

Multi-Instance Mode

You can deploy the Secure Firewall 3100 as a single device (*appliance mode*) or as multiple container instances (*multi-instance mode*). This chapter describes how to deploy the device in multi-instance mode.

- Multi-instance mode, on page 1
- Licenses for instances, on page 13
- Requirements and prerequisites for instances, on page 13
- Guidelines and limitations for instances, on page 14
- Instance configuration, on page 17
- Multi-instance mode, on page 61
- Multi-instance mode feature and version history, on page 64

Multi-instance mode

Multi-instance mode is a deployment mode that enables multiple container instances on a single chassis that act as completely independent devices.

Multi-instance modes and appliance modes

A multi-instance mode and an appliance mode are deployment types for network devices that

- determine whether the device operates as a single instance or supports multiple, independent instances,
- offer distinct management and configuration options at both the chassis and instance levels, and
- provide different approaches to resource allocation, interface assignment, and monitoring.

You can run the device in either appliance mode or multi-instance mode.

Appliance mode

Appliance mode is the default. The device runs the native Firewall Threat Defense image and acts as a single device. The only chassis-level configuration available (on the **Chassis Manager** page) is for network module management (breakout ports or enabling/disabling a network module).

Multi-instance mode

If you change to multi-instance mode, the device runs the Secure Firewall eXtensible Operating System (FXOS) on the chassis, while each instance runs separate Firewall Threat Defense images. You can configure the mode using the FXOS CLI.

Because multiple instances run on the same chassis, you need to perform chassis-level management of:

- CPU and memory resources using resource profiles.
- Interface configuration and assignment.
- Deployment and monitoring of instances.

For a multi-instance device, you add the *chassis* to the Firewall Management Center and configure chassis-level settings on the **Chassis Manager** page.

Chassis management interface

A chassis management interface is a network hardware feature that

- provides a dedicated pathway for device management and control,
- supports multi-instance modes by allowing each software instance its own management IP address, and
- separates management, eventing, and application access to streamline security, access, and monitoring.

Chassis management

The chassis uses the dedicated Management interface on the device. Multi-instance mode does not support using a data interface for chassis management or DHCP addressing for the Management interface.

You can only configure the chassis Management interface at the Firewall Threat Defense CLI (at initial setup) or FXOS CLI (after you convert to multi-instance mode). Refer to Enable multi-instance mode, on page 17 for initial setup. Refer to Change chassis management settings at the FXOS CLI, on page 58 to change Management interface settings in multi-instance mode.

Note By default, SSH is not allowed to this interface in multi-instance mode unless you enable the SSH server and an SSH access list. This difference means that you can connect to the application-mode threat defense Management interface using SSH, but after you convert to multi-instance mode, you can no longer connect using SSH by default. For more information, refer to Configure SSH and SSH access list, on page 43.

Instance management

All instances share the chassis management interface, and each instance has its own IP address on the Management network. After you add the instance and specify the IP address, you can make changes to the network settings at the Firewall Threat Defense CLI.

The instance Management IP address allows SSH by default.

Instance interfaces

A chassis interface is a network connection point that

- allows flexible assignment to instances through physical connections or VLAN subinterfaces,
- enables sharing (physical or VLAN) between multiple instances for efficient resource use, and
- restricts certain interface features (like subinterfaces or sharing) in multi-instance clusters, except for control links.

To provide flexible physical interface use for instances, you can create VLAN subinterfaces on the chassis and also share interfaces (VLAN or physical) between multiple instances. Refer to Shared interface scalability, on page 5 and Configure a subinterface, on page 29.



Note This chapter discusses *chassis* VLAN subinterfaces only. You can separately create subinterfaces within the Firewall Threat Defense instance. For more information, refer to Chassis interfaces vs. instance interfaces, on page 3.

Interface types

An interface type is a network configuration element that

- determines how physical interfaces, VLAN subinterfaces, and EtherChannel interfaces are used in deployment, and
- defines whether interfaces can be shared among instances or dedicated to specific traffic.

Physical interfaces, VLAN subinterfaces, and EtherChannel interfaces can be one of these types:

- Data: Use for regular data or the failover link. Data interfaces cannot be shared between instances, and instances cannot communicate over the backplane to other instances. For traffic on Data interfaces, all traffic must exit the chassis on one interface and return on another interface to reach another instance. You can add VLAN subinterfaces to a data interface to provide separate failover links per High Availability pair.
- Data-sharing: Use for regular data. These data interfaces can be shared by one or more instances. Each instance can communicate over the backplane with all other instances that share this interface. Shared interfaces can affect the number of instances you can deploy. Shared interfaces are not supported for bridge group member interfaces (in transparent mode or routed mode), inline sets, passive interfaces, or failover links.

Chassis interfaces vs. instance interfaces

Chassis interfaces vs. instance interfaces is an interface management model that

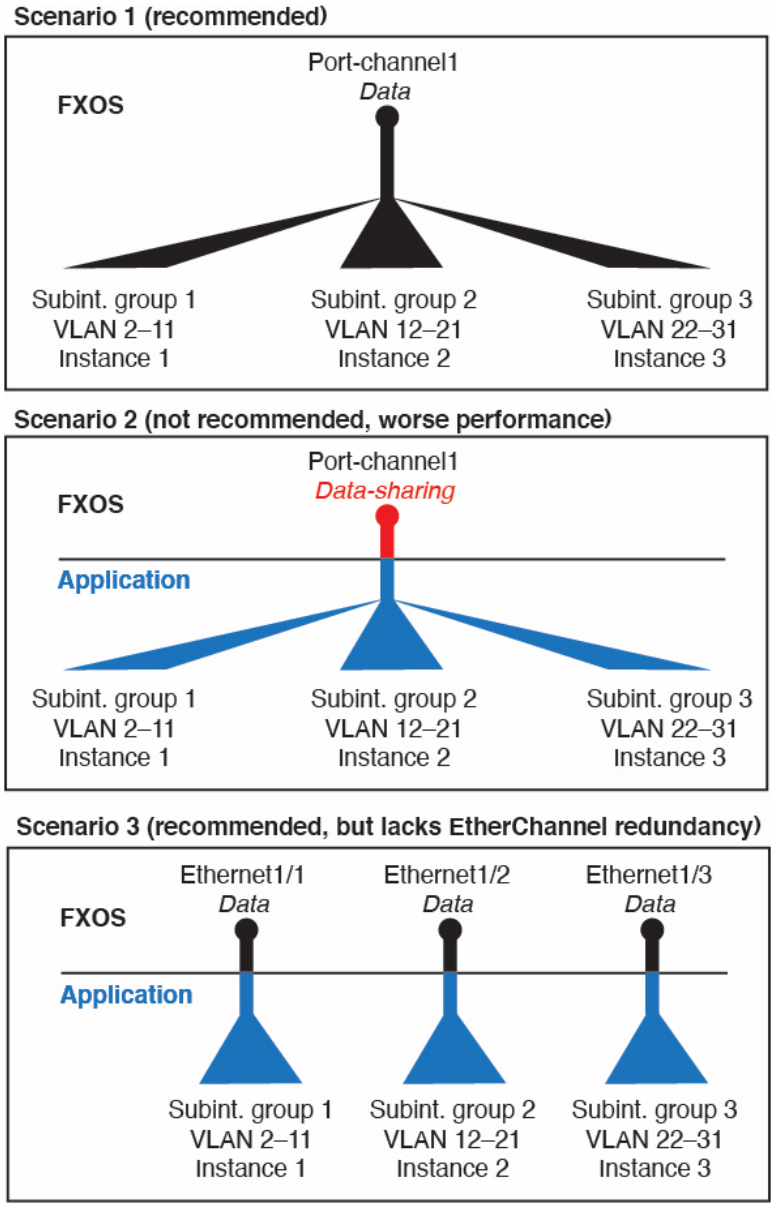
- manages basic Ethernet settings of physical interfaces, VLAN subinterfaces for instances, and EtherChannel interfaces at the chassis level,
- configures higher level settings within the instance, such as IP addresses for EtherChannels, and
- allows independent administrative control of interface states in both chassis and instance locations.

VLAN subinterfaces

You can create VLAN subinterfaces within the instance, just as you would for any device.

You can *also* create VLAN subinterfaces in the chassis. The instance-defined subinterfaces are not subject to the chassis limit. Choosing in which location to create subinterfaces depends on your network deployment and personal preference. For example, to share a subinterface, you must create the subinterface on the chassis. Another scenario that favors chassis subinterfaces comprises allocating separate subinterface groups on a single interface to multiple instances. For example, you want to use Port-channel1 with VLAN 2–11 on instance A, VLAN 12–21 on instance B, and VLAN 22–31 on instance C. If you create these subinterfaces in the instance, then you would have to share the parent interface in the chassis, which may not be desirable. See this illustration that shows the three ways you can accomplish this scenario:

Figure 1: VLANs in the chassis vs. the instance



Independent interface states in the chassis and in the instance

You can administratively enable and disable interfaces in both the chassis and in the instance. For an interface to be operational, the interface must be enabled in both locations. Because the interface state is controlled independently, you may have a mismatch between the chassis and instance.

The default state of an interface within the instance depends on the type of interface. For example, the physical interface or EtherChannel is disabled by default within the instance, but a subinterface is enabled by default.

Shared interface scalability

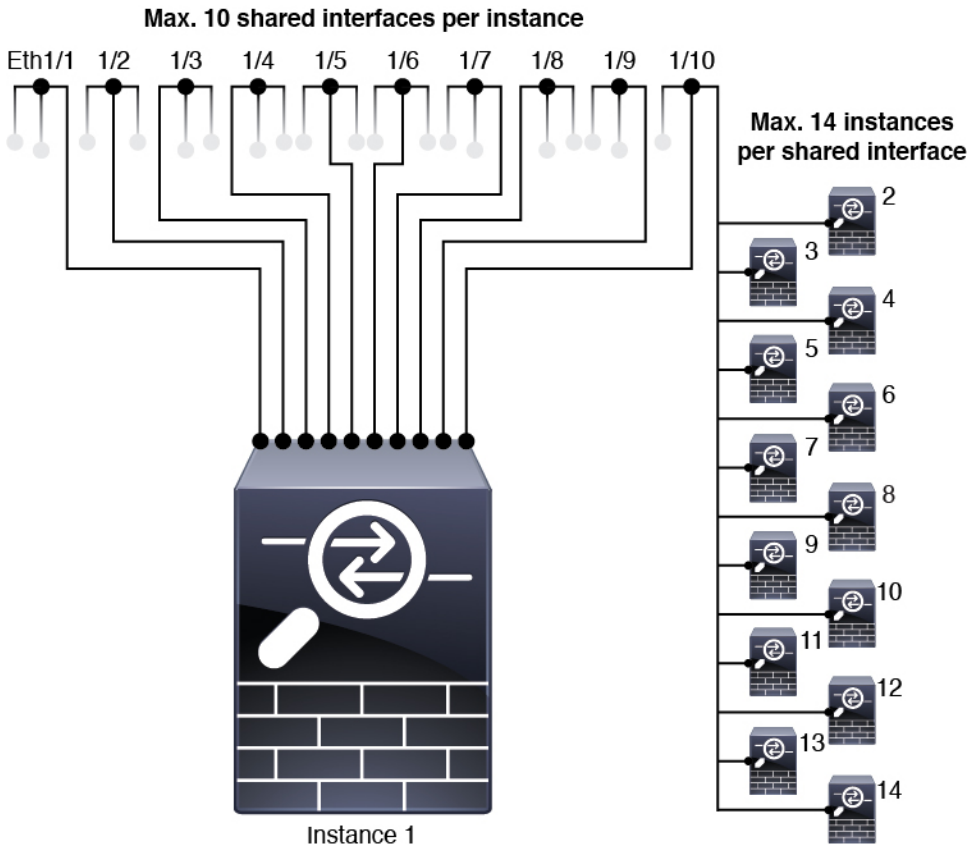
A shared interface is a network resource that

- allows multiple logical connections to transmit data concurrently,
- adapts bandwidth allocation based on real-time demand, and
- enhances scalability to support growing numbers of devices and traffic flows.

Instances can share data-sharing type interfaces. This capability lets you conserve physical interface usage as well as support flexible networking deployments. When you share an interface, the chassis uses unique MAC addresses to forward traffic to the correct instance. However, shared interfaces can cause the forwarding table to grow large due to the need for a full mesh topology within the chassis (every instance must be able to communicate with every other instance that is sharing the same interface). Therefore, there are limits to how many interfaces you can share.

In addition to the forwarding table, the chassis maintains a VLAN group table for VLAN subinterface forwarding. You can create up to 500 VLAN subinterfaces.

See the following limits for shared interface allocation:



Shared interface best practices

This topic outlines the best practices to follow when configuring shared interfaces to maximize performance and security.

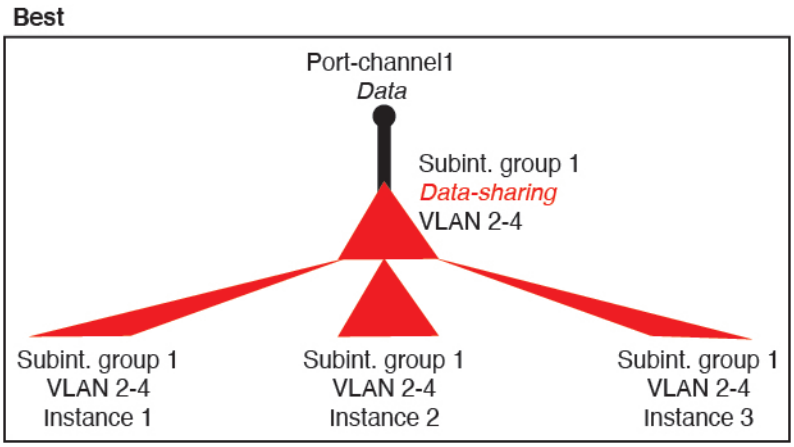
For optimal scalability of the forwarding table, share as few interfaces as possible. Instead, you can create up to 500 VLAN subinterfaces on one or more physical interfaces and then divide the VLANs among the container instances.

When sharing interfaces, follow these practices in the order of most scalable to least scalable:

1. Best: Share subinterfaces under a single parent, and use the same set of subinterfaces with the same group of instances.

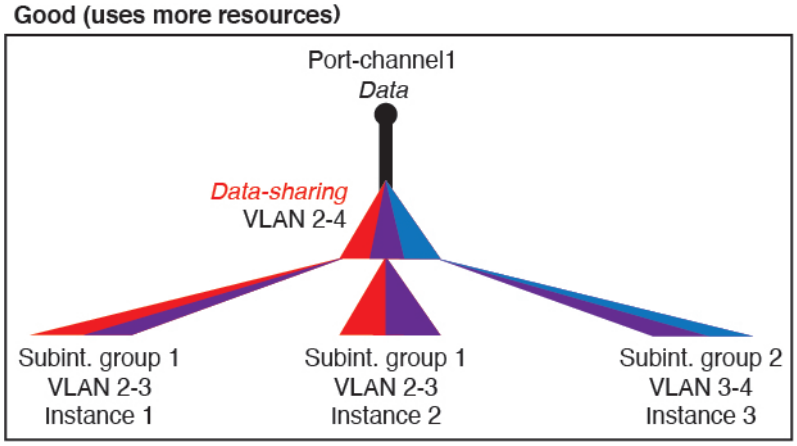
   For example, create a large EtherChannel to bundle all of your like-kind interfaces together, and then share subinterfaces of that EtherChannel: Port-Channel1.2, 3, and 4 instead of Port-Channel2, Port-Channel3, and Port-Channel4. When you share subinterfaces from a single parent, the VLAN group table provides better scaling of the forwarding table than when sharing physical/EtherChannel interfaces or subinterfaces across parents.

Figure 2: Best: Shared Subinterface Group on One Parent



If you do not share the same set of subinterfaces with a group of instances, your configuration can cause more resource usage (more VLAN groups). For example, share Port-Channel1.2, 3, and 4 with instances 1, 2, and 3 (one VLAN group) instead of sharing Port-Channel1.2 and 3 with instances 1 and 2, while sharing Port-Channel1.3 and 4 with instance 3 (two VLAN groups).

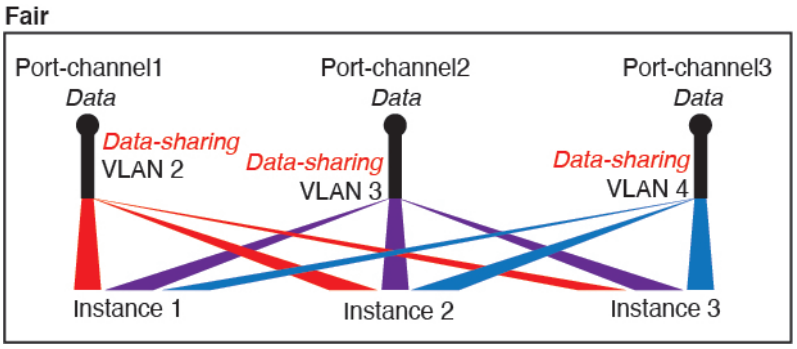
Figure 3: Good: Sharing Multiple Subinterface Groups on One Parent



2. Fair: Share subinterfaces across parents.

   For example, share Port-Channel1.2, Port-Channel2.3, and Port-Channel3.4 instead of Port-Channel2, Port-Channel4, and Port-Channel4. Although this usage is not as efficient as only sharing subinterfaces on the same parent, it still takes advantage of VLAN groups.

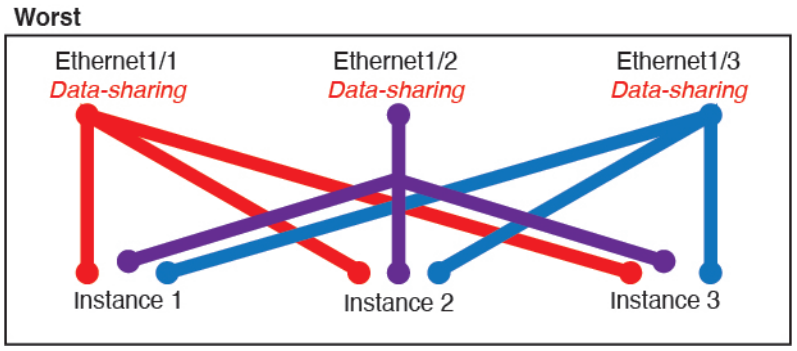
Figure 4: Fair: Shared Subinterfaces on Separate Parents



3. Worst: Share individual parent interfaces (physical or EtherChannel).

   This method uses the most forwarding table entries.

Figure 5: Worst: Shared Parent Interfaces



How the chassis classifies packets

The chassis processes incoming and outgoing packets as they pass through a device, ensuring that each packet is handled according to its type, destination, and security requirements.

Each packet that enters the chassis must be classified so that the chassis can determine to which instance to send a packet.

- Unique Interfaces: If only one instance is associated with the ingress interface, the chassis classifies the packet into that instance. For bridge group member interfaces (in transparent mode or routed mode), inline sets, or passive interfaces, this method is used to classify packets at all times.
- Unique MAC Addresses: The chassis automatically generates unique MAC addresses for all interfaces, including shared interfaces. If multiple instances share an interface, then the classifier uses unique MAC addresses assigned to the interface in each instance. An upstream router cannot route directly to an instance without unique MAC addresses. You can also set the MAC addresses manually when you configure each interface within the application.

Note If the destination MAC address is a multicast or broadcast MAC address, the packet is duplicated and delivered to each instance.

Classification examples

This reference provides examples to help you understand different classification scenarios and their implementation.

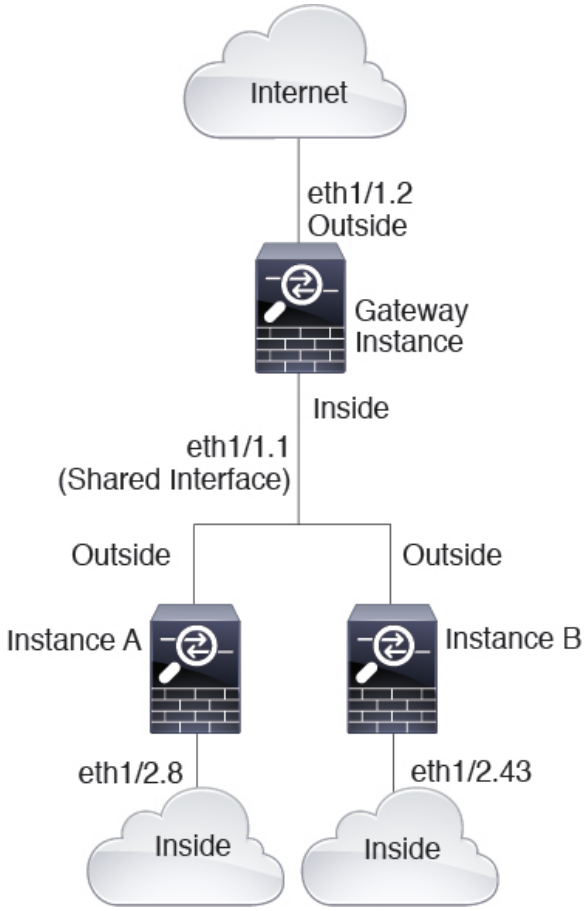
Cascading container instances

Cascading instances connect one instance's outside interface to another's inside interface, simplifying configuration by sharing parameters in the top instance. Avoid using shared interfaces with High Availability to prevent MAC address conflicts; use unique interfaces and an external switch instead.

Placing an instance directly in front of another instance is called *cascading instances*; the outside interface of one instance is the same interface as the inside interface of another instance. You might want to cascade instances if you want to simplify the configuration of some instances by configuring shared parameters in the top instance.

The following figure shows a gateway instance with two instances behind the gateway.

Figure 6: Cascading Instances



Note Do not use cascading instances (using a shared interface) with High Availability. After a failover occurs and the standby unit rejoins, MAC addresses can overlap temporarily and cause an outage. You should instead use unique interfaces for the gateway instance and inside instance using an external switch to pass traffic between the instances.

Typical multi-instance deployments

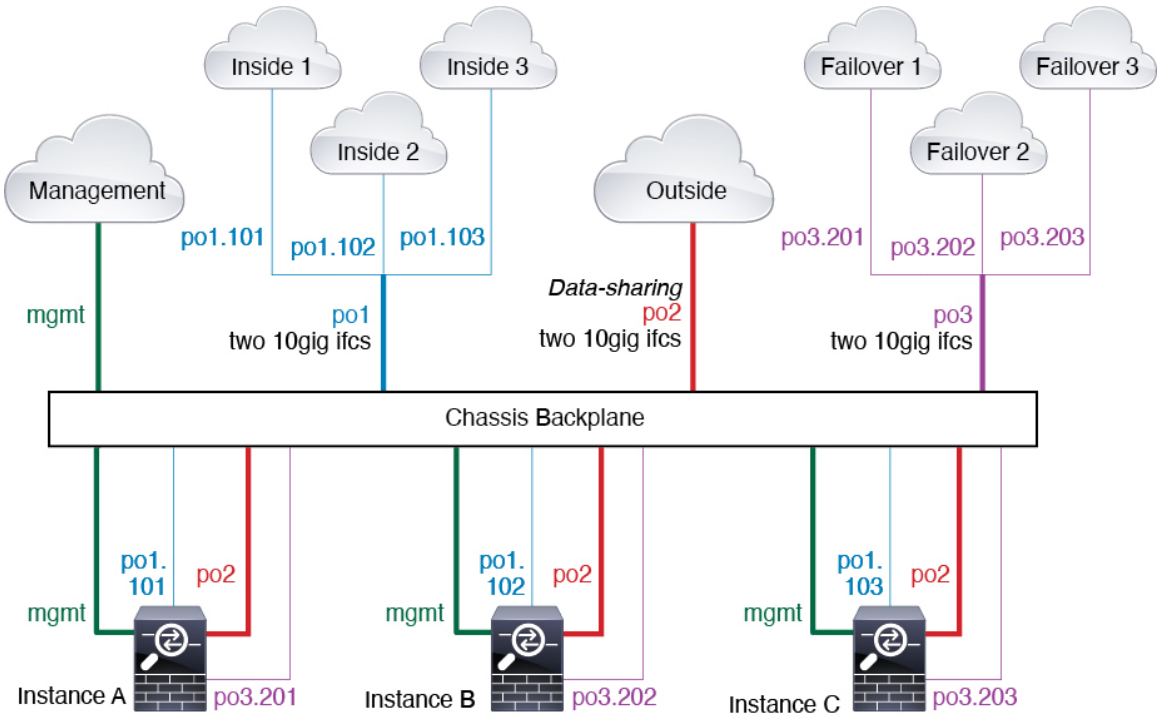
A typical multi-instance deployment is a firewall configuration that

- supports multiple container instances operating simultaneously,
- configures distinct interfaces for management, inside, outside, and failover connectivity, and
- utilizes dedicated subinterfaces and EtherChannels to enhance performance and reliability.

The following example includes three container instances in routed firewall mode. They include the following interfaces:

- Management: All instances and the chassis use the dedicated Management interface. Within each instance (and the chassis), the interface uses a unique IP address on the same management network.
- Inside: Each instance uses a subinterface on Port-Channel1 (data type). This EtherChannel includes two 10 Gigibit Ethernet interfaces. Each subinterface is on a separate network.
- Outside: All instances use the Port-Channel2 interface (data-sharing type). This EtherChannel includes two 10 Gigibit Ethernet interfaces. Within each application, the interface uses a unique IP address on the same outside network.
- Failover: Each instance uses a subinterface on Port-Channel3 (data type). This EtherChannel includes two 10 Gigibit Ethernet interfaces. Each subinterface is on a separate network.

Figure 7: Typical multi-instance deployment



Automatic MAC addresses for instance interfaces

An automatic MAC address for instance interfaces is a network identifier that is automatically generated and assigned to instance interfaces in multi-instance deployments.

The chassis automatically generates MAC addresses for instance interfaces, and guarantees that a shared interface in each instance uses a unique MAC address.

If you manually assign a MAC address to a shared interface within the instance, then the manually-assigned MAC address is used. If you later remove the manual MAC address, the autogenerated address is used. In the rare circumstance that the generated MAC address conflicts with another private MAC address in your network, we suggest that you manually set the MAC address for the interface within the instance.

Because autogenerated addresses start with A2, you should not start manual MAC addresses with A2 due to the risk of overlapping addresses.

The chassis generates the MAC address using the following format:

A2*xx.yyzz.zzzz*

Where *xx.yy* is a user-defined prefix or a system-defined prefix, and *zz.zzzz* is an internal counter generated by the chassis. The system-defined prefix matches the lower 2 bytes of the first MAC address in the burned-in MAC address pool that is programmed into the IDPROM. Use **connect fxos**, then **show module** to view the MAC address pool. For example, if the range of MAC addresses shown for module 1 is b0aa.772f.f0b0 to b0aa.772f.f0bf, then the system prefix will be f0b0.

The user-defined prefix is an integer that is converted into hexadecimal. For an example of how the user-defined prefix is used, if you set a prefix of 77, then the chassis converts 77 into the hexadecimal value 004D (*yyxx*). When used in the MAC address, the prefix is reversed (*xxyy*) to match the chassis native form:

A2**4D.00***zz.zzzz*

For a prefix of 1009 (03F1), the MAC address is:

A2**F1.03***zz.zzzz*

Performance scaling factors for multi-instance modes

A performance scaling factor is a throughput adjustment metric that

- enables calculation of effective throughput in multi-instance appliance modes,
- considers the percentage of CPU cores assigned to each instance, and
- accounts for efficiency reductions compared to single-instance (appliance mode) operation.

The maximum throughput (connections, VPN sessions) for a platform is calculated for an appliance mode device's use of memory and CPU (and this value is shown in **show resource usage**). If you use multiple instances, then you need to calculate the throughput based on the percentage of CPU cores that you assign to the instance. For example, if you use an instance with 50% of the cores, then you should initially calculate 50% of the throughput. Moreover, the throughput available to an instance may be less than that available to an appliance.

For detailed instructions on calculating the throughput for instances, refer to https://www.cisco.com/c/en/us/products/collateral/security/firewalls/white-paper-c11-744750.html.

Instances and high availability

A high availability pair is a deployment option that

- uses two instances on separate chassis to ensure service continuity,
- allows for multiple high availability pairs to be created from available instances, and
- permits standalone instances to exist alongside high availability pairs within the same chassis.

You can use High Availability using an instance on 2 separate chassis; for example, if you have 2 chassis, each with 10 instances, you can create 10 High Availability pairs. You can also have standalone instances on the same chassis as High Availability instances. For more information about the requirements, refer to Requirements and prerequisites for instances, on page 13.

Note Clustering is not supported.

Licenses for instances

This reference provides information about how licenses are assigned and consumed for chassis and feature instances.

All licenses are consumed per chassis and not per instance. Refer to these details.

- Essentials licenses are assigned to the chassis as a whole, one per chassis.
- Feature licenses are assigned to each instance, but you only *consume* one license per feature per chassis.

Requirements and prerequisites for instances

This reference provides the requirements and prerequisites for deploying instances on Secure Firewall models, including supported models, upgrade paths, resource limitations per model, software requirements, high availability requirements, and management center requirements.

Model support

- Secure Firewall 3110
- Secure Firewall 3120
- Secure Firewall 3130
- Secure Firewall 3140

Note The Secure Firewall 3105 is not supported.

Maximum container instances and resources per model

For each container instance, you can specify the number of CPU cores (or more specifically, threads) to assign to the instance. We use the term "core" generically to account for different hardware architecture. RAM is dynamically allocated according to the number of cores, and disk space is set to 40 GB per instance.

Table 1: Maximum container instances and resources per model

Model	Max. Container Instances	Available CPU Cores (Threads)
Secure Firewall 3110	3	22
Secure Firewall 3120	5	30

Model	Max. Container Instances	Available CPU Cores (Threads)
Secure Firewall 3130	7	46
Secure Firewall 3140	10	62

Software requirements

- You can run different versions of Firewall Threat Defense software on each instance as long as they are all compatible with the version of FXOS running on the chassis.
- You cannot deploy an instance with a patch version of the Firewall Threat Defense software because it is not a complete bundle. You need to first deploy the major or maintenance version, and then patch it after deployment.

High availability requirements

- The two instances in a High Availability configuration must:
  - Be on separate chassis.
  - Be on the same model.
  - Have the same interfaces assigned. All interfaces must be preconfigured in the chassis identically before you enable High Availability.
  - Use the same resource profile attributes. The profile names can be different, but the definitions need to match.

Firewall Management Center requirements

For chassis management and all instances on the chassis, you must use the same Firewall Management Center due to the licensing implementation.

Guidelines and limitations for instances

This reference provides guidelines and limitations for managing instances on chassis, including general requirements, interface configurations, VLAN subinterfaces, EtherChannels, data-sharing interfaces, and MAC address considerations.

General guidelines

- A single Firewall Management Center must manage all instances on a chassis, as well as manage the chassis itself.
- For instances, these features are not supported:
  - TLS crypto acceleration
  - Clustering
  - Firewall Management Center UCAPL/CC mode

- Flow offload to hardware
- LLDP

- Primary management of the chassis by Security Cloud Control cloud-delivered Firewall Management Center and separate analytics-only management of the chassis by an on-prem Firewall Management Center is not supported. You can however add Security Cloud Control-managed instances to an analytics-only on-prem Firewall Management Center.

Management interface

- No support for a data interface for chassis management; only the dedicated Management interface can be used
- No DHCP addressing for the Management interface
- The management connection requires TCP port 8305; do not change the value from the default.

VLAN subinterfaces

- This document discusses *chassis* VLAN subinterfaces only. You can separately create subinterfaces within the instance.
- If you assign a parent interface to an instance, it only passes untagged (non-VLAN) traffic. Do not assign the parent interface unless you intend to pass untagged traffic.
- Subinterfaces are supported on Data or Data-sharing type interfaces.
- You can create up to 500 VLAN IDs.
- You cannot use subinterfaces for an inline set or as a passive interface.
- If you use a subinterface for the failover link, then all subinterfaces on that parent, and the parent itself, are restricted for use as failover links. You cannot use some subinterfaces as failover links, and some as regular data interfaces.

EtherChannels

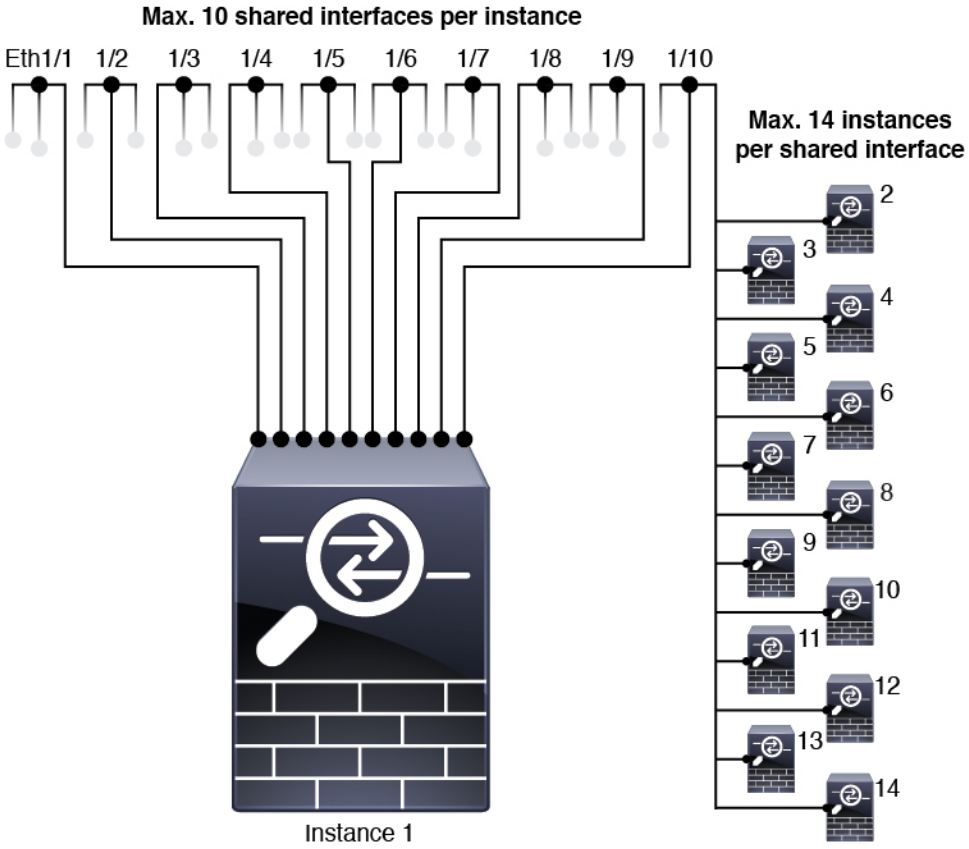
- You can configure up to 48 EtherChannels, limited by the number of physical interfaces.
- The EtherChannel can have up to 8 active interfaces.
- All interfaces in the EtherChannel must be the same media type and speed capacity. The media type can be either RJ-45 or SFP; SFPs of different types (copper and fiber) can be mixed. You cannot mix interface capacities (for example 1GB and 10GB interfaces) by setting the speed to be lower on the larger-capacity interface, unless you set the speed to **Detect SFP**; in this case, you can use different interface capacities, and the lowest common speed is used.
- The chassis does not support LACPDUs that are VLAN-tagged. If you enable native VLAN tagging on the neighboring switch using the Cisco IOS **VLAN dot1Q tag native** command, then the chassis will drop the tagged LACPDUs. Be sure to disable native VLAN tagging on the neighboring switch.
- In Cisco IOS software versions earlier than 15.1(1)S2, the chassis did not support connecting an EtherChannel to a switch stack. With default switch settings, if the chassis EtherChannel is connected cross stack, and if the primary switch is powered down, then the EtherChannel connected to the remaining

switch will not come up. To improve compatibility, set the **stack-MAC persistent timer** command to a large enough value to account for reload time; for example, 8 minutes or 0 for indefinite. Or, you can upgrade to more a more stable switch software version, such as 15.1(1)S2.

Data-sharing interfaces

- Maximum 14 instances per shared interface. For example, you can allocate Ethernet1/1 to Instance1 through Instance14.

  Maximum 10 shared interfaces per instance. For example, you can allocate Ethernet1/1.1 through Ethernet1/1.10 to Instance1.



- You cannot use a data-sharing interface with a transparent firewall mode instance.
- You cannot use a data-sharing interface with inline sets or passive interfaces.
- You cannot use a data-sharing interface for the failover link.

Default MAC addresses

- MAC addresses for all interfaces are taken from a MAC address pool. For subinterfaces, if you decide to manually configure MAC addresses, make sure you use unique MAC addresses for all subinterfaces on the same parent interface to ensure proper classification. For more information, refer to Automatic MAC addresses for instance interfaces, on page 11.

Instance configuration

Prepare the chassis and management environment to enable instance configuration on your device.

Before you configure instances, you need to enable multi-instance mode, add the chassis to the Firewall Management Center, and configure chassis interfaces. You can also customize the chassis settings.

Enable multi-instance mode

You need to connect to the Firewall Threat Defense CLI at the console port to enable multi-instance mode. After you configure the mode, you can add it to the Firewall Management Center.

Note Although you can connect to SSH on the management port, we recommend using the console port to avoid multiple disconnections. This procedure covers the console port.

Procedure

Step 1 Connect to the chassis console port.

The console port connects to the FXOS CLI.

Step 2 Log in with the username **admin** and the password **Admin123**.

The first time you log in to FXOS, you are prompted to change the password.

Note
If the password was already changed, and you do not know it, you must reimage the device to reset the password to the default. Refer to the FXOS troubleshooting guide for the reimage procedure.

Example:

```
firepower login: admin
Password: Admin123
Successful login attempts for user 'admin' : 1

[...]

Hello admin. You must change your password.
Enter new password: ********
Confirm new password: ********
Your password was updated successfully.

[...]

firepower#
```

Step 3 Check your current mode, Native or Container. If the mode is Native, you can continue with this procedure to convert to multi-instance (Container) mode.

show system detail

Example:

```
firepower # show system detail

Systems:
    Name: firepower
    Mode: Stand Alone
    System IP Address: 172.16.0.50
    System IPv6 Address: ::
    System Owner:
    System Site:
    Deploy Mode: Native
    Description for System:
firepower #
```

Step 4 Connect to the Firewall Threat Defense CLI.

connect ftd

Example:

```
firepower# connect ftd
>
```

Step 5 The first time you log in to the Firewall Threat Defense, you are prompted to accept the General Terms. You are then presented with the CLI setup script.

The setup script lets you set the Management interface IP address and other settings. However, when you convert to multi-instance mode, the only settings that are retained are the following:

- Admin password (that you set at initial login)
- DNS servers
- Search domains

You will reset the Management IP address and gateway as part of the multi-instance mode command. After you convert to multi-instance mode, you can change Management settings at the FXOS CLI. For more information, refer to Change chassis management settings at the FXOS CLI, on page 58.

Step 6 Enable multi-instance mode, set the chassis management interface settings, and identify the Firewall Management Center. You can use IPv4 and/or IPv6 static addressing; DHCP is not supported. After you enter the command, you will be prompted to erase the configuration and reboot. Enter **ERASE** (all caps). The system reboots and, as part of changing the mode, erases the configuration with the exception of the Management network settings you set in the command and the admin password. The chassis hostname is set to "firepower-*model*."

IPv4:

configure multi-instance network ipv4 *ip_address network_mask gateway_ip_address* **manager** *manager_name* {*hostname* | *ipv4_address* | **DONTRESOLVE**} *registration_key nat_id*

IPv6:

configure multi-instance network ipv6 *ipv6_address prefix_length gateway_ip_address* **manager** *manager_name* {*hostname* | *ipv6_address* | **DONTRESOLVE**} *registration_key nat_id*

See the following **manager** components:

- {*hostname* | *ipv4_address* | **DONTRESOLVE**}: Specifies either the FQDN or IP address of the Firewall Management Center. At least one of the devices, either the Firewall Management Center or the chassis, must have a reachable IP address to establish the two-way, SSL-encrypted communication channel between the two devices. If you don't specify a manager hostname or IP address in this command, then enter **DONTRESOLVE**; in this case, the chassis must have a reachable IP address or hostname, and you must specify the *nat_id*.
- *registration_key*: Enter a one-time registration key of your choice that you will also specify on the Firewall Management Center when you register the chassis. The registration key must not exceed 37 characters. Valid characters include alphanumerical characters (A–Z, a–z, 0–9) and the hyphen (-).
- *nat_id*: Specifies a unique, one-time string of your choice that you will also specify on the Firewall Management Center when you register the chassis when one side does not specify a reachable IP address or hostname. It is required if you do not specify a manager address or hostname, however, we recommend that you always set the NAT ID even when you specify a hostname or IP address. The NAT ID must not exceed 37 characters. Valid characters include alphanumerical characters (A–Z, a–z, 0–9) and the hyphen (-). This ID cannot be used for any other devices registering to the Firewall Management Center.

To change the mode back to appliance mode, you must use the FXOS CLI and enter **scope system** and then **set deploymode native**. For more information, refer to Change chassis management settings at the FXOS CLI, on page 58.

Example:

```
> configure multi-instance network ipv4 172.16.0.104 255.255.255.0 172.16.0.1 manager
fmc1 172.16.0.103 impala67 winchester1
WARNING: This command will discard any FTD configuration (except admin's credentials).
Make sure you back up your content. All previous content will be lost. System is going to
be re-initialized.
Type ERASE to confirm:ERASE
Exit...
>
```

Add a multi-instance chassis to the Firewall Management Center

Add the multi-instance chassis to the Firewall Management Center to enable centralized management and configuration of all chassis settings and instances.

The management center and the chassis share a separate management connection using the chassis MGMT interface.

You can use the Firewall Management Center to configure all chassis settings as well as instances. The Secure Firewall Chassis Manager or configuration at the FXOS CLI is not supported.

Before you begin

Convert the chassis to multi-instance mode. For more information, refer to Enable multi-instance mode, on page 17.

Procedure

Step 1 In the Firewall Management Center, add the chassis using the chassis management IP address or hostname.

a) Choose **Devices** > **Device Management**, and then from **Add** drop-down list, choose **Chassis**.

Figure 8: Add chassis

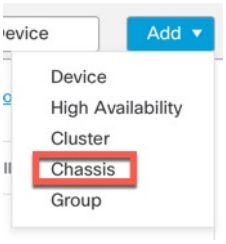
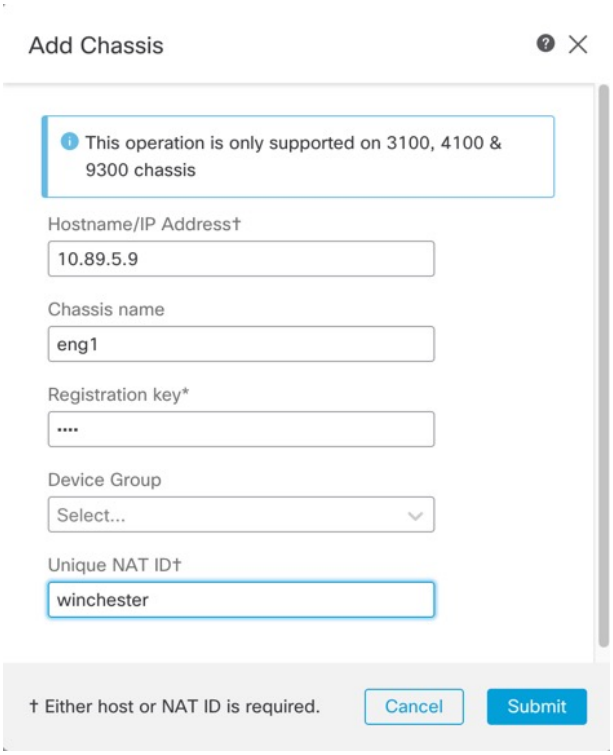


Figure 9: Add chassis



b) In the **Hostname/IP Address** field, enter the IP address or the hostname of the chassis you want to add.

If you don't know the hostname or IP address, you can leave this field blank specify the **Unique NAT ID**.

c) In the **Chassis Name** field, enter a name for the chassis as you want it to display in the Firewall Management Center.

d) In the **Registration Key** field, enter the same registration key that you used when you configured the chassis to be managed by the Firewall Management Center.

The registration key is a one-time-use shared secret. The key can include alphanumeric characters and hyphens (-).

e) In a multidomain deployment, regardless of your current domain, assign the chassis to a leaf **Domain**.

If your current domain is a leaf domain, the chassis is automatically added to the current domain. If your current domain is not a leaf domain, post-registration, you must switch to the leaf domain to configure the chassis. A chassis can only belong to one domain.

f) (Optional) Add the chassis to a **Device Group**.

g) If you used a NAT ID during chassis setup, expand enter the same NAT ID in the **Unique NAT ID** field.

The NAT ID can include alphanumeric characters and hyphens (-).

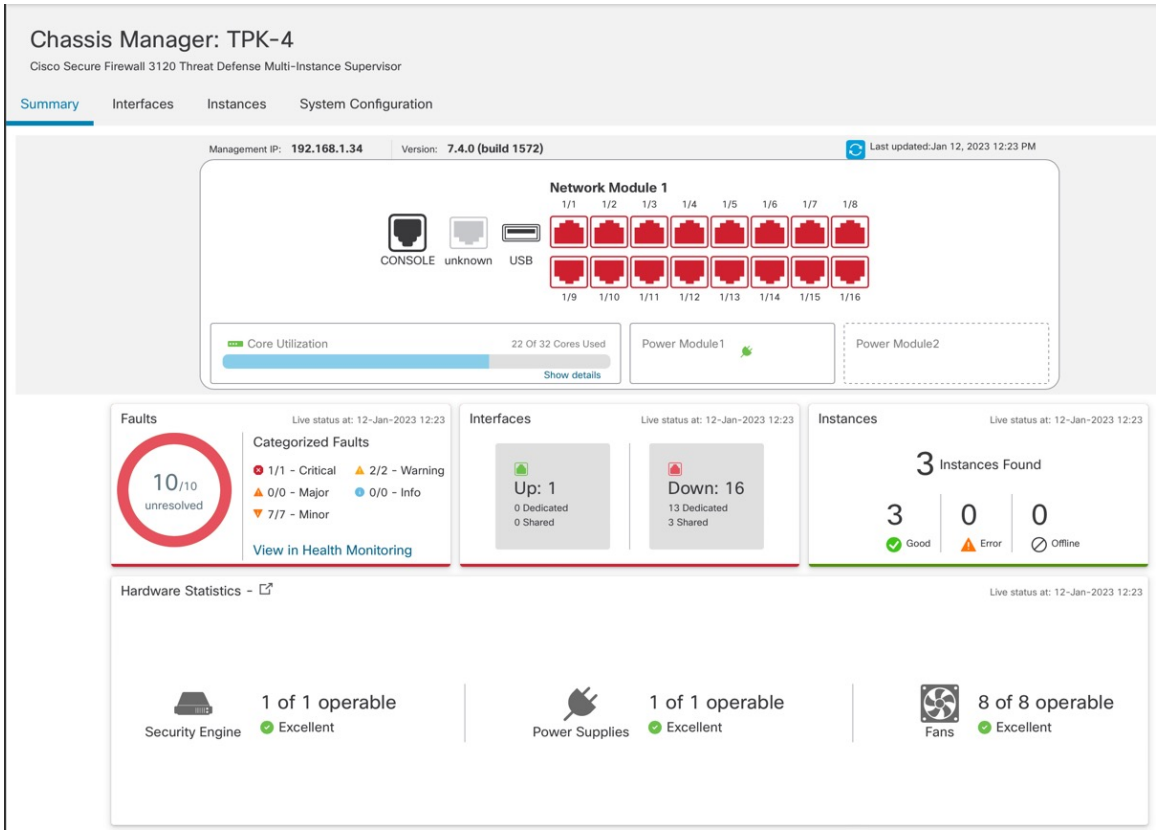
h) Click **Submit**.

The chassis is added to the **Devices** > **Device Management** page.

Step 2 To view and configure the chassis, click **Manage** in the **Chassis** column, or click **Edit** ( ).

The **Chassis Manager** page opens for the chassis to the **Summary** page.

Figure 10: Chassis summary



Chassis interface configuration

A chassis interface is a physical network interface that

- enables connection to the firewall's internal and external networks,
- supports configuration of basic Ethernet settings, VLAN subinterfaces, and EtherChannel grouping, and
- defaults to a disabled state until explicitly enabled by an administrator.

Additional information

Note To configure breakout ports and perform other network module operations, refer to Network module management for Secure Firewall 3100/4200.

Note For information about the **Sync Device** button, refer to Sync Interface Changes with the Firewall Management Center.

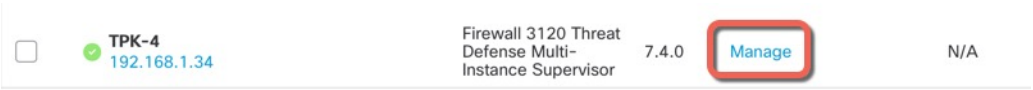
Configure a physical interface

You can physically enable and disable interfaces, as well as set the interface speed and duplex and other hardware settings. To use an interface, it must be physically enabled for the chassis and logically enabled in the instance. By default, physical interfaces are disabled. For VLAN subinterfaces, the admin state is inherited from the parent interface.

Procedure

Step 1 From **Devices** > **Device Management**, click **Manage** in the **Chassis** column or click **Edit** ( ).

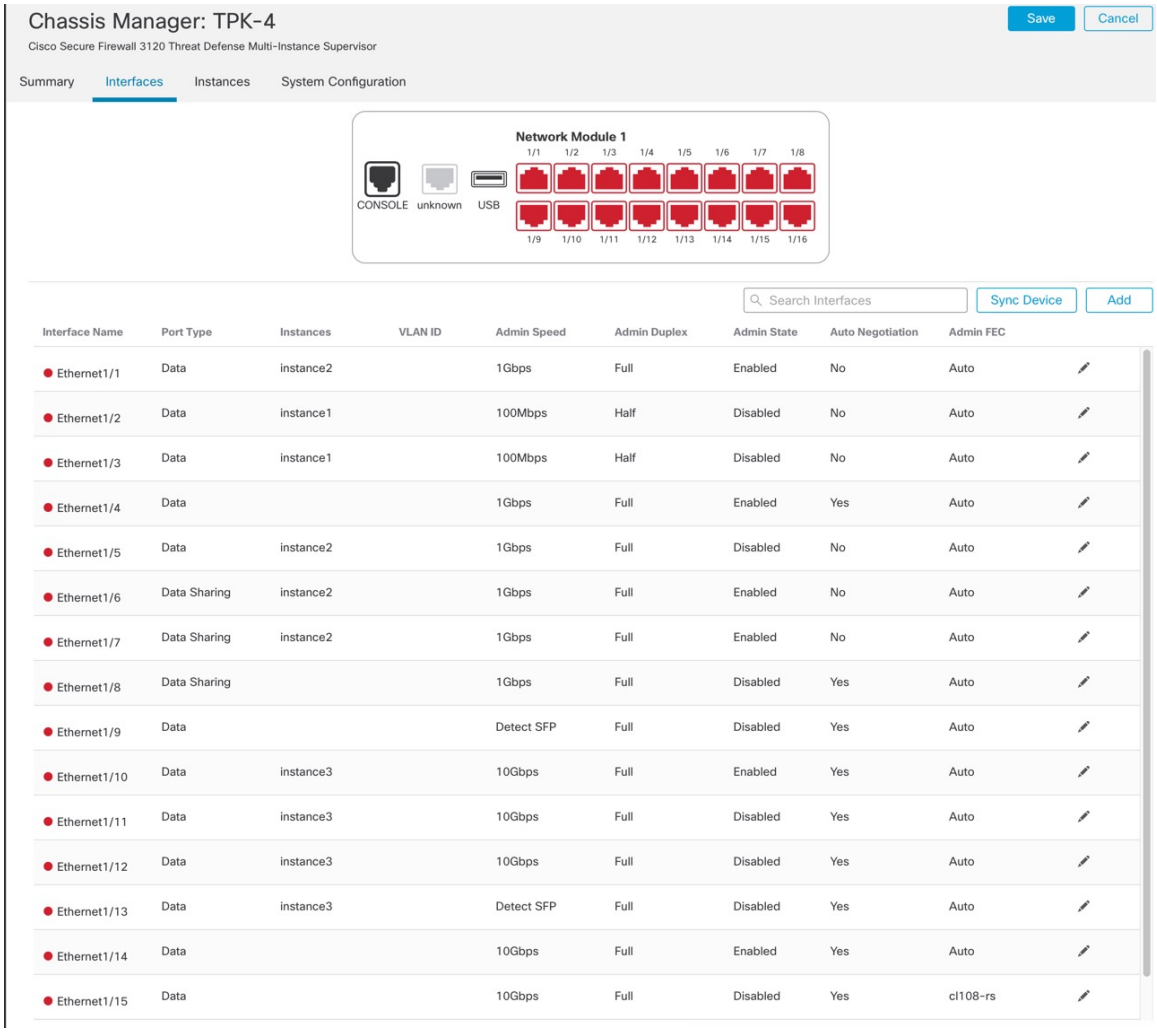
Figure 11: Manage Chassis



The **Chassis Manager** page opens for the chassis to the **Summary** page.

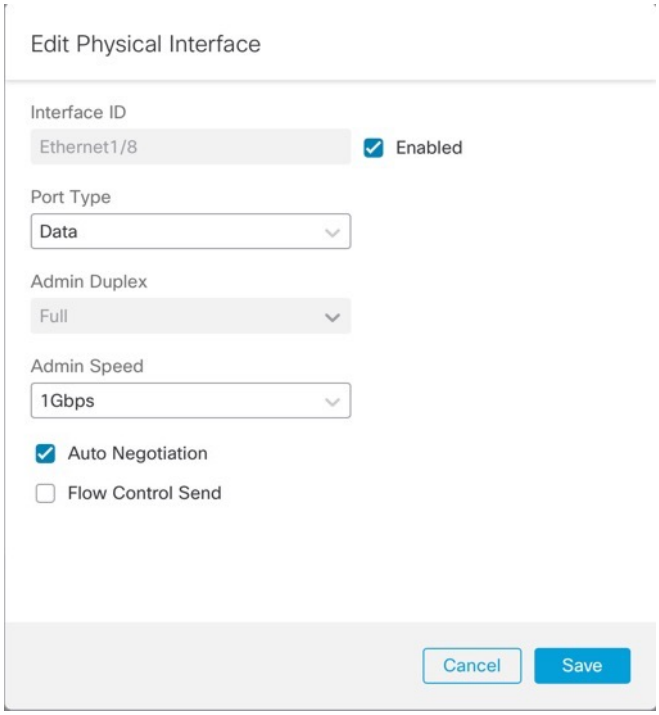
Step 2 Click **Interfaces**.

Figure 12: Interfaces



Step 3 Click **Edit** (✎) for the interface you want to edit.

Figure 13: Edit Physical Interface



Step 4 Enable the interface by checking the **Enabled** check box.

- For the **Port Type**, choose **Data** or **Data Sharing**.

  Figure 14: Port Type

- Set the **Admin Duplex**.

  Speeds of 1Gbps and higher only support **Full** duplex. SFP interfaces only support **Full** duplex.

- Set the **Admin Speed**.

  For SFPs, Choose **Detect SFP** to detect the speed of the installed SFP module and use the appropriate speed. Duplex is always Full, and auto-negotiation is always enabled. This option is useful if you later change the network module to a different model, and want the speed to update automatically.

Step 5 (Optional) Check **Flow Control Send** to enable pause (XOFF) frames for flow control.

Flow control enables connected Ethernet ports to control traffic rates during congestion by allowing congested nodes to pause link operation at the other end. If the threat defense port experiences congestion (exhaustion of queuing resources on the internal switch) and cannot receive any more traffic, it notifies the other port by sending a pause frame to stop sending until the condition clears. Upon receipt of a pause frame, the sending device stops sending any data packets, which prevents any loss of data packets during the congestion period.

Note
The Firewall Threat Defense supports transmitting pause frames so that the remote peer can rate-control the traffic.

However, receiving of pause frames is not supported.

The internal switch has a global pool of 8000 buffers of 250 bytes each, and the switch allocates buffers dynamically to each port. A pause frame is sent out every interface with flow control enabled when the buffer usage exceeds the global high-water mark (2 MB (8000 buffers)); and a pause frame is sent out of a particular interface when its buffer exceeds the port high-water mark (.3125 MB (1250 buffers)). After a pause is sent, an XON frame can be sent when the buffer usage is reduced below the low-water mark (1.25 MB globally (5000 buffers); .25 MB per port (1000 buffers)). The link partner can resume traffic after receiving an XON frame.

Only flow control frames defined in 802.3x are supported. Priority-based flow control is not supported.

Step 6 (Optional) Check **Auto Negotiation** to set the interface to negotiate the speed, link status, and flow control. You cannot edit this setting for speeds lower than 1Gbps. For SFP interfaces, you can only disable auto-negotiation when the speed is set to 1Gbps.

Step 7 Click **Save** and then **Save** in the top right of the **Interfaces** page.

You can now **Deploy** the policy to the chassis. The changes are not active until you deploy them.

Configure an EtherChannel

An EtherChannel (also known as a port channel) can include up to 8 member interfaces of the same media type and capacity, and must be set to the same speed and duplex. The media type can be either RJ-45 or SFP; SFPs of different types (copper and fiber) can be mixed. You cannot mix interface capacities (for example 1GB and 10GB interfaces) by setting the speed to be lower on the larger-capacity interface, unless you set the speed to **Detect SFP**; in this case, you can use different interface capacities, and the lowest common speed is used.

The Link Aggregation Control Protocol (LACP) aggregates interfaces by exchanging the Link Aggregation Control Protocol Data Units (LACPDUs) between two network devices. LACP coordinates the automatic addition and deletion of links to the EtherChannel without user intervention. It also handles misconfigurations and checks that both ends of member interfaces are connected to the correct channel group. "On" mode cannot use standby interfaces in the channel group when an interface goes down, and the connectivity and configurations are not checked.

When the chassis creates an EtherChannel, the EtherChannel stays in a **Suspended** state for Active LACP mode or a **Down** state for On LACP mode until you assign it to a logical device, even if the physical link is up. The EtherChannel will be brought out of this **Suspended** state when it is added to an instance.

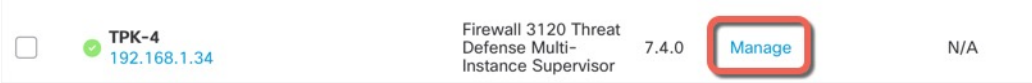
Before you begin

Enable physical interfaces and set hardware parameters. For more information, refer to Configure a physical interface, on page 22.

Procedure

Step 1 From **Devices** > **Device Management**, click **Manage** in the **Chassis** column or click **Edit** ( ).

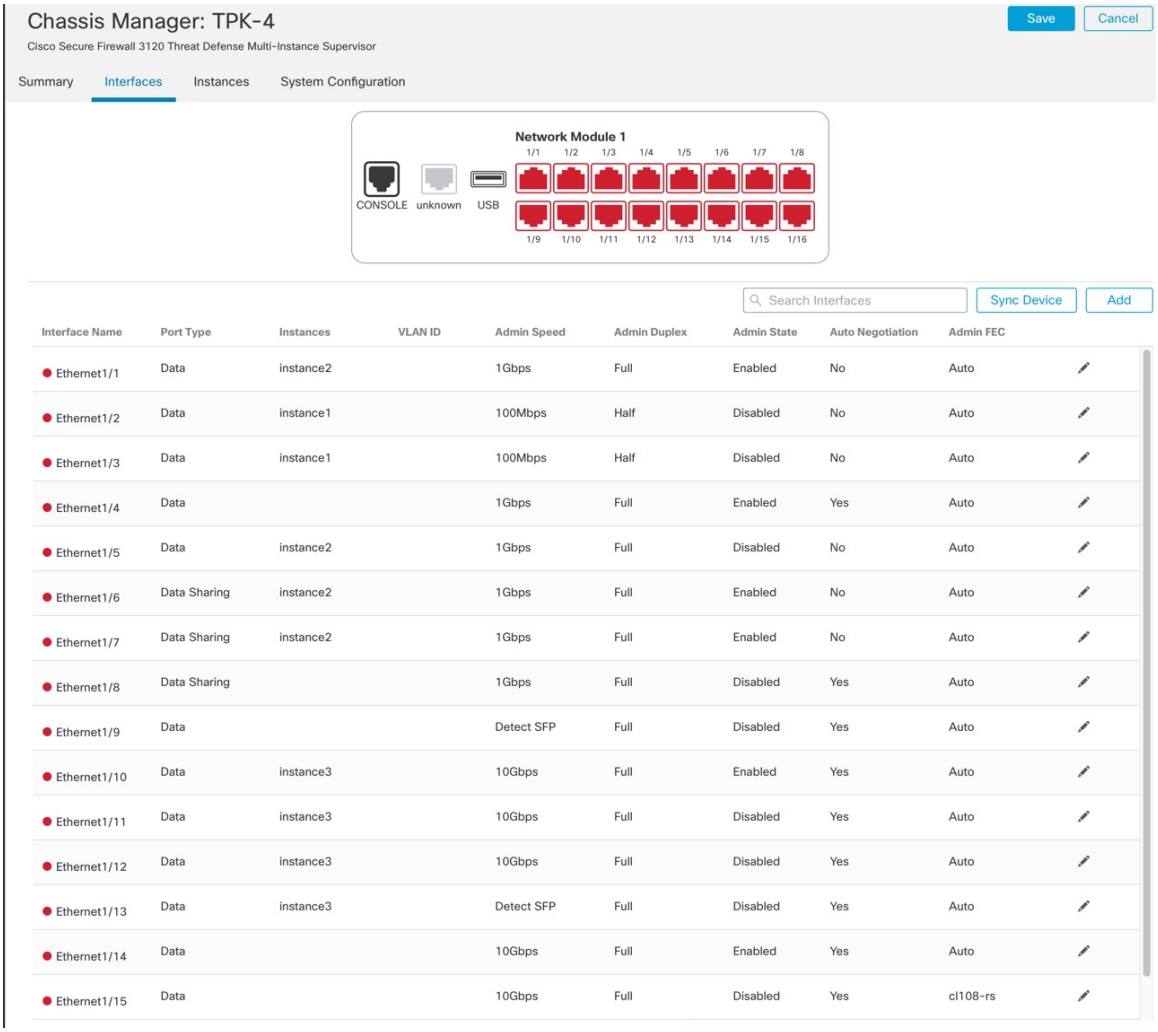
Figure 15: Manage Chassis



The **Chassis Manager** page opens for the chassis to the **Summary** page.

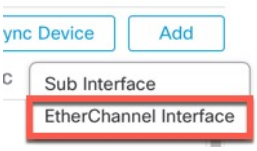
Step 2 Click **Interfaces**.

Figure 16: Interfaces



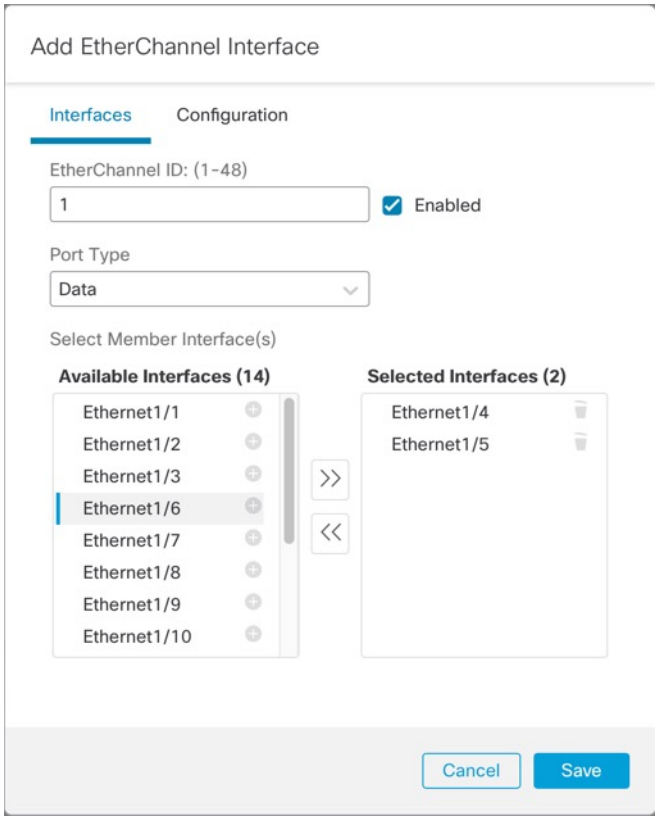
Step 3 Click **Add** > **EtherChannel Interface**.

Figure 17: Add EtherChannel



Step 4 Set the following **Interfaces** parameters.

Figure 18: Interfaces settings



a) For the **EtherChannel ID**, specify an ID between 1 and 48.
b) Check **Enabled**.
c) For the **Port Type**, choose **Data** or **Data Shared**.

   For information about the port type, refer to Interface types, on page 3.

d) To add a physical interface to the EtherChannel, select click **Add** (⊕) in the **Available Interfaces** list to move it to the **Selected Interfaces** list.

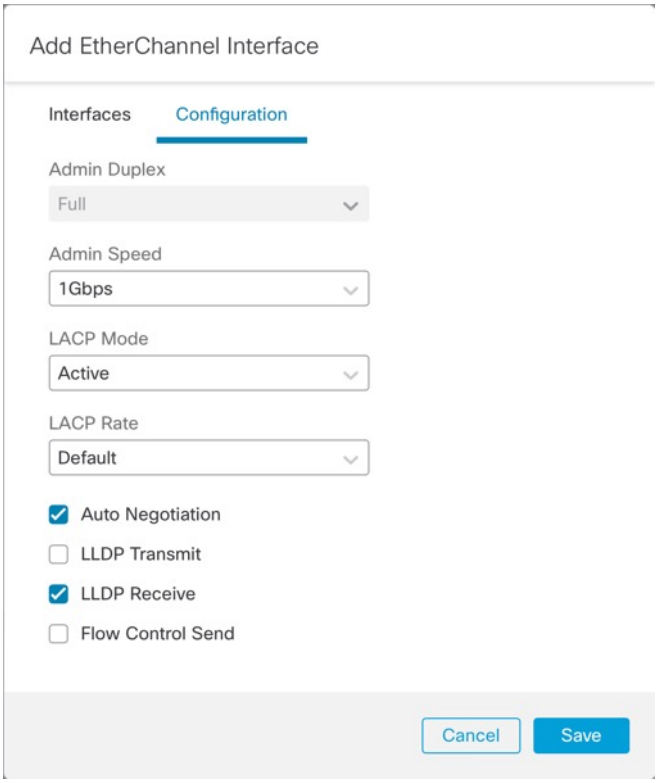
   To add or remove all interfaces, click the double arrow button.

   Note
   You cannot add an interface that is already assigned to an instance.

Step 5 (Optional) Set the following **Configuration** parameters.

Many of these settings (excluding the LACP settings) set the requirements for interfaces to be included in the EtherChannel; they do not override the settings of member interfaces. So if you check **LLDP Transmit**, for example, you should only add interfaces that have that setting. If you set the **Admin Speed** to 1 Gbps, then only 1Gbps interfaces can be included.

Figure 19: Configuration settings



a) Choose the required **Admin Duplex** for the member interfaces, **Full Duplex** or **Half Duplex**.

If you add a member interface that is configured with the specified duplex, it will not successfully join the port channel.

b) Choose the required **Admin Speed** for the member interfaces from the drop-down list.

If you add a member interface that is not at the specified speed, it will not successfully join the port channel.

c) Choose the **LACP Mode**, **Active** or **On**.

- **Active**—Sends and receives LACP updates. An active EtherChannel can establish connectivity with either an active or a passive EtherChannel. You should use the active mode unless you need to minimize the amount of LACP traffic.
- **On**—The EtherChannel is always on, and LACP is not used. An "on" EtherChannel can only establish a connection with another "on" EtherChannel.

Note
It may take up to three minutes for an EtherChannel to come up to an operational state if you change its mode from On to Active or from Active to On.

d) Choose the **LACP Rate**, **Default**, **Fast**, or **Normal**.

The default is **Fast**.

e) Choose the required Link Layer Discovery Protocol (LLDP) settings for member interfaces by checking **LLDP Transmit** and/or **LLDP Receive**.
f) Check the required **Flow Control Send** setting for member interfaces.

Step 6 Click **Save** and then **Save** in the top right of the **Interfaces** page.

You can now **Deploy** the policy to the chassis. The changes are not active until you deploy them.

Configure a subinterface

You can add up to 500 subinterfaces to your chassis.

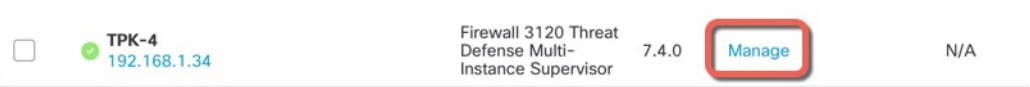
VLAN IDs per interface must be unique, and within an instance, VLAN IDs must be unique across all assigned interfaces. You can reuse VLAN IDs on *separate* interfaces as long as they are assigned to different instances. However, each subinterface still counts towards the limit even though it uses the same ID.

This section discusses *FXOS* VLAN subinterfaces only. You can separately create subinterfaces within the instance. For more information, refer to Chassis interfaces vs. instance interfaces, on page 3.

Procedure

Step 1 From **Devices** > **Device Management**, click **Manage** in the **Chassis** column or click **Edit** ( ).

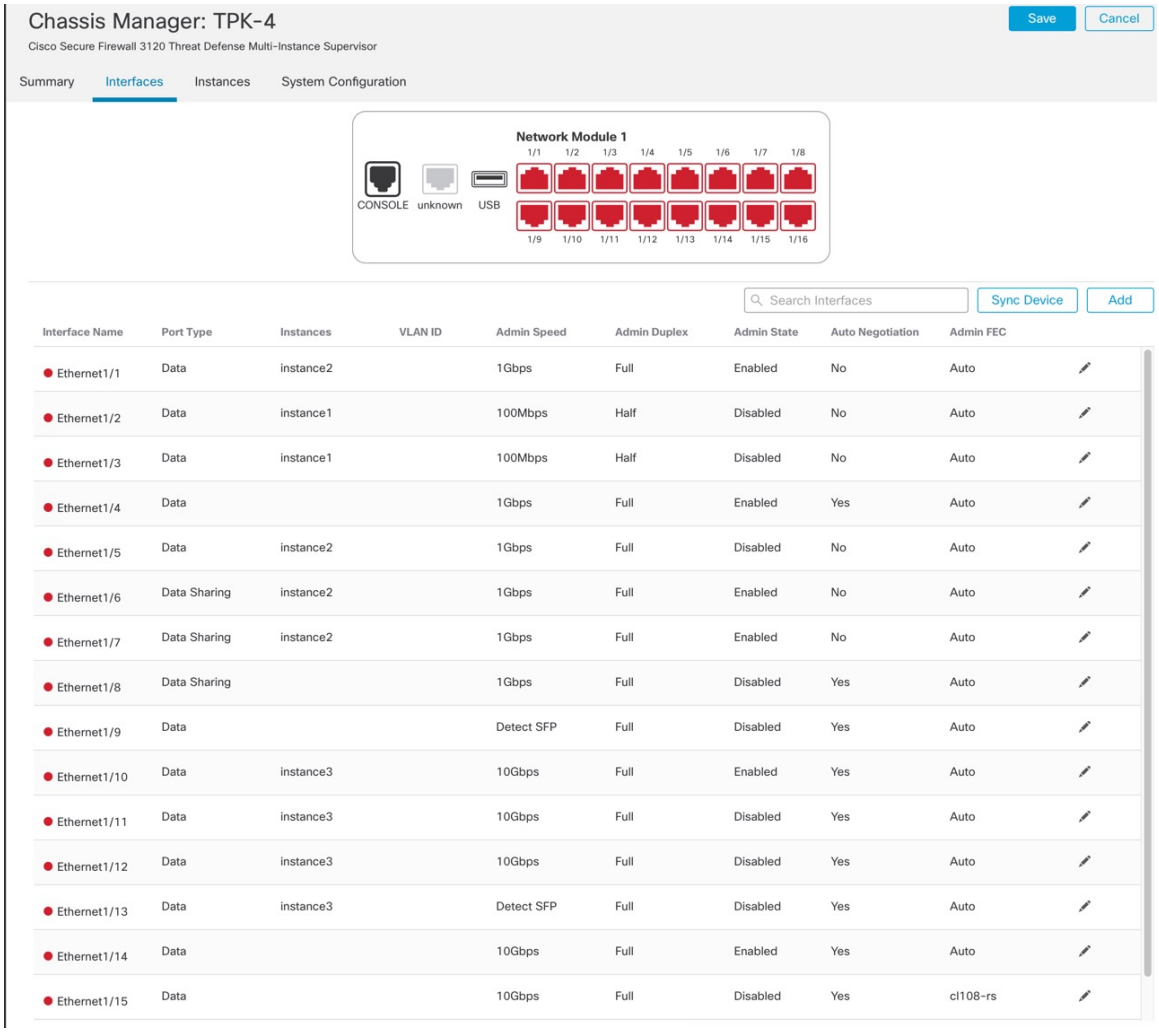
Figure 20: Manage Chassis



The **Chassis Manager** page opens for the chassis to the **Summary** page.

Step 2 Click **Interfaces**.

Figure 21: Interfaces



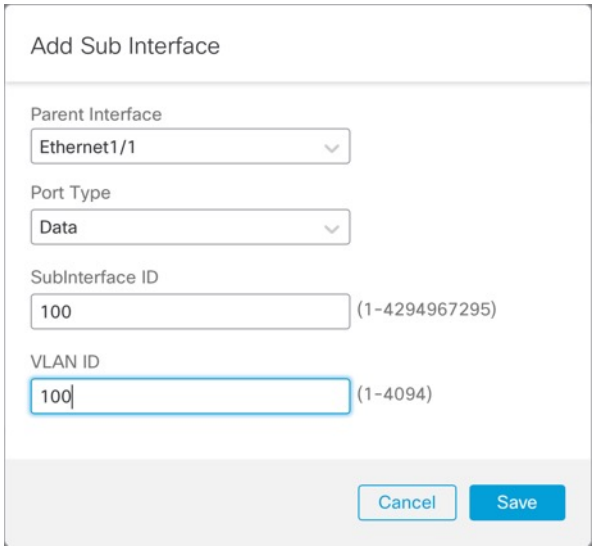
Step 3 Click **Add** > **Subinterface**.

Figure 22: Add subinterface



Step 4 Set the following parameters.

Figure 23: Subinterface settings



Step 5 Click **Save** and then **Save** in the top right of the **Interfaces** page.

You can now **Deploy** the policy to the chassis. The changes are not active until you deploy them.

Add an instance

Use this task when you need to add one or more instances to a chassis already in multi-instance mode. The number of supported instances depends on your model; refer to Requirements and prerequisites for instances, on page 13.

Before you begin

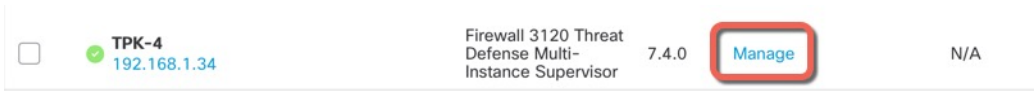
Enable multi-instance mode, on page 17 and Add a multi-instance chassis to the Firewall Management Center, on page 19.

Follow these steps to add an instance to a chassis in multi-instance mode:

Procedure

Step 1 From **Devices** > **Device Management**, click **Manage** in the **Chassis** column or click **Edit** ( ).

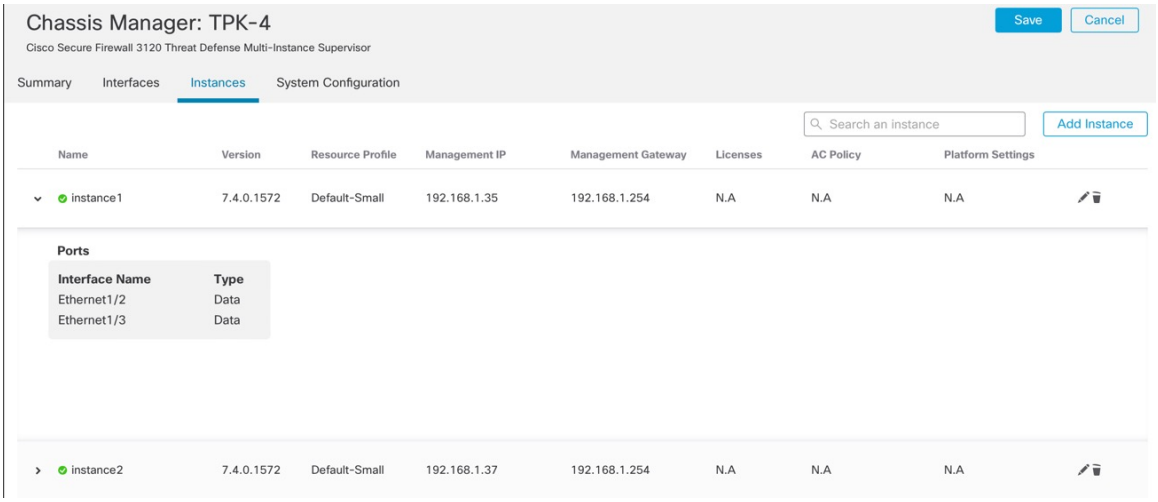
Figure 24: Manage Chassis



The **Chassis Manager** page opens for the chassis to the **Summary** page.

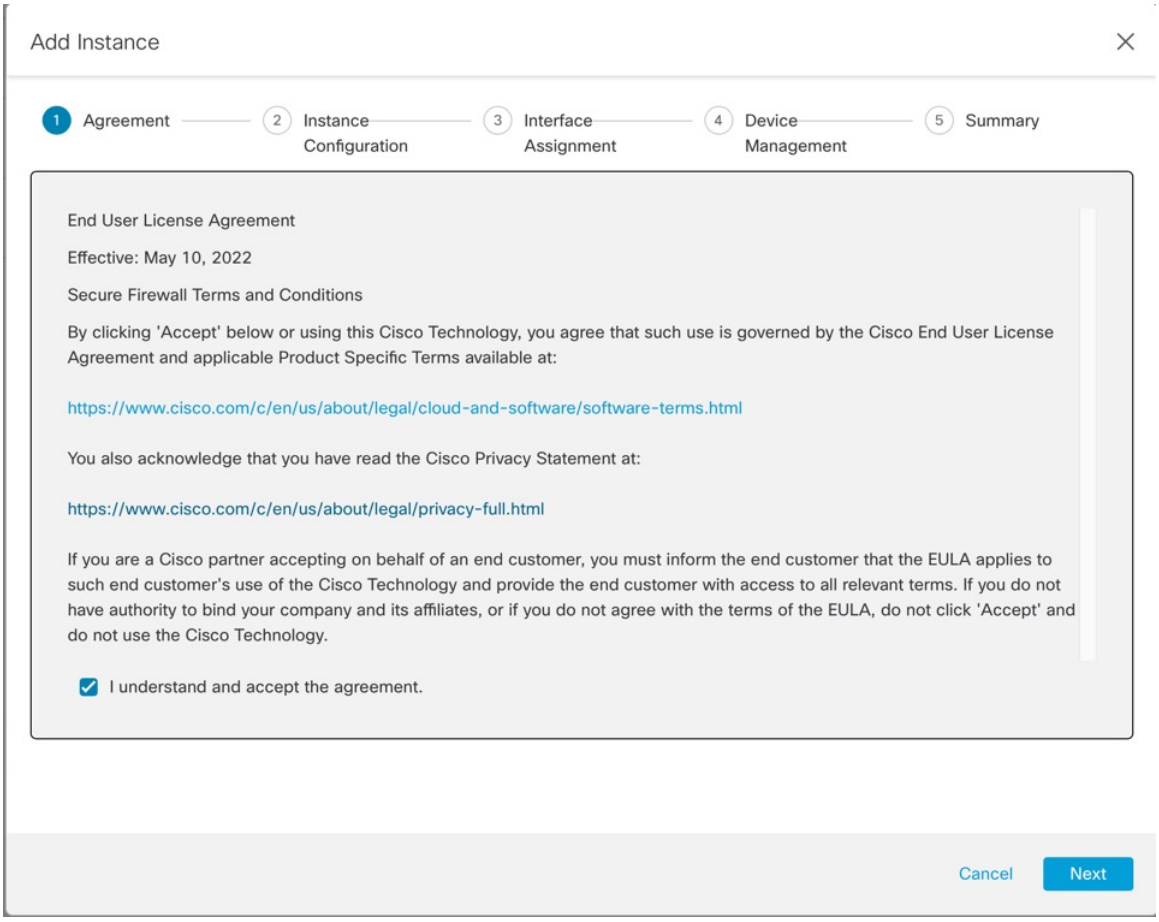
Step 2 Click **Instances**, and click **Add Instance**.

Figure 25: Instances



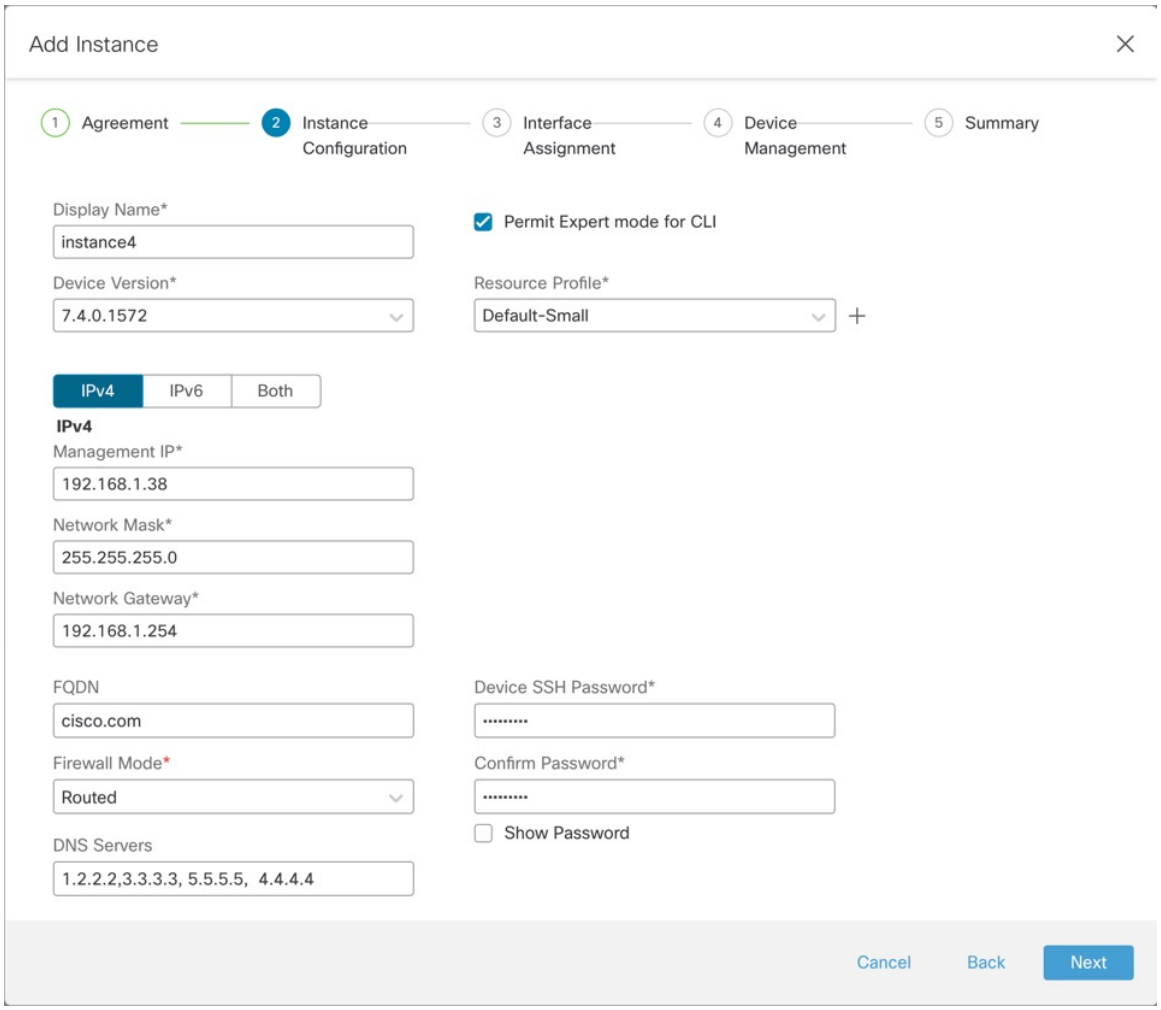
Step 3 On **Agreement**, check **I understand and accept the agreement**, then click **Next**.

Figure 26: Agreement



Step 4 On **Instance Configuration**, set the instance parameters, then click **Next**.

Figure 27: Instance configuration



- **Display Name**
- **Device Version**: Versions listed are packages currently downloaded to the chassis. Patch versions are not listed and cannot be used because they don't contain the entire bundle. To upgrade to a new package, refer to **Devices** > **Upgrade** > **Chassis Upgrade**. When you upgrade, both the old Firewall Threat Defense version and the new Firewall Threat Defense version will be listed in the menu. To download an older package, you need to use the FXOS CLI. Note: Both FXOS and Firewall Threat Defense images are included in the same package. For more information, refer to troubleshooting guide.

  For example:

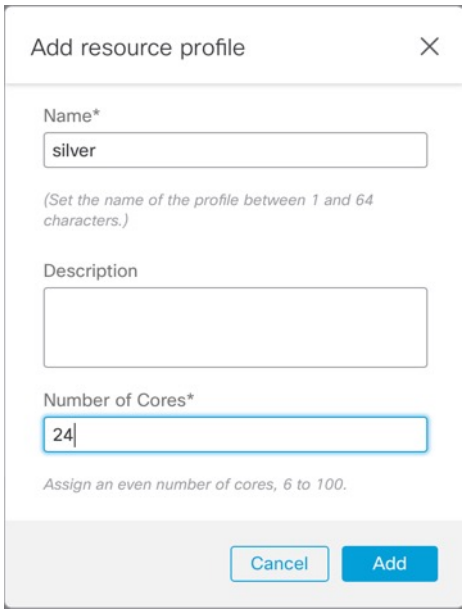
```
firepower-3110# scope firmware
firepower-3110# download image
https://10.10.7.89/Cisco_FTD_SSP_FP3K_Upgrade-7.4.1-1.sh.DEV.tar
Please use the command 'show download-task' or 'show download-task detail' to check
download progress.
% Download-task Cisco_FTD_SSP_FP3K_Upgrade-7.4.1-1.sh.DEV.tar : completed successfully.
```

- **IPv4**, **IPv6**, or **Both**: Set a **Management IP** address on the same network as the chassis Management interface. Set the **Network Mask** and gateway (likely the same gateway as the chassis). The chassis Management interface is shared with each instance, and each instance has its own IP address on the network. You can SSH to this IP address by default to reach the Firewall Threat Defense CLI.
- (Optional) **FQDN**
- **Firewall Mode**: **Routed** or **Transparent**. For more information about the firewall mode, refer to Transparent or Routed Firewall Mode.
- **DNS Servers**: Enter a comma-separated list of DNS servers for management traffic only.
- (Optional) **Permit Expert Mode for CLI**: Expert Mode provides Firewall Threat Defense shell access for advanced troubleshooting.

  If you enable this option, then users who access the instance directly from an SSH session can enter Expert Mode. If you disable this option, then only users who access the instance from the FXOS CLI can enter Expert Mode. We recommend disabling this option to increase isolation between instances.

  Use Expert Mode only if a documented procedure tells you it is required, or if the Cisco Technical Assistance Center asks you to use it. To enter this mode, use the **expert** command in the Firewall Threat Defense CLI.
- **Resource Profile**: The resource profile sets the number of CPU cores; RAM is dynamically allocated according to the number of cores, and disk space is set to 40 GB per instance. The chassis includes these default resource profiles: Default-Small, Default-Medium, and Default-Large. You can add additional profiles for this chassis by clicking **Add** (+). You cannot later edit the resource profile.

Figure 28: Add resource profile



  - The minimum number of cores is 6.

  Note

Instances with a smaller number of cores might experience relatively higher CPU utilization than those with larger numbers of cores. Instances with a smaller number of cores are more sensitive to traffic load changes. If you experience traffic drops, try assigning more cores.

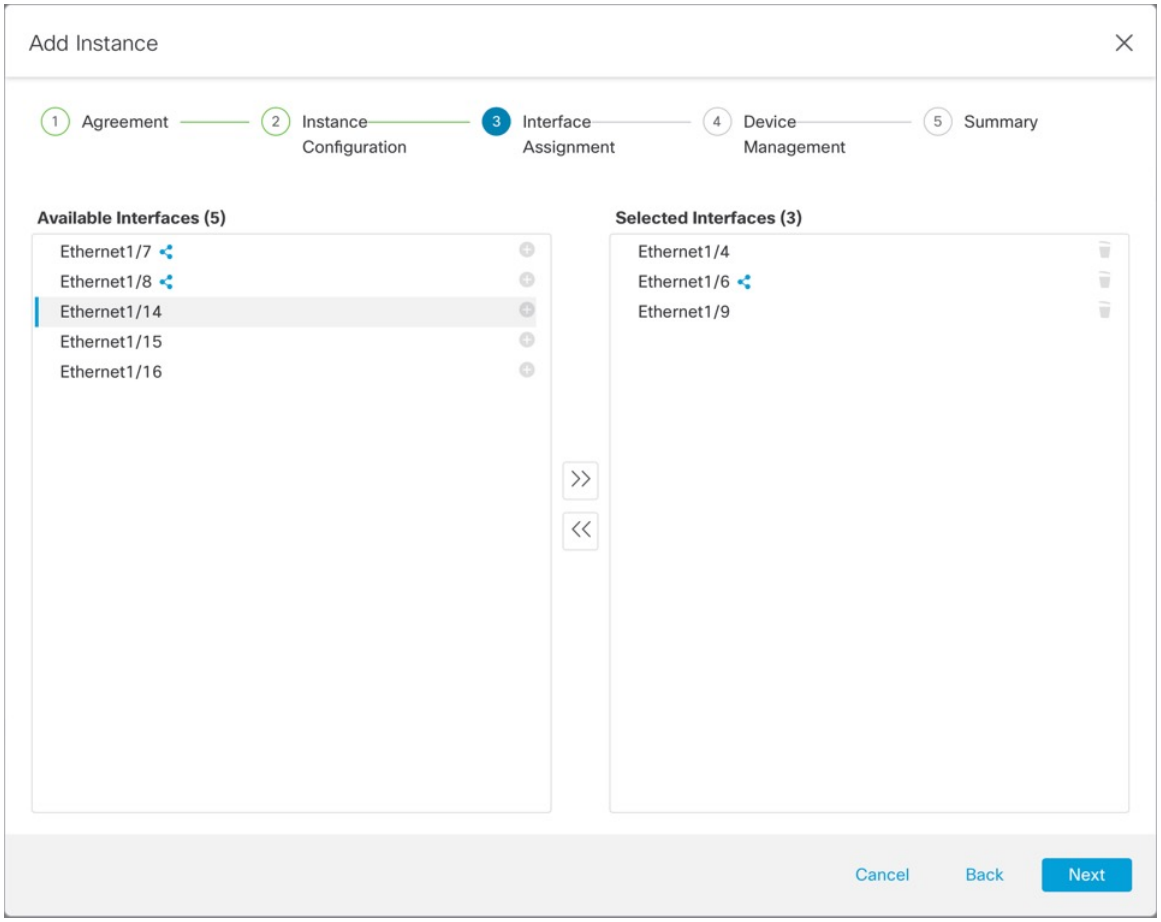
- You can assign cores as an even number (6, 8, 10, 12, 14 etc.) up to the maximum.
- The maximum number of cores available depends on the model; refer to Requirements and prerequisites for instances, on page 13.

If you later assign a different resource profile, then the instance will reload, which can take approximately 5 minutes. Note that for an established High Availability pair , if you assign a different-sized resource profile, be sure to make all members the same size as soon as possible.

- **Device SSH Password**: Set the Firewall Threat Defense admin user password for CLI access, either SSH or console. Repeat the password in the **Confirm Password** field.

Step 5 On **Interface Assignment**, assign the chassis interfaces to the instance, then click **Next**.

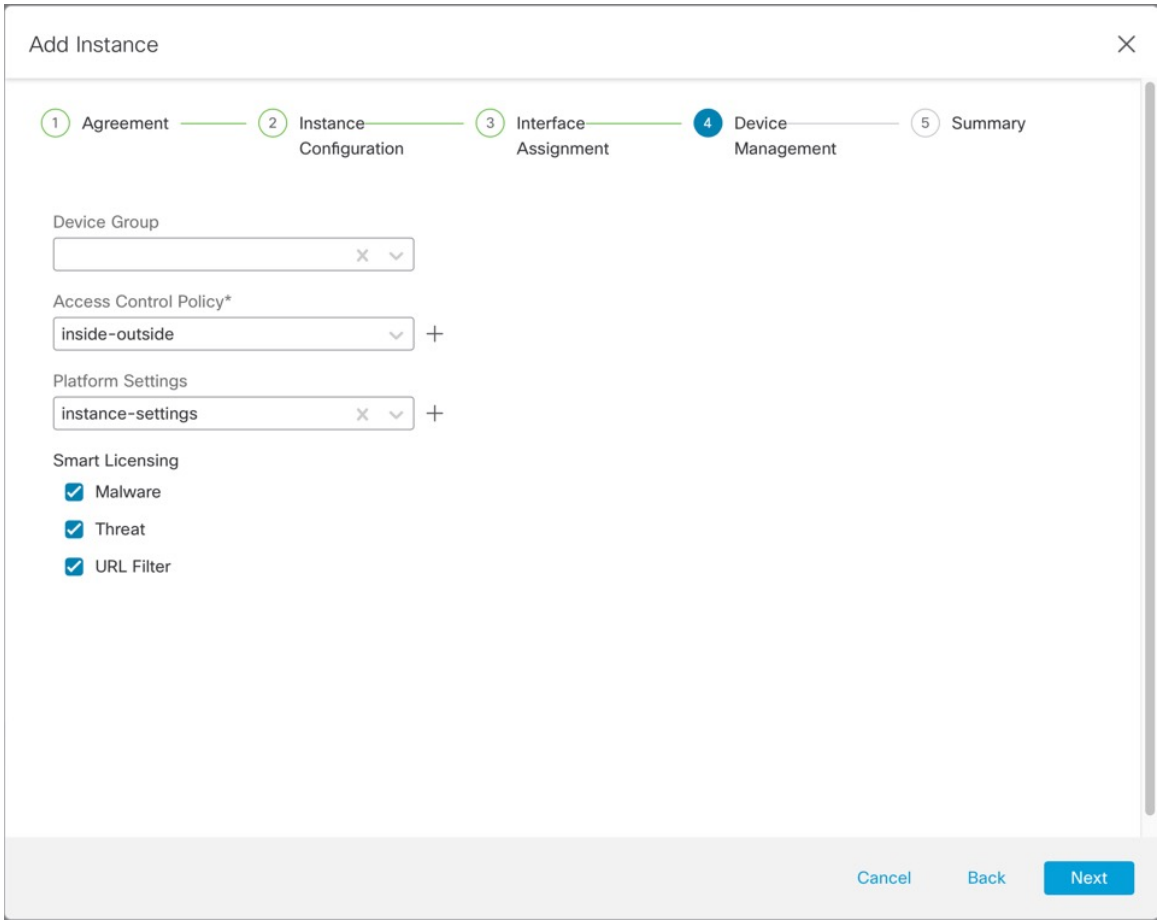
Figure 29: Interface assignment



Shared interfaces show the sharing icon ( ).

Step 6 On **Device Management**, set the device-specific settings, then click **Next**.

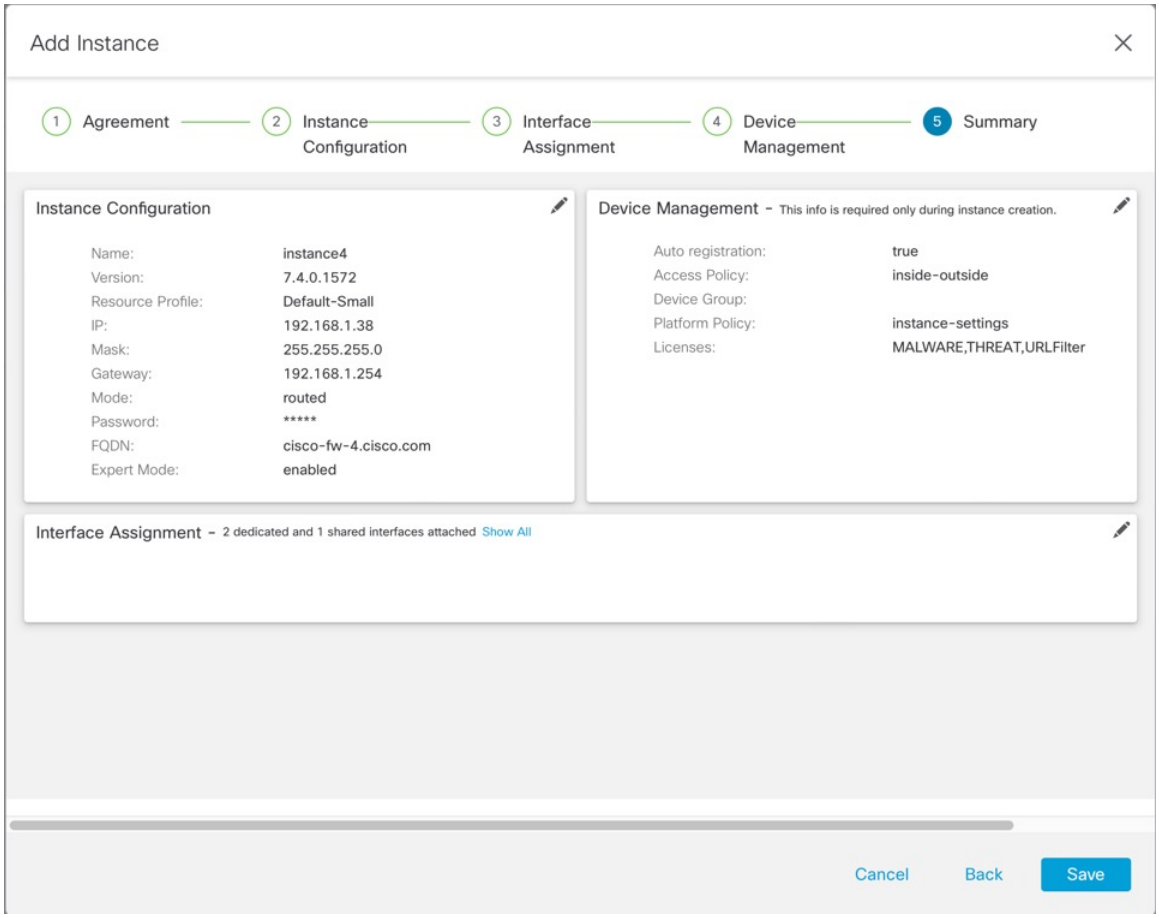
Figure 30: Device management



- **Device Group**
- **Access Control Policy**: Choose an existing access control policy, or create a new policy.
- **Platform Settings**: Choose an existing platform setting policy, or create a new policy.
- **Smart Licensing**

Step 7 On **Summary**, confirm your settings, then click **Save**.

Figure 31: Summary



You can edit any settings on this screen before saving the instance. After you save, the instance is added to the **Instances** screen.

Step 8 On the **Instances** screen, click **Save**.

Deploy the chassis configuration.

After deployment, the instance will be added as a device on the **Device Management** page.

Note
Before you delete an instance, you must first unregister it on the **Device Management** page. If you delete the instance without unregistering it, the instance will remain as a stale entry in **Device Management**.

System configuration customization

You can configure chassis-level settings such as SNMP. You can also import or export the chassis FXOS configuration.

Configure SNMP

You can access chassis-level MIBs through the data interface of one of the instances, which you specify in the chassis system configuration. You can only use this instance for chassis SNMP information. You cannot access SNMP through the chassis Management interface.

Before you begin

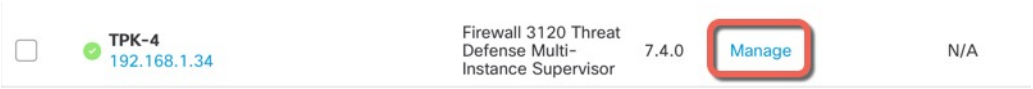
Configure SNMP for one of the instances. For more information, refer to Configure SNMP for a security device.

Follow these steps to configure SNMP:

Procedure

Step 1 From **Devices** > **Device Management**, click **Manage** in the **Chassis** column or click **Edit** ( ).

Figure 32: Manage Chassis

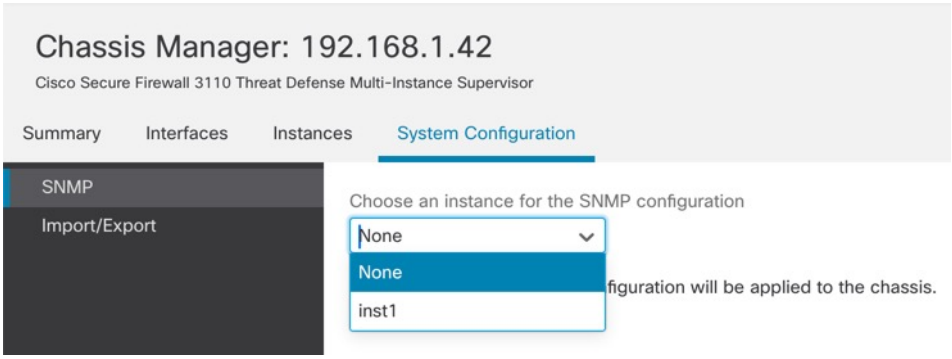


The **Chassis Manager** page opens for the chassis to the **Summary** page.

Step 2 Click **System Configuration**.

Step 3 Click **SNMP**, and choose the instance from the drop-down list.

Figure 33: SNMP



SNMP on the chassis will be accessible from the selected instance.

Step 4 Click **Save**.

Step 5 Deploy the chassis configuration.

Import or export the chassis configuration

You can use the configuration export feature to export an XML file containing chassis configuration settings to your local computer. You can later import that configuration file to quickly apply the configuration settings to your chassis to return to a known good configuration or to recover from a system failure.

Note You can't import the chassis configuration to a new chassis.

When exporting, only chassis configurations are exported; instance configuration settings are not exported. Instances need to be backed up separately using the device backup/restore feature.

When importing, all existing configurations on the chassis will be replaced by the configuration in the import file.

Before you begin

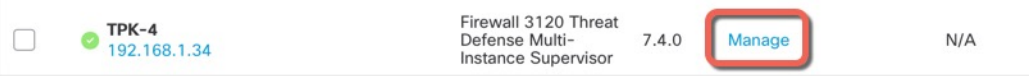
When you import the configuration to the same chassis, make sure the chassis matches for the following:

- Same chassis software version
- Same Firewall Threat Defense instance images
- Same network modules

Procedure

Step 1 From **Devices** > **Device Management**, click **Manage** in the **Chassis** column or click **Edit** ( ).

Figure 34: Manage Chassis



The **Chassis Manager** page opens for the chassis to the **Summary** page.

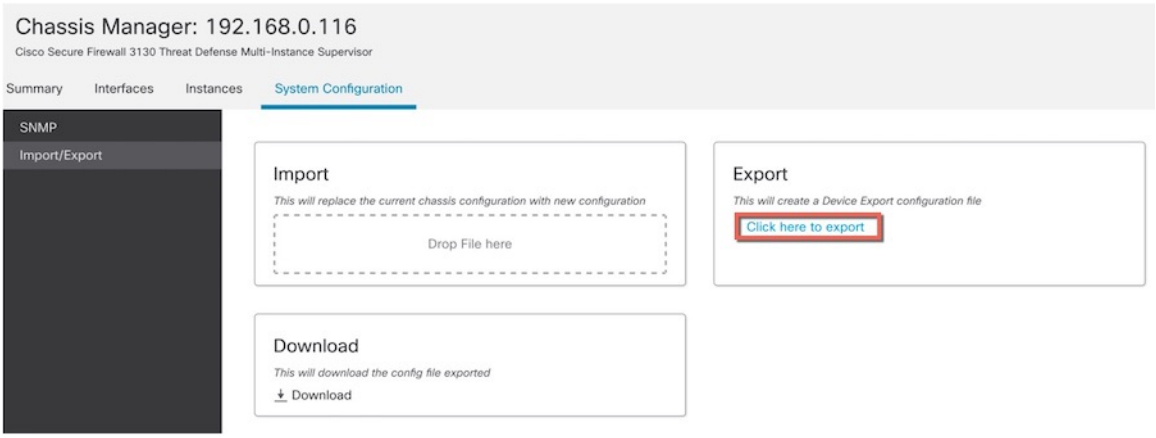
Step 2 Click **System Configuration**.

Step 3 Click **Import/Export**.

Step 4 To export the configuration, follow these steps.

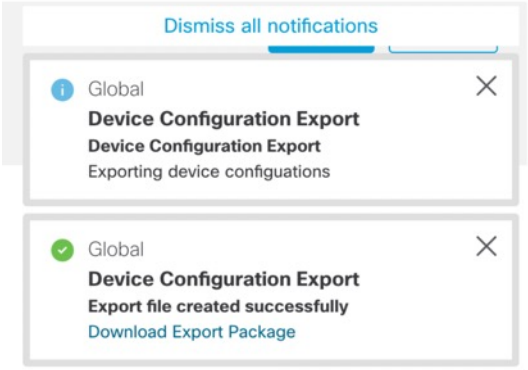
a) In the **Export** area, click **Click here to export**.

Figure 35: Create Export File



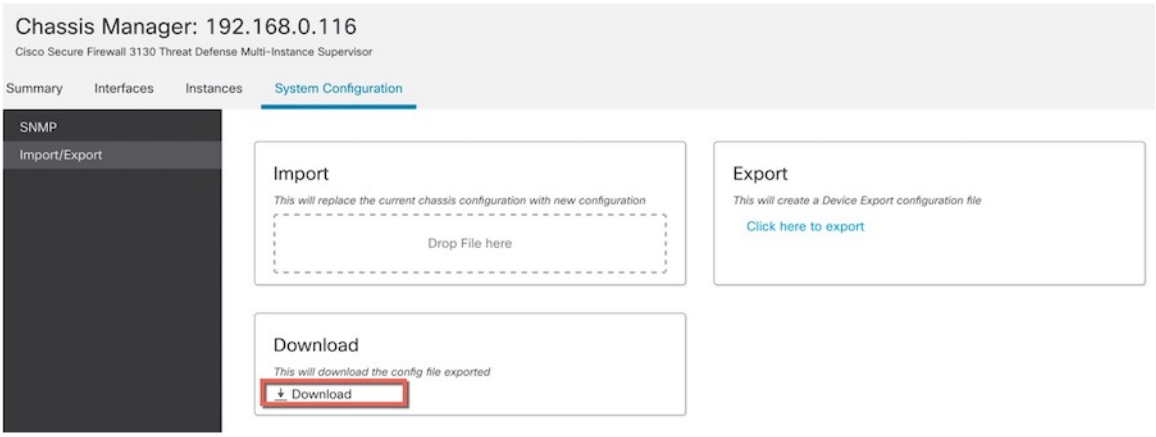
b) Monitor the notifications for the **Export file created successfully** message.

Figure 36: Export File Created Successfully



c) Download the export file by clicking the notification message (**Download Export Package**) or by clicking **Download**.

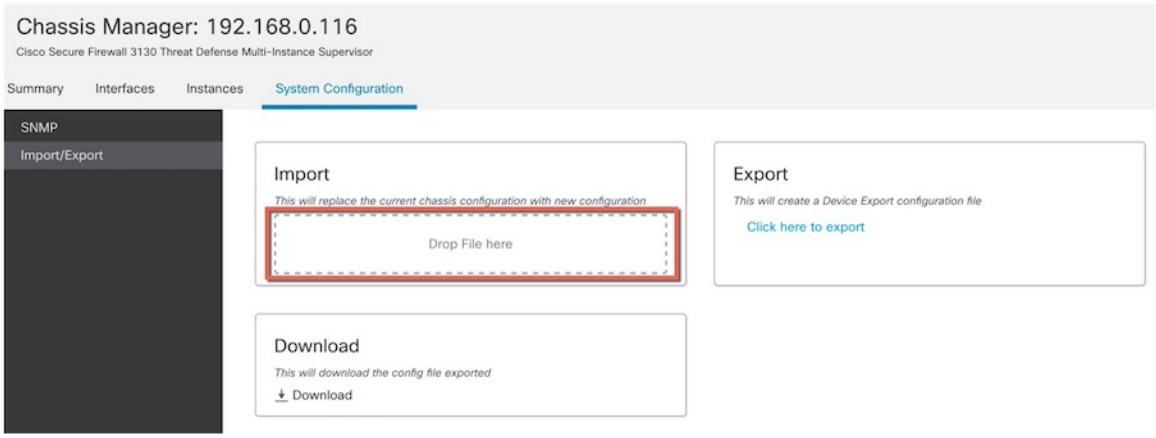
Figure 37: Download



The file is saved with the .sfo extension.

Step 5 To import a configuration, drag the **.sfo** file on the **Import** > **Drop File here** area.

Figure 38: Import



Chassis platform settings

Chassis platform settings are configuration features that configure a range of features for managing the chassis and can be shared among multiple chassis.

Chassis platform settings configure a range of features for managing the chassis. You can share the policy among multiple chassis. If you want different settings per chassis, you must create multiple policies.

Create a chassis platform settings policy

Use the **Platform Settings** page (**Devices** > **Platform Settings**) to manage platform settings policies. This page indicates the type of device for each policy. The **Status** column shows the device targets for the policy.

Procedure

Step 1 Choose **Devices** > **Platform Settings**.

Step 2 For an existing policy, you can **Copy** ( ), **Edit** ( ), or **Delete** ( ) the policy.

Caution
You should not delete a policy that is the last-deployed policy on any of its target devices, even if it is out of date. Before you delete the policy completely, it is good practice to deploy a different policy to those targets.

Step 3 To create a new policy, click **New Policy**.

a) Choose **Chassis Platform Settings** from the drop-down list.
b) Enter a **Name** for the new policy and optionally, a **Description**.
c) Optionally, choose the **Available Chassis** where you want to apply the policy and click **Add** to add the selected chassis.
d) Click **Save**.

The system creates the policy and opens it for editing.

Step 4 To change the target chassis for a policy, click **Edit** ( ) next to the platform settings policy that you want to edit.

a) Click **Policy Assignment**.
b) To assign a chassis to the policy, select it in the **Available Chassis** list and click **Add**.
c) To remove a chassis assignment, click **Delete** ( ) next to a chassis in the **Selected Chassis** list.
d) Click **OK**.

Configure DNS

You need to specify a DNS server if the chassis requires resolution of hostnames to IP addresses. These chassis DNS settings are separate from the DNS settings per instance, which are configured in the device platform settings.

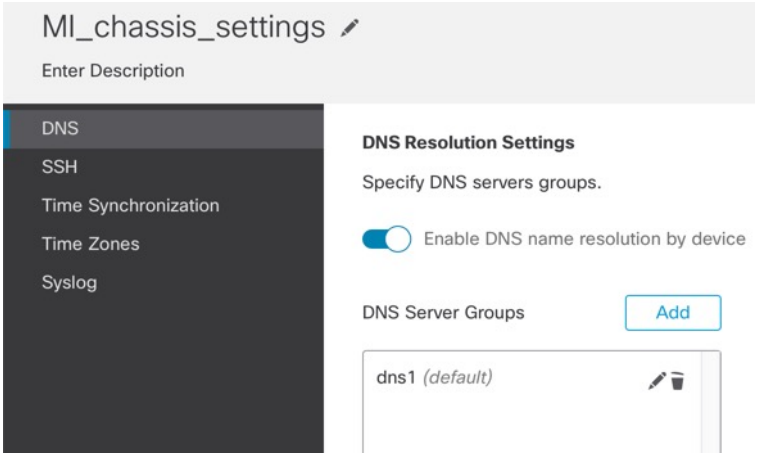
When you configure multiple DNS servers, the chassis uses servers in a random order. You can configure up to four servers across four DNS server groups. For example, you can configure a single server group with four servers, or you can configure four server groups with one server each.

Procedure

Step 1 Choose **Devices** > **Platform Settings** and create or edit the chassis policy.

Step 2 Choose **DNS**.

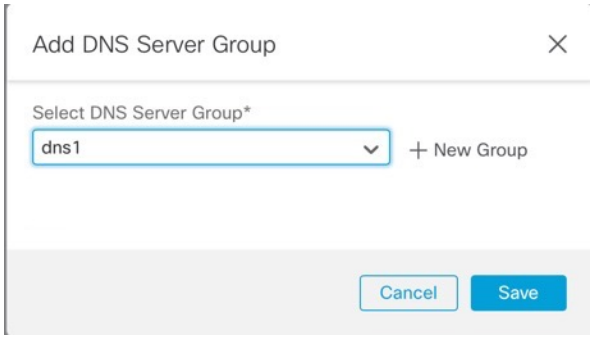
Figure 39: DNS



Step 3 Enable the **Enable DNS name resolution by device** slider.

Step 4 Click **Add** to add a DNS server group.

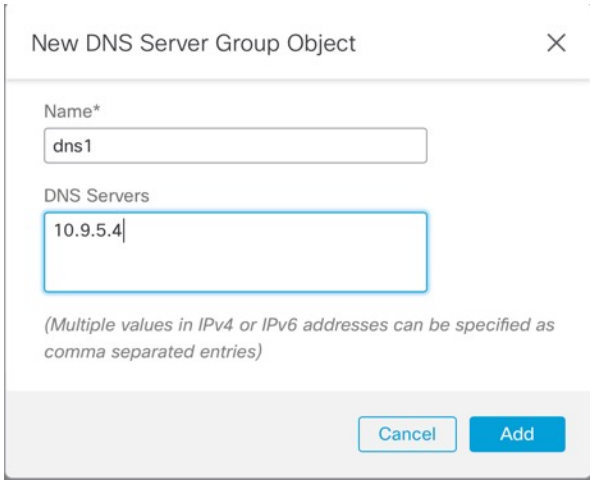
Figure 40: Add DNS server group



Step 5 Either select an existing DNS server group (refer to Create a DNS server group object), or click (+) **New Group**.

If you add a new group, you see this dialog box. Provide a name and up to four DNS server IP addresses as comma-separated values, and click **Add**.

Figure 41: New DNS server group object



Step 6 Click **Save** to add the DNS server to the list.

Step 7 Repeat these steps to add additional server groups.

Make sure you only identify a maximum of four DNS servers in all groups combined.

Step 8 Click **Save** to save all policy changes.

Configure SSH and SSH access list

To allow SSH sessions from the admin user to the chassis on the Management interface, enable the SSH server and configure the allowed networks.

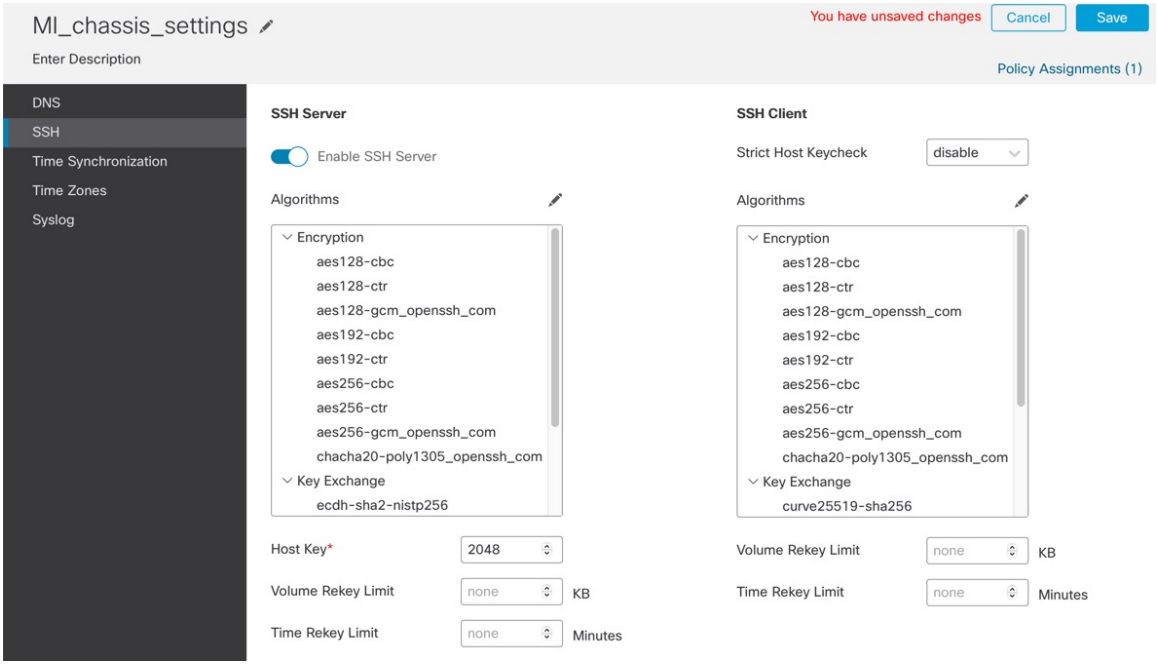
Procedure

Step 1 Choose **Devices** > **Platform Settings** and create or edit the chassis policy.

Step 2 Choose **SSH**.

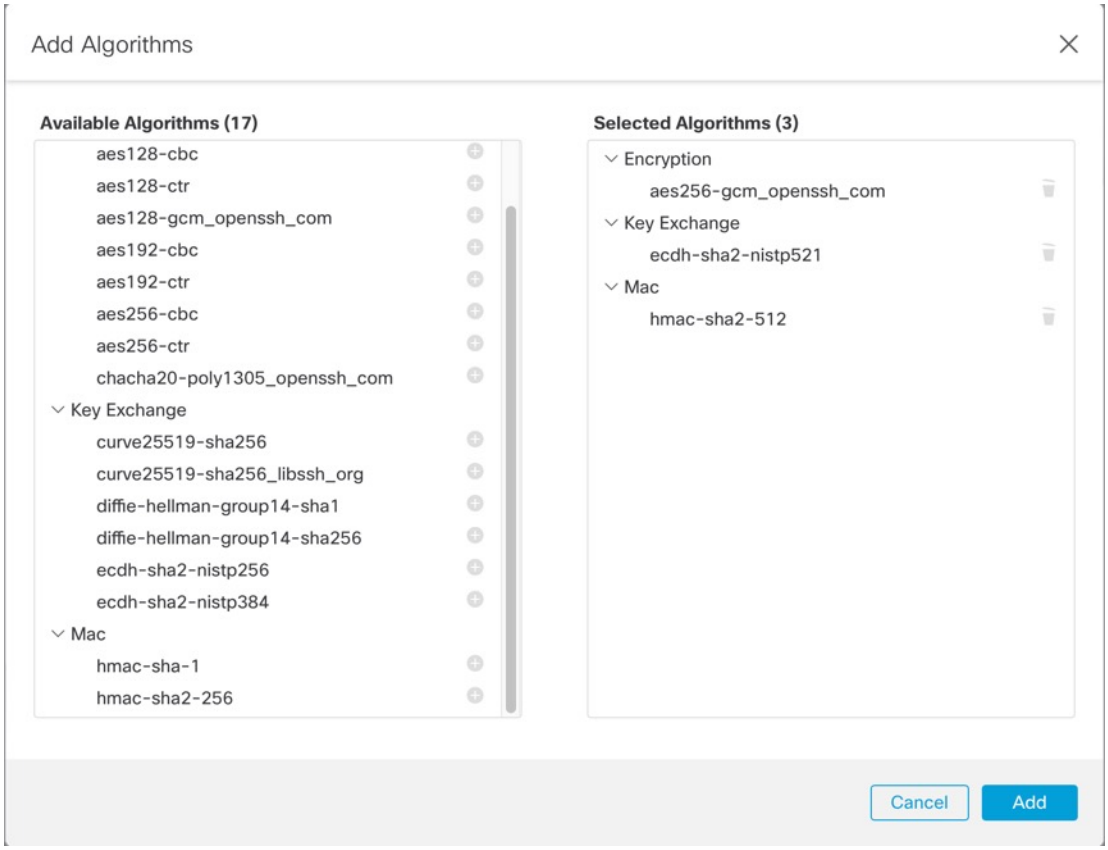
Step 3 To enable SSH access to the chassis, enable the **Enable SSH Server** slider.

Figure 42: SSH



Step 4 To set the allowed **Algorithms**, click **Edit** ( ).

Figure 43: Add Algorithms



a) Select the **Encryption** algorithms.
b) Select the **Key Exchange** algorithms.

   The key exchange provides a shared secret that cannot be determined by either party alone. The key exchange is combined with a signature and the host key to provide host authentication. This key-exchange method provides explicit server authentication.

c) Select the **Mac** integrity algorithms.

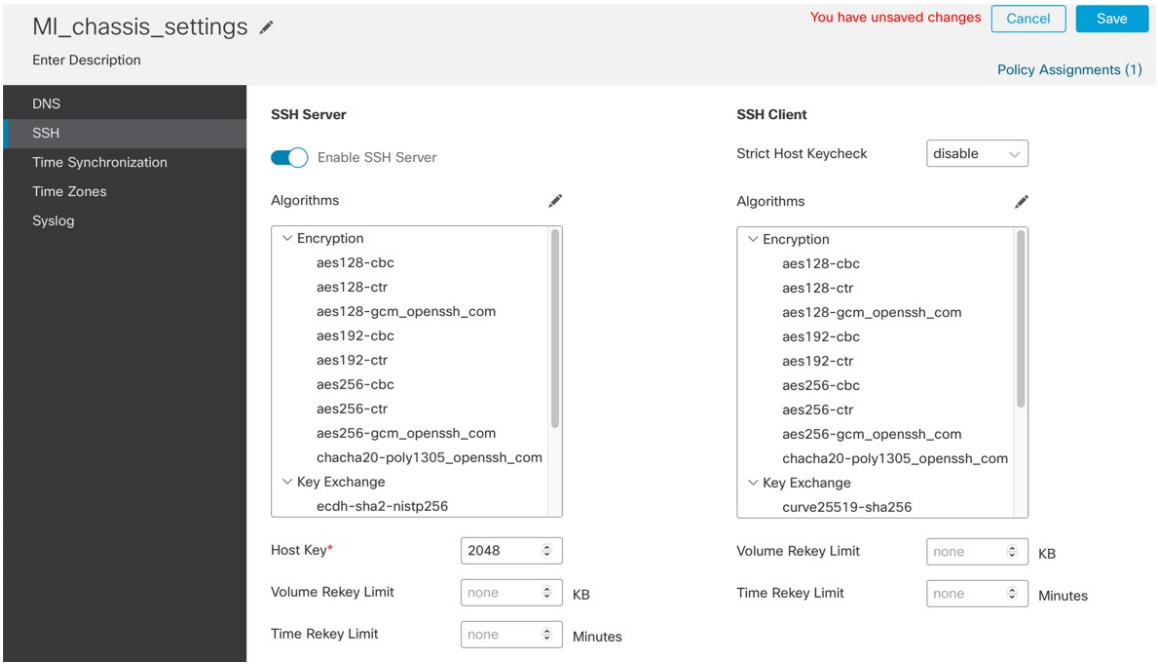
Step 5 For **Host Key**, enter the modulus size for the RSA key pairs.

The modulus value (in bits) is in multiples of 8 from 1024 to 2048. The larger the key modulus size you specify, the longer it takes to generate an RSA key pair. We recommend a value of 2048.

Step 6 For the server **Volume Rekey Limit**, set the amount of traffic in KB allowed over the connection before FXOS disconnects from the session.. For the server **Time Rekey Limit**, set the minutes for how long an SSH session can be idle before FXOS disconnects the session.

Step 7 For the **SSH Client**, configure the following settings.

Figure 44: SSH

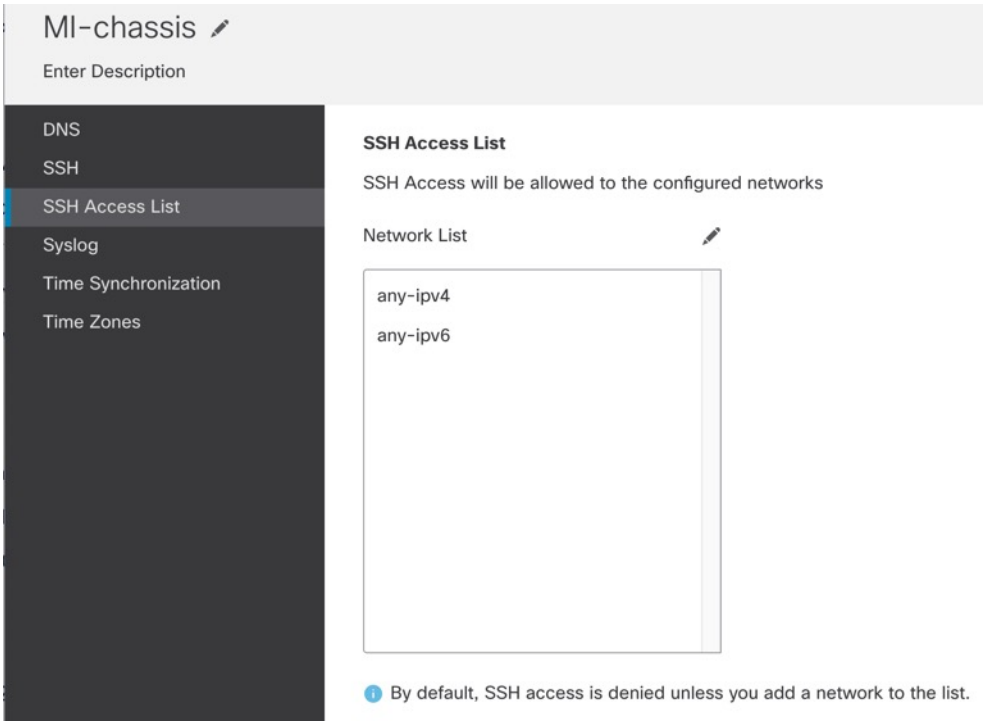


- **Strict Host Keycheck**—Choose **enable**, **disable**, or **prompt** to control SSH host key checking.
  - **enable**—The connection is rejected if the host key is not already in the FXOS known hosts file. You must manually add hosts at the FXOS CLI using the **enter ssh-host** command in the system/services scope.
  - **prompt**—You are prompted to accept or reject the host key if it is not already stored on the chassis.
  - **disable**—(The default) The chassis accepts the host key automatically if it was not stored before.
- **Algorithms**: Click **Edit** (✎). and select the **Encryption**, **Key Exchange**, and **Mac** algorithms.
- **Volume Rekey Limit**: Set the amount of traffic in KB allowed over the connection before FXOS disconnects from the session.
- **Time Rekey Limit**: Set the minutes for how long an SSH session can be idle before FXOS disconnects the session.

Step 8 Choose **SSH Access List**. You need to allow access to IP addresses or networks before you can use SSH.

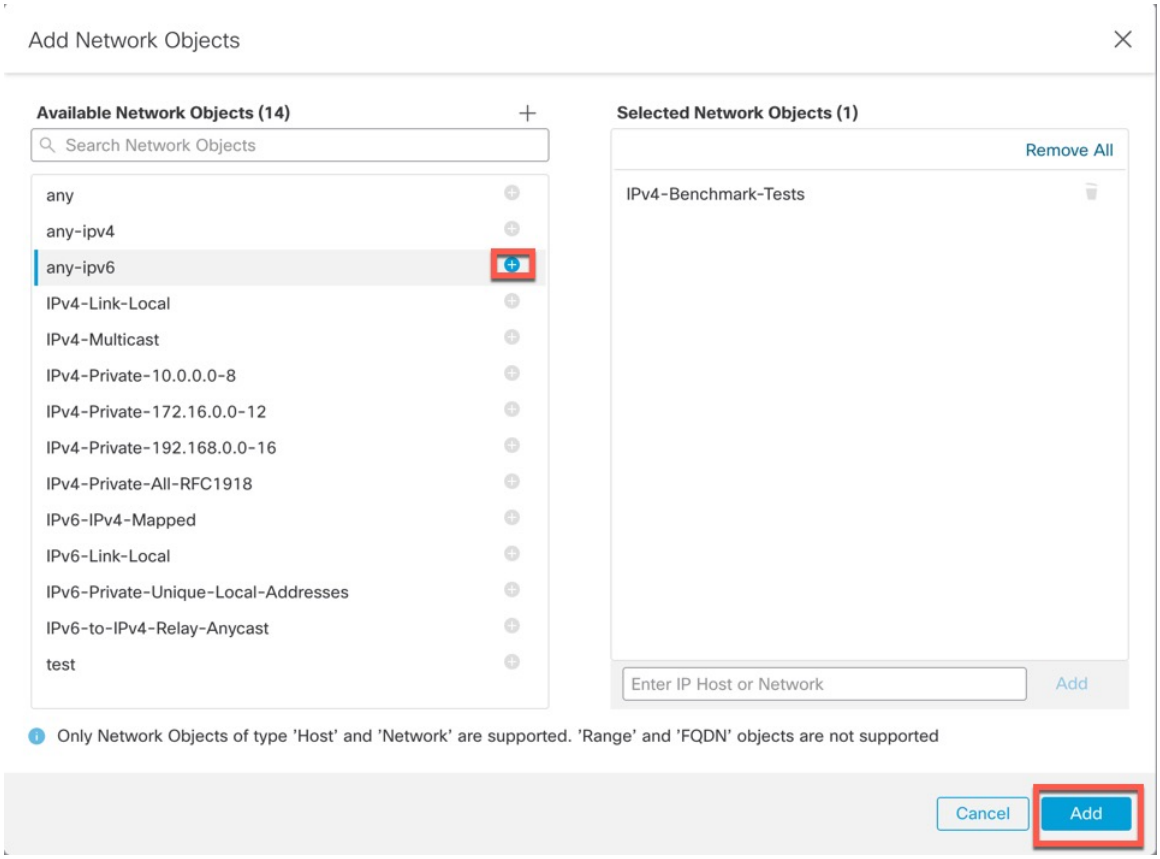
You can add up to 25 access lists.

Figure 45: SSH Access List



Step 9 Click **Edit** (✎) to add network objects and click **Save**. You can also manually enter IP addresses. Click **Save** to save all policy changes.

Figure 46: Network Objects



Configure syslog

You can enable syslogs from the chassis. These syslogs come from the chassis' FXOS operating system.

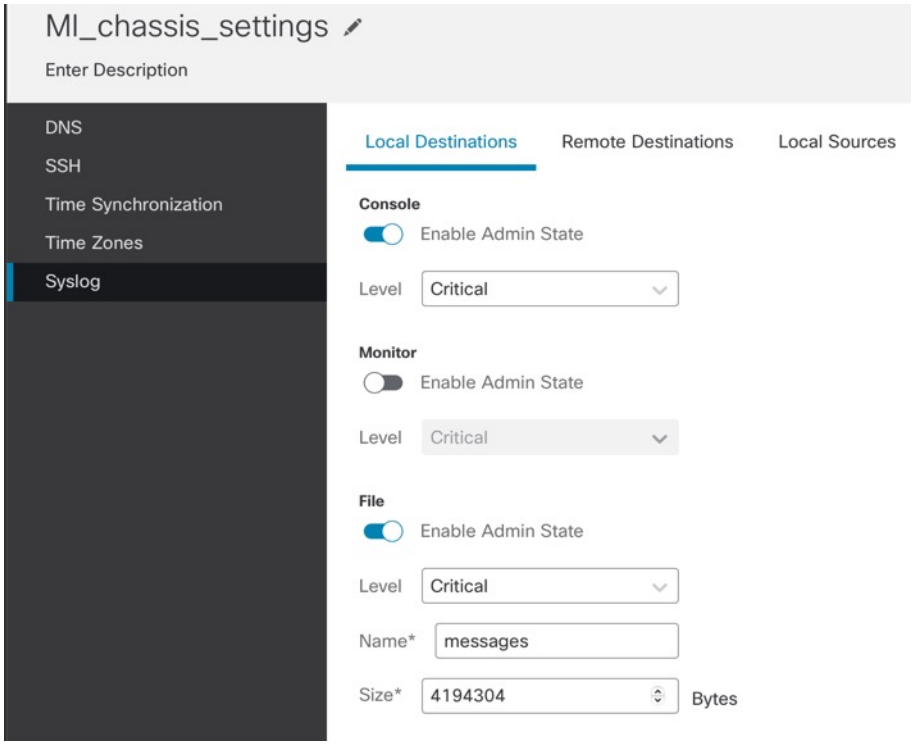
Procedure

Step 1 Choose **Devices** > **Platform Settings** and create or edit the chassis policy.

Step 2 Choose **Syslog**.

Step 3 Click the **Local Destinations** tab, and complete the following fields.

Figure 47: Syslog local destinations

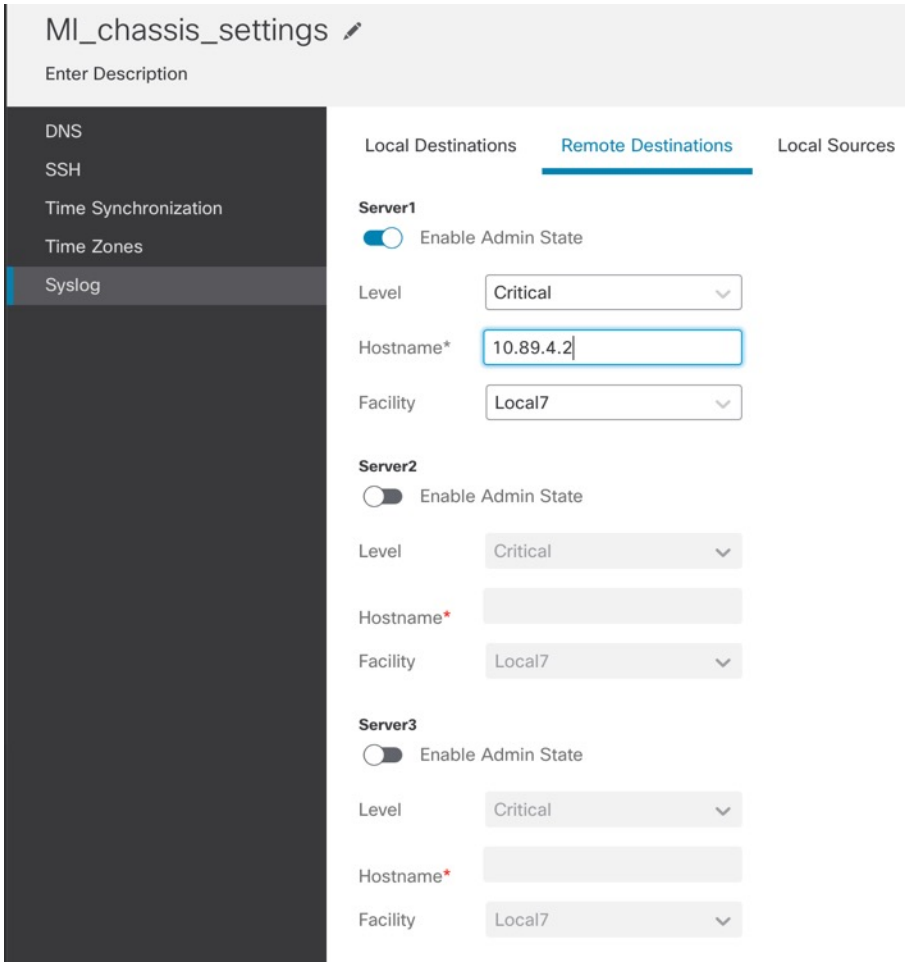


Name	Description
Console Section	
Admin State field	Whether the chassis displays syslog messages on the console. Check the **Enable** check box if you want to have syslog messages displayed on the console as well as added to the log. If the **Enable** check box is unchecked, syslog messages are added to the log but are not displayed on the console.
Level field	If you checked the **Enable** check box for **Console - Admin State**, select the lowest message level that you want displayed on the console. The chassis displays that level and above on the console. This can be one of the following: • **Emergencies** • **Alerts** • **Critical**
Monitor Section	

Name	Description
Admin State field	Whether the chassis displays syslog messages on the monitor. Check the **Enable** check box if you want to have syslog messages displayed on the monitor as well as added to the log. If the **Enable** check box is unchecked, syslog messages are added to the log but are not displayed on the monitor.
Level drop-down list	If you checked the **Enable** check box for **Monitor - Admin State**, select the lowest message level that you want displayed on the monitor. The system displays that level and above on the monitor. This can be one of the following: • **Emergencies** • **Alerts** • **Critical** • **Errors** • **Warnings** • **Notifications** • **Information** • **Debugging**

Step 4 On the **Remote Destinations** tab, complete the following fields for up to three external logs that can store messages generated by the chassis:

Figure 48: Syslog remote destinations



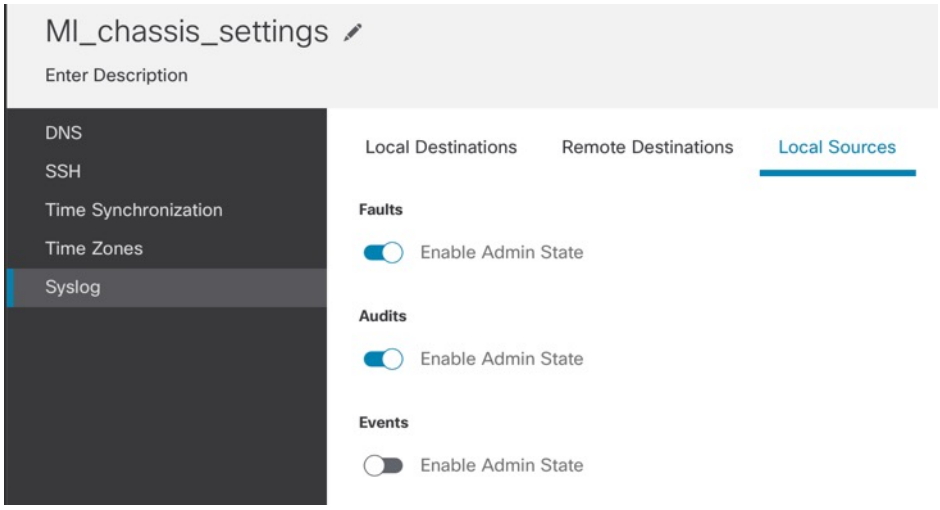
By sending syslog messages to a remote destination, you can archive messages according to the available disk space on the external syslog server, and manipulate logging data after it is saved. For example, you could specify actions to be executed when certain types of syslog messages are logged, extract data from the log and save the records to another file for reporting, or track statistics using a site-specific script.

Name	Description
Admin State field	Check the **Enable** check box if you want to have syslog messages stored in a remote log file.

Name	Description
Level drop-down list	Select the lowest message level that you want the system to store. The system stores that level and above in the remote file. This can be one of the following: • **Emergencies** • **Alerts** • **Critical** • **Errors** • **Warnings** • **Notifications** • **Information** • **Debugging**
Hostname/IP Address field	The hostname or IP address on which the remote log file resides. **Note** You must configure a DNS server if you use a hostname rather than an IP address.
Facility drop-down list	Choose a system log facility for syslog servers to use as a basis to file messages. This can be one of the following: • **Local0** • **Local1** • **Local2** • **Local3** • **Local4** • **Local5** • **Local6** • **Local7**

Step 5 Click the **Local Sources** tab, and complete the following fields.

Figure 49: Syslog local sources



Name	Description
Faults > **Enable Admin State**	Enable system fault logging.
Audits > **Enable Admin State**	Enable audit logging.
Events > **Enable Admin State**	Enable system event logging.

Step 6 Click **Save** to save all policy changes.

Configure time synchronization

NTP is used to implement a hierarchical system of servers that provide precisely synchronized time among network systems. This kind of accuracy is required for time-sensitive operations, such as validating CRLs, which include a precise time stamp. You can configure up to four NTP servers.

Note

- FXOS uses NTP version 3.
- If the stratum value of an external NTP server is 13 or greater, the application instance cannot sync to the NTP server on the FXOS chassis. Each time a NTP client syncs to a NTP server, the stratum value increases by one.

  If you have set up your own NTP server, you can find its stratum value in the /etc/ntp.conf file on the server. If the NTP server has stratum value of 13 or greater you can either change the stratum value in the ntp.conf file and restart the server, or use a different NTP server (for example: pool.ntp.org).

Before you begin

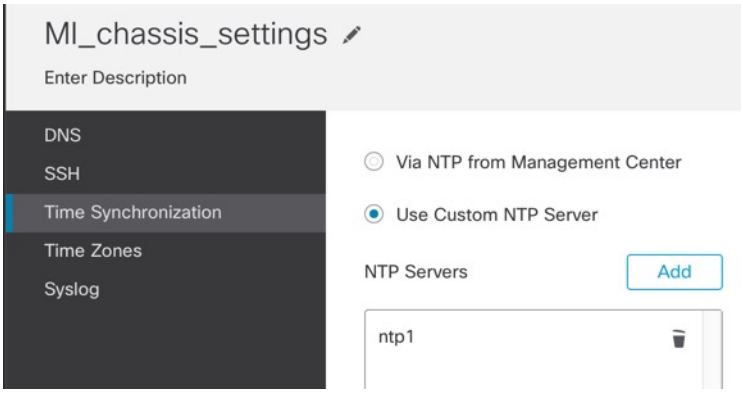
If you use a hostname for the NTP server, you must configure a DNS server. For more information, refer to Configure DNS, on page 42.

Procedure

Step 1 Choose **Devices** > **Platform Settings** and create or edit the chassis policy.

Step 2 Choose **Time Synchronization**.

Figure 50: Time synchronization



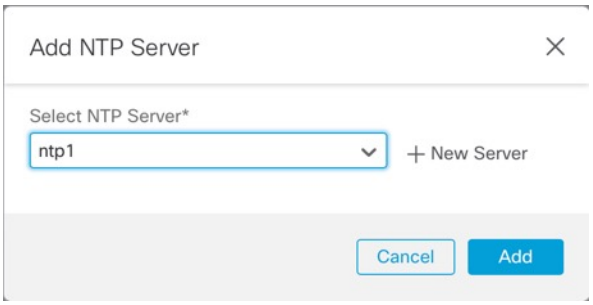
Step 3 If you want to obtain the time from the Firewall Management Center, click **Via NTP from Management Center**.

This option ensures both the chassis and the Firewall Management Center have the same time.

Step 4 To use an external NTP server, click **Use Custom NTP Server**.

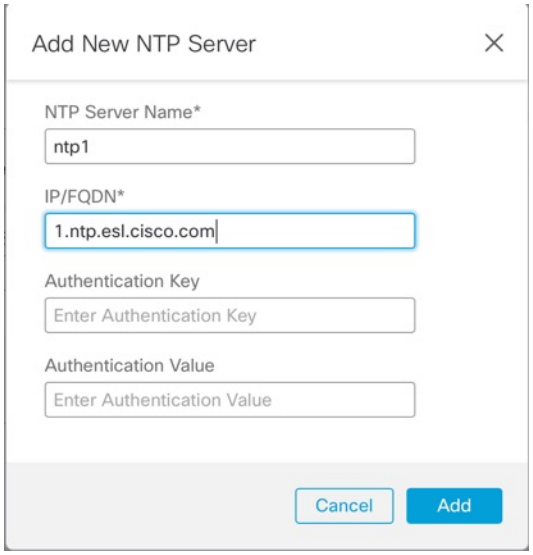
a) Click **Add** to add a server.

Figure 51: Add NTP server



b) Choose any already-defined servers from the drop-down menu and click **Add**, or click **New Server** to add a new server.

Figure 52: Add new NTP server



c) For a new server, enter the following fields, and click **Add**.

- **NTP Server Name**: A name to identify this server.
- **IP/FQDN**: The IP address or hostname of the server.
- **Authentication Key** and **Authentication Value**: Obtain the key ID and value from the NTP server. For example, to generate the SHA1 key on NTP server Version 4.2.8p8 or later with OpenSSL installed, enter the **NTP-keygen -M** command, and then view the key ID and value in the NTP.keys file. The key is used to tell both the client and server which value to use when computing the message digest.

  Only SHA1 is supported for NTP server authentication.

Step 5 Click **Save** to save all policy changes.

Configure time zones

Time zone configuration is essential for maintaining accurate timestamps in system logs and ensuring proper time-based functionality across the device.

Before you begin

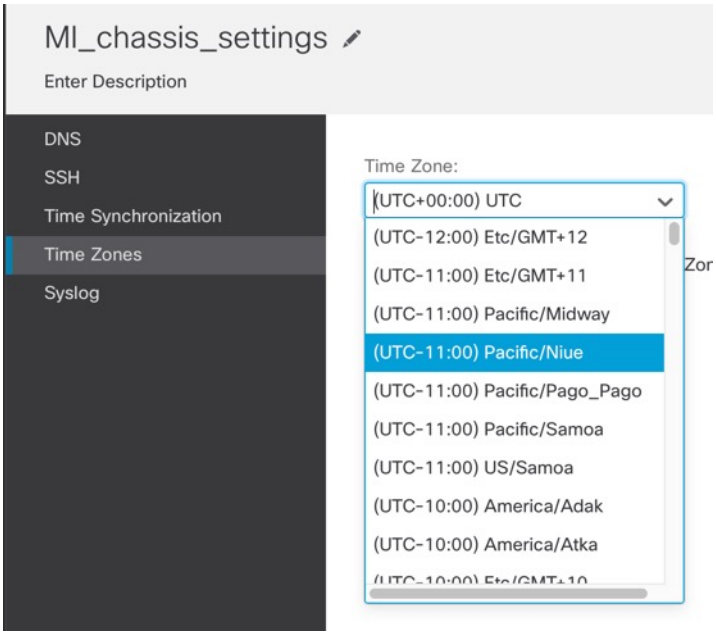
Follow these steps to configure the time zone for the chassis:

Procedure

Step 1 Choose **Devices** > **Platform Settings** and create or edit the chassis policy.

Step 2 Choose **Time Zones**.

Figure 53: Time zones



Step 3 Choose your **Time Zone** from the drop-down menu.

Step 4 Click **Save** to save all policy changes.

Multi-instance mode management

This section describes less common tasks, including changing settings at the FXOS CLI or changing interfaces assigned to the chassis.

Change interfaces assigned to an instance

You can allocate or unallocate an interface on the instance. Adding a new interface, or deleting an unused interface, has minimal impact on the instance configuration. You can also edit the membership of an allocated EtherChannel without affecting the instance.

Deleting an interface that is used in your security policy will impact the configuration. Interfaces can be referenced directly in many places in the instance configuration, including access rules, NAT, SSL, identity rules, VPN, DHCP server, and so on. Deleting an interface will delete any configuration associated with that interface.

Policies that refer to security zones are not affected.

Note For high availability, you need to make the same interface changes for the other unit. Otherwise, high availability might not operate correctly.

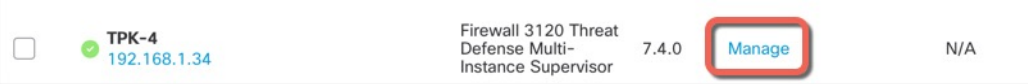
Before you begin

- Configure your interfaces according to Instance configuration, on page 17.
- If you plan to add an already-allocated interface to an EtherChannel, you need to unallocate the interface from the instance first, then add the interface to the EtherChannel. For a new EtherChannel, you can then allocate the EtherChannel to the instance.

Procedure

Step 1 From **Devices** > **Device Management**, click **Manage** in the **Chassis** column or click **Edit** ( ).

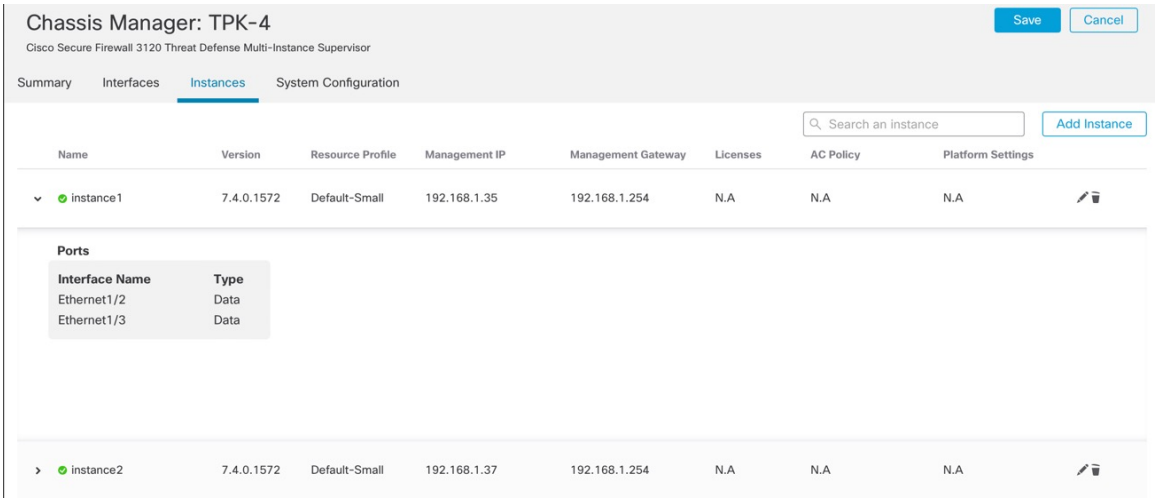
Figure 54: Manage Chassis



The **Chassis Manager** page opens for the chassis to the **Summary** page.

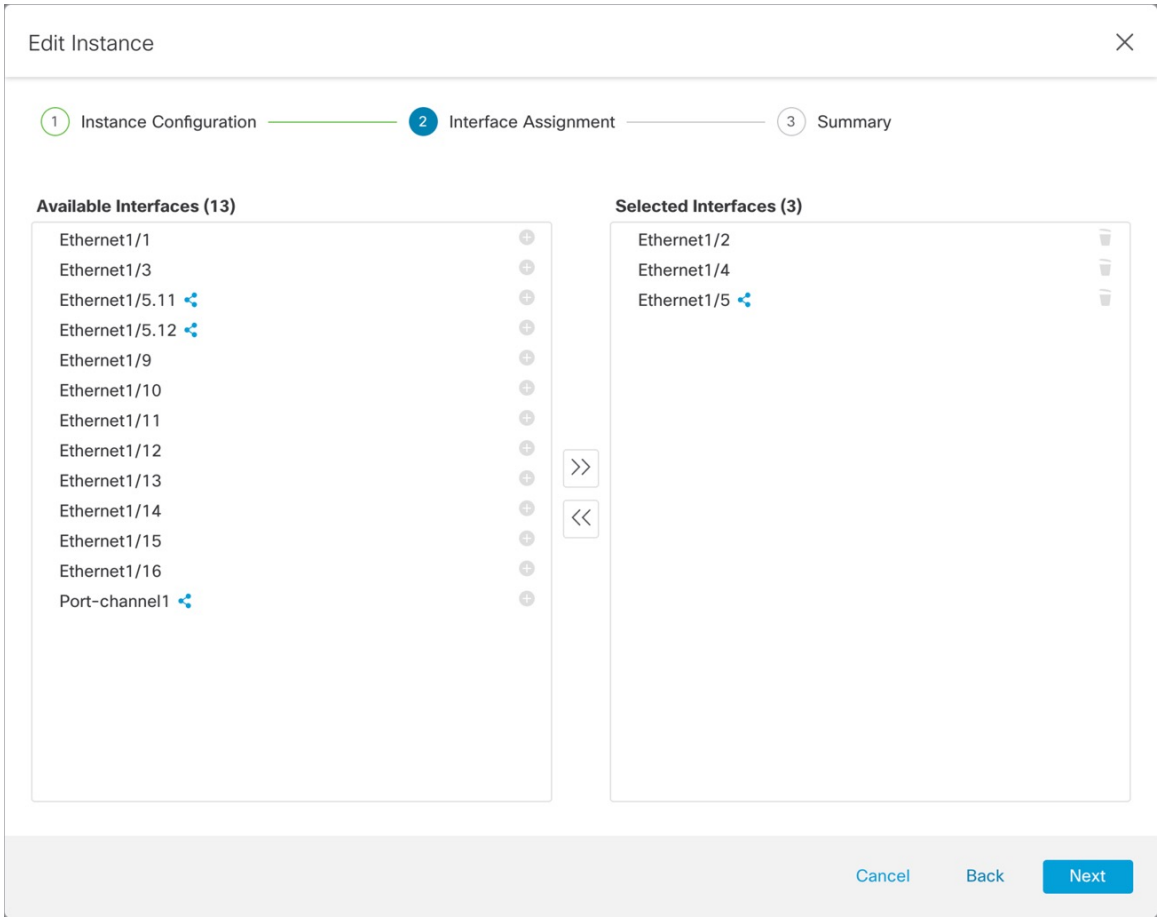
Step 2 Click **Instances**, and click **Edit** ( ) next to the instance for which you want to change interfaces.

Figure 55: Instances



Step 3 Click **Next** until you get to the **Interface Assignment** screen.

Figure 56: Interface assignment



Shared interfaces show the sharing icon ( ).

Step 4 Make your interface changes, and then click **Next**.

Step 5 Click **Save** on the **Summary** screen.

Step 6 For high availability, you need to make the same interface changes for the other unit. Otherwise, high availability might not operate correctly.

Change chassis management settings at the FXOS CLI

If you want to change the chassis management interface IP address and gateway, change the Firewall Management Center to a new manager, change the admin password, or disable multi-instance mode, you can do so from the FXOS CLI.

Procedure

Step 1 Connect to the chassis console port.

The console port connects to the FXOS CLI.

Note
We recommend using the console port. You can also connect using SSH to the management interface, if configured in the chassis platform settings in the Firewall Management Center; however, if you change the management IP address, you will be disconnected.

Step 2 Log in with the username **admin** and the password you set during initial setup.

Step 3 Change the Management IP address. You can use a static IPv4 and/or IPv6 address.

IPv4:

scope fabric-interconnect

set out-of-band static IP *ip_address* **netmask** *network_mask* **gw** *gateway_ip_address*

IPv6:

scope fabric-interconnect

scope ipv6-config

set out-of-band static ipv6 *ipv6_address* **ipv6-prefix** *prefix_length* **ipv6-gw** *gateway_address*

Example:

IPv4:

```
firepower-3110# scope fabric-interconnect
firepower-3110 /fabric-interconnect # set out-of-band static ip 10.5.23.8 netmask
255.255.255.0
gw 10.5.23.1
```

IPv6:

```
firepower-3110# scope fabric-interconnect
firepower-3110 / fabric-interconnect # scope ipv6-config
firepower-3110 / fabric-interconnect /ipv6-config # set out-of-band static ipv6 2001:DB8::34
ipv6-prefix 64 ipv6-gw 2001:DB8::1
```

Step 4 Change the Firewall Management Center.

You should first unregister the chassis from the current Firewall Management Center.

enter device-manager *manager_name* [**hostname** {*hostname* | *ipv4_address* | *ipv6_address*}] [**NAT-ID** *nat_id*]

You are prompted for the registration key.

You can enter this command from any scope.

- **hostname** {*hostname* | *ipv4_address* | *ipv6_address*}: Specifies either the FQDN or IP address of the Firewall Management Center. At least one of the devices, either the Firewall Management Center or the chassis, must have a reachable IP address to establish the two-way, TLS-1.3-encrypted communication channel between the two devices. If you do not specify a **hostname**, then the chassis must have a reachable IP address or hostname and you must specify the **NAT-ID**.

- **NAT-ID** *nat_id*: Specifies a unique, one-time string of your choice that you will also specify on the Firewall Management Center when you register the chassis when one side does not specify a reachable IP address or hostname. It is required if you do not specify a **hostname**, however we recommend that you always set the NAT ID even when you specify a hostname or IP address. The NAT ID must not exceed 37 characters. Valid characters include alphanumerical characters (A–Z, a–z, 0–9) and the hyphen (-). This ID cannot be used for any other devices registering to the Firewall Management Center.
- **Registration Key:** *reg_key*: You will be prompted for a one-time registration key of your choice that you will also specify on the Firewall Management Center when you register the chassis. The registration key must not exceed 37 characters. Valid characters include alphanumerical characters (A–Z, a–z, 0–9) and the hyphen (-).

Example:

```
firepower-3110# enter device-manager boulder_fmc hostname 10.89.5.35 nat-id 93002
(Valid registration key characters: [a-z],[A-Z],[0-9],[-]. Length: [2-36])
Registration Key: Impala67
```

Step 5 Change the admin password.

scope security

set password

Enter a password: *password*

Confirm the password: *password*

Example:

```
firepower-3110# scope security
firepower-3110 /security # set password
Enter new password: Sw@nsong67
Confirm new password: Sw@nsong67
firepower-3110 /security #
```

Step 6 Disable multi-instance mode and set the system back to appliance mode.

scope system

set deploymode native

You are prompted to reboot.

Example:

```
firepower-3110# scope system
firepower-3110 /system # set deploymode native
All configuration and bootable images will be lost and system will reboot.
If there was out of band upgrade, it might reboot with the base version and
need to re-image to get the expected running version.
Do you still want to change deploy mode? (yes/no):yes
firepower-3110 /system #
```

To change the mode back to multi-instance mode, enter **set deploymode container**. You can check the current mode using the **show system detail** command.

Multi-instance mode

This section helps you troubleshoot and diagnose your multi-instance mode chassis and instances.

Multi-instance setups

A multi-instance setup is a system configuration approach that

- enables running multiple independent instances on a single hardware platform,
- allows flexible allocation of resources to each instance, and
- supports both container and native (appliance) operating modes.

show system detail

This FXOS command shows the current mode, Native or Container. If the mode is Native (also known as appliance mode), you can convert to multi-instance (Container) mode. Note that the prompt/name in multi-instance mode is generic "firepower-<model>" while the prompt in appliance mode is the hostname you set for the Firewall Threat Defense (by default "firepower")

```
firepower # show system detail

Systems:
    Name: firepower
    Mode: Stand Alone
    System IP Address: 172.16.0.50
    System IPv6 Address: ::
    System Owner:
    System Site:
    Deploy Mode: Native
    Description for System:
firepower #
```

scope system > show

This FXOS command shows the current mode in a table format. Note that the prompt/name in multi-instance mode is generic "firepower-<model>" while the prompt in appliance mode is the hostname you set for the Firewall Threat Defense.

```
firepower-3110# scope system
firepower-3110 /system # show

Systems:
    Name       Mode        Deploy Mode System IP Address System IPv6 Address
    ---------- ----------- ----------- ----------------- -------------------
```

```
    firepower-3110
               Stand Alone Container   10.89.5.42        ::

3110-1# scope system
3110-1 /system # show

Systems:
    Name       Mode        Deploy Mode System IP Address System IPv6 Address
    ---------- ----------- ----------- ----------------- -------------------
    3110-1     Stand Alone Native      10.89.5.41        ::
3110-1 /system #
```

Instance interfaces

An instance interface is a physical or virtual network interface on a device that

- enables connectivity between multiple logical instances within a system,
- facilitates assignment of interfaces to one or more instances, and
- supports configuration of forwarding rules using ECMP and multicast groups.

show portmanager switch forward-rules hardware mac-filter

This command shows the internal switch-forwarding rule for two instances with a dedicated physical interface assigned to each instance. Ethernet 1/2 is assigned to ftd1 and Ethernet 1/1 is assigned to ftd2.

ECMP group 1540 is assigned to ftd1 and ECMP group 1541 is assigned to ftd2.

```
secfw-3140(local-mgmt)# show portmanager switch forward-rules hardware mac-filter
      VLAN    SRC_PORT    PC_ID    SRC_ID    DST_PORT      PKT_CNT       DMAC
1        0          17        0        17          19         29164  0:0:0:0:0:0
2        0          19        0        19          17         67588  0:0:0:0:0:0
3        0           1        0       101        1541             0  a2:5b:83:0:0:15
4        0           1        0       101        1541          8181  ff:ff:ff:ff:ff:ff
5        0           2        0       102        1540             0  a2:5b:83:0:0:18
6        0           2        0       102        1540           431  ff:ff:ff:ff:ff:ff
7        0          17        0         0           0         11133  0:0:0:0:0:0
8        0          17        0         0           0             0  0:0:0:0:0:0
```

This command shows the internal switch-forwarding rule for two instances with shared physical interfaces assigned to two instances. Ethernet 1/1 is shared between ftd1 and ftd2.

ECMP group 1540 is assigned to ftd1 and ECMP group 1541 is assigned to ftd2.

MCAST group 4096 is used for replicating broadcast traffic between ftd1 and ftd2.

```
firepower-3140(local-mgmt)# show portmanager switch forward-rules hardware mac-filter
      VLAN    SRC_PORT    PC_ID    SRC_ID    DST_PORT      PKT_CNT       DMAC
1        0          17        0        17          19          2268  0:0:0:0:0:0
2        0          19        0        19          17          4844  0:0:0:0:0:0
3        0           1        0       101        1541             0  a2:5b:83:0:0:9
4        0           1        0       101        4096           546  ff:ff:ff:ff:ff:ff
5        0           1        0       101        1540             0  a2:5b:83:0:0:c
6        0          17        0         0           0          1263  0:0:0:0:0:0
7        0          17        0         0           0             0  0:0:0:0:0:0
```

This command shows the internal switch-forwarding rule for two instances with shared subinterfaces assigned to both instances. Ethernet 1/1.2452 is shared between ftd1 and ftd2.

ECMP group 1540 is assigned to ftd1 and ECMP group 1541 is assigned to ftd2.

MCAST group 4097 is used for replicating broadcast traffic between ftd1 and ftd2.

```
firepower-3140(local-mgmt)# show portmanager switch forward-rules hardware mac-filter
      VLAN   SRC_PORT   PC_ID   SRC_ID   DST_PORT     PKT_CNT      DMAC
1        0         17       0       17         19       21305  0:0:0:0:0:0
2        0         19       0       19         17       50976  0:0:0:0:0:0
3     2452          1       0      101       1541         430  a2:5b:83:0:0:f
4     2452          1       0      101       4097           0  ff:ff:ff:ff:ff:ff
5     2452          1       0      101       1540           0  a2:5b:83:0:0:12
6        0         17       0        0          0       11038  0:0:0:0:0:0
7        0         17       0        0          0           0  0:0:0:0:0:0
```

show portmanager switch ecmp-groups detail

Use this command to list each Instance Ecmp-Vport-Physical port mapping detail.

Note Physical-Port 18 is the backplane uplink interface between the internal switch and the instance.

```
firepower-3140(local-mgmt)# show portmanager switch ecmp-groups detail
       ECMP-GROUP  VPORT      PHYSICAL-PORT
1      1536        256        18
2      1537        257        18
3      1538        258        18
4      1539        259        18
5      1540        260        18
6      1541        261        18
7      1542        262        18
8      1543        263        18
9      1544        264        18
10     1545        265        18
```

show portmanager switch mcast-groups detail

Use this command to list MCAST group membership details.

```
firepower-3140(local-mgmt)# show portmanager switch mcast-groups detail
       MCAST-GROUP
1      4096
       Member-ports
       Ethernet 1/1
       ECMP-ID 1541
       ECMP-ID 1540
```

show portmanager counters mcast-group

Use this command to check the MCAST group packet counter.

```
firepower-3140(local-mgmt)# show portmanager counters mcast-group 4096
PKT_CNT: 8106
```

show portmanager counters ecmp

Use this command to check the ECMP group packet counter.

```
firepower-3140(local-mgmt)# show portmanager counters ecmp 1541
PKT_CNT: 430
```

Multi-instance mode feature and version history

This reference documents the history of multi-instance mode capabilities, including instance limit increases, platform support expansions, and management interface improvements for Secure Firewall devices.

Feature	Minimum Firewall Management Center	Minimum Firewall Threat Defense	Details
Multi-instance mode for the Secure Firewall 3100.	7.4.1	7.4.1	You can deploy the Secure Firewall 3100 as a single device (*appliance mode*) or as multiple container instances (*multi-instance mode*). In multi-instance mode, you can deploy multiple container instances on a single chassis that act as completely independent devices. Note that in multi-instance mode, you upgrade the operating system and the firmware (*chassis upgrade*) separately from the container instances (*Firewall Threat Defense upgrade*). New/modified screens: • **Devices** > **Device Management** > **Add** > **Chassis** • **Devices** > **Device Management** > **Device** > **Chassis Manager** • **Devices** > **Platform Settings** > **New Policy** > **Chassis Platform Settings** • **Devices** > **Chassis Upgrade** New/modified Firewall Threat Defense CLI commands: **configure multi-instance network ipv4**, **configure multi-instance network ipv6** New/modified FXOS CLI commands: **create device-manager**, **set deploymode** Platform restrictions: Not supported on the Secure Firewall 3105.