process (rpd).

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 routing protocol process 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 routing protocol process to be running first before installing the destination prefix to the routing and forwarding tables.

Due to a system limitation, do not configure the destination address specified in the backup-router as 0.0.0.0/0 or ::/0. The mask has to be a nonzero value.

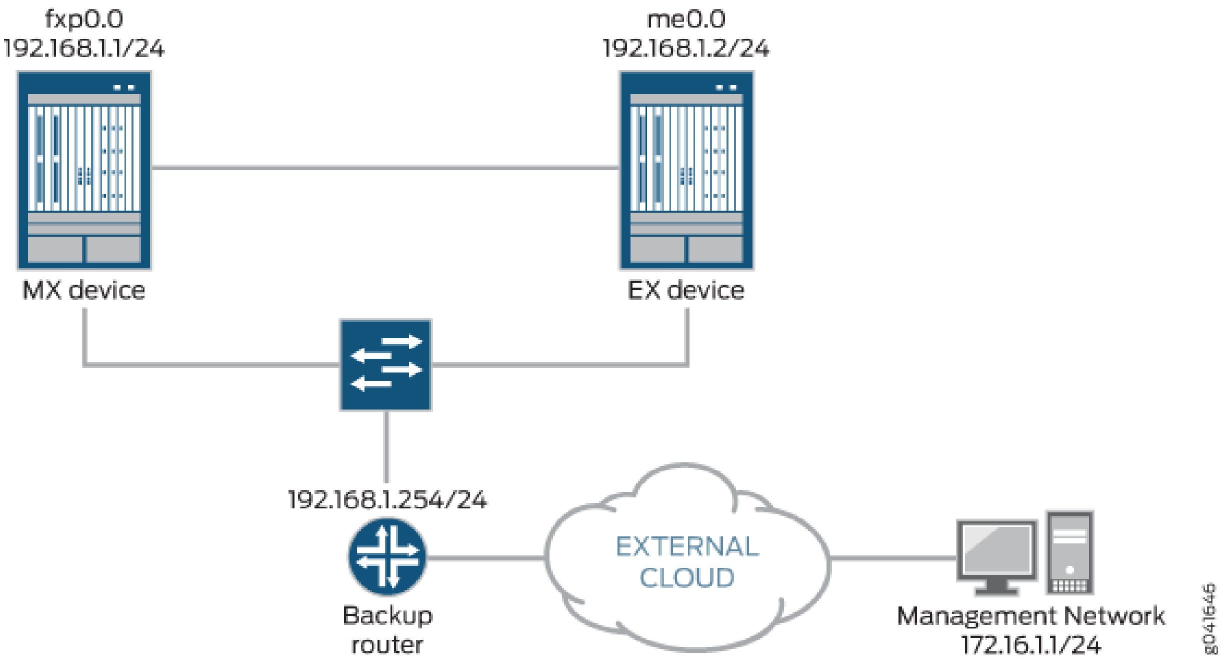
Active routes and more specific routes take precedence over destination prefixes defined with the **backup-router** statement.

Configuring a Backup Router Running IPv4 for Routers

In the example shown in Figure 1 on page 59, 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 only uses the backup router 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 routing protocol process 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
```

Configuring 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 8:3::1 destination abcd::/48;
   ```

   ```
   [edit groups re1 system]
   inet6-backup-router 8:3::1 destination abcd::/48;
   ```

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

   The software only uses the backup router 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 routing protocol process is running.

   ```
   routing-options {
       rib inet6.0 {
           static {
               route abcd::/48 {
                   next-hop 8:3::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
```

Configuring 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, via fxp0.

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 is intended to facilitate the management access on the backup node only. The access to the primary node is enabled via the routing on the primary node. When the backup router configuration is done, a route is injected into the forwarding table on the secondary node. It is not possible to see the routing table on the secondary, as the routing subsystem does not run on the secondary. This example uses groups **node 0** 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
```

RELATED DOCUMENTATION

Understanding Backup Routers | 57

Configuring Junos OS for the First Time on a Device with a Single Routing Engine

Configuring Junos OS for the First Time on a Device with Dual Routing Engines

Requirements for Routers with a Backup Router Configuration