Logical Devices on the Firepower 4100/9300

The Firepower 4100/9300 is a flexible security platform on which you can install one or more *logical devices*. Before you can add the threat defense to the management center, you must configure chassis interfaces, add a logical device, and assign interfaces to the device on the Firepower 4100/9300 chassis using the Secure Firewall chassis manager or the FXOS CLI. This chapter describes basic interface configuration and how to add a standalone or High Availability logical device using the Secure Firewall chassis manager. To use the FXOS CLI, see the FXOS CLI configuration guide. For more advanced FXOS procedures and troubleshooting, see the FXOS configuration guide.

About Interfaces

The Firepower 4100/9300 chassis supports physical interfaces, VLAN subinterfaces for container instances, and EtherChannel (port-channel) interfaces. EtherChannel interfaces can include up to 16 member interfaces of the same type.

Chassis Management Interface

The chassis management interface is used for management of the FXOS Chassis by SSH or chassis manager. This interface is separate from the mgmt-type interface that you assign to the logical devices for application management.

To configure parameters for this interface, you must configure them from the CLI. To view information about this interface in the FXOS CLI, connect to local management and show the management port:

Firepower # **connect local-mgmt**

Firepower(local-mgmt) # **show mgmt-port**

Note that the chassis management interface remains up even if the physical cable or SFP module are unplugged, or if the **mgmt-port shut** command is performed.

Note The chassis management interface does not support jumbo frames.

Interface Types

Physical interfaces, VLAN subinterfaces for container instances, and EtherChannel (port-channel) interfaces can be one of the following types:

- Data—Use for regular data. Data interfaces cannot be shared between logical devices, and logical devices cannot communicate over the backplane to other logical devices. For traffic on Data interfaces, all traffic must exit the chassis on one interface and return on another interface to reach another logical device.
- Data-sharing—Use for regular data. Only supported with container instances, these data interfaces can be shared by one or more logical devices/container instances (threat defense-using-management center only). Each container instance can communicate over the backplane with all other instances that share this interface. Shared interfaces can affect the number of container instances you can deploy. Shared interfaces are not supported for bridge group member interfaces (in transparent mode or routed mode), inline sets, passive interfaces, clusters, or failover links.
- Mgmt—Use to manage application instances. These interfaces can be shared by one or more logical devices to access external hosts; logical devices cannot communicate over this interface with other logical devices that share the interface. You can only assign one management interface per logical device. Depending on your application and manager, you can later enable management from a data interface; but you must assign a Management interface to the logical device even if you don't intend to use it after you enable data management. For information about the separate chassis management interface, see Chassis Management Interface, on page 1.

  Note Mgmt interface change will cause reboot of the logical device, for example one change mgmt from e1/1 to e1/2 will cause the logical device to reboot to apply the new management.

- Eventing—Use as a secondary management interface for threat defense-using-management center devices. To use this interface, you must configure its IP address and other parameters at the threat defense CLI. For example, you can separate management traffic from events (such as web events). See the management center configuration guide for more information. Eventing interfaces can be shared by one or more logical devices to access external hosts; logical devices cannot communicate over this interface with other logical devices that share the interface. If you later configure a data interface for management, you cannot use a separate eventing interface.

  Note A virtual Ethernet interface is allocated when each application instance is installed. If the application does not use an eventing interface, then the virtual interface will be in an admin down state.

  ```
  Firepower # show interface Vethernet775
  Firepower # Vethernet775 is down (Administratively down)
  Bound Interface is Ethernet1/10
  Port description is server 1/1, VNIC ext-mgmt-nic5
  ```

- Cluster—Use as the cluster control link for a clustered logical device. By default, the cluster control link is automatically created on Port-channel 48. The Cluster type is only supported on EtherChannel interfaces. For multi-instance clustering, you cannot share a Cluster-type interface across devices. You can add VLAN subinterfaces to the Cluster EtherChannel to provide separate cluster control links per cluster. If you add subinterfaces to a Cluster interface, you cannot use that interface for a native cluster. The device manager and Security Cloud Control does not support clustering.

Note This chapter discusses *FXOS* VLAN subinterfaces only. You can separately create subinterfaces within the threat defense application. See FXOS Interfaces vs. Application Interfaces, on page 4 for more information.

See the following table for interface type support for the threat defense and ASA applications in standalone and cluster deployments.

Table 1: Interface Type Support

Application		Data	Data: Subinterface	Data-Sharing	Data-Sharing: Subinterface	Mgmt	Eventing	Cluster (EtherChannel only)	Cluster: Subinterface
Threat Defense	Standalone Native Instance	**Yes**	—	—	—	**Yes**	**Yes**	—	—
	Standalone Container Instance	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	—	—
	Cluster Native Instance	**Yes** (EtherChannel only for inter-chassis cluster)	—	—	—	**Yes**	**Yes**	**Yes**	—
	Cluster Container Instance	**Yes** (EtherChannel only for inter-chassis cluster)	—	—	—	**Yes**	**Yes**	**Yes**	**Yes**
ASA	Standalone Native Instance	**Yes**	—	—	—	**Yes**	—	**Yes**	—
	Cluster Native Instance	**Yes** (EtherChannel only for inter-chassis cluster)	—	—	—	**Yes**	—	**Yes**	—

FXOS Interfaces vs. Application Interfaces

The Firepower 4100/9300 manages the basic Ethernet settings of physical interfaces, VLAN subinterfaces for container instances, and EtherChannel (port-channel) interfaces. Within the application, you configure higher level settings. For example, you can only create EtherChannels in FXOS; but you can assign an IP address to the EtherChannel within the application.

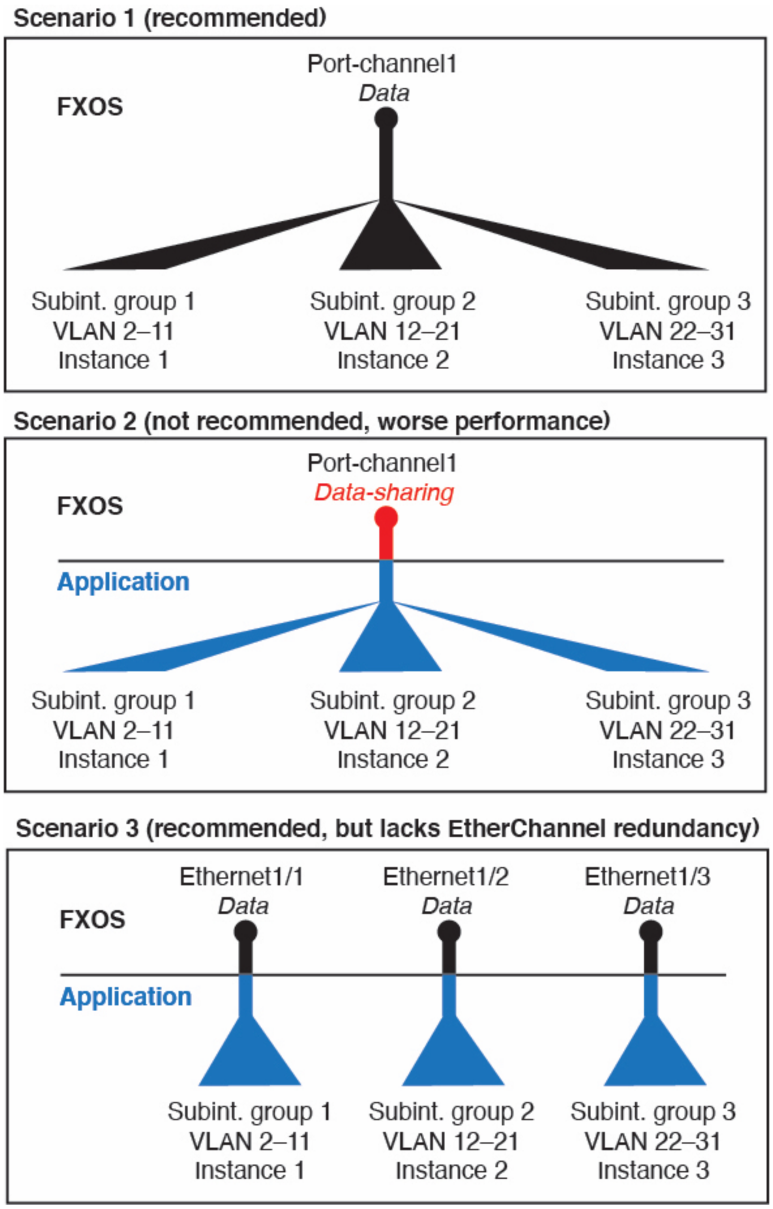
The following sections describe the interaction between FXOS and the application for interfaces.

VLAN Subinterfaces

For all logical devices, you can create VLAN subinterfaces within the application.

For container instances in standalone mode only, you can *also* create VLAN subinterfaces in FXOS. Multi-instance clusters do not support subinterfaces in FXOS except on the Cluster-type interface. Application-defined subinterfaces are not subject to the FXOS limit. Choosing in which operating system to create subinterfaces depends on your network deployment and personal preference. For example, to share a subinterface, you must create the subinterface in FXOS. Another scenario that favors FXOS 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 within the application, then you would have to share the parent interface in FXOS, which may not be desirable. See the following illustration that shows the three ways you can accomplish this scenario:

Figure 1: VLANs in FXOS vs. the Application for Container Instances



Independent Interface States in the Chassis and in the Application

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

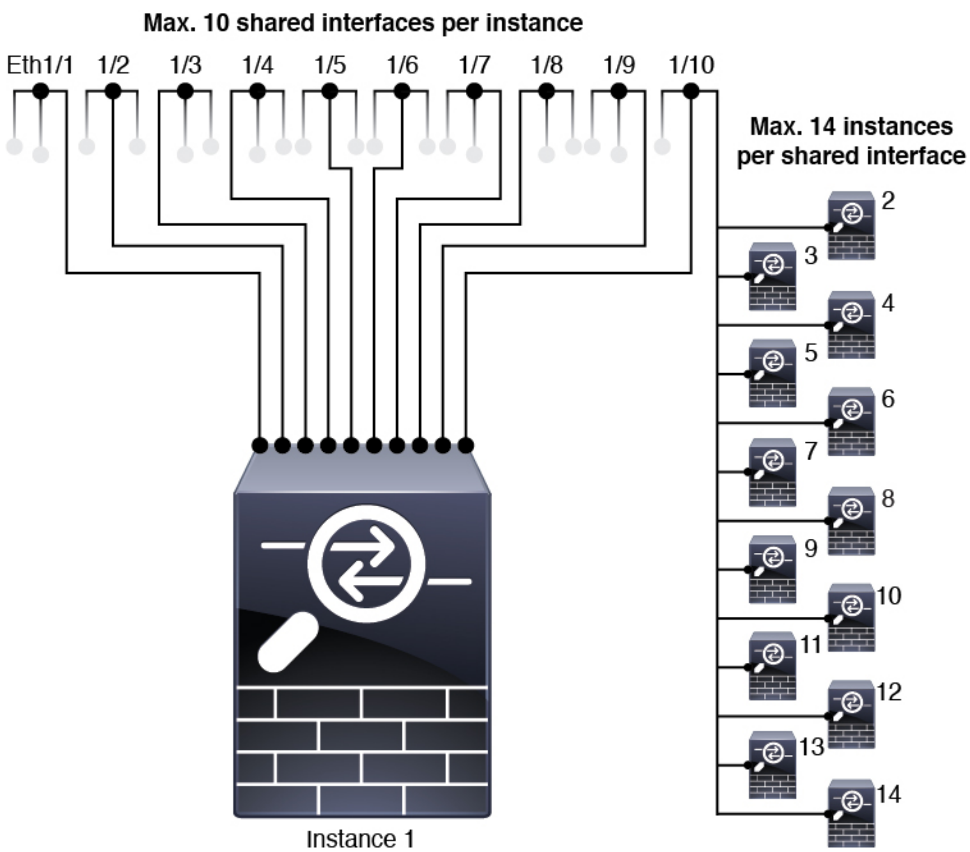
The default state of an interface within the application depends on the type of interface. For example, the physical interface or EtherChannel is disabled by default within the application, but a subinterface is enabled by default.

Shared Interface Scalability

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

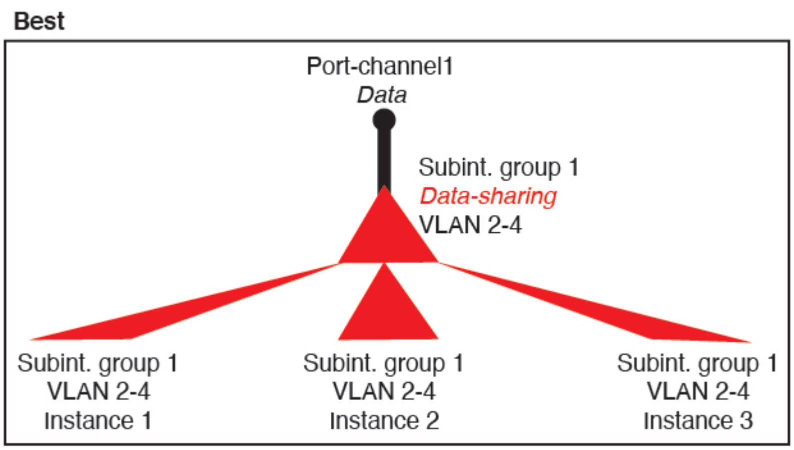
For optimal scalability of the forwarding table, share as few interfaces as possible. Instead, you can create up to 500 VLAN subinterfaces on one or more physical interfaces and then divide the VLANs among the container instances.

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.

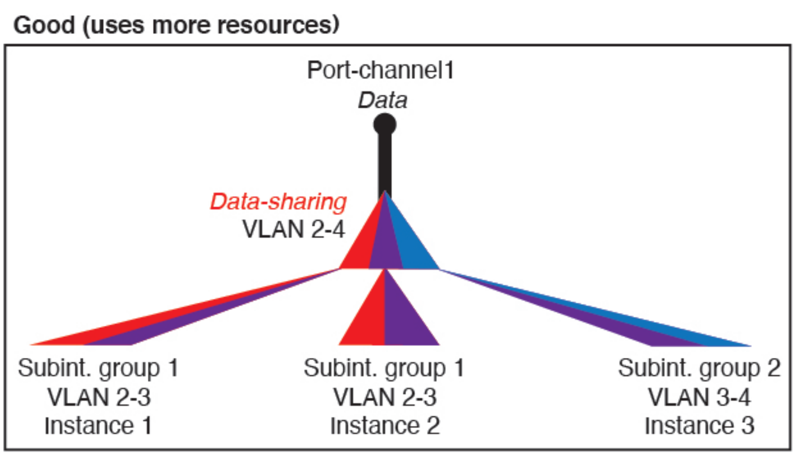
For example, create a large EtherChannel to bundle all of your like-kind interfaces together, and then share subinterfaces of that EtherChannel: Port-Channel1.2, 3, and 4 instead of Port-Channel2, Port-Channel3, and Port-Channel4. When you share subinterfaces from a single parent, the VLAN group table 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).

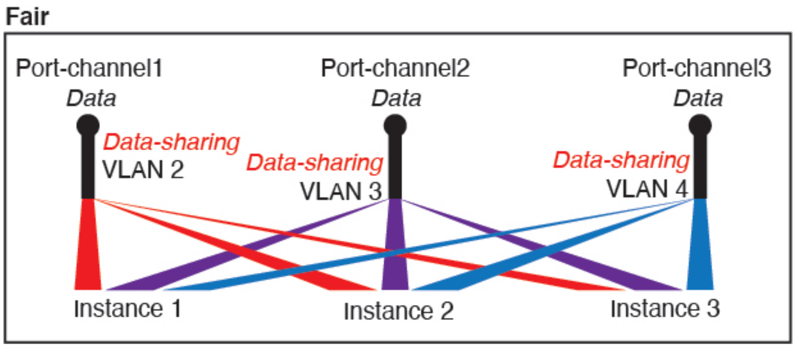
Figure 3: Good: Sharing Multiple Subinterface Groups on One Parent



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.

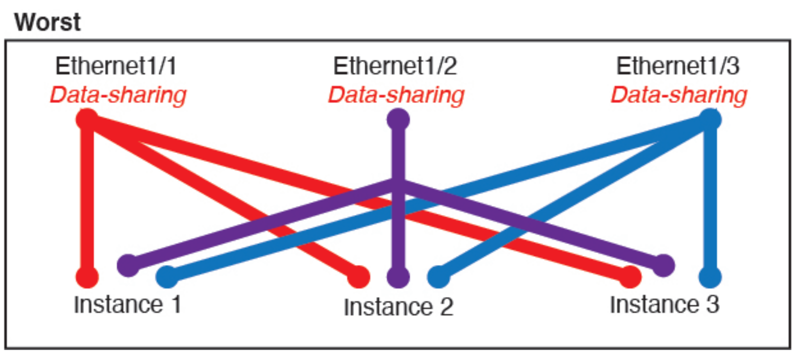
Figure 4: Fair: Shared Subinterfaces on Separate Parents



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

   This method uses the most forwarding table entries.

Figure 5: Worst: Shared Parent Interfaces



Shared Interface Usage Examples

See the following tables for examples of interface sharing and scalability. The below scenarios assume use of one physical/EtherChannel interface for management shared across all instances, and another physical or EtherChannel interface with dedicated subinterfaces for use with High Availability.

- Table 2: Physical/EtherChannel Interfaces and Instances on a Firepower 9300 with Three SM-44s, on page 9
- Table 3: Subinterfaces on One Parent and Instances on a Firepower 9300 with Three SM-44s, on page 10
- Table 4: Physical/EtherChannel Interfaces and Instances on a Firepower 9300 with One SM-44, on page 12
- Table 5: Subinterfaces on One Parent and Instances on a Firepower 9300 with One SM-44, on page 13

Firepower 9300 with Three SM-44s

The following table applies to three SM-44 security modules on a 9300 using only physical interfaces or EtherChannels. Without subinterfaces, the maximum number of interfaces are limited. Moreover, sharing multiple physical interfaces uses more forwarding table resources than sharing multiple subinterfaces.

Each SM-44 module can support up to 14 instances. Instances are split between modules as necessary to stay within limits.

Table 2: Physical/EtherChannel Interfaces and Instances on a Firepower 9300 with Three SM-44s

Dedicated Interfaces	Shared Interfaces	Number of Instances	% Forwarding Table Used
32: • 8 • 8 • 8 • 8	**0**	**4:** • Instance 1 • Instance 2 • Instance 3 • Instance 4	16%
30: • 15 • 15	**0**	**2:** • Instance 1 • Instance 2	14%
14: • 14 (1 ea.)	**1**	**14:** • Instance 1-Instance 14	46%
33: • 11 (1 ea.) • 11 (1 ea.) • 11 (1 ea.)	**3:** • 1 • 1 • 1	**33:** • Instance 1-Instance 11 • Instance 12-Instance 22 • Instance 23-Instance 33	98%
33: • *11 (1 ea.)* • *11 (1 ea.)* • *12 (1 ea.)*	***3:*** • *1* • *1* • *1*	***34:*** • *Instance 1-Instance 11* • *Instance 12-Instance 22* • *Instance 23-Instance 34*	*102%* DISALLOWED
30: • 30 (1 ea.)	**1**	**6:** • Instance 1-Instance 6	25%
30: • 10 (5 ea.) • 10 (5 ea.) • 10 (5 ea.)	**3:** • 1 • 1 • 1	**6:** • Instance 1-Instance2 • Instance 2-Instance 4 • Instance 5-Instance 6	23%

Dedicated Interfaces	Shared Interfaces	Number of Instances	% Forwarding Table Used
30: • 30 (6 ea.)	**2**	**5:** • Instance 1-Instance 5	28%
30: • 12 (6 ea.) • 18 (6 ea.)	**4:** • 2 • 2	**5:** • Instance 1-Instance2 • Instance 2-Instance 5	26%
24: • 6 • 6 • 6 • 6	**7**	**4:** • Instance 1 • Instance 2 • Instance 3 • Instance 4	44%
24: • 12 (6 ea.) • 12 (6 ea.)	**14:** • 7 • 7	**4:** • Instance 1-Instance2 • Instance 2-Instance 4	41%

The following table applies to three SM-44 security modules on a 9300 using subinterfaces on a single parent physical interface. For example, create a large EtherChannel to bundle all of your like-kind interfaces together, and then share subinterfaces of that EtherChannel. Sharing multiple physical interfaces uses more forwarding table resources than sharing multiple subinterfaces.

Each SM-44 module can support up to 14 instances. Instances are split between modules as necessary to stay within limits.

Table 3: Subinterfaces on One Parent and Instances on a Firepower 9300 with Three SM-44s

Dedicated Subinterfaces	Shared Subinterfaces	Number of Instances	% Forwarding Table Used
168: • 168 (4 ea.)	**0**	**42:** • Instance 1-Instance 42	33%
224: • 224 (16 ea.)	**0**	**14:** • Instance 1-Instance 14	27%
14: • 14 (1 ea.)	**1**	**14:** • Instance 1-Instance 14	46%

Dedicated Subinterfaces	Shared Subinterfaces	Number of Instances	% Forwarding Table Used
33: • 11 (1 ea.) • 11 (1 ea.) • 11 (1 ea.)	**3:** • 1 • 1 • 1	**33:** • Instance 1-Instance 11 • Instance 12-Instance 22 • Instance 23-Instance 33	98%
70: • 70 (5 ea.)	**1**	**14:** • Instance 1-Instance 14	46%
165: • 55 (5 ea.) • 55 (5 ea.) • 55 (5 ea.)	**3:** • 1 • 1 • 1	**33:** • Instance 1-Instance 11 • Instance 12-Instance 22 • Instance 23-Instance 33	98%
70: • 70 (5 ea.)	**2**	**14:** • Instance 1-Instance 14	46%
165: • 55 (5 ea.) • 55 (5 ea.) • 55 (5 ea.)	**6:** • 2 • 2 • 2	**33:** • Instance 1-Instance 11 • Instance 12-Instance 22 • Instance 23-Instance 33	98%
70: • 70 (5 ea.)	**10**	**14:** • Instance 1-Instance 14	46%
165: • *55 (5 ea.)* • *55 (5 ea.)* • *55 (5 ea.)*	***30:*** • *10* • *10* • *10*	***33:*** • *Instance 1-Instance 11* • *Instance 12-Instance 22* • *Instance 23-Instance 33*	*102%* DISALLOWED

Firepower 9300 with One SM-44

The following table applies to the Firepower 9300 with one SM-44 using only physical interfaces or EtherChannels. Without subinterfaces, the maximum number of interfaces are limited. Moreover, sharing multiple physical interfaces uses more forwarding table resources than sharing multiple subinterfaces.

The Firepower 9300 with one SM-44 can support up to 14 instances.

Table 4: Physical/EtherChannel Interfaces and Instances on a Firepower 9300 with One SM-44

Dedicated Interfaces	Shared Interfaces	Number of Instances	% Forwarding Table Used
32: • 8 • 8 • 8 • 8	**0**	**4:** • Instance 1 • Instance 2 • Instance 3 • Instance 4	16%
30: • 15 • 15	**0**	**2:** • Instance 1 • Instance 2	14%
14: • 14 (1 ea.)	**1**	**14:** • Instance 1-Instance 14	46%
14: • 7 (1 ea.) • 7 (1 ea.)	**2:** • **1** • **1**	**14:** • Instance 1-Instance 7 • Instance 8-Instance 14	37%
32: • 8 • 8 • 8 • 8	**1**	**4:** • Instance 1 • Instance 2 • Instance 3 • Instance 4	21%
32: • 16 (8 ea.) • 16 (8 ea.)	**2**	**4:** • Instance 1-Instance 2 • Instance 3-Instance 4	20%
32: • 8 • 8 • 8 • 8	**2**	**4:** • Instance 1 • Instance 2 • Instance 3 • Instance 4	25%

Dedicated Interfaces	Shared Interfaces	Number of Instances	% Forwarding Table Used
32: • 16 (8 ea.) • 16 (8 ea.)	**4:** • 2 • 2	**4:** • Instance 1-Instance 2 • Instance 3-Instance 4	24%
24: • 8 • 8 • 8	**8**	**3:** • Instance 1 • Instance 2 • Instance 3	37%
10: • 10 (2 ea.)	**10**	**5:** • Instance 1-Instance 5	69%
10: • 6 (2 ea.) • 4 (2 ea.)	**20:** • 10 • 10	**5:** • Instance 1-Instance 3 • Instance 4-Instance 5	59%
14: • *12 (2 ea.)*	***10***	***7:*** • *Instance 1-Instance 7*	*109%* DISALLOWED

The following table applies to the Firepower 9300 with one SM-44 using subinterfaces on a single parent physical interface. For example, create a large EtherChannel to bundle all of your like-kind interfaces together, and then share subinterfaces of that EtherChannel. Sharing multiple physical interfaces uses more forwarding table resources than sharing multiple subinterfaces.

The Firepower 9300 with one SM-44 can support up to 14 instances.

Table 5: Subinterfaces on One Parent and Instances on a Firepower 9300 with One SM-44

Dedicated Subinterfaces	Shared Subinterfaces	Number of Instances	% Forwarding Table Used
112: • 112 (8 ea.)	**0**	**14:** • Instance 1-Instance 14	17%
224: • 224 (16 ea.)	**0**	**14:** • Instance 1-Instance 14	17%
14: • 14 (1 ea.)	**1**	**14:** • Instance 1-Instance 14	46%

Dedicated Subinterfaces	Shared Subinterfaces	Number of Instances	% Forwarding Table Used
14: • 7 (1 ea.) • 7 (1 ea.)	**2:** • 1 • 1	**14:** • Instance 1-Instance 7 • Instance 8-Instance 14	37%
112: • 112 (8 ea.)	**1**	**14:** • Instance 1-Instance 14	46%
112: • 56 (8 ea.) • 56 (8 ea.)	**2:** • 1 • 1	**14:** • Instance 1-Instance 7 • Instance 8-Instance 14	37%
112: • 112 (8 ea.)	**2**	**14:** • Instance 1-Instance 14	46%
112: • 56 (8 ea.) • 56 (8 ea.)	**4:** • 2 • 2	**14:** • Instance 1-Instance 7 • Instance 8-Instance 14	37%
140: • 140 (10 ea.)	**10**	**14:** • Instance 1-Instance 14	46%
140: • 70 (10 ea.) • 70 (10 ea.)	**20:** • 10 • 10	**14:** • Instance 1-Instance 7 • Instance 8-Instance 14	37%

Viewing Shared Interface Resources

To view forwarding table and VLAN group usage, . For example:

Inline Set Link State Propagation for the Threat Defense

An inline set acts like a bump on the wire, and binds two interfaces together to slot into an existing network. This function allows the system to be installed in any network environment without the configuration of adjacent network devices. Inline interfaces receive all traffic unconditionally, but all traffic received on these interfaces is retransmitted out of an inline set unless explicitly dropped.

When you configure an inline set in the threat defense application and enable link state propagation, the threat defense sends inline set membership to the FXOS chassis. Link state propagation means that the chassis automatically brings down the second interface in the inline interface pair when one of the interfaces in an inline set goes down. When the downed interface comes back up, the second interface automatically comes back up, also. In other words, if the link state of one interface changes, the chassis senses the change and updates the link state of the other interface to match it. Note that the chassis requires up to 4 seconds to propagate link state changes. Link state propagation is especially useful in resilient network environments where routers are configured to reroute traffic automatically around network devices that are in a failure state.

Note Do not enable Hardware Bypass and link state propagation for the same inline set.

About Logical Devices

A logical device lets you run one application instance (either ASA or threat defense) and also one optional decorator application (Radware DefensePro) to form a service chain.

When you add a logical device, you also define the application instance type and version, assign interfaces, and configure bootstrap settings that are pushed to the application configuration.

Note For the Firepower 9300, you can install different application types (ASA and threat defense) on separate modules in the chassis. You can also run different versions of an application instance type on separate modules.

Standalone and Clustered Logical Devices

You can add the following logical device types:

- Standalone—A standalone logical device operates as a standalone unit or as a unit in a High Availability pair.
- Cluster—A clustered logical device lets you group multiple units together, providing all the convenience of a single device (management, integration into a network) while achieving the increased throughput and redundancy of multiple devices. Multiple module devices, like the Firepower 9300, support intra-chassis clustering. For the Firepower 9300, all three modules must participate in the cluster, for both native and container instances. The device manager does not support clustering.

Logical Device Application Instances: Container and Native

Application instances run in the following deployment types:

- Native instance—A native instance uses all of the resources (CPU, RAM, and disk space) of the security module/engine, so you can only install one native instance.
- Container instance—A container instance uses a subset of resources of the security module/engine, so you can install multiple container instances. Multi-instance capability is only supported for the threat defense using management center; it is not supported for the ASA or the threat defense using device manager.

Note Multi-instance capability is similar to ASA multiple context mode, although the implementation is different. Multiple context mode partitions a single application instance, while multi-instance capability allows independent container instances. Container instances allow hard resource separation, separate configuration management, separate reloads, separate software updates, and full threat defense feature support. Multiple context mode, due to shared resources, supports more contexts on a given platform. Multiple context mode is not available on the threat defense.

For the Firepower 9300, you can use a native instance on some modules, and container instances on the other module(s).

Container Instance Interfaces

To provide flexible physical interface use for container instances, you can create VLAN subinterfaces in FXOS and also share interfaces (VLAN or physical) between multiple instances. Native instances cannot use VLAN subinterfaces or shared interfaces. A multi-instance cluster cannot use VLAN subinterfaces or shared interfaces. An exception is made for the cluster control link, which can use a subinterface of the Cluster EtherChannel. See Shared Interface Scalability, on page 6 and Add a VLAN Subinterface for Container Instances, on page 35.

Note This document discusses *FXOS* VLAN subinterfaces only. You can separately create subinterfaces within the threat defense application. See FXOS Interfaces vs. Application Interfaces, on page 4 for more information.

How the Chassis Classifies Packets

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.

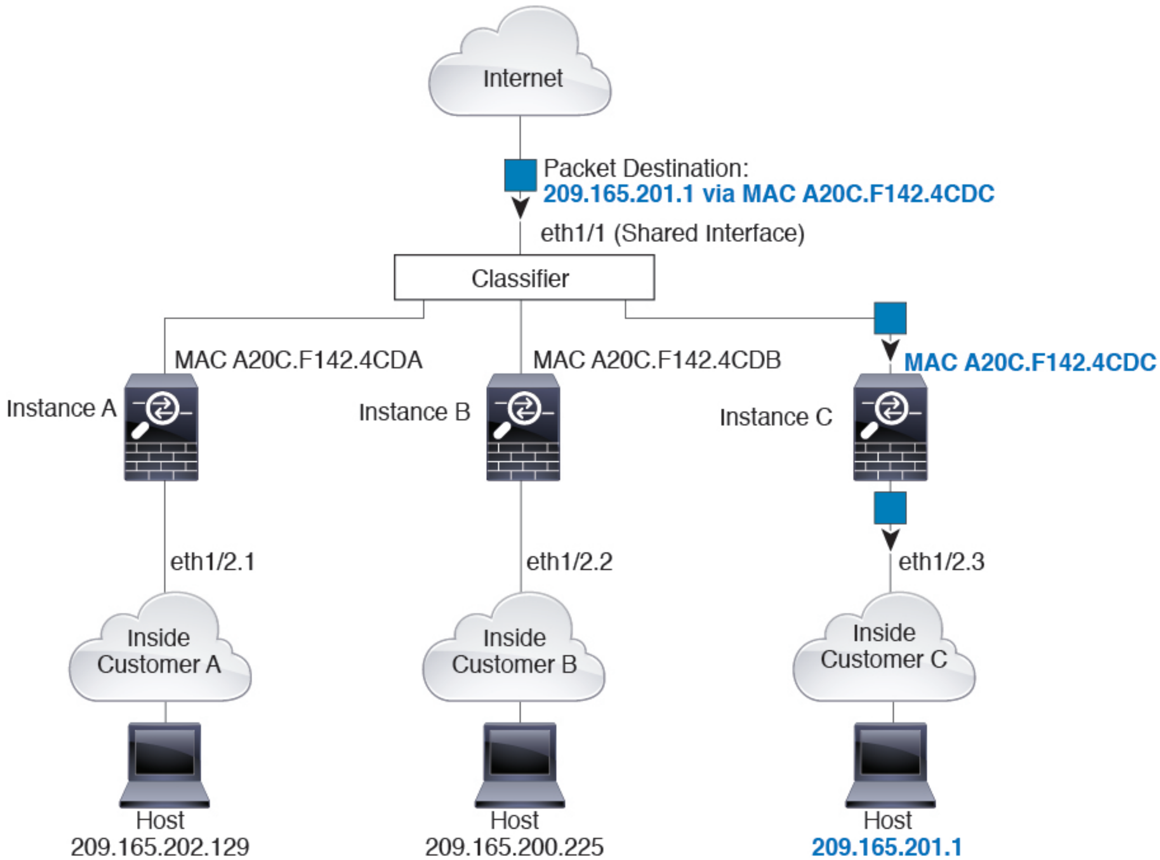
Note If the destination MAC address is a multicast or broadcast MAC address, the packet is duplicated and delivered to each instance.

Classification Examples

Packet Classification with a Shared Interface Using MAC Addresses

The following figure shows multiple instances sharing an outside interface. The classifier assigns the packet to Instance C because Instance C includes the MAC address to which the router sends the packet.

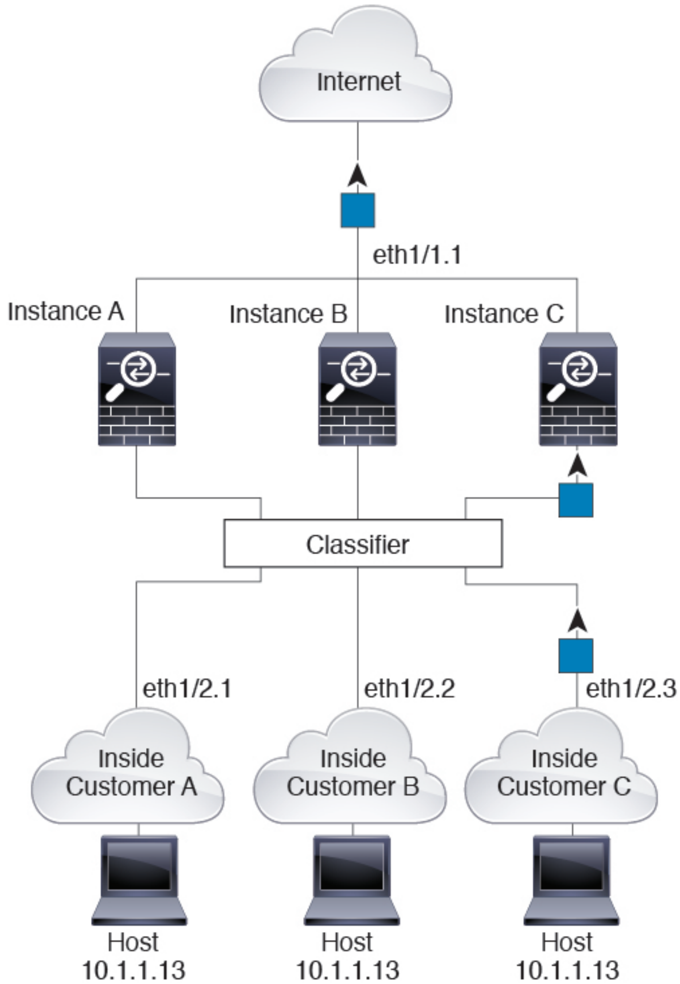
Figure 6: Packet Classification with a Shared Interface Using MAC Addresses



Incoming Traffic from Inside Networks

Note that all new incoming traffic must be classified, even from inside networks. The following figure shows a host on the Instance C inside network accessing the internet. The classifier assigns the packet to Instance C because the ingress interface is Ethernet 1/2.3, which is assigned to Instance C.

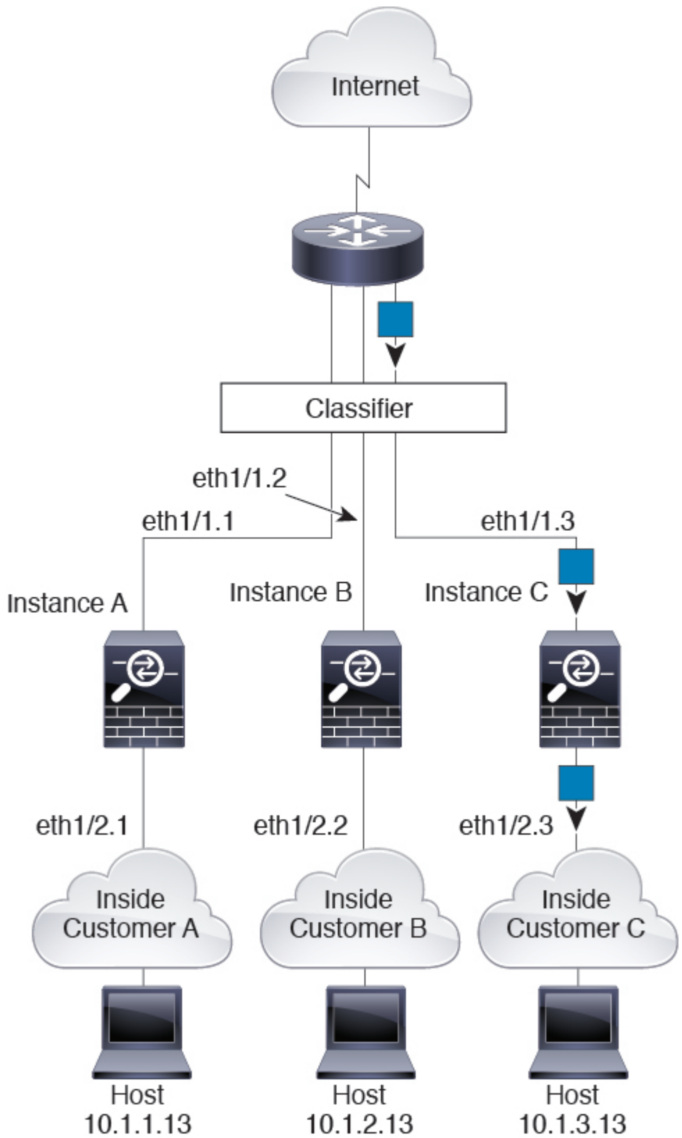
Figure 7: Incoming Traffic from Inside Networks



Transparent Firewall Instances

For transparent firewalls, you must use unique interfaces. The following figure shows a packet destined to a host on the Instance C inside network from the internet. The classifier assigns the packet to Instance C because the ingress interface is Ethernet 1/2.3, which is assigned to Instance C.

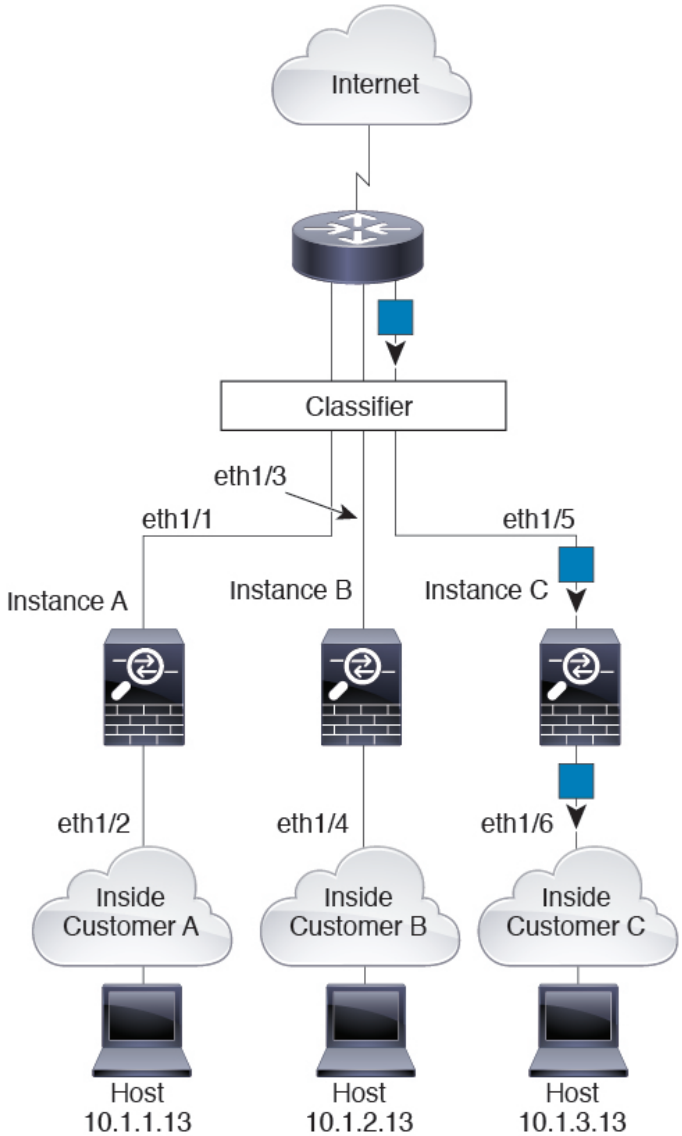
Figure 8: Transparent Firewall Instances



Inline Sets

For inline sets, you must use unique interfaces and they must be physical interfaces or EtherChannels. The following figure shows a packet destined to a host on the Instance C inside network from the internet. The classifier assigns the packet to Instance C because the ingress interface is Ethernet 1/5, which is assigned to Instance C.

Figure 9: Inline Sets

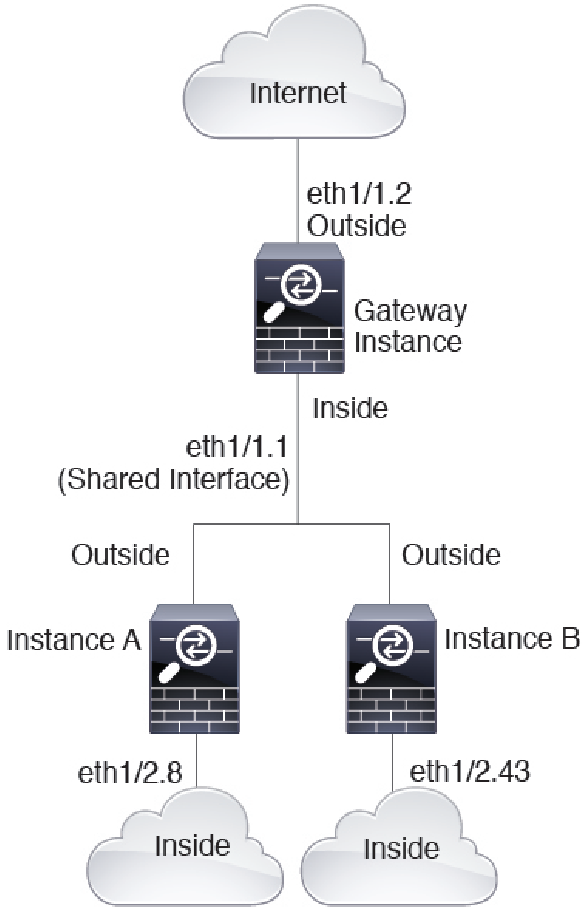


Cascading Container Instances

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 10: Cascading Instances



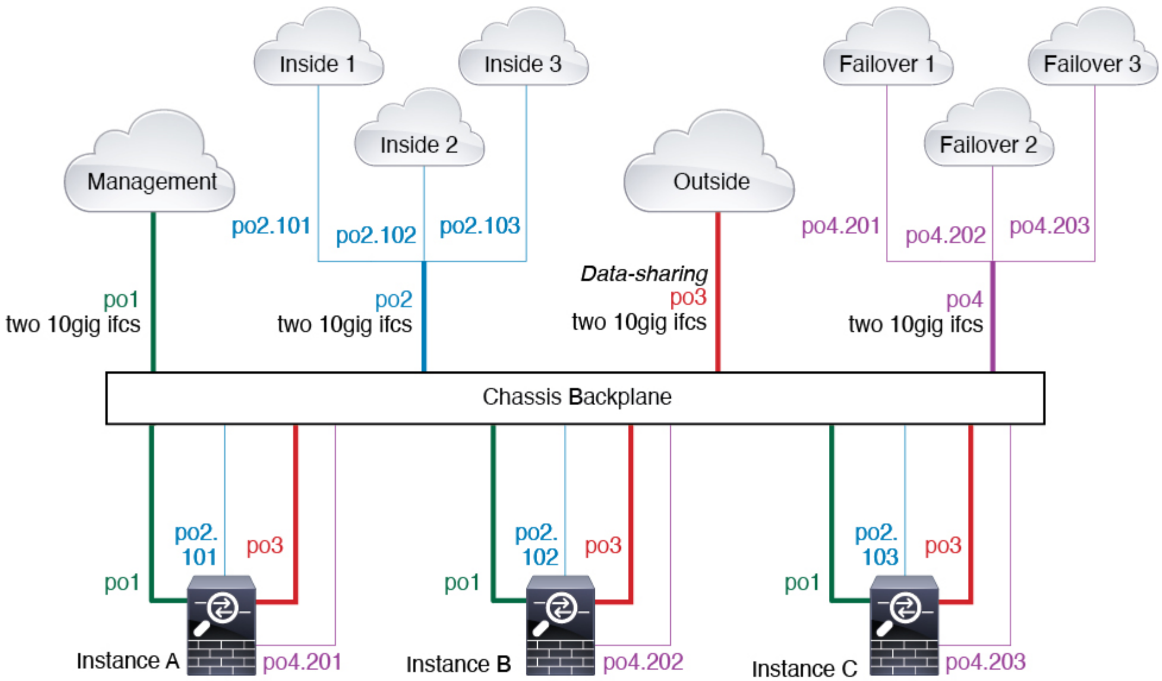
> **Note** Do not use cascading instances (using a shared interface) with High Availability. After a failover occurs and the standby unit rejoins, MAC addresses can overlap temporarily and cause an outage. You should instead use unique interfaces for the gateway instance and inside instance using an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

Typical Multi-Instance Deployment

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

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

- Outside—All instances use the Port-Channel3 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-Channel4 (data type). This EtherChannel includes two 10 Gigibit Ethernet interfaces. Each subinterface is on a separate network.



Automatic MAC Addresses for Container Instance Interfaces

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

Container Instance Resource Management

To specify resource usage per container instance, create one or more resource profiles in FXOS. When you deploy the logical device/application instance, you specify the resource profile that you want to use. 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. To view the available resources per model, see Requirements and Prerequisites for Container Instances, on page 26. To add a resource profile, see Add a Resource Profile for Container Instances, on page 36.

Performance Scaling Factor for Multi-Instance Capability

The maximum throughput (connections, VPN sessions, and TLS proxy sessions) for a platform is calculated for a native instance'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 a container instance with 50% of the cores, then you should initially calculate 50% of the throughput. Moreover, the throughput available to a container instance may be less than that available to a native instance.

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

Container Instances and High Availability

You can use High Availability using a container instance on 2 separate chassis; for example, if you have 2 chassis, each with 10 instances, you can create 10 High Availability pairs. Note that High Availability is not configured in FXOS; configure each High Availability pair in the application manager.

For detailed requirements, see Requirements and Prerequisites for High Availability, on page 27 and Add a High Availability Pair, on page 39.

Container Instances and Clustering

You can create a cluster of container instances using one container instance per security module/engine.

Licenses for Container Instances

All licenses are consumed per security engine/chassis (for the Firepower 4100) or per security module (for the Firepower 9300), and not per container instance. See the following details:

- Essentials licenses are automatically assigned: one per security module/engine.
- Feature licenses are manually assigned to each instance; but you only consume one license per feature per security module/engine. For example, for the Firepower 9300 with 3 security modules, you only need

one URL Filtering license per module for a total of 3 licenses, regardless of the number of instances in use.

For example:

Table 6: Sample License Usage for Container Instances on a Firepower 9300

Firepower 9300	Instance	Licenses
Security Module 1	Instance 1	Essentials, URL Filtering, Malware Defense
	Instance 2	Essentials, URL Filtering
	Instance 3	Essentials, URL Filtering
Security Module 2	Instance 4	Essentials, IPS
	Instance 5	Essentials, URL Filtering, Malware Defense, IPS
Security Module 3	Instance 6	Essentials, Malware Defense, IPS
	Instance 7	Essentials, IPS

Table 7: Total Number of Licenses

Essentials	URL Filtering	Malware Defense	IPS
3	2	3	2

Requirements and Prerequisites for Logical Devices

See the following sections for requirements and prerequisites.

Requirements and Prerequisites for Hardware and Software Combinations

The Firepower 4100/9300 supports multiple models, security modules, application types, and high availability and scalability features. See the following requirements for allowed combinations.

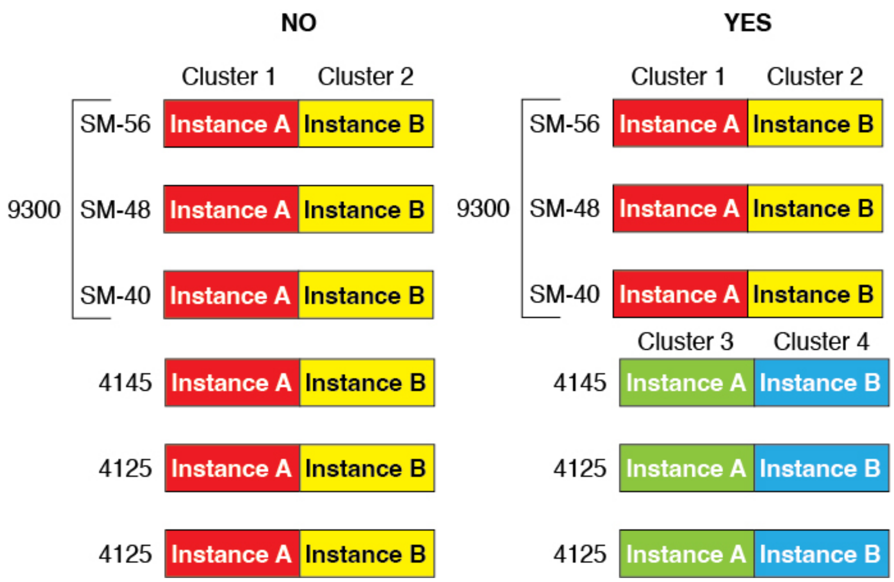
Firepower 9300 Requirements

The Firepower 9300 includes 3 security module slots and multiple types of security modules. See the following requirements:

- Security Module Types—You can install modules of different types in the Firepower 9300. For example, you can install the SM-48 as module 1, SM-40 as module 2, and SM-56 as module 3.
- Native instance Clustering—All security modules in the cluster, whether it is intra-chassis or inter-chassis, must be the same type. You can have different quantities of installed security modules in each chassis, although all modules present in the chassis must belong to the cluster including any empty slots. For

example, you can install 2 SM-40s in chassis 1, and 3 SM-40s in chassis 2. You cannot use clustering if you install 1 SM-48 and 2 SM-40s in the same chassis.

- Container instance Clustering—You can create a cluster using instances on different model types. For example, you can create a cluster using an instance on a Firepower 9300 SM-56, SM-48, and SM-40. You *cannot* mix the Firepower 9300 and the Firepower 4100 in the same cluster, however.



- High Availability—High Availability is only supported between same-type modules on the Firepower 9300. However, the two chassis can include mixed modules. For example, each chassis has an SM-40, SM-48, and SM-56. You can create High Availability pairs between the SM-40 modules, between the SM-48 modules, and between the SM-56 modules.

- ASA and threat defense application types—You can install different application types on separate modules in the chassis. For example, you can install ASA on module 1 and module 2, and threat defense on module 3.

- ASA or threat defense versions—You can run different versions of an application instance type on separate modules, or as separate container instances on the same module. For example, you can install the threat defense 6.3 on module 1, threat defense 6.4 on module 2, and threat defense 6.5 on module 3.

Firepower 4100 Requirements

The Firepower 4100 comes in multiple models. See the following requirements:

- Native and Container instances—When you install a container instance on a Firepower 4100, that device can only support other container instances. A native instance uses all of the resources for a device, so you can only install a single native instance on the device.

- Native instance Clustering—All chassis in the cluster must be the same model.

- Container instance Clustering—You can create a cluster using instances on different model types. For example, you can create a cluster using an instance on a Firepower 4145 and a 4125. You *cannot* mix the Firepower 9300 and the Firepower 4100 in the same cluster, however.

- High Availability—High Availability is only supported between same-type models.
- ASA and threat defense application types—The Firepower 4100 can only run a single application type.
- The threat defense container instance versions—You can run different versions of threat defense as separate container instances on the same module.

Requirements and Prerequisites for Container Instances

For information about high-availability or clustering requirements with multi-instance, see Requirements and Prerequisites for High Availability, on page 27 and see Requirements and Prerequisites for Clustering, on page 27.

Supported Application Types

- The threat defense using management center

Maximum Container Instances and Resources per Model

For each container instance, you can specify the number of CPU cores to assign to the instance. RAM is dynamically allocated according to the number of cores, and disk space is set to 40 GB per instance.

Table 8: Maximum Container Instances and Resources per Model

Model	Max. Container Instances	Available CPU Cores	Available RAM	Available Disk Space
Firepower 4112	3	22	78 GB	308 GB
Firepower 4115	7	46	162 GB	308 GB
Firepower 4125	10	62	162 GB	644 GB
Firepower 4140	7	70	222 GB	311.8 GB

Model	Max. Container Instances	Available CPU Cores	Available RAM	Available Disk Space
Firepower 4145	14	86	344 GB	608 GB
Firepower 9300 SM-40 security module	13	78	334 GB	1359 GB
Firepower 9300 SM-48 security module	15	94	334 GB	1341 GB
Firepower 9300 SM-56 security module	18	110	334 GB	1314 GB

Management Center Requirements

For all instances on a Firepower 4100 chassis or Firepower 9300 module, you must use the same management center due to the licensing implementation.

Requirements and Prerequisites for High Availability

- The two units in a High Availability Failover configuration must:
  - Be on a separate chassis; intra-chassis High Availability for the Firepower 9300 is not supported.
  - Be the same model.
  - Have the same interfaces assigned to the High Availability logical devices.
  - Have the same number and types of interfaces. All interfaces must be preconfigured in FXOS identically before you enable High Availability.
- High Availability is only supported between same-type modules on the Firepower 9300; but the two chassis can include mixed modules. For example, each chassis has an SM-56, SM-48, and SM-40. You can create High Availability pairs between the SM-56 modules, between the SM-48 modules, and between the SM-40 modules.
- For container instances, each unit must use the same resource profile attributes.
- For container instances: 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.
- For other High Availability system requirements, see High Availability System Requirements.

Requirements and Prerequisites for Clustering

Cluster Model Support

The Threat Defense supports clustering on the following models:

- Firepower 9300— You can include up to 16 nodes in the cluster. For example, you can use 1 module in 16 chassis, or 2 modules in 8 chassis, or any combination that provides a maximum of 16 modules. Supports clustering with multiple chassis and clustering isolated to security modules within one chassis.
- Firepower 4100—Supported for up to 16 nodes using clustering with multiple chassis.

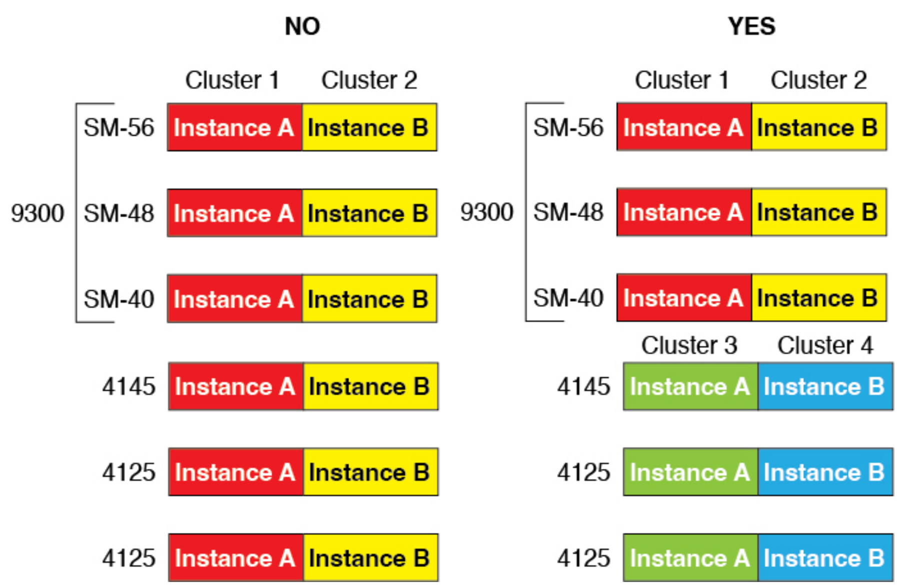
User Roles

- Admin
- Access Admin
- Network Admin

Clustering Hardware and Software Requirements

All chassis in a cluster:

- Native instance clustering—For the Firepower 4100: All chassis must be the same model. For the Firepower 9300: All security modules must be the same type. For example, if you use clustering, all modules in the Firepower 9300 must be SM-40s. You can have different quantities of installed security modules in each chassis, although all modules present in the chassis must belong to the cluster including any empty slots.
- Container instance clustering—We recommend that you use the same security module or chassis model for each cluster instance. However, you can mix and match container instances on different Firepower 9300 security module types or Firepower 4100 models in the same cluster if required. You cannot mix Firepower 9300 and 4100 instances in the same cluster. For example, you can create a cluster using an instance on a Firepower 9300 SM-56, SM-48, and SM-40. Or you can create a cluster on a Firepower 4145 and a 4125.

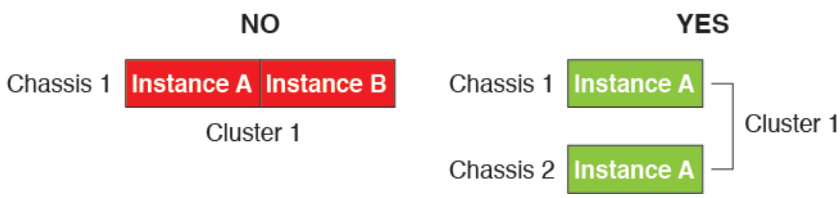


- Must run the identical FXOS and application software except at the time of an image upgrade. Mismatched software versions can lead to poor performance, so be sure to upgrade all nodes in the same maintenance window.

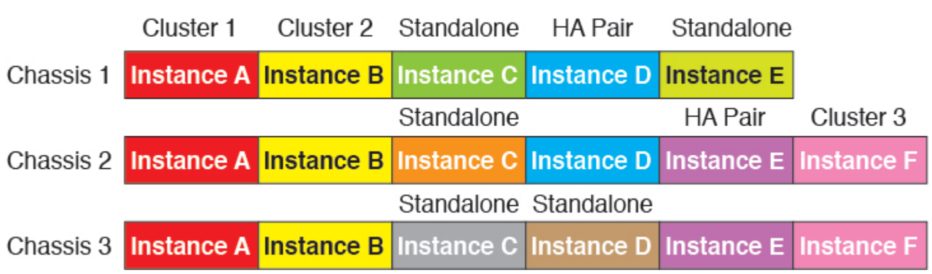
- Must include the same interface configuration for interfaces you assign to the cluster, such as the same Management interface, EtherChannels, active interfaces, speed and duplex, and so on. You can use different network module types on the chassis as long as the capacity matches for the same interface IDs and interfaces can successfully bundle in the same spanned EtherChannel. Note that all data interfaces must be EtherChannels in clusters with multiple chassis. If you change the interfaces in FXOS after you enable clustering (by adding or removing interface modules, or configuring EtherChannels, for example), then perform the same changes on each chassis, starting with the data nodes, and ending with the control node.
- Must use the same NTP server. For threat defense, the management center must also use the same NTP server. Do not set the time manually.

Multi-Instance Clustering Requirements

- No intra-security-module/engine clustering—For a given cluster, you can only use a single container instance per security module/engine. You cannot add 2 container instances to the same cluster if they are running on the same module.

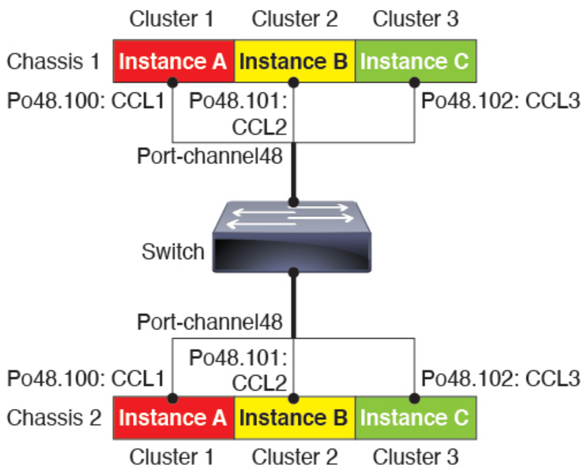


- Mix and match clusters and standalone instances—Not all container instances on a security module/engine need to belong to a cluster. You can use some instances as standalone or High Availability nodes. You can also create multiple clusters using separate instances on the same security module/engine.



- All 3 modules in a Firepower 9300 must belong to the cluster—For the Firepower 9300, a cluster requires a single container instance on all 3 modules. You cannot create a cluster using instances on module 1 and 2, and then use a native instance on module 3, or example.

- Match resource profiles—We recommend that each node in the cluster use the same resource profile attributes; however, mismatched resources are allowed when changing cluster nodes to a different resource profile, or when using different models.
- Dedicated cluster control link—For clusters with multiple chassis, each cluster needs a dedicated cluster control link. For example, each cluster can use a separate subinterface on the same cluster-type EtherChannel, or use separate EtherChannels.



- No shared interfaces—Shared-type interfaces are not supported with clustering. However, the same Management and Eventing interfaces can used by multiple clusters.
- No subinterfaces—A multi-instance cluster cannot use FXOS-defined VLAN subinterfaces. An exception is made for the cluster control link, which can use a subinterface of the Cluster EtherChannel.
- Mix chassis models—We recommend that you use the same security module or chassis model for each cluster instance. However, you can mix and match container instances on different Firepower 9300 security module types or Firepower 4100 models in the same cluster if required. You cannot mix Firepower 9300 and 4100 instances in the same cluster. For example, you can create a cluster using an instance on a Firepower 9300 SM-56, SM-48, and SM-40. Or you can create a cluster on a Firepower 4145 and a 4125.

- Maximum 6 nodes—You can use up to six container instances in a cluster.

Switch Requirements

- Be sure to complete the switch configuration and successfully connect all the EtherChannels from the chassis to the switch(es) before you configure clustering on the Firepower 4100/9300 chassis.
- For supported switch characteristics, see Cisco FXOS Compatibility.

Guidelines and Limitations for Logical Devices

See the following sections for guidelines and limitations.

Guidelines and Limitations for Interfaces

VLAN Subinterfaces

- This document discusses *FXOS* VLAN subinterfaces only. You can separately create subinterfaces within the threat defense application. See FXOS Interfaces vs. Application Interfaces, on page 4 for more information.
- Subinterfaces (and the parent interfaces) can only be assigned to container instances.

Note If you assign a parent interface to a container instance, it only passes untagged (non-VLAN) traffic. Do not assign the parent interface unless you intend to pass untagged traffic. For Cluster type interfaces, the parent interface cannot be used.

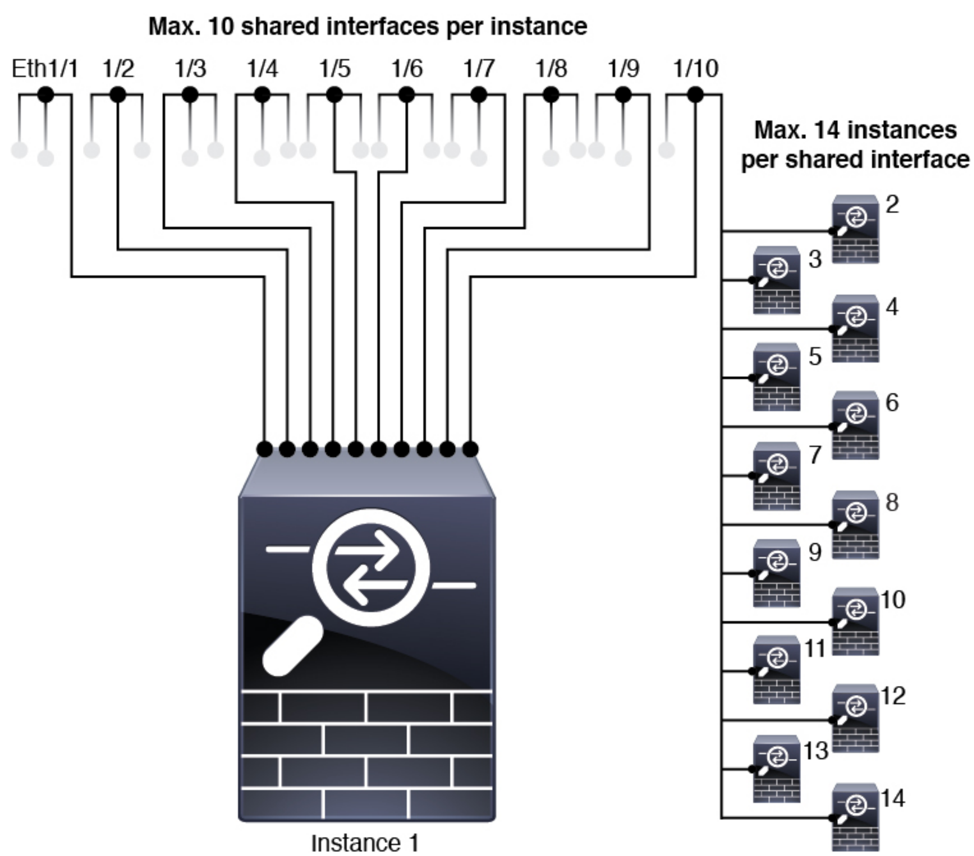
- Subinterfaces are supported on Data or Data-sharing type interfaces, as well as Cluster type interfaces. If you add subinterfaces to a Cluster interface, you cannot use that interface for a native cluster.

- For multi-instance clustering, FXOS subinterfaces are not supported on Data interfaces. However, subinterfaces are supported for the cluster control link, so you can use either a dedicated EtherChannel or a subinterface of an EtherChannel for the cluster control link. Note that *application*-defined subinterfaces are supported for Data interfaces.
- You can create up to 500 VLAN IDs.
- See the following limitations within the logical device application; keep these limitations in mind when planning your interface allocation.
  - You cannot use subinterfaces for an threat defense 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.

Data-sharing Interfaces

- You cannot use a data-sharing interface with a native instance.
- 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 in a cluster.

- See the following limitations within the logical device application; keep these limitations in mind when planning your interface allocation.
  - You cannot use a data-sharing interface with a transparent firewall mode device.
  - You cannot use a data-sharing interface with threat defense inline sets or passive interfaces.
  - You cannot use a data-sharing interface for the failover link.

Inline Sets for Threat Defense

- Supported for physical interfaces (both regular and breakout ports) and EtherChannels. Subinterfaces are not supported.
- Link state propagation is supported.
- Do not enable Hardware Bypass and link state propagation for the same inline set.

Hardware Bypass

- Supported for the threat defense; you can use them as regular interfaces for the ASA.
- The threat defense only supports Hardware Bypass with inline sets.
- Hardware Bypass-capable interfaces cannot be configured for breakout ports.
- You cannot include Hardware Bypass interfaces in an EtherChannel and use them for Hardware Bypass; you can use them as regular interfaces in an EtherChannel.
- Hardware Bypass is not supported with High Availability.
- Do not enable Hardware Bypass and link state propagation for the same inline set.

Default MAC Addresses

For native instances:

Default MAC address assignments depend on the type of interface.

- Physical interfaces—The physical interface uses the burned-in MAC address.
- EtherChannels—For an EtherChannel, all interfaces that are part of the channel group share the same MAC address. This feature makes the EtherChannel transparent to network applications and users, because they only see the one logical connection; they have no knowledge of the individual links. The port-channel interface uses a unique MAC address from a pool; interface membership does not affect the MAC address.

For container instances:

- 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. See Automatic MAC Addresses for Container Instance Interfaces, on page 22.

General Guidelines and Limitations

Firewall Mode

You can set the firewall mode to routed or transparent in the bootstrap configuration for the threat defense.

High Availability

- Configure high availability within the application configuration.
- You can use any data interfaces as the failover and state links. Data-sharing interfaces are not supported.

Multi-Instance

- Multi-instance capability with container instances is only available for the threat defense using management center.
- For threat defense container instances, a single management center must manage all instances on a security module/engine.
- For threat defense container instances, the following features are not supported:
  - Radware DefensePro link decorator
  - Management Center UCAPL/CC mode
  - Flow offload to hardware

Configure Interfaces

By default, physical interfaces are disabled. You can enable interfaces, add EtherChannels, add VLAN subinterfaces, and edit interface properties.

Configure a Physical Interface

You can physically enable and disable interfaces, as well as set the interface speed and duplex. To use an interface, it must be physically enabled in FXOS and logically enabled in the application.

Note For QSFPH40G-CUxM, auto-negotiation is always enabled by default and you cannot disable it.

Before you begin

- Interfaces that are already a member of an EtherChannel cannot be modified individually. Be sure to configure settings before you add it to the EtherChannel.

Add an EtherChannel (Port Channel)

An EtherChannel (also known as a port channel) can include up to 16 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. The Link Aggregation Control Protocol (LACP) aggregates interfaces by exchanging the Link Aggregation Control Protocol Data Units (LACPDUs) between two network devices.

You can configure each physical Data or Data-sharing interface in an EtherChannel to be:

- 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.

Non-data interfaces only support active mode.

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 Firepower 4100/9300 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 in the following situations:

- The EtherChannel is added as a data or management interface for a standalone logical device
- The EtherChannel is added as a management interface or cluster control link for a logical device that is part of a cluster
- The EtherChannel is added as a data interface for a logical device that is part of a cluster and at least one unit has joined the cluster

Note that the EtherChannel does not come up until you assign it to a logical device. If the EtherChannel is removed from the logical device or the logical device is deleted, the EtherChannel will revert to a **Suspended** or **Down** state.

Add a VLAN Subinterface for Container Instances

You can add between 250 and 500 VLAN subinterfaces to the chassis, depending on your network deployment. You can add up to 500 subinterfaces to your chassis.

For multi-instance clustering, you can only add subinterfaces to the Cluster-type interface; subinterfaces on data interfaces are not supported.

VLAN IDs per interface must be unique, and within a container 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 container instances. However, each subinterface still counts towards the limit even though it uses the same ID.

This document discusses *FXOS* VLAN subinterfaces only. You can separately create subinterfaces within the threat defense application. For more information on when to use FXOS subinterfaces vs. application subinterfaces, see FXOS Interfaces vs. Application Interfaces, on page 4.

Configure Logical Devices

Add a standalone logical device or a High Availability pair on the Firepower 4100/9300.

Add a Resource Profile for Container Instances

To specify resource usage per container instance, create one or more resource profiles. When you deploy the logical device/application instance, you specify the resource profile that you want to use. 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 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 security module/chassis model; see Requirements and Prerequisites for Container Instances, on page 26.

The chassis includes a default resource profile called "Default-Small," which includes the minimum number of cores. You can change the definition of this profile, and even delete it if it is not in use. Note that this profile is created when the chassis reloads and no other profile exists on the system.

Changing the resource profile after you assign it is disruptive. See the following guidelines:

- You cannot change the resource profile settings if it is currently in use. You must disable any instances that use it, then change the resource profile, and finally reenable the instance.
- If you change the resource profile settings after you add the threat defense instance to the management center, then update the inventory for each unit on the management center **Devices** > **Device Management** > **Device** > **System** > **Inventory** dialog box.
- If you assign a different profile to an instance, it reboots.
- If you assign a different profile to instances in an established high-availability pair, which requires the profile to be the same on both units, you must:
  1. Break high availability.

2. Assign the new profile to both units.

3. Re-establish high availability.

- If you assign a different profile to instances in an established cluster, which allows mismatched profiles, then apply the new profile on the data nodes first; after they all come back up, you can apply the new profile to the control node.

Add a Standalone Threat Defense

Standalone logical devices work either alone or in a High Availability pair. On the Firepower 9300 with multiple security modules, you can deploy either a cluster or standalone devices. The cluster must use all modules, so you cannot mix and match a 2-module cluster plus a single standalone device, for example.

You can use native instances on some modules, and container instances on the other module(s).

Before you begin

- Download the application image you want to use for the logical device from Cisco.com, and then that image to the Firepower 4100/9300 chassis.

Note For the Firepower 9300, you can install different application types (ASA and threat defense) on separate modules in the chassis. You can also run different versions of an application instance type on separate modules.

- Configure a management interface to use with the logical device. The management interface is required. Note that this management interface is not the same as the chassis management port that is used only for chassis management.
- You can later enable management from a data interface; but you must assign a Management interface to the logical device even if you don't intend to use it after you enable data management. See the **configure network management-data-interface** command in the FTD command reference for more information.
- You must also configure at least one Data type interface. Optionally, you can also create a firepower-eventing interface to carry all event traffic (such as web events). See Interface Types, on page 2 for more information.
- For container instances, if you do not want to use the default profile, add a resource profile according to Add a Resource Profile for Container Instances, on page 36.
- For container instances, before you can install a container instance for the first time, you must reinitialize the security module/engine so that the disk has the correct formatting. An existing logical device will be deleted and then reinstalled as a new device, losing any local application configuration. If you are replacing a native instance with container instances, you will need to delete the native instance in any case. You cannot automatically migrate a native instance to a container instance.
- Gather the following information:
  - Interface IDs for this device
  - Management interface IP address and network mask

- Gateway IP address
- management center IP address and/or NAT ID of your choosing
- DNS server IP address
- threat defense hostname and domain name

Procedure

See the management center configuration guide to add the threat defense as a managed device and start configuring your security policy.

Add a Standalone Threat Defense for the Cisco Security Cloud Control

You can use Security Cloud Control with both native and container instances. Standalone logical devices work either alone or in a High Availability pair.

Before you begin

- Download the application image you want to use for the logical device from Cisco.com, and then that image to the Firepower 4100/9300 chassis.

> **Note** For the Firepower 9300, you can install different application types (ASA and threat defense) on separate modules in the chassis. You can also run different versions of an application instance type on separate modules.

- Configure a management interface to use with the logical device. The management interface is required. Note that this management interface is not the same as the chassis management port that is used only for chassis management.
- You must also configure at least one Data type interface.
- You must onboard the FTD device in Security Cloud Control.
- Gather the following information:
  - Interface IDs for this device
  - Management interface IP address and network mask
  - Gateway IP address
  - DNS server IP address
  - Threat Defense hostname and domain name
  - Security Cloud Control onboard string
  - Threat Defense hostname and domain name

Procedure

Step 1 Save the configuration.

commit-buffer

The chassis deploys the logical device by downloading the specified software version and pushing the bootstrap configuration and management interface settings to the application instance. Check the status of the deployment using the **show app-instance** command. The application instance is running and ready to use when the **Admin State** is **Enabled** and the **Oper State** is **Online**.

Example:

```
Firepower /ssa/logical-device* # commit-buffer
Firepower /ssa/logical-device # exit
Firepower /ssa # show app-instance
App Name   Identifier Slot ID    Admin State Oper State       Running Version Startup Version
 Deploy Type Profile Name Cluster State   Cluster Role
---------- ---------- ---------- ----------- ---------------- --------------- ---------------
 ----------- ------------ --------------- ------------
asa        asa1       2          Disabled    Not Installed                    9.12.1
    Native                    Not Applicable  None
ftd        ftd1       1          Enabled     Online           7.3.0            7.3.0
  Container   Default-Small Not Applicable   None
```

Step 2 See the Security Cloud Control configuration guide to start configuring your security policy.

Add a High Availability Pair

Threat Defense High Availability (also known as failover) is configured within the application, not in FXOS. However, to prepare your chassis for high availability, see the following steps.

Before you begin

See Requirements and Prerequisites for High Availability, on page 27.

Procedure

Step 1 Allocate the same interfaces to each logical device.

Step 2 Allocate 1 or 2 data interfaces for the failover and state link(s).

These interfaces exchange high availability traffic between the 2 chassis. We recommend that you use a 10 GB data interface for a combined failover and state link. If you have available interfaces, you can use separate failover and state links; the state link requires the most bandwidth. You cannot use the management-type interface for the failover or state link. We recommend that you use a switch between the chassis, with no other device on the same network segment as the failover interfaces.

For container instances, data-sharing interfaces are not supported for the failover link. We recommend that you create subinterfaces on a parent interface or EtherChannel, and assign a subinterface for each instance to use as a failover link. Note that you must use all subinterfaces on the same parent as failover links. You cannot use one subinterface as a failover link and then use other subinterfaces (or the parent interface) as regular data interfaces.

Step 3 Enable High Availability on the logical devices. See High Availability.

Step 4 If you need to make interface changes after you enable High Availability, perform the changes on the standby unit first, and then perform the changes on the active unit.

Change an Interface on a Threat Defense Logical Device

You can allocate or unallocate an interface on the threat defense logical device. You can then sync the interface configuration in the management centerthe .

Adding a new interface, or deleting an unused interface has minimal impact on the threat defense configuration. However, deleting an interface that is used in your security policy will impact the configuration. Interfaces can be referenced directly in many places in the threat defense configuration, including access rules, NAT, SSL, identity rules, VPN, DHCP server, and so on. Policies that refer to security zones are not affected. You can also edit the membership of an allocated EtherChannel without affecting the logical device or requiring a sync on the management centerthe .

Deleting an interface will delete any configuration associated with that interface.

Before you begin

- Configure your interfaces, and add any EtherChannels according to Configure a Physical Interface, on page 34 and Add an EtherChannel (Port Channel), on page 35.
- If you want to add an already-allocated interface to an EtherChannel (for example, all interfaces are allocated by default to a cluster), you need to unallocate the interface from the logical device first, then add the interface to the EtherChannel. For a new EtherChannel, you can then allocate the EtherChannel to the device.
- For clustering or High Availability, make sure you add or remove the interface on all units before you sync the configuration in the management centerthe . We recommend that you make the interface changes on the data/standby unit(s) first, and then on the control/active unit. Note that new interfaces are added in an administratively down state, so they do not affect interface monitoring.
- In mult-instance mode, for changing a sub-interface with an another sub-interface with the same vlan tag, you must first remove all the configuration (including nameif config) of the interface and then unalloacte the interface from chassis manager. Once unallocated, add the new interface and then use sync interfaces from the management center.

Procedure

Step 1 Sync the interfaces in the management center.

a) Log into the management center.

b) Select **Devices** > **Device Management** and click **Edit** (✎) for your threat defense device. The **Interfaces** page is selected by default.

c) Click the **Sync Device** button on the top left of the **Interfaces** page.
d) After the changes are detected, you will see a red banner on the **Interfaces** page indicating that the interface configuration has changed. Click the **Click to know more** link to view the interface changes.
e) If you plan to delete an interface, manually transfer any interface configuration from the old interface to the new interface.

   Because you have not yet deleted any interfaces, you can refer to the existing configuration. You will have additional opportunity to fix the configuration after you delete the old interface and re-run the validation. The validation will show you all locations in which the old interface is still used.

f) Click **Validate Changes** to make sure your policy will still work with the interface changes.

   If there are any errors, you need to change your policy and rerun the validation.

g) Click **Save**.
h) Click **Deploy** > **Deployment**.
i) Select the devices and click **Deploy** to deploy the policy to the assigned devices. The changes are not active until you deploy them.

Step 2 Sync the interfaces again in the management centerthe .

Connect to the Console of the Application

Use the following procedure to connect to the console of the application.

Procedure

Step 1 Connect to the module CLI using a console connection or a Telnet connection.

connect module *slot_number* {**console** | **telnet**}

To connect to the security engine of a device that does not support multiple security modules, always use **1** as the *slot_number*.

The benefits of using a Telnet connection is that you can have multiple sessions to the module at the same time, and the connection speed is faster.

Example:

```
Firepower# connect module 1 console
Telnet escape character is '~'.
Trying 127.5.1.1...
Connected to 127.5.1.1.
Escape character is '~'.

CISCO Serial Over LAN:
Close Network Connection to Exit

Firepower-module1>
```

Step 2 Connect to the application console.

connect ftd *name*

To view the instance names, enter the command without a name.

Example:

```
Firepower-module1> connect ftd ftd1
Connecting to ftd(ftd-native) console... enter exit to return to bootCLI
[...]
>
```

Step 3 Exit the application console to the FXOS module CLI.

- Threat Defense—Enter **exit**

Step 4 Return to the supervisor level of the FXOS CLI.

Exit the console:

a) Enter **~**

   You exit to the Telnet application.

b) To exit the Telnet application, enter:

   telnet>**quit**

Exit the Telnet session:

a) Enter **Ctrl-], .**