Convert to Container (MI Mode) in Firepower 4200 with FTD 7.6

Contents

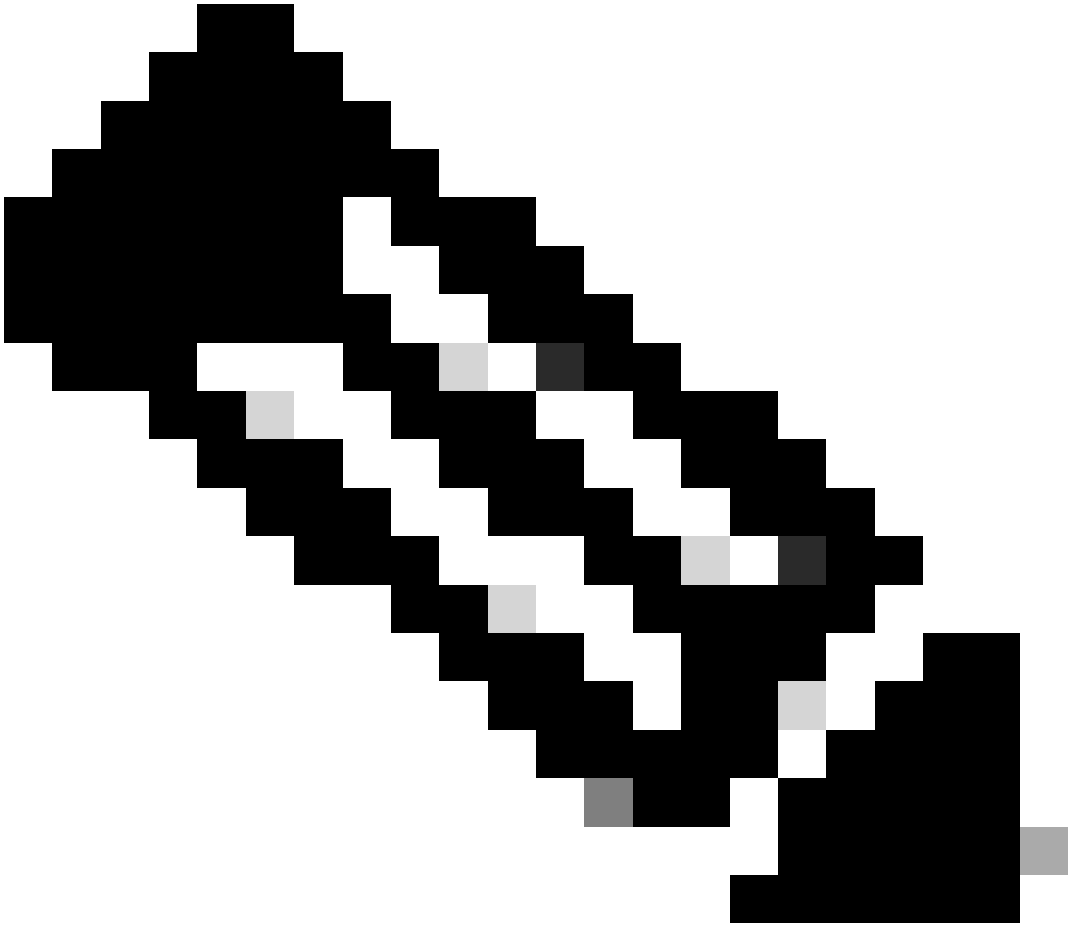
Introduction

This document describes how to configure a container (multi-instance mode) in Firepower 4200 firewall series with FTD 7.6 and related details.

Prerequisites, Supported Platforms, Licensing

Minimum Software & Hardware Platforms

Manager(s) and Version (s)	Application (ASA/FTD) and Minimum Version of Application	Supported Platforms
• FMC 7.6.0	• FTD 7.6.0	4200 Series 4215, 4225, 4245



Note: Multi-Instance is not supported with FDM on any platform.

Licensing

- Feature licenses are manually assigned to each instance, but you only consume one license per feature per 4200-series device.
  - For example, for one 4200 series with 3 FTD instances, you only need one URL license, regardless of the number of instances in use, provided you are on same FMC.
- All licenses are consumed per 4200 Series device and not per container instance, provided they are on same FMC. Therefore, for all instances on a 4200 Series devices, you are recommended to use the same FMC due to the licensing implementation.

Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

- FTD already supports Multi-Instance (MI) on 3100 models (as well as the 9300 and 4100 Series), but there is no support for 4200 series.
- 4200 models are supported only in Native Mode in FMC.
- There is no provision to create multiple instances in 7.4.x in 4200.
- Multi-Instance (MI) on 3100 was supported as of 7.4.1.
    - Instances can be created and managed using FMC (unlike the 9300 and 4100 Series, where FCM must be used).
    - FXOS can be updated, when in MI mode, via FMC's Upgrade Chassis GUI.
    - Converting to MI mode is done via a CLI.

What's New?

- You have the capability to provision and manage MI instances on the 4200 series.
- FMC - Single management solution for 4200 Series (MI mode) and FTD instances
- Allow for single and bulk conversion of native devices to MI mode on FMC for 3100 and 4200 series devices.
- Target Market: Enterprise/Large Enterprise - Internet Edge, Data Center

Platforms with FTD Multi-Instance Support

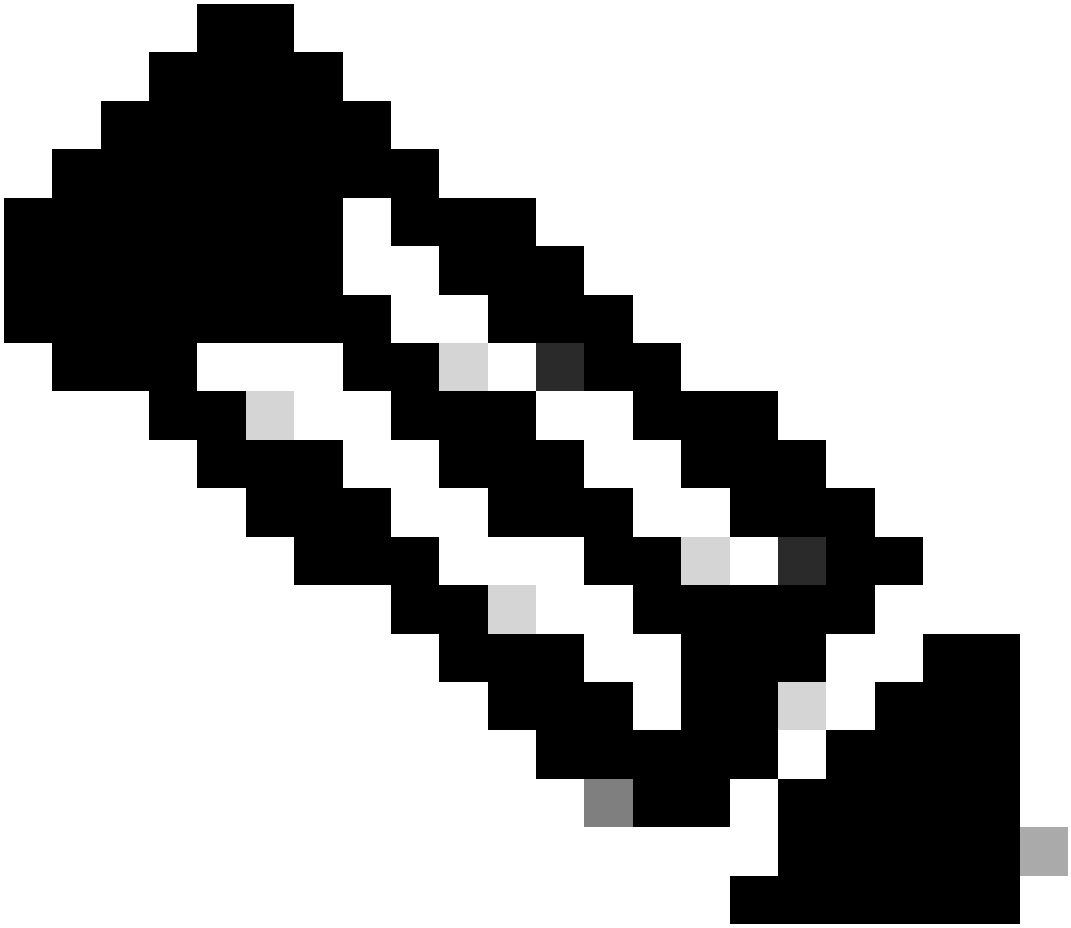
Platform	FTD Version	FTD Multi-Instance Support	Management Solution
Virtual	-	No	-
FPR1000	-	No	-
FPR2100	(not supported in 7.6)	No	-
3105		No	
3110, 3120, 3130, 3140	FTD 7.4.1	Yes	FMC
FPR4100	FTD 6.3.0	Yes	FCM & FMC
4215, 4225, 4245	FTD 7.6.0	Yes	FMC
FPR9300	FTD 6.3.0	Yes	FCM & FMC

Differences Between 3100 and 4200 Series

- 4200 has two management interfaces, allowing using one for management and the other for eventing.
    - Both Management1/1 and Management1/2 interfaces are bootstrapped to all FTD container instances.
    - One or both management interfaces can be used in MI mode.
        - Management1/1 for both Management and Events, or
        - Management1/1 could be used for management and Management1/2 for events, in which case:
            - Static routes need to be defined to route traffic using the Management 1/2 interface.
- Because of the larger size, more instances can be created on the 4200 than on the 3100.

Supported Deployments

- Manage 4200 Series (MI mode) with Standalone FTD Instance(s)
- Manage 4200 Series (MI mode) with HA FTD Instance(s)*



Note: For the FPR4100 Series, in case of FTD-HA, primary and secondary nodes must be on two different 4200 Series (MI mode) devices. Additionally, MI Clustering is not supported in this release.

Feature Description and Walkthrough

Changes to Multi-Instance Configuration in 7.6.0:

- Support for the 4200 Series in MI mode
- Changes in FMC, which pertain to MI Mode management of the 3100 Series as well:
  - Conversion of device from Native to MI mode in FMC
  - Readiness Checks to check if device can be converted to MI mode
  - Auto-register FTD instance in FMC after conversion

4200 Series Instance Specifications

Max Instances Support

Platform	Maximum Instance Count	Maximum Logical CPU Cores Supported
FP4215	10	62
FP4225	15	126
FP4245	34	254

Instance density is driven by 2 main factors:

1. The amount of CPU cores and the amount of disk space on a given platform

2. How many of these resources are available to provision to instances. The smallest instance size requires 3 physical CPU (6 logical) cores and 48 GB of disk space.

FTD Instance Sizes

Platform	4215	4225	4245
Total CPU cores	32	64	128
Available CPU cores for FTD	30	62	126
Total RAM (GiB)	222	445	875
FXOS RAM (GiB)	6	6	6
DMA RAM (GiB)	11	39	78
Available RAM for FTD (GiB)	7	7	7
Available Disk space for FTD (GiB)	660	864	1794
Max Instances	10	15	34

Lina (Data Plane) Snort Core Allocations

	4215	**4225**	**4245**			
Instance Size	**Data Plane Cores**	**Snort Cores**	**Data Plane Cores**	**Snort Cores**	**Data Plane Cores**	**Snort Cores**
6	2	2	2	2	2	2

8	2	4	2	4	2	4
10	4	4	4	4	4	4
12	4	6	4	6	4	6
14	6	8	6	6	6	6
16	6	8	6	6	8	8
18	8	10	8	8	8	10
20	8	10	8	8	10	10
22	10	12	10	10	10	12
24	12	12	10	10	10	12
26	12	14	12	12	12	12
28	14	14	12	14	12	14
30	14	16	14	14	14	14
32	14	16	14	16	14	16
34	16	16	16	16	16	16
36	16	18	16	18	16	18
38	18	18	18	18	18	18
40	18	20	18	20	18	20
42	20	20	20	20	20	20
44	20	22	20	22	20	22

46	22	22	22	22	22	22
48	22	24	22	24	22	24
50	24	24	24	24	24	24
52	24	26	24	26	24	26
54	26	26	26	26	24	26
56	26	28	26	28	26	28
58	28	28	28	28	28	28
60	28	30	28	39	28	30
62	30	30	30	30	30	30
64			30	32	30	32
66			30	34	30	34
68			32	34	32	34
70			32	36	32	36
72			34	36	34	36
74			34	38	34	38
76			36	38	36	38
78			36	40	36	40
80			38	40	38	40
82			38	42	38	42

84			40	42	40	42
86			40	44	40	44
88			42	44	42	44
90			42	46	42	46
92			44	46	44	46
94			44	48	44	48
96			46	48	46	48
98			46	50	46	50
100			48	50	48	50
102			48	52	48	52
104			50	52	50	52
106			50	54	50	54
108			52	54	52	54
110			52	56	52	56
112			54	56	54	56
114			54	58	54	58
116			56	58	56	58
118			56	60	56	60
120			58	60	58	60

122			58	62	58	62
124			60	62	60	62
128					60	64
130					60	66
132					62	66
134					62	68
136					64	68
138					64	70
140					66	70
142					66	72
144					68	72
146					68	74
148					70	74
150					70	76
152					72	76
154					72	78
156					74	78
158					74	80
254					120	130

Configure

Overview of Configuration

1. Register 4200 Series (Native mode) device in FMC.
2. New! On FMC, select and convert the device from Native to MI mode.
3. New! MI chassis auto registers to FMC after conversion.
4. Update Physical Interface(s).
5. Create FTD instance(s) and assign interface(s).
6. Create/Update/Delete Port channel and sub interfaces from FMC.
7. Configure platform settings.
8. Deploy configuration changes to device.
9. FTD instance(s) auto registers to FMC.

Convert 4200 Series to Multi-Instance Mode in FMC

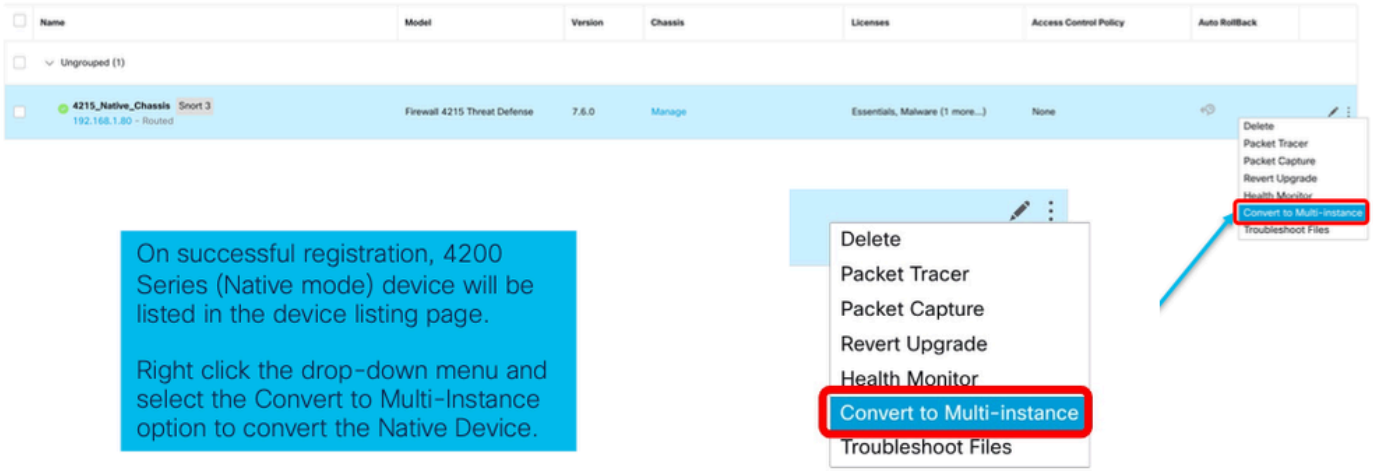
By default, 4200s are in native mode. To convert 4200 series to multi-instance mode in FMC:

1. Connect to the device and create a manager (already documented).
2. Register the native device to the FMC (already documented).
3. Convert to Multi-Instance using FMC.
4. On FMC, select the device(s) that needs to be converted to Multi-Instance and trigger the conversion. One or more than one device can be picked.

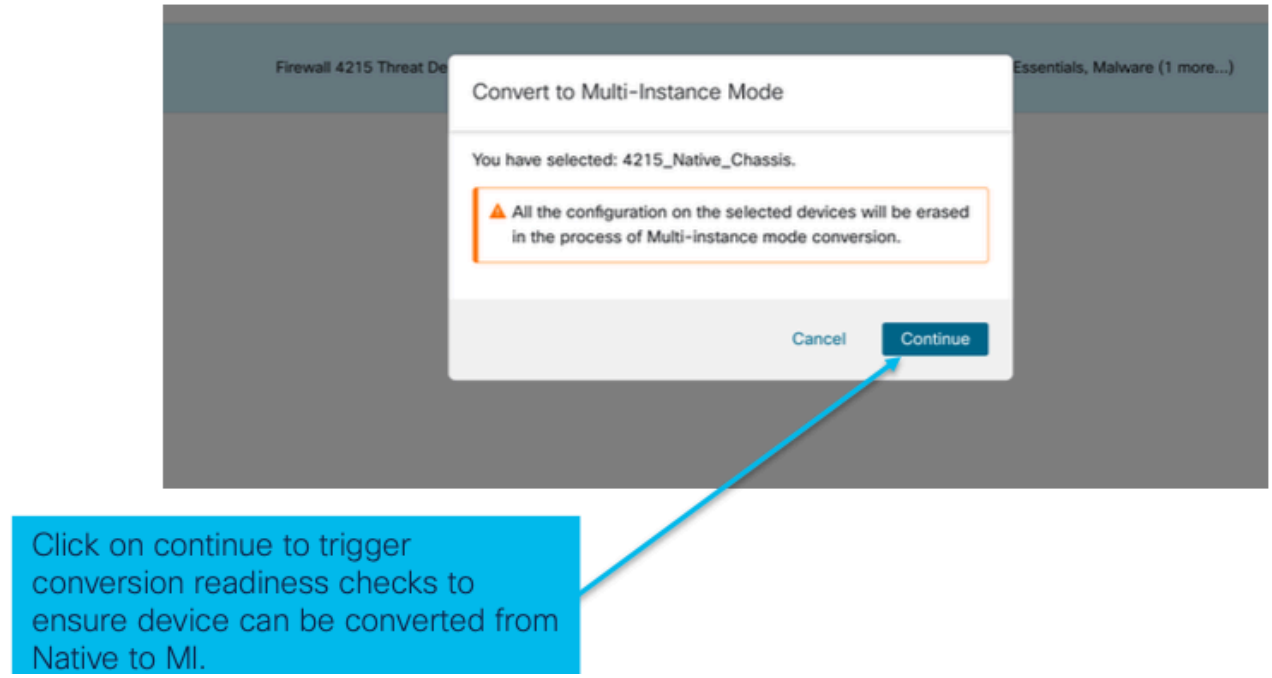
Note: Switching between native to MI mode resets ALL the configuration on the chassis. Converting from MI Mode to Native Mode is still via CLI.

Convert a Single Device

1. To start the conversion, navigate to **Devices > Device management**.

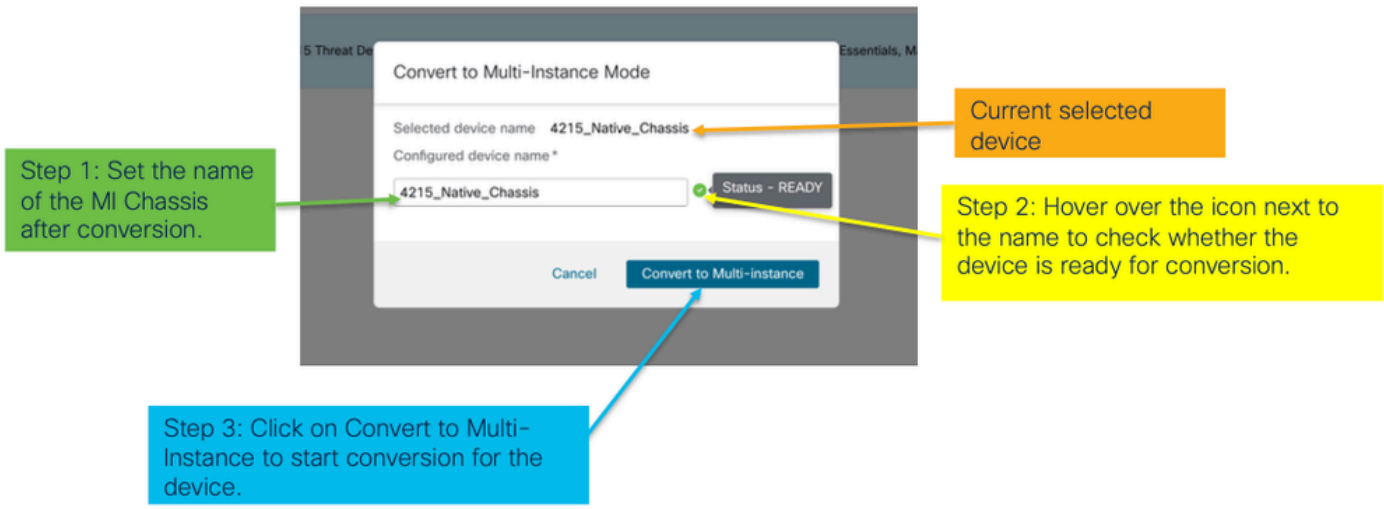


2. Validate selected device and click on **Continue**:



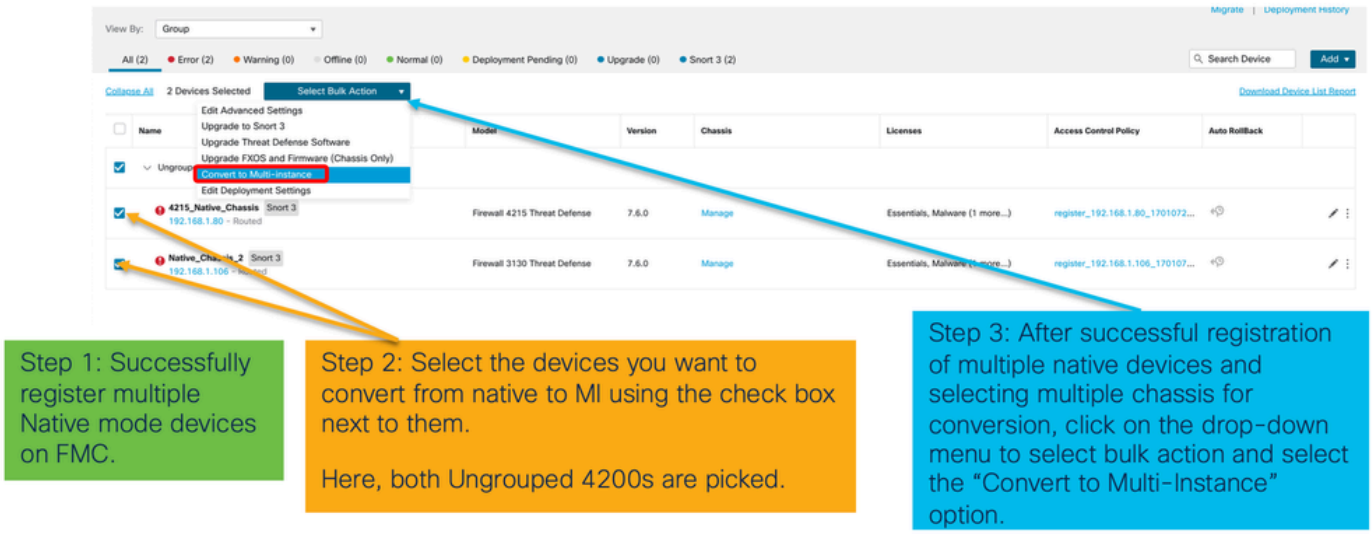
validate selected devices

3. Readiness check and initial conversion:

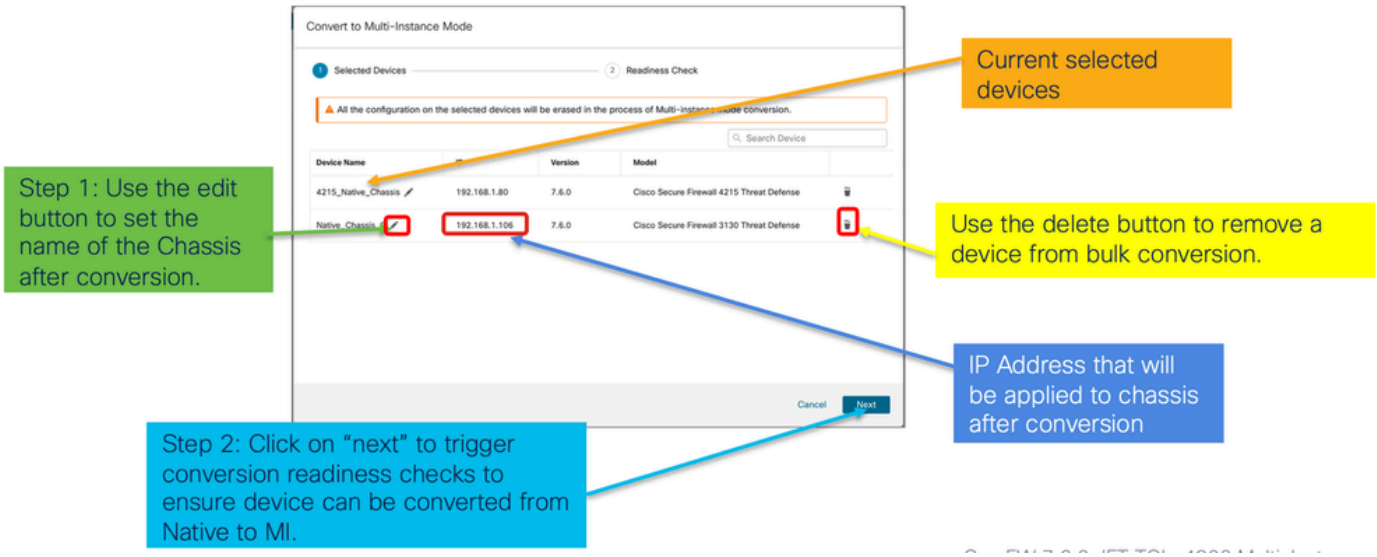


Convert More than One Device (Bulk Conversion)

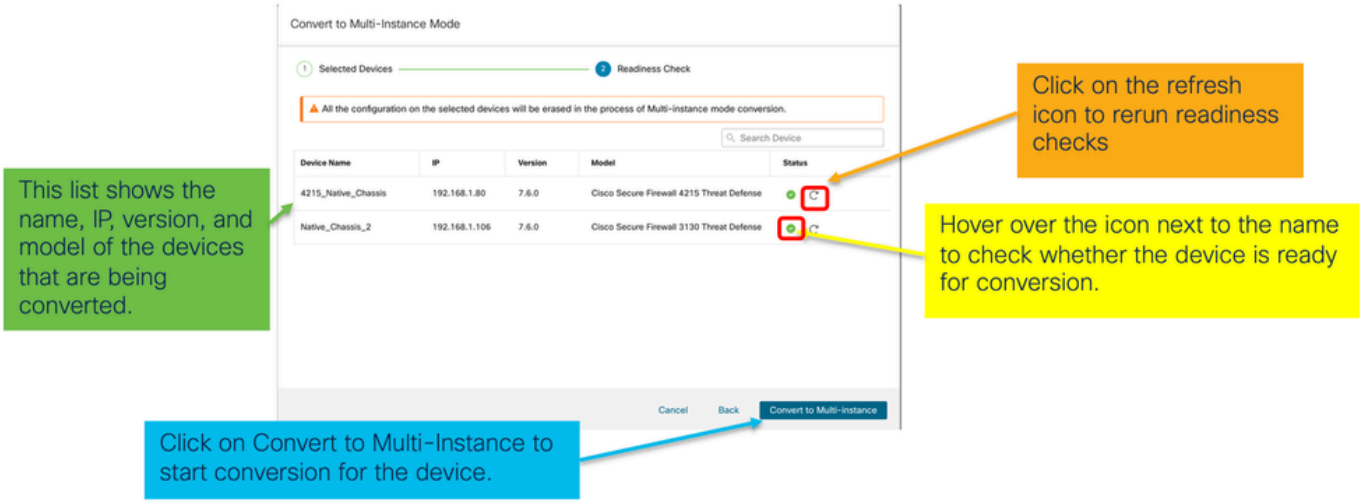
1. Select devices:



2. Confirm selection:

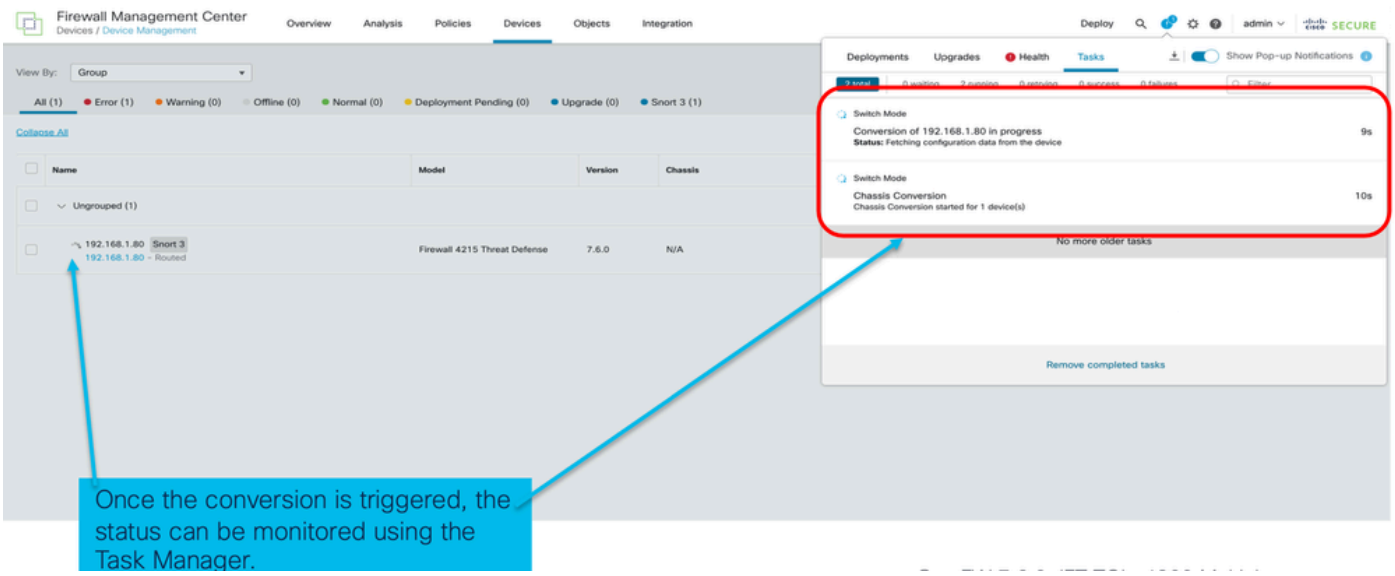


3. Readiness check and initiate conversion:

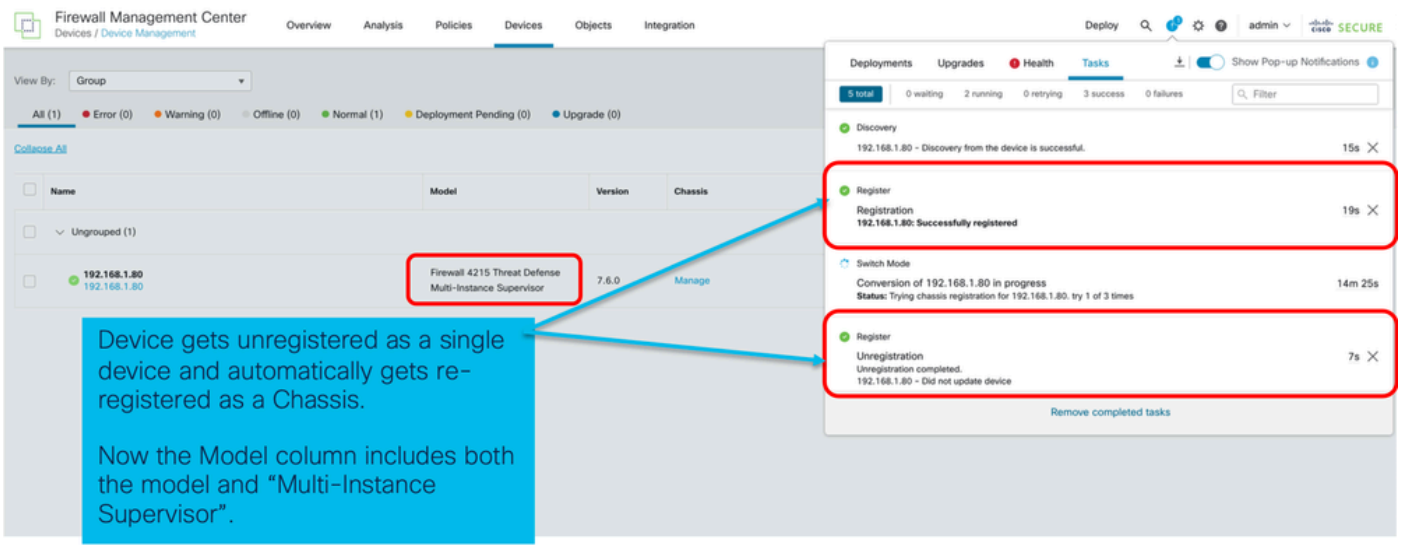


Monitoring Progress and Finishing UP

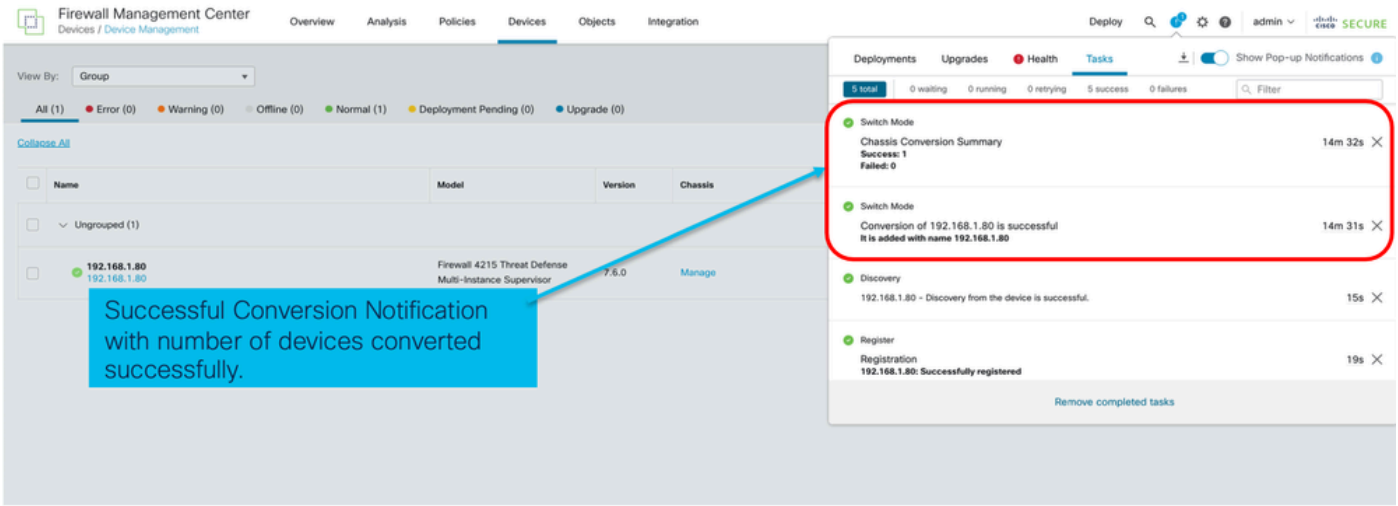
1. Conversion starting notification:



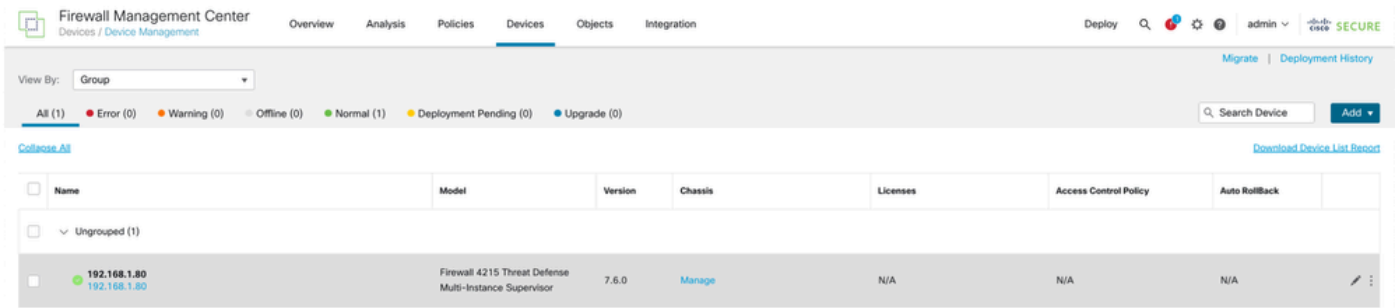
2. Auto-registration of the chassis:



3. Post-conversion notification:



Resulting device management page listing 4200 series (MI mode) devices:



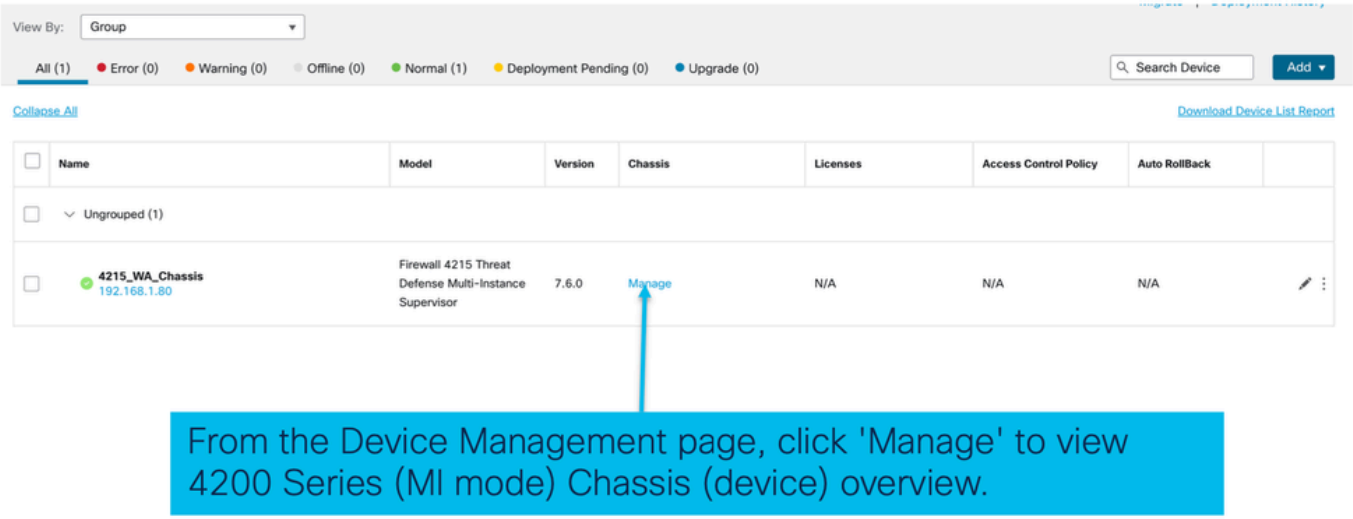
FMC Chassis Overview Page

Overview of the FMC Chassis Overview Page

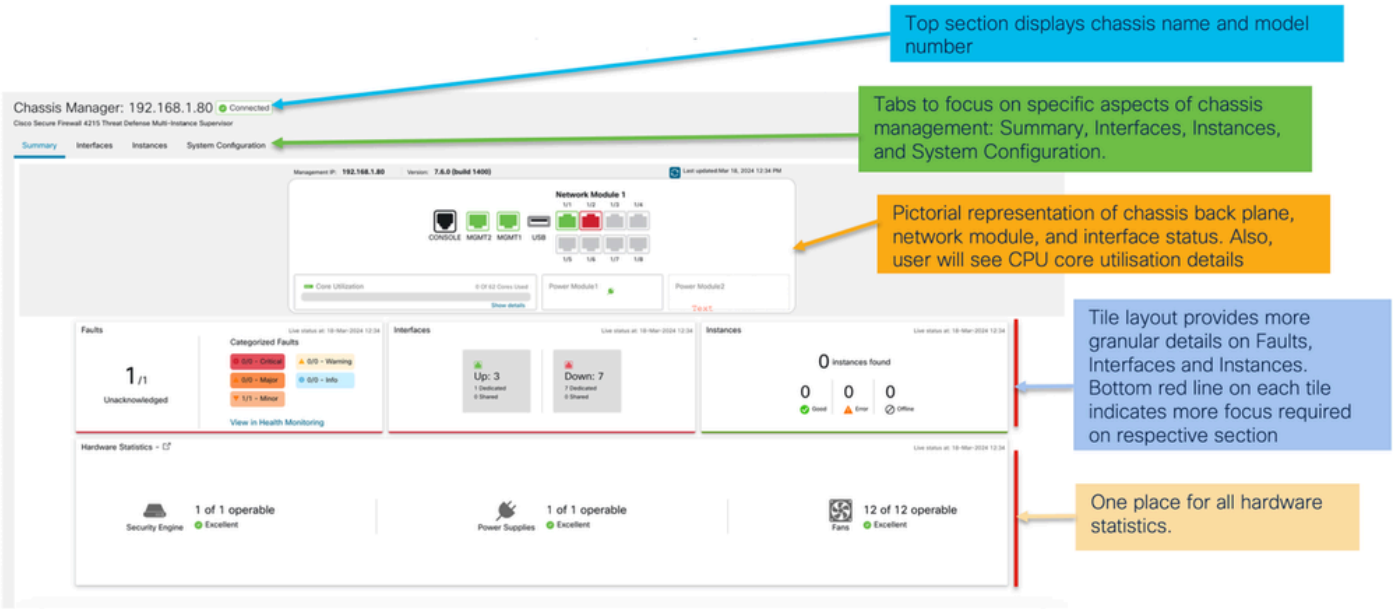
The FMC Chassis Overview page gives a complete summary of 4200 Series (MI mode) device. It includes:

- Pictorial back panel view of the device, including available network modules.
- Faults summary, with their criticality.
- Interface summary, status.
- FTD instance summary, status.
- Hardware statistics - including FAN, Power supply, memory, CPU usage, and storage.

Click **Manage** to navigate to **Chassis Overview**:



Chassis page summary tab:

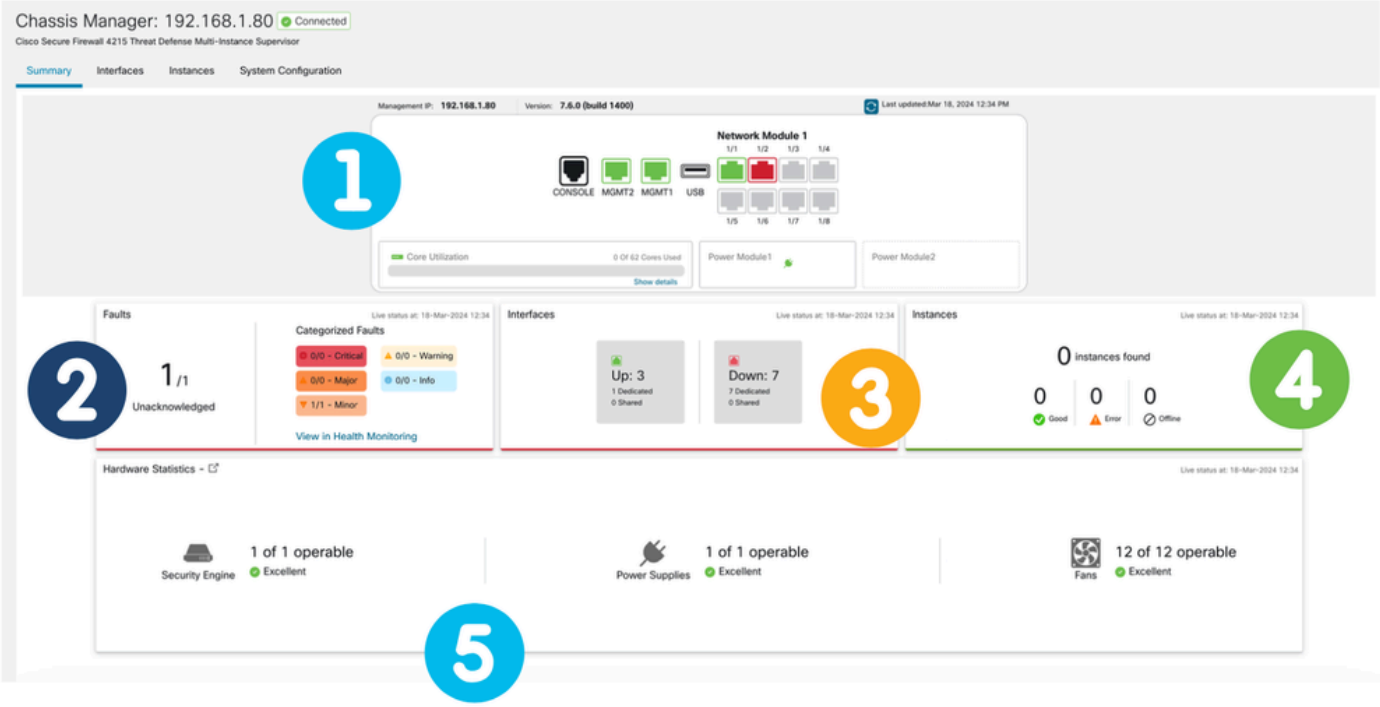


Chassis Page Summary Tab Sections

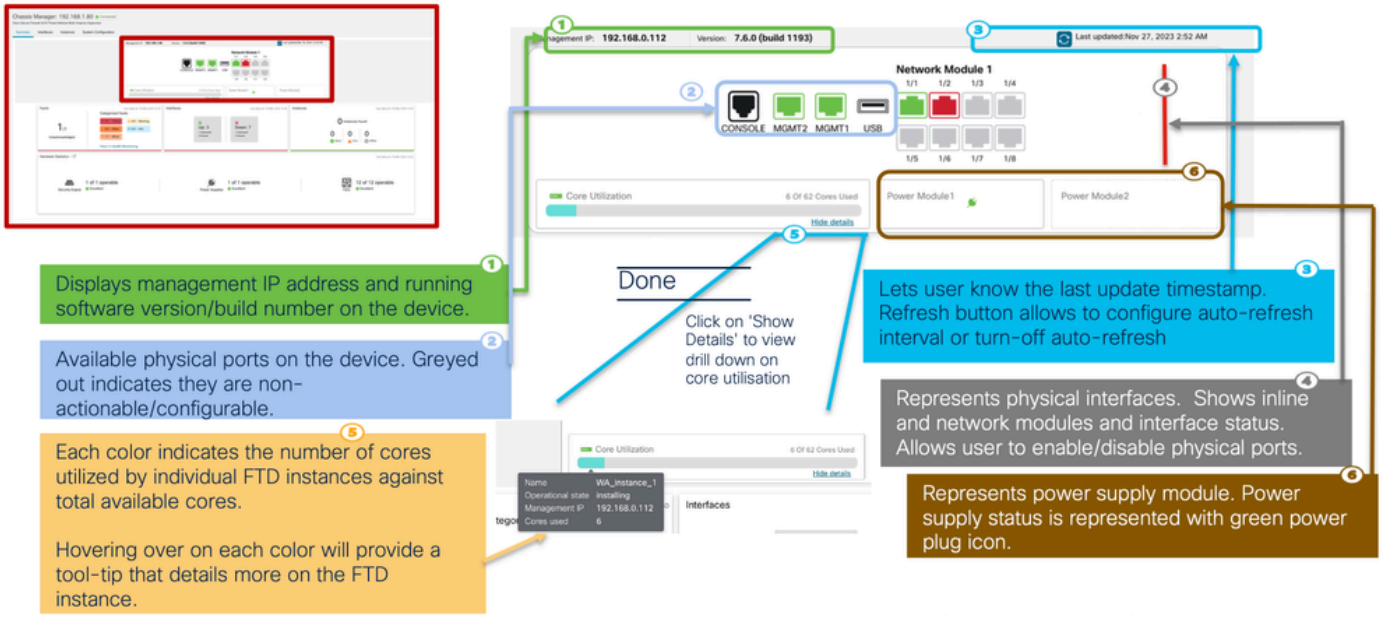
The Summary tab contains sections. Click to get more details:

- Back plane
- Faults
- Interfaces
- Instances
- Hardware Statistics

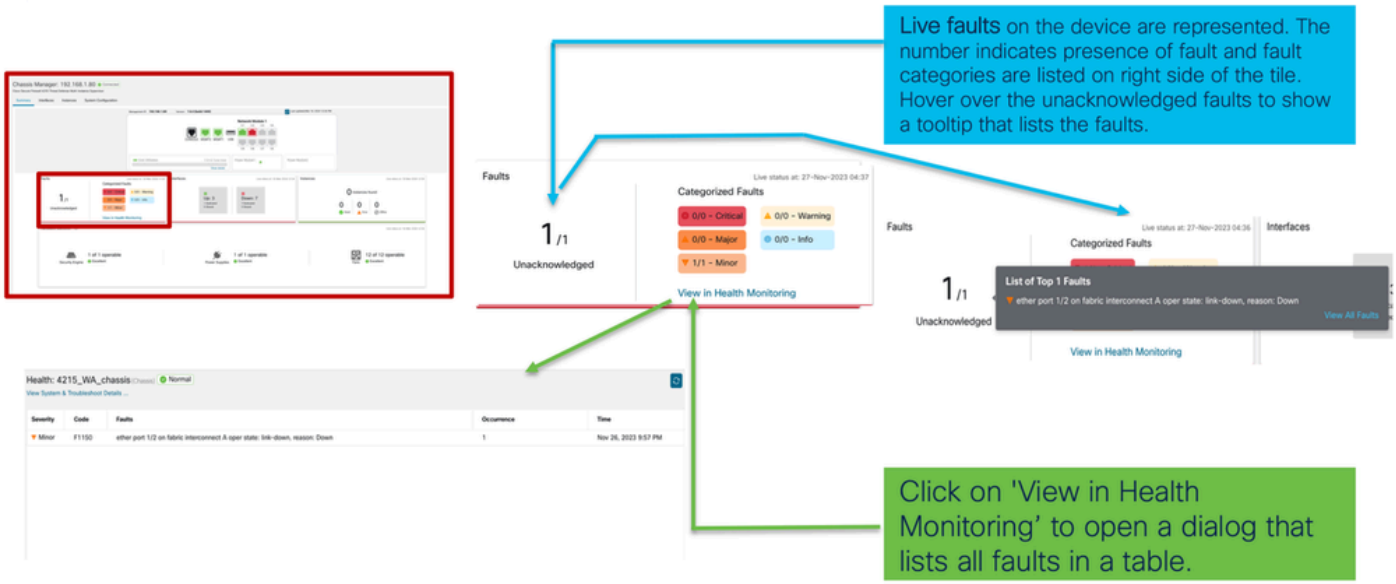
Sections are mapped by number as shown in this image:



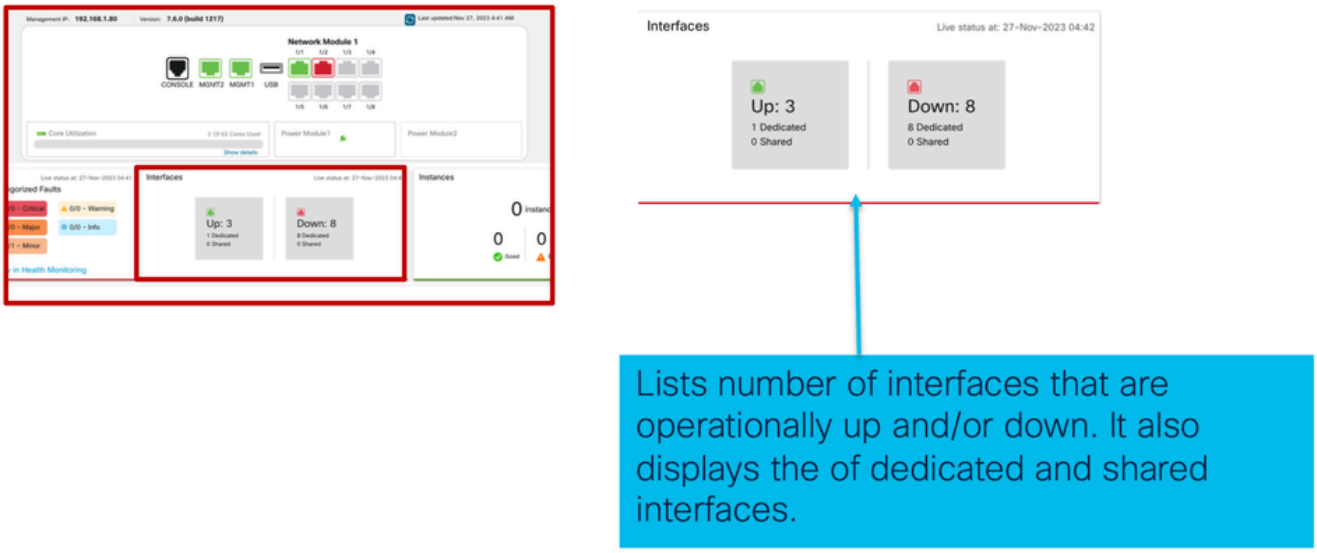
1. Back plane view:



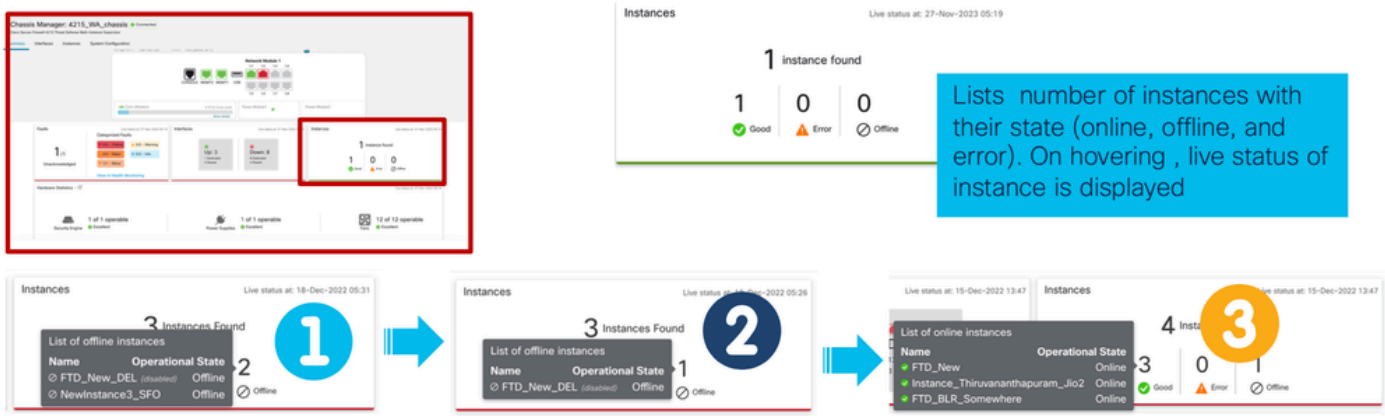
2. Faults section:



3. Interfaces section:



4 .Instances section:

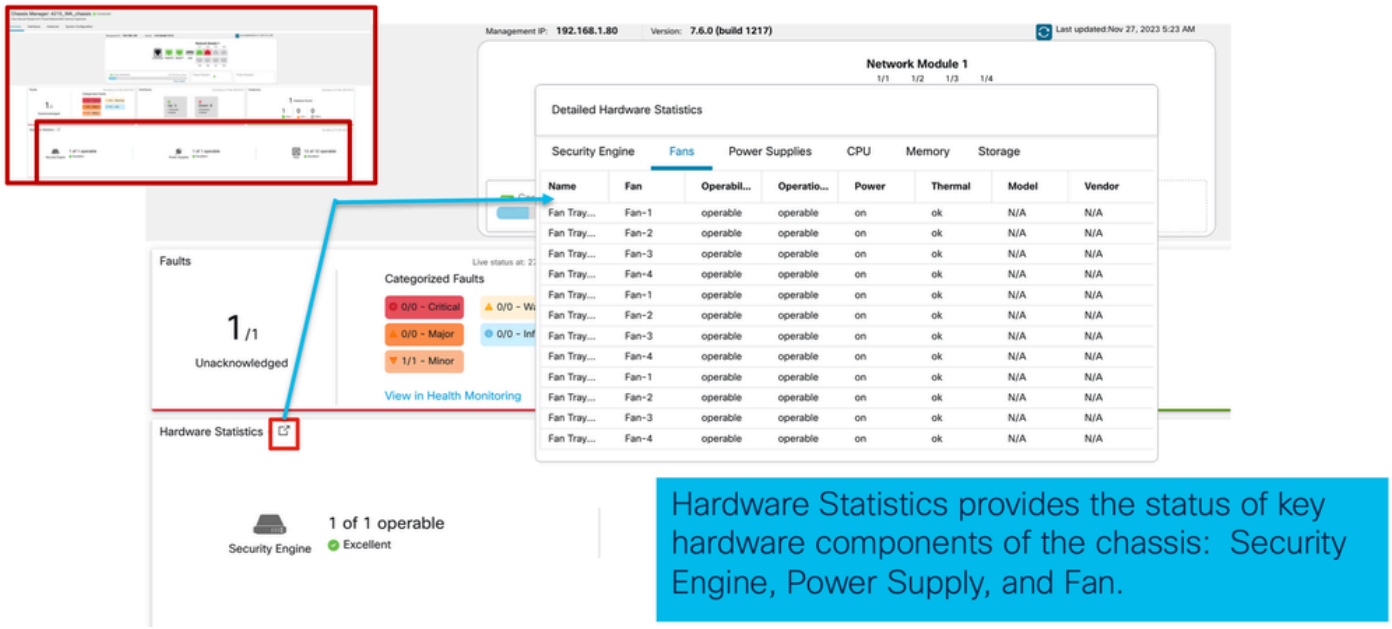


The transition of instances from offline to online is shown in the preceding image.

- Once provisioned (1)

- The instance is offline until it comes online (2)
- Intermediate states are also reflected (3)

5. Hardware statistics:

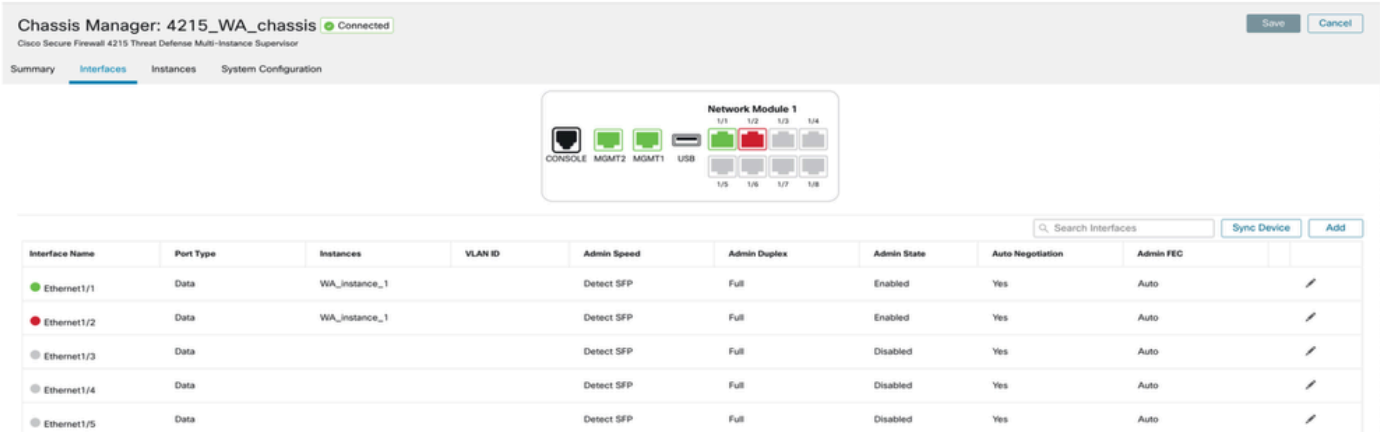


Manage Interfaces

Operations Supported from Interfaces tab:

- Update of Physical interface.
- Create/Update/Delete of Sub-interfaces.
- Create/Update/Delete of EtherChannel interfaces.
- Sync Interface configurations.
- OIR of Network module.
- Break/Join of Physical interface.

Interfaces Tab Summary

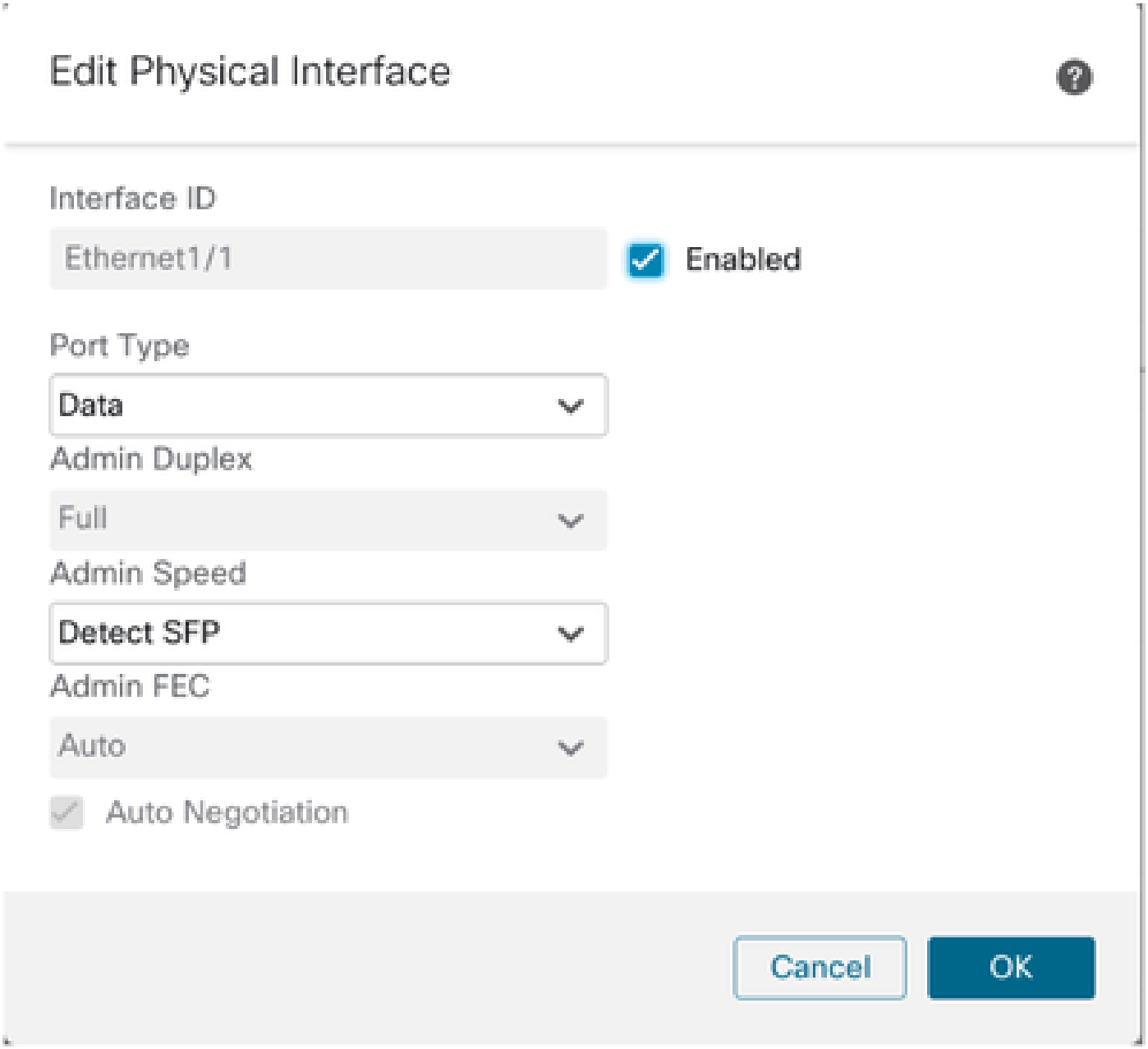


The landing page of the Interfaces tab shows all the types of interfaces that are managed for a chassis, such as physical interfaces, sub interfaces, and EtherChannel's, and EtherChannel sub interfaces.

Modify Physical Interface Configurations

These attributes of a physical interface can be updated:

- State (Enabled/Disabled)
- Port Type (Data | Data-Sharing)
- Admin Duplex
- Admin Speed
- Auto Negotiation



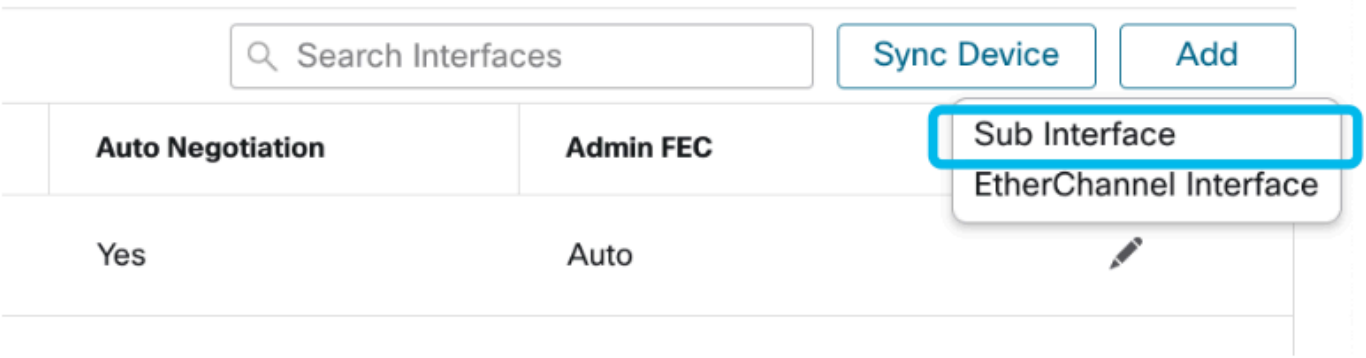
Manage Sub Interface

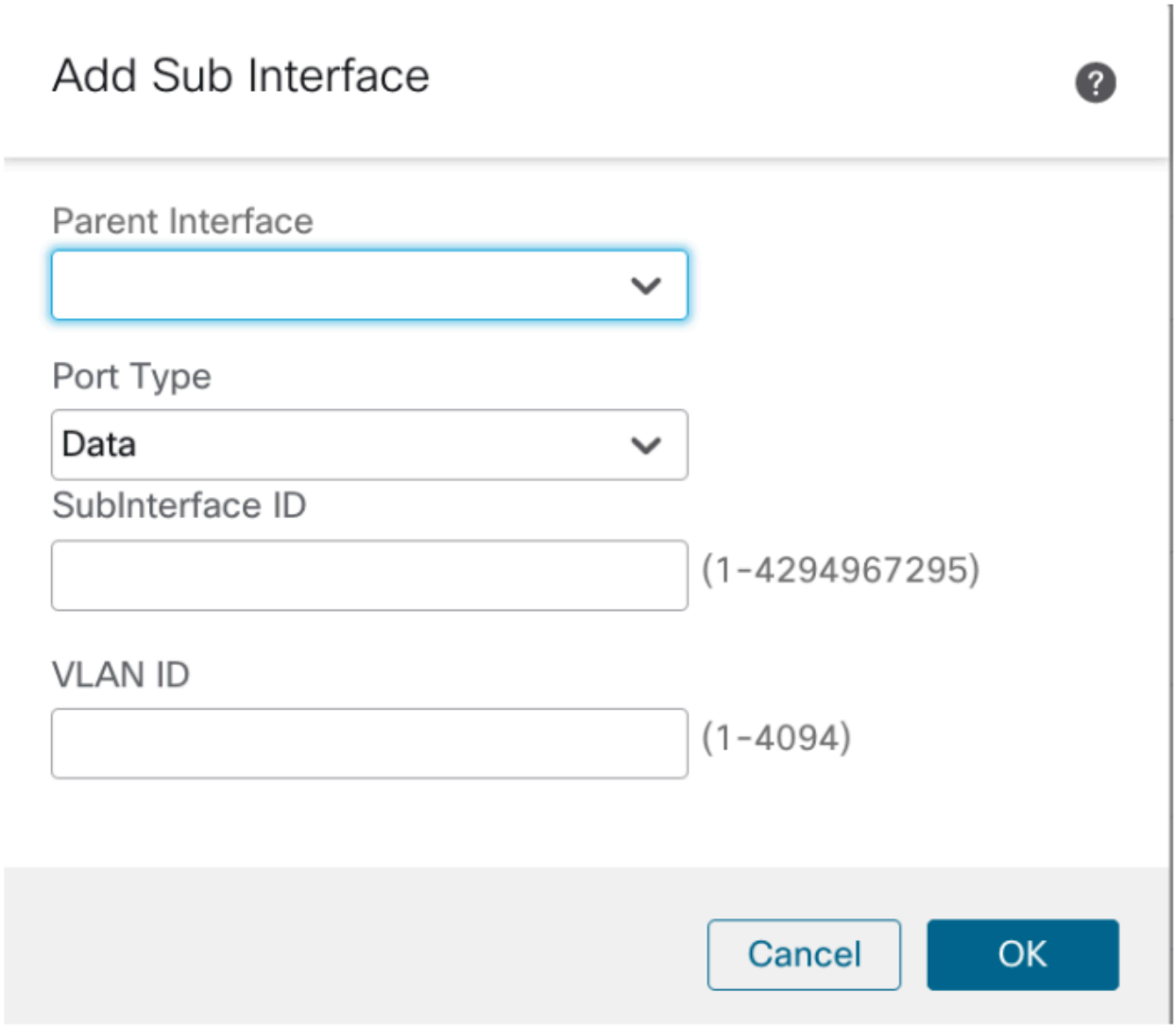
Pick the sub-interface option from the **Add** button to add a new interface.

These attributes of a sub-interface can be modified:

- Parent Interface
- Port Type (Data / Data-Sharing)

- SubInterface ID
- VLAN ID





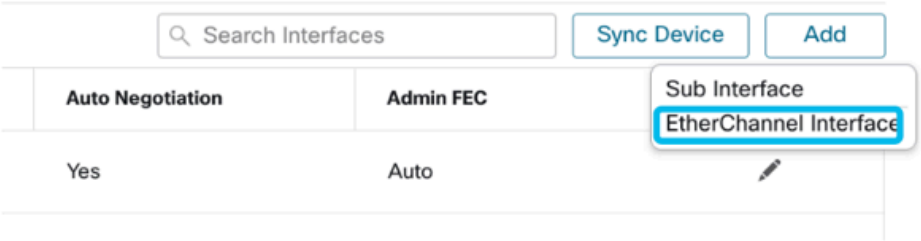
Manage EtherChannel

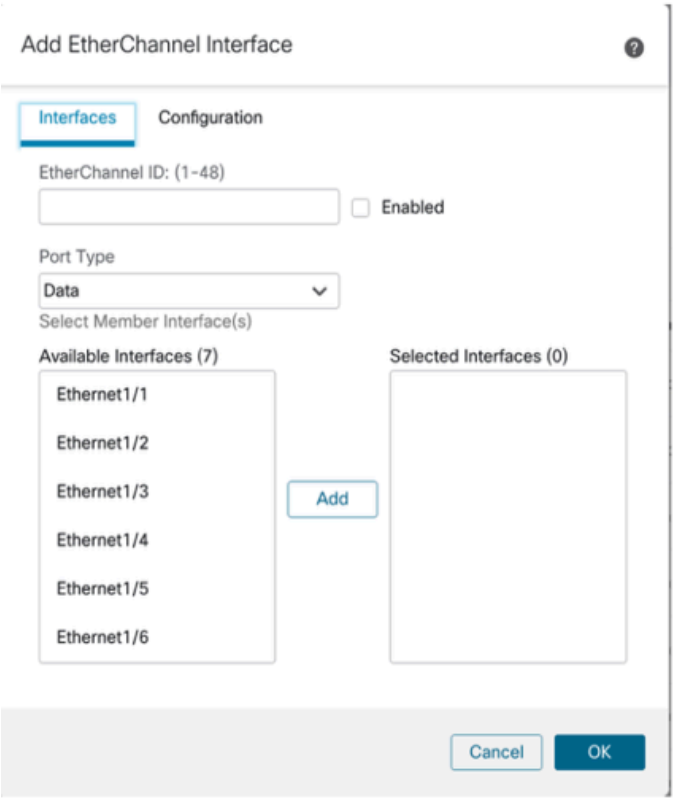
To create a new EtherChannel interface, use the “EtherChannel interface” under the **Add** button.

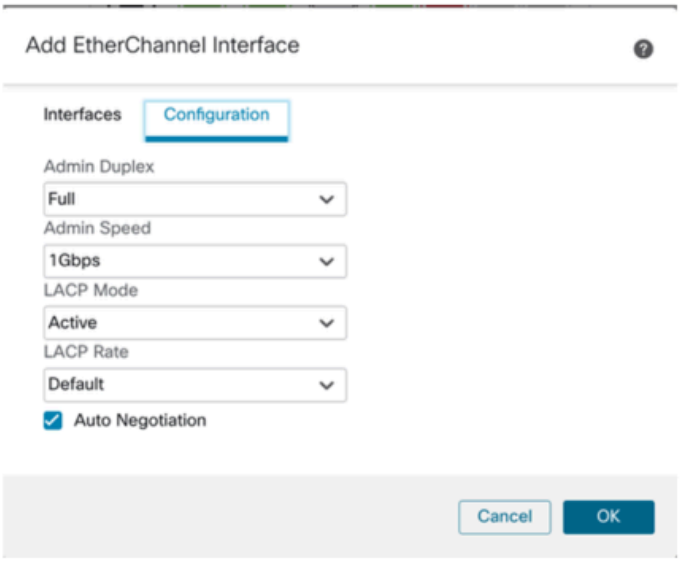
Attributes which can be configured for an EtherChannel are:

- EtherChannel ID
- Port-Type (Data/ Data-Sharing)

- Member interfaces
- Admin Speed
- Admin Duplex
- LACP Mode
- LACP Rate
- Auto Negotiation

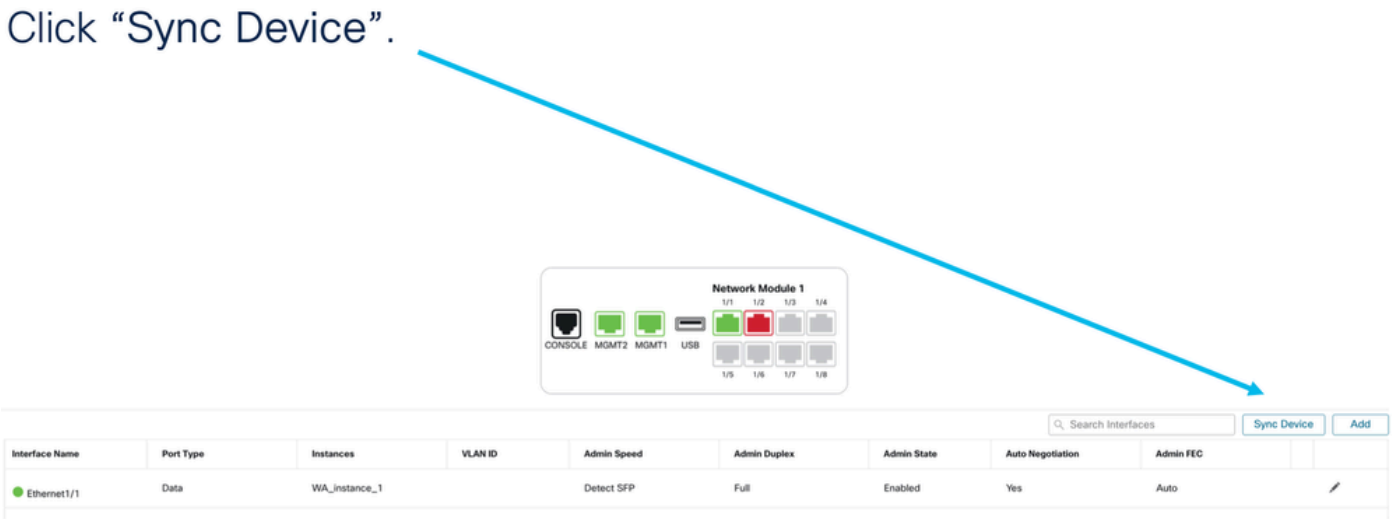






Sync Device Configurations

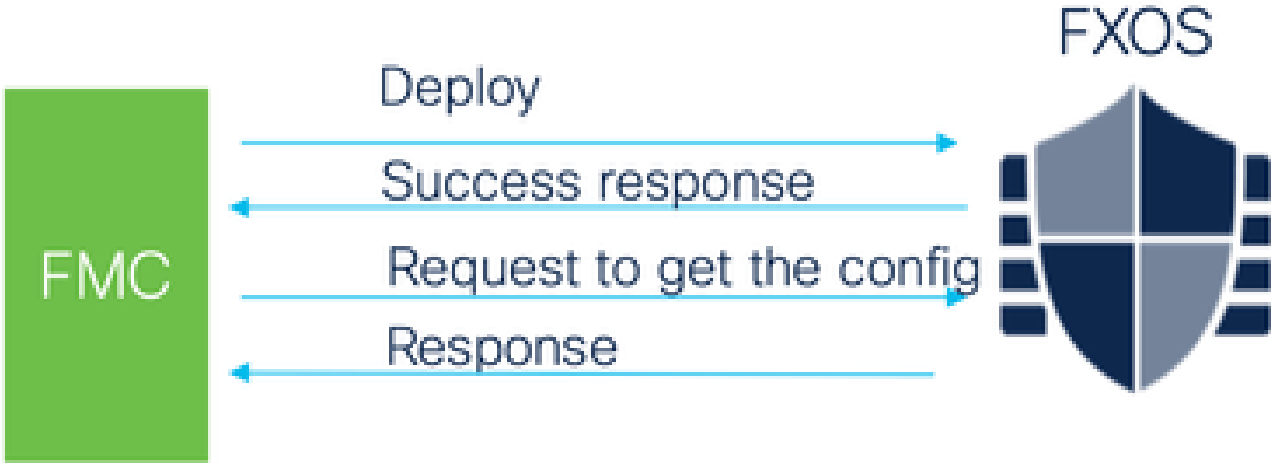
There are cases when the FMC configuration and the device configuration can go out of sync. One case is when a user removes or inserts a netmod. Sync device can be done in such cases.



Netmod Hot Swap / Break-Out Support

“Hot Swap”, used in your docs, is referred to as Online Insertion and Removal or OIR in other internal documentation.

There is an immediate deploy upon Enable/Disable of Network Module or Break or Join of interfaces. Multi-Instance mode is same as 4200 Series in native mode.



FMC compares the response received against the current configuration and then creates interface change notification for user to acknowledge.

4200 Native Supports EPM Hot Swap and Breakout

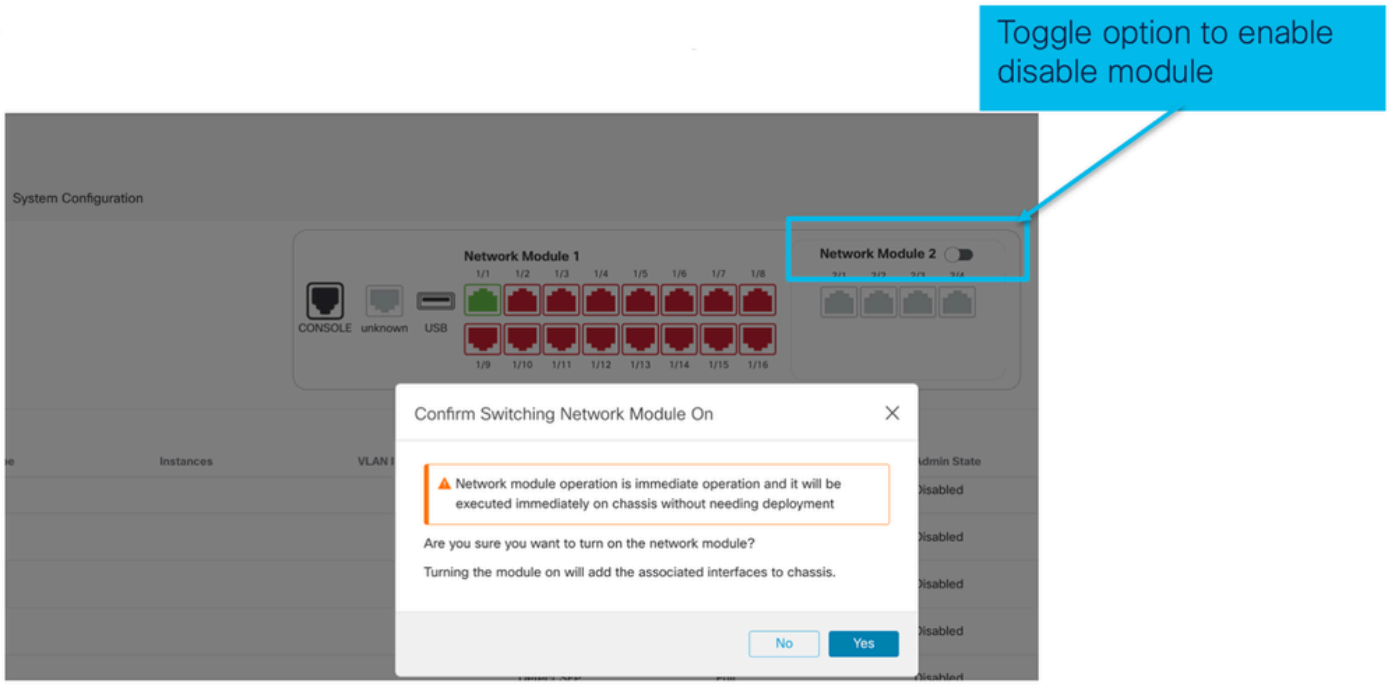
EPM OIR and Breakout are already supported on the standalone, native mode Secure Firewall 4200 Series standalone.

4200 Series EPM OIR and Breakout FMC documentation:

- https://www.cisco.com/c/en/us/td/docs/security/secure-firewall/hardware/4200/fw-4200-install/m-overview.html

OIR: Enable/Disable EPM Confirmation

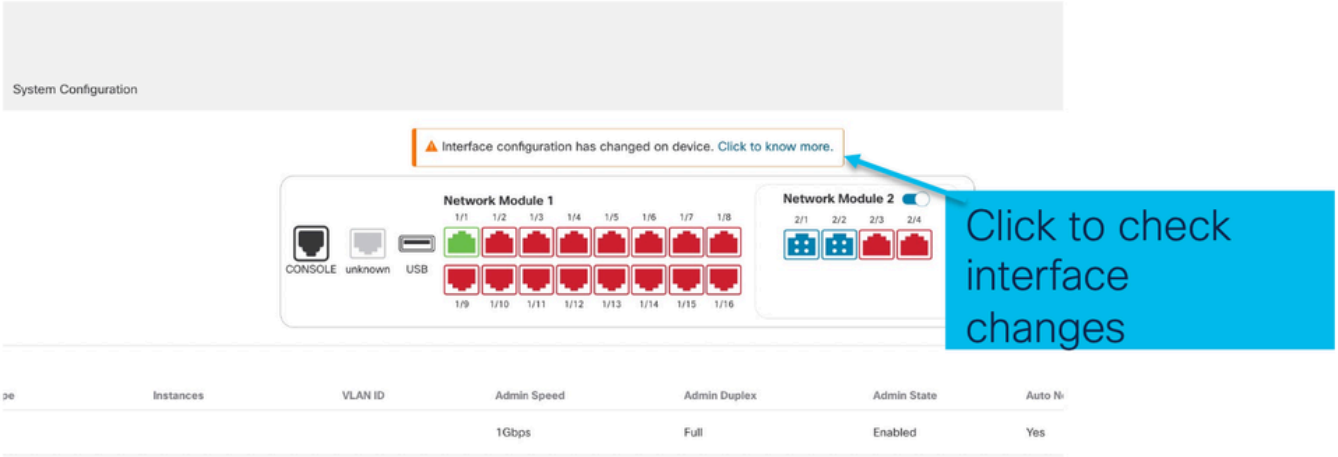
When the user toggles to enable module, a warning is shown to make sure this is not an accidental click.



EPM Enable Complete: Interface Notification Received

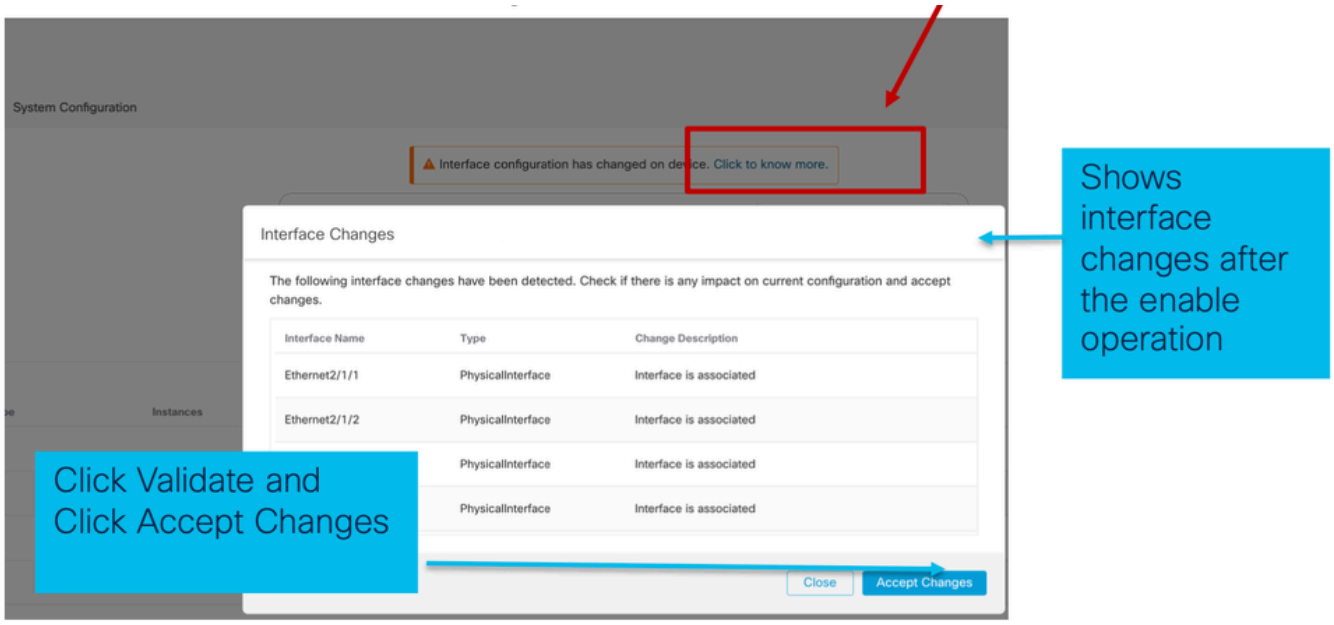
- When enabling an EPM, new interfaces are associated on the device.
- FMC receives the notification about the associated interfaces.
- On FMC, the user has to accept the changes.

This screenshot shows the option to see the associated interfaces:

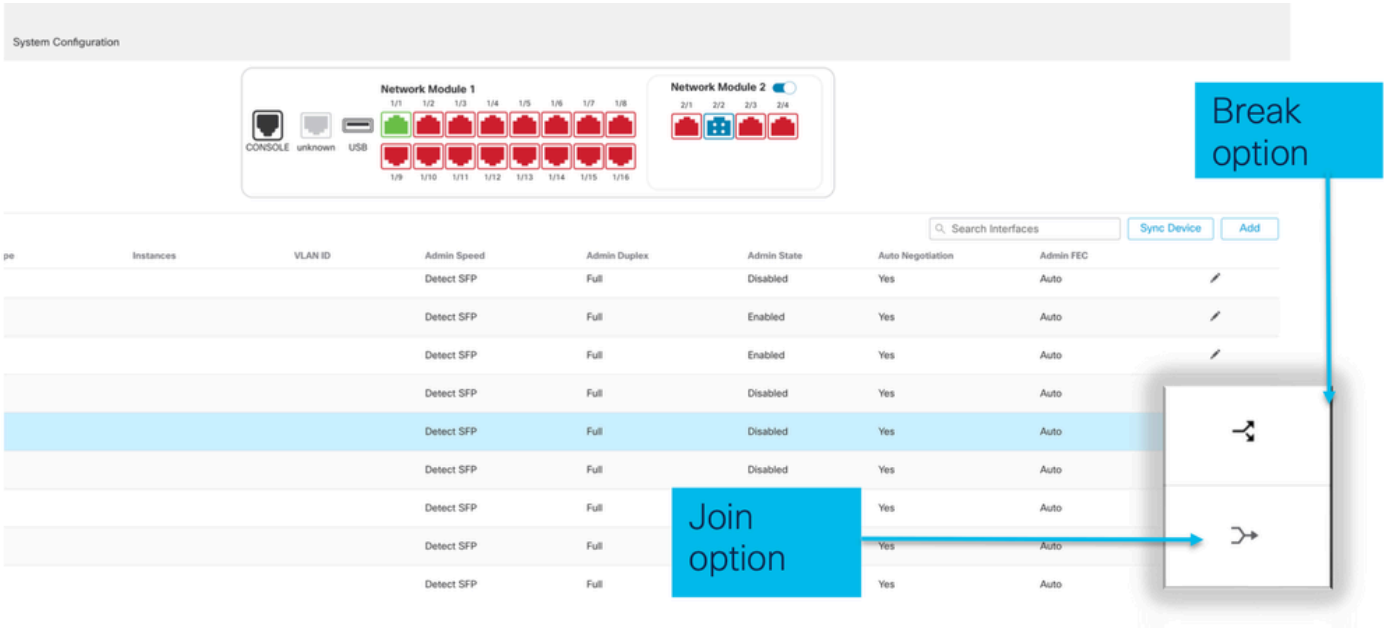


EPM Interface Change Notification

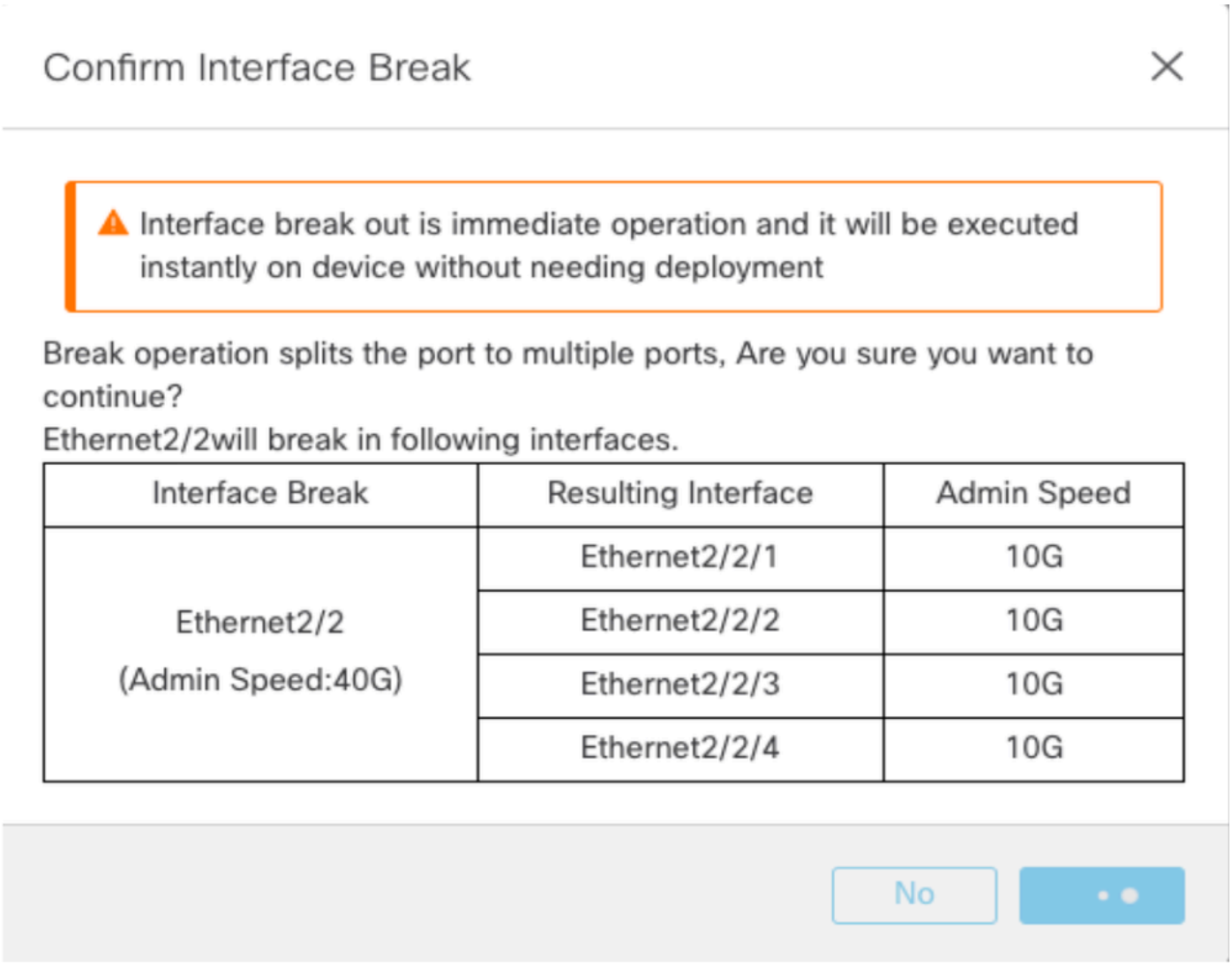
The interface listing page lists the interfaces which are added when EPM is enabled. **Click to know more l**aunches the Interface Changes dialog.
Click to know more is not available after saving.



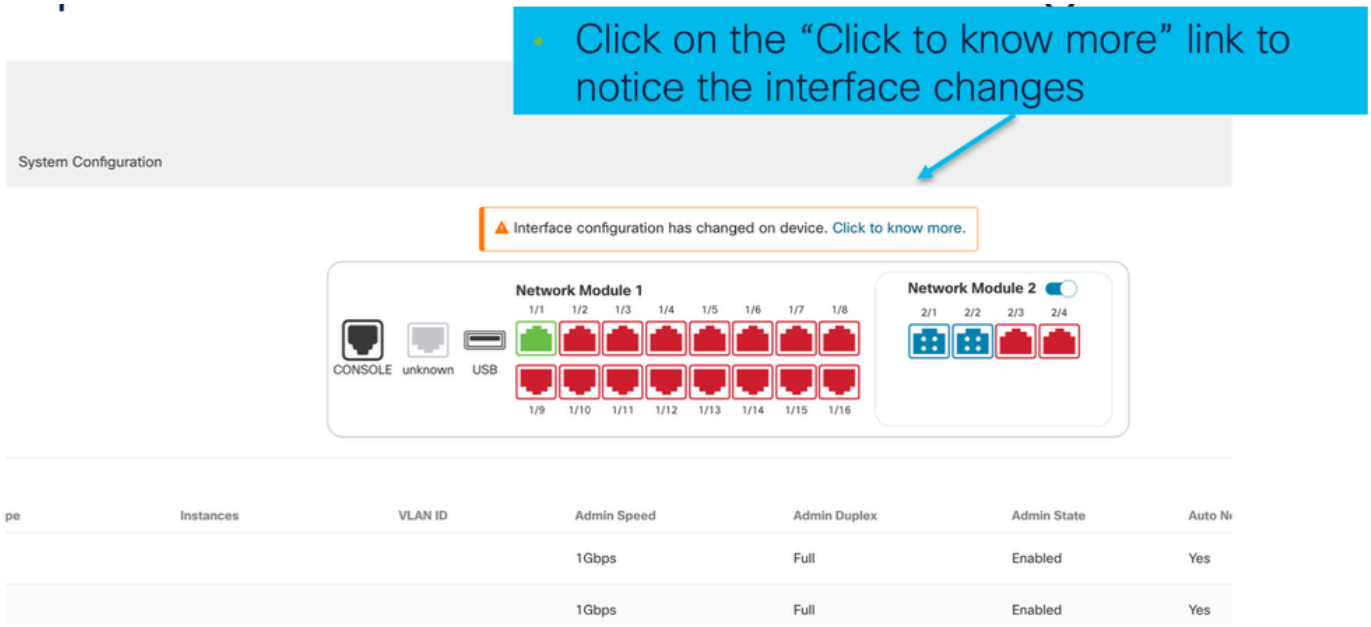
Break/Join Options in Chassis Page



The interface break confirmation wizard opens up on break option is triggered.

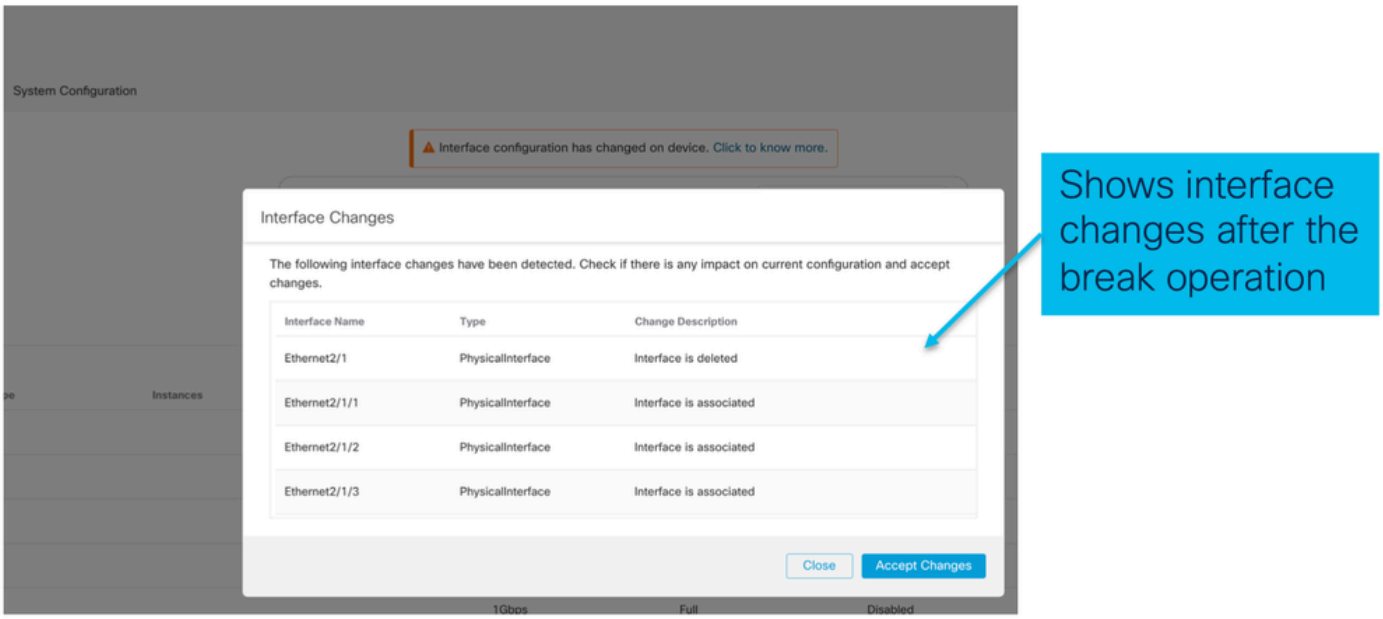


The interface update notification is visible on the chassis page after the interface break is confirmed.



Interface Changes after Break/Join

Upon clicking **Accept Changes**, these interfaces become available in the FMC to be used:



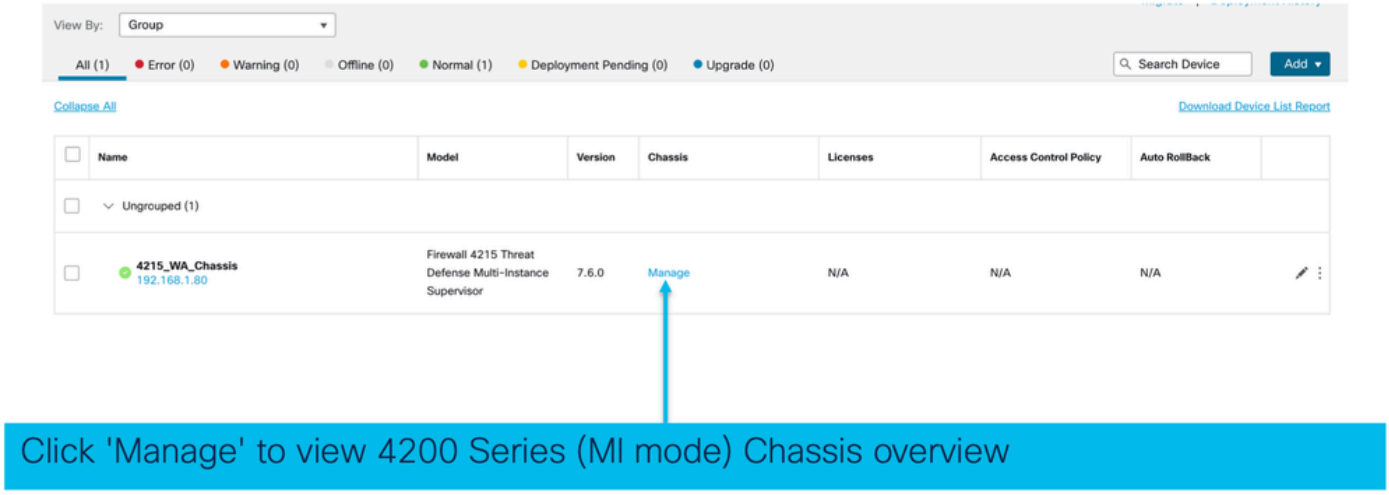
Impact of Interface Changes on Instance

Change	Behavior
Change a dedicated interface to shared	No validation error
Change a shared interface used in multiple instance to dedicated	Validation error will block the change
Disable of Network module with interfaces assigned to Instance	No validation error during the disable operation, but error will be thrown in case user tries to accept the notifications without removing the assignment from the instance
Break/Join of interfaces assigned to instance	• Validation error will be thrown to initiate such operation • User needs to unassign the interfaces from the Logical Device before initiating Break/Join operation

Instance Management

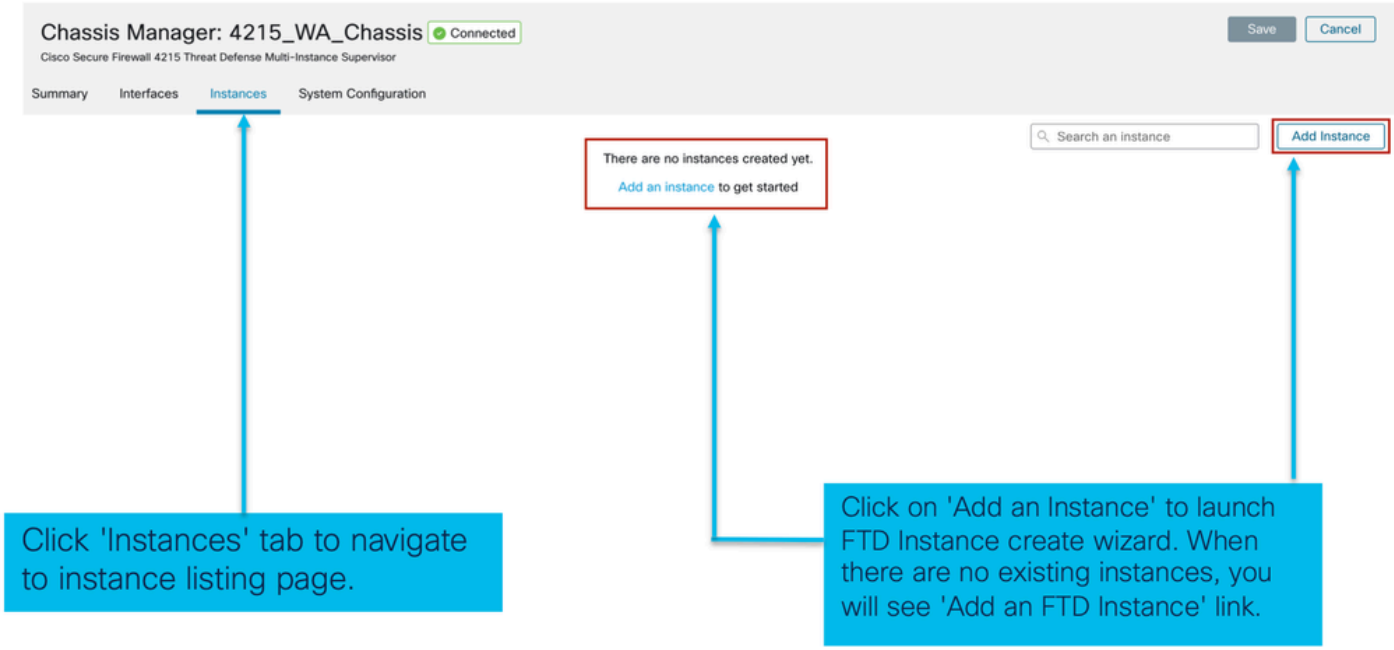
Instance Management enables you to:

- View all existing FTD instances and their details on a 4200 Series (MI mode) device.
- Create/Update FTD instances with desired CPU core and software version.
- Delete an existing FTD instance.
- Allows user to choose FTD policies – Access policy and Platform Settings policy for FTD instance.
- Auto-register FTD instance to FMC once it comes online.

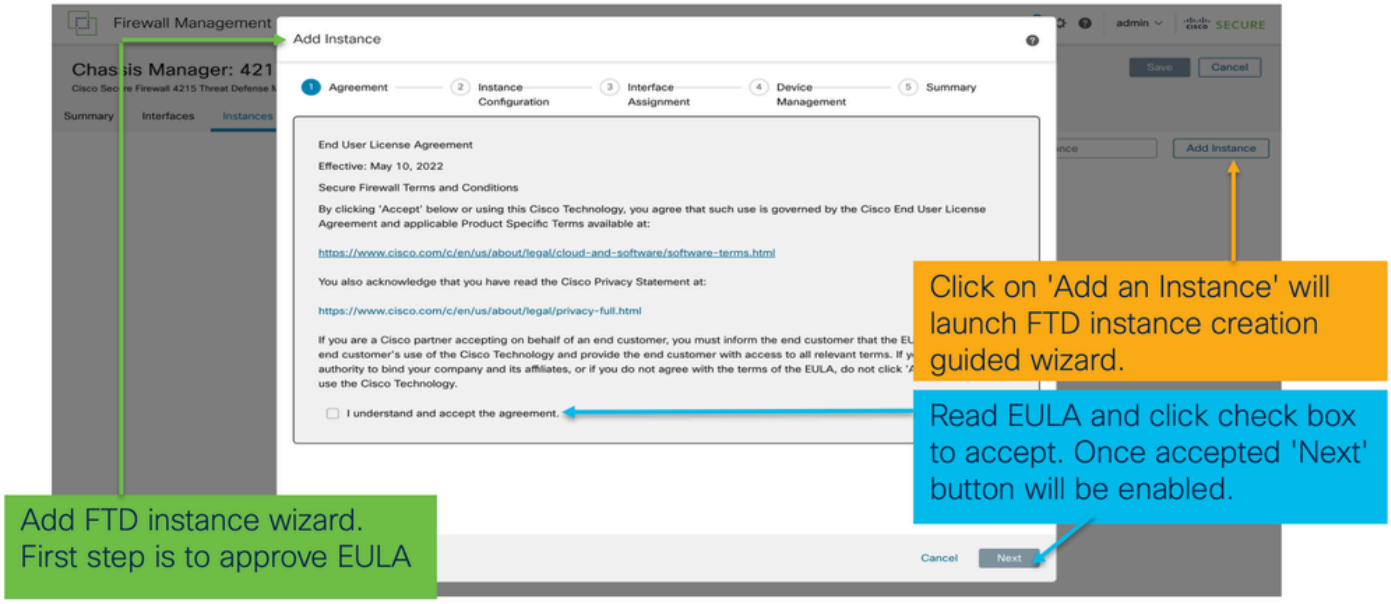


Create an Instance

Launch the wizard by clicking on **Add Instance**.

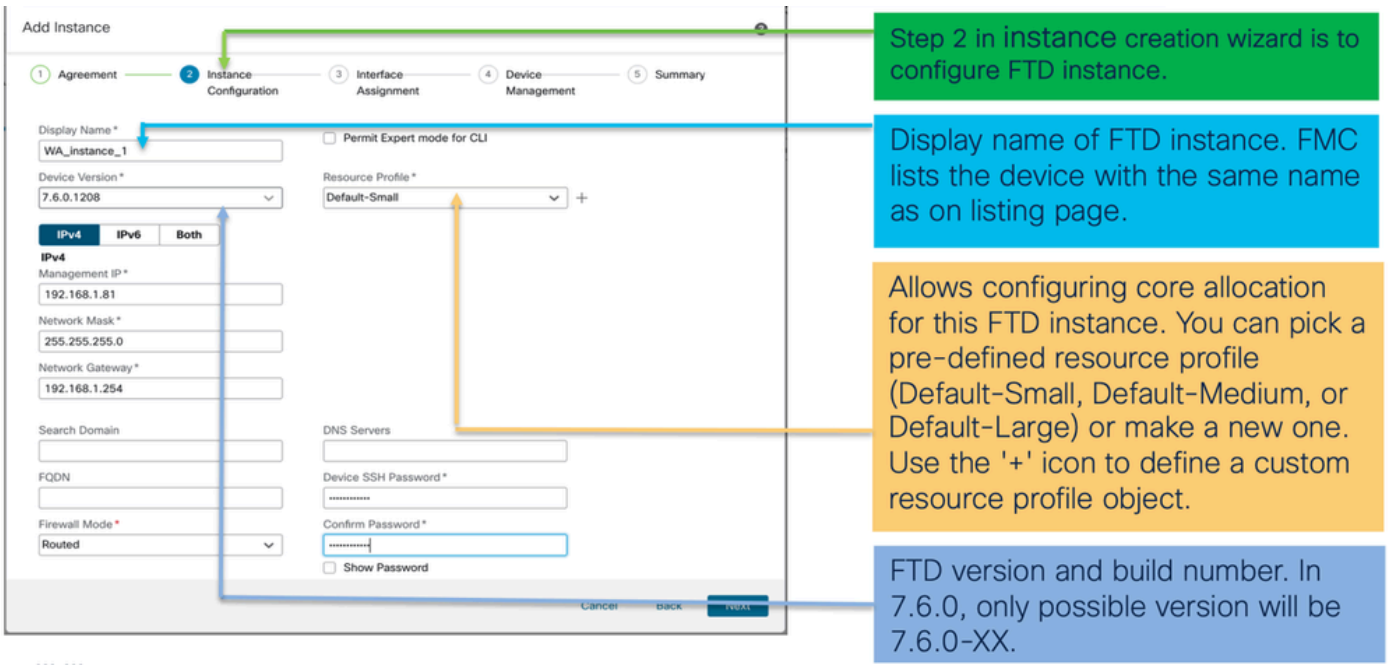


Step 1. Agreement:

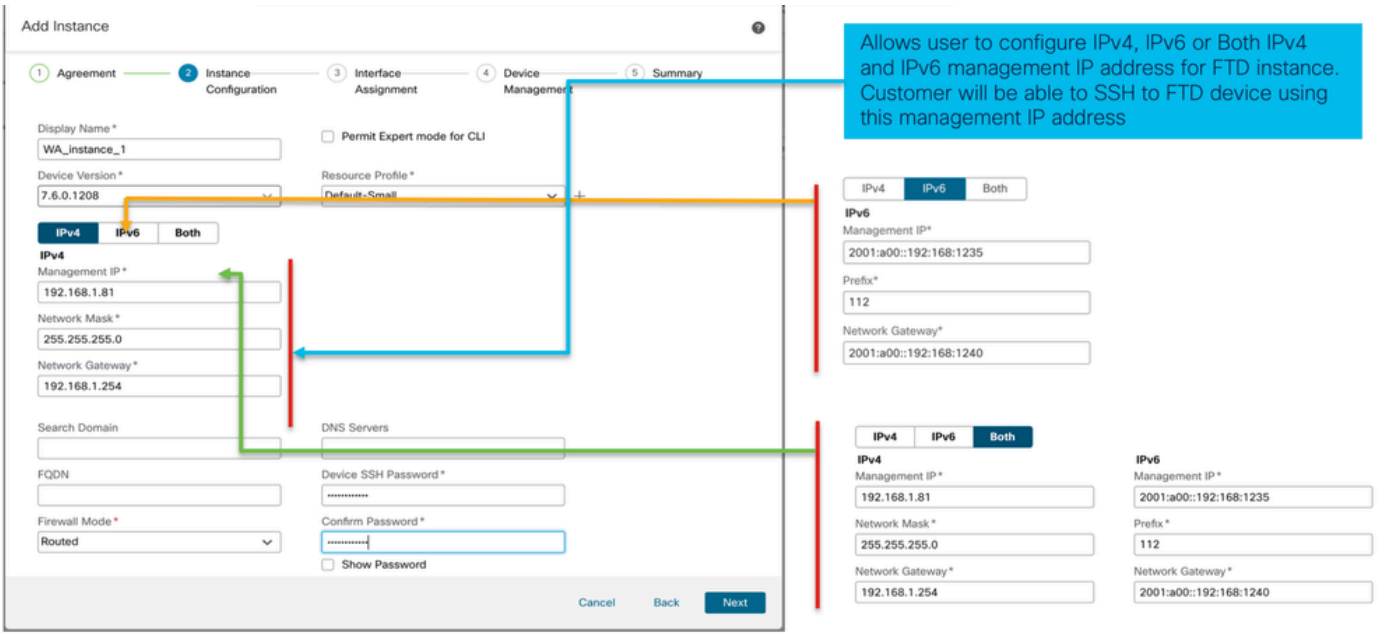


Step 2.

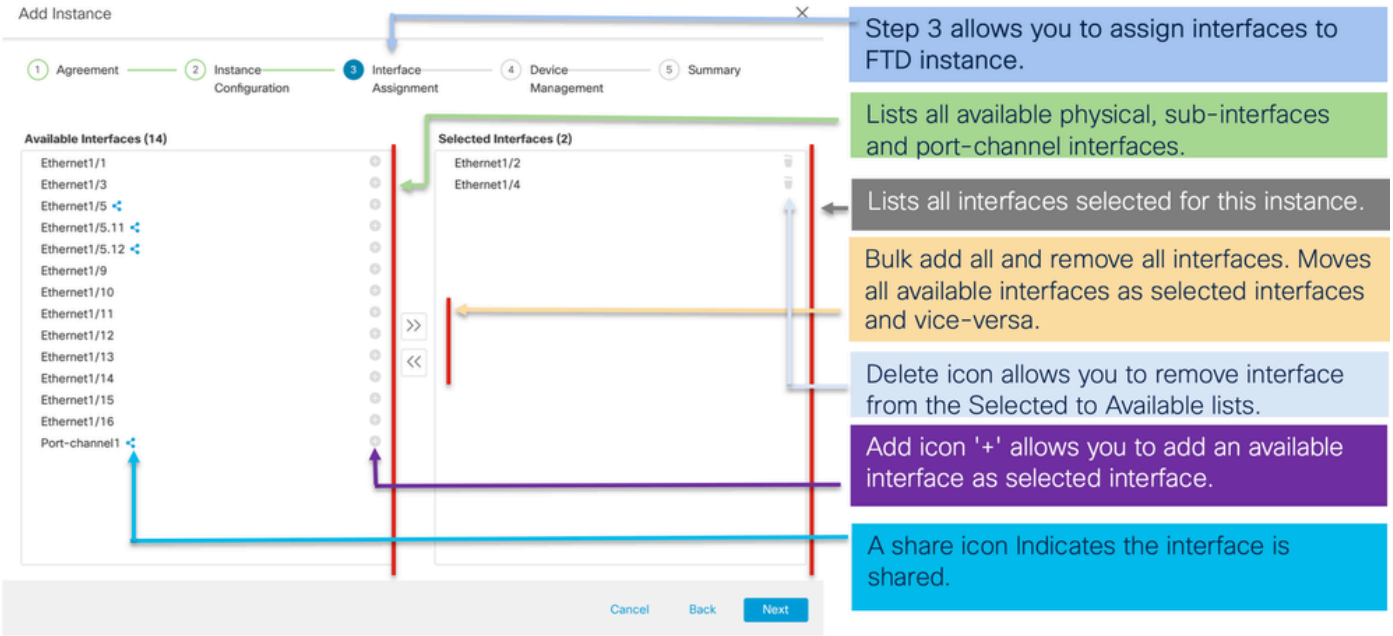
- Instance configuration basics:



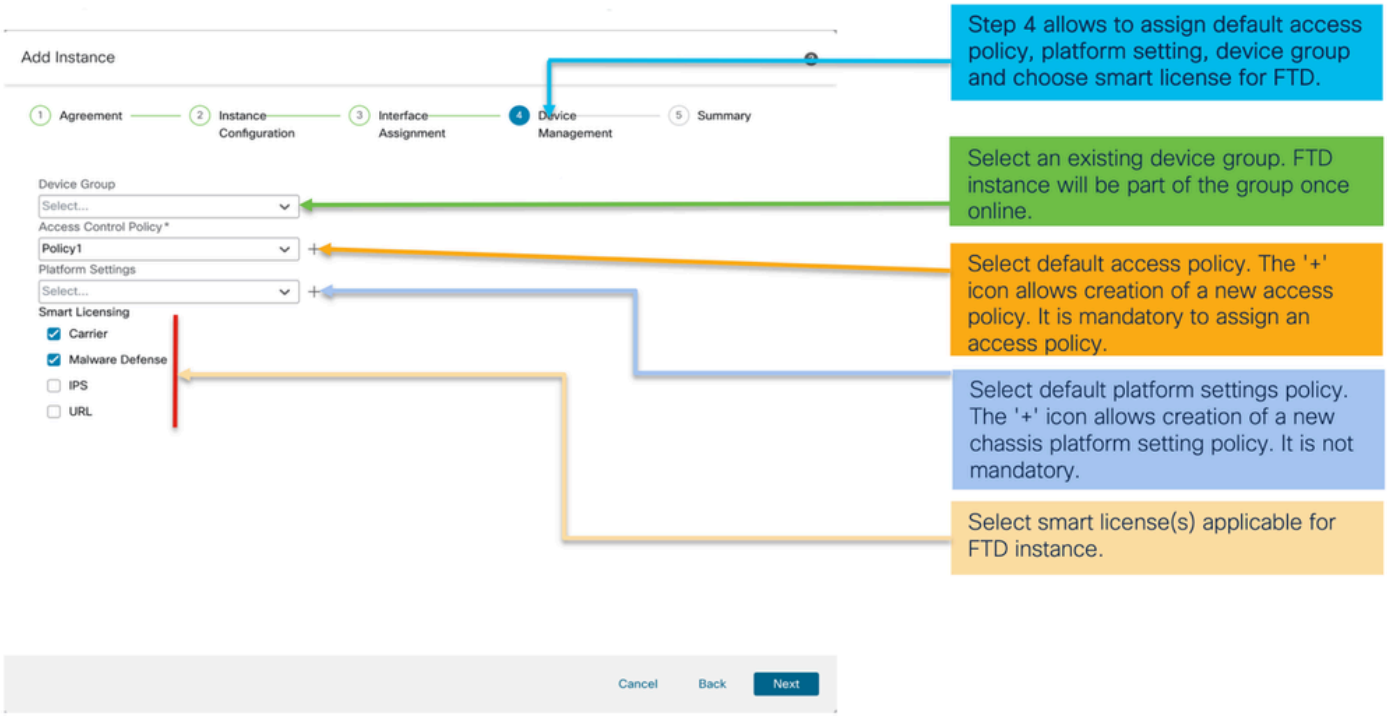
- Instance Configuration IPs:



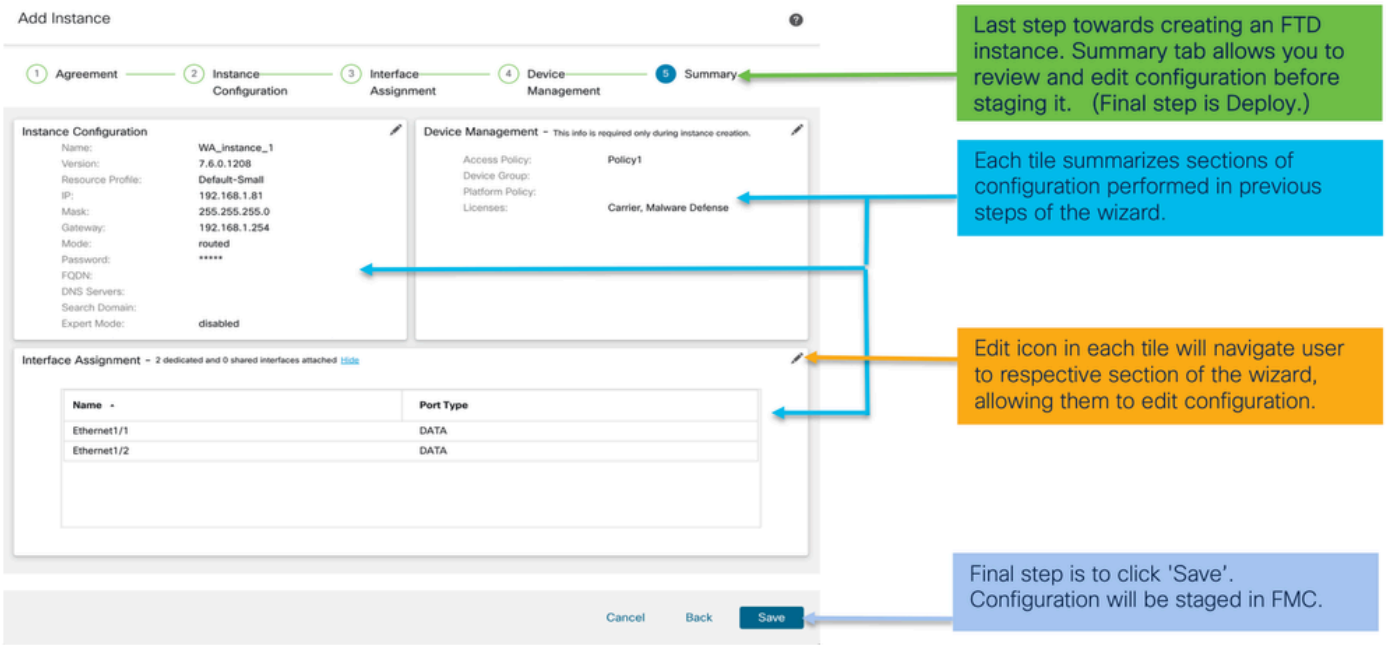
Step 3. Interface assignments:



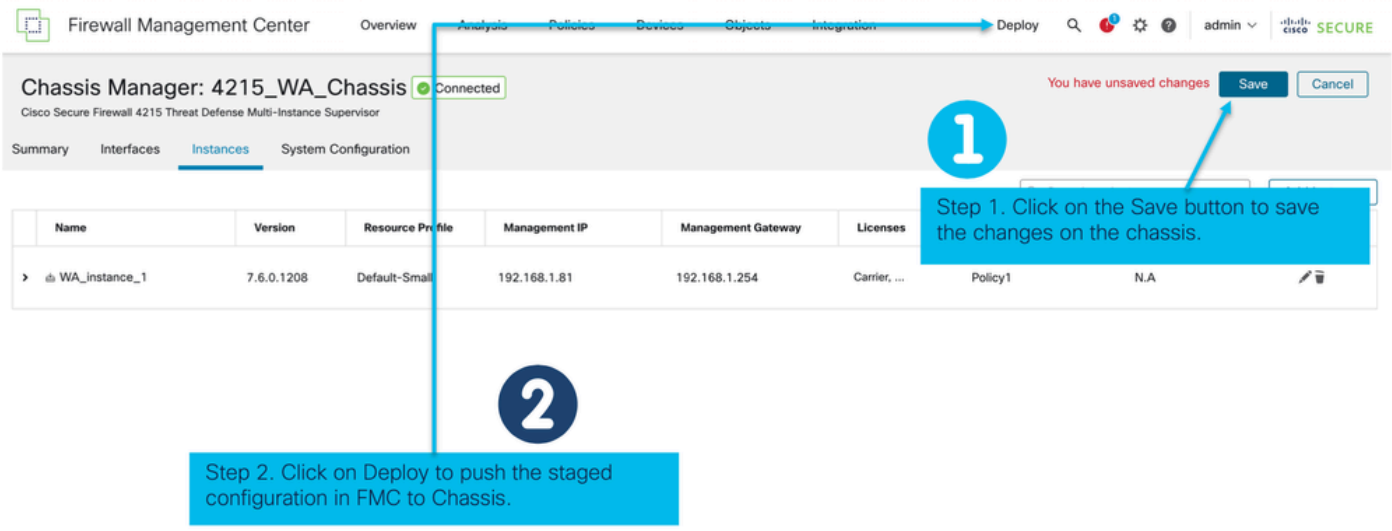
Step 4. Device management:

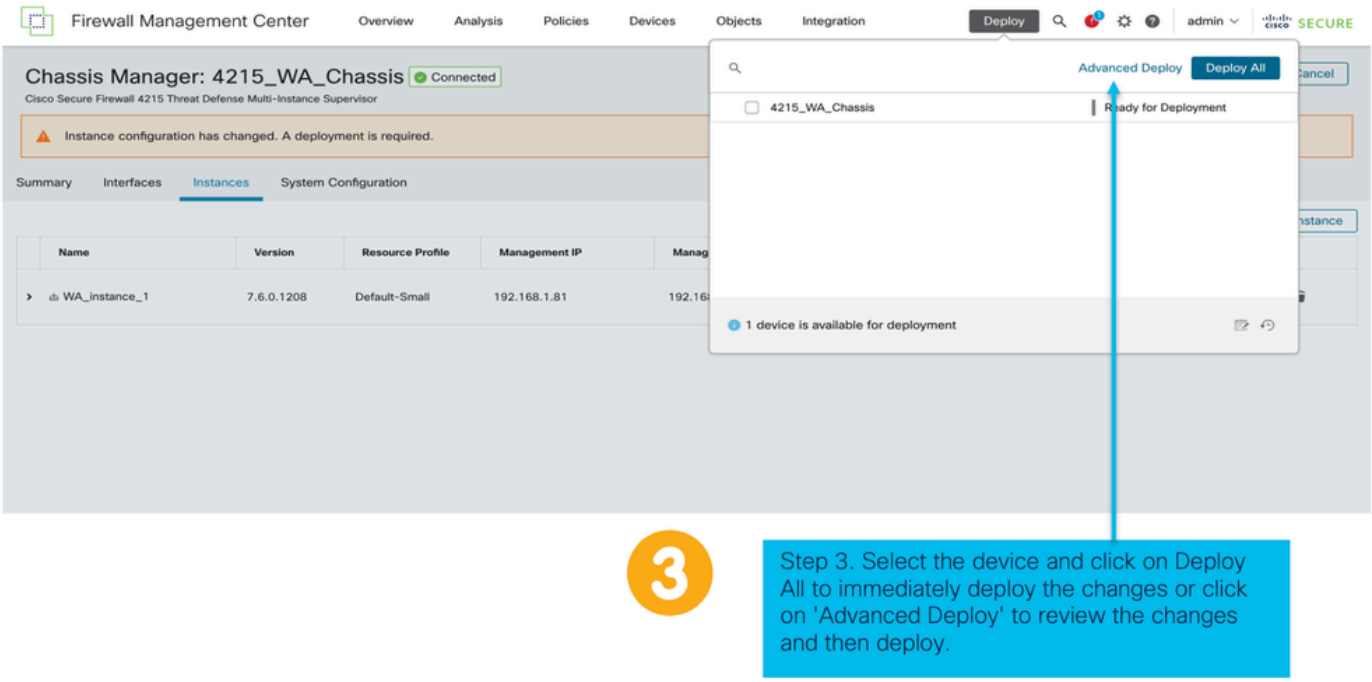


Step 5. Summary:

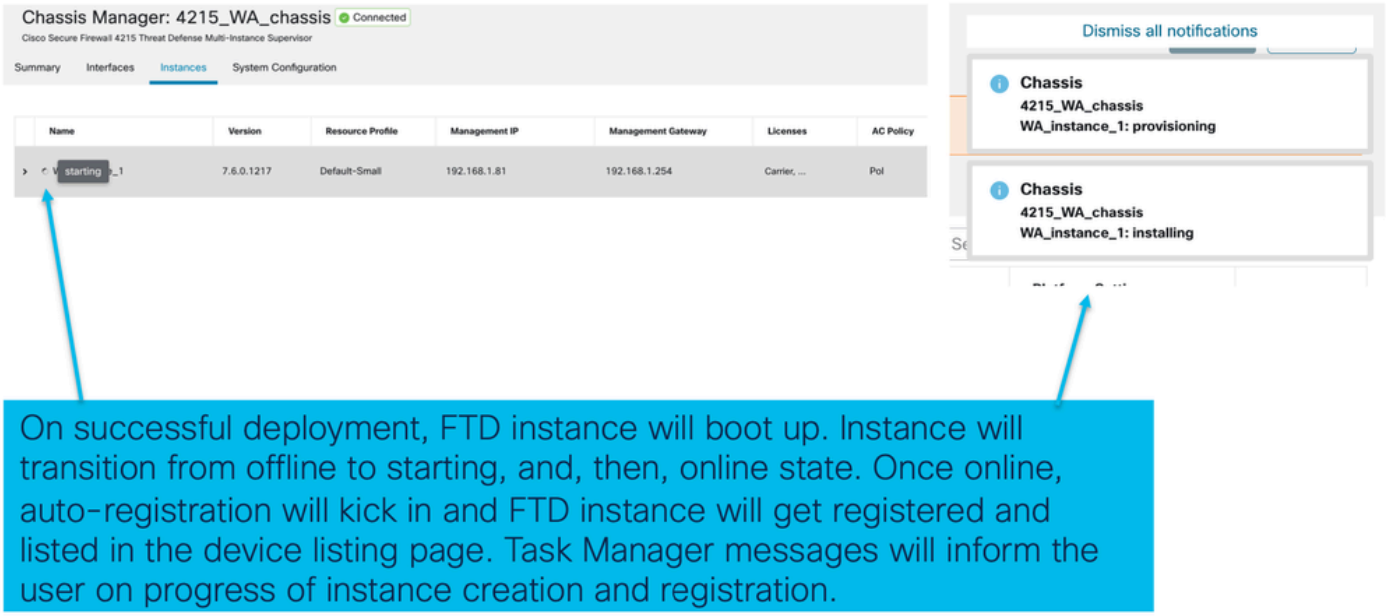


To complete configuration, **Save** and **Deploy**.

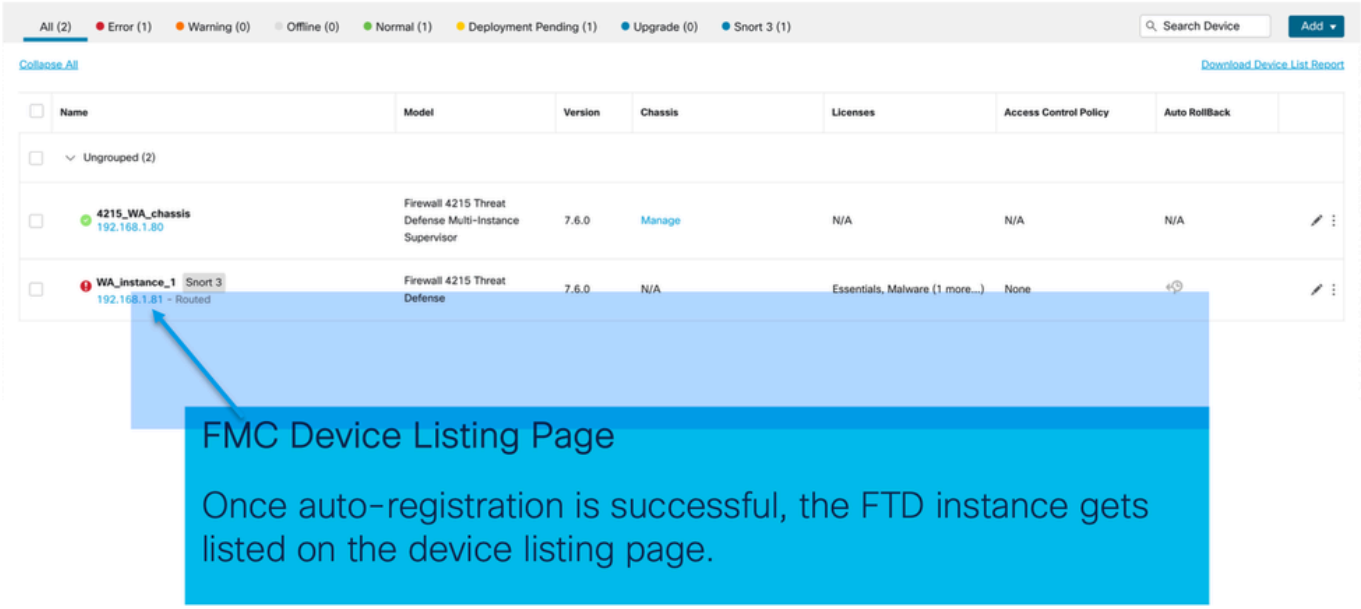




Auto-registration of an FTD instance after successful deployment:

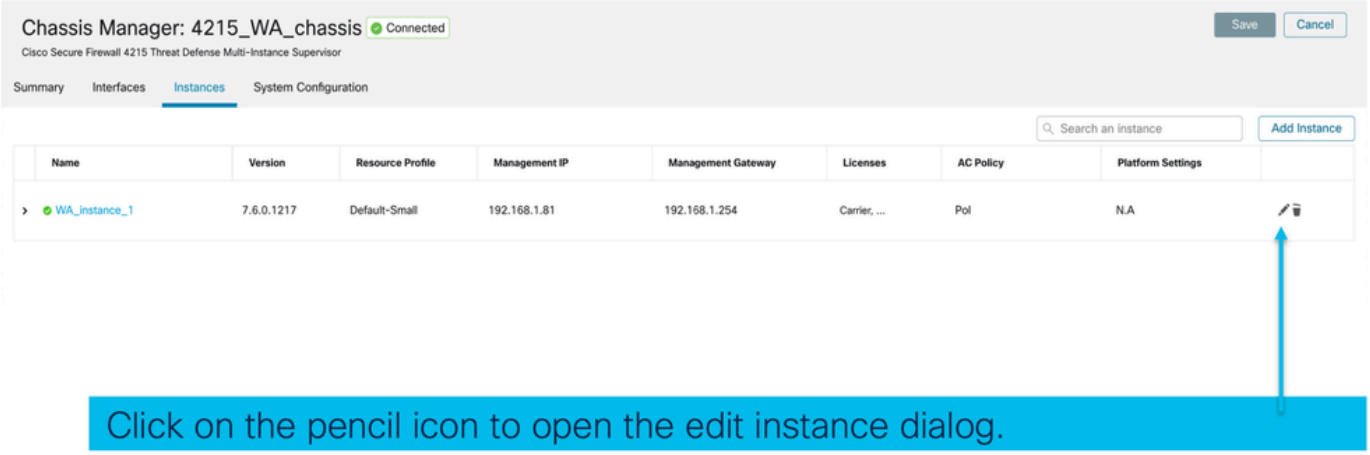


Instance registered to Management Center:

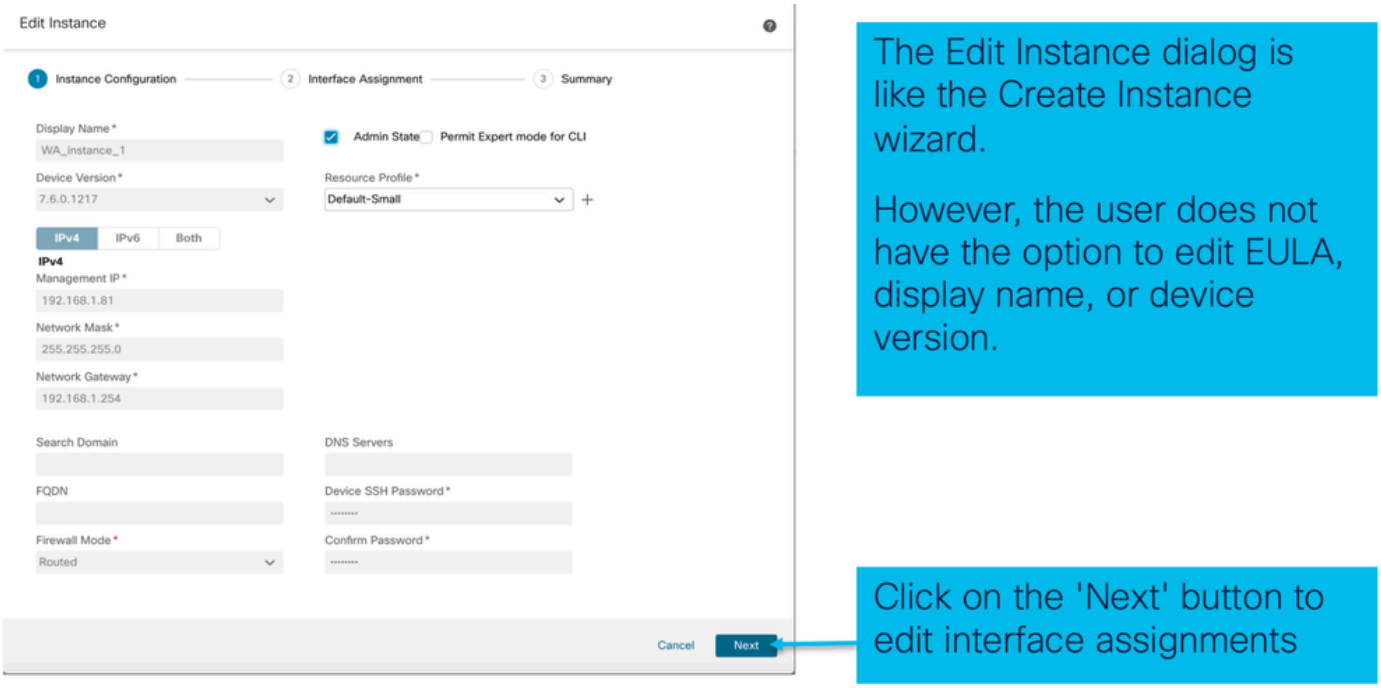


Edit an Instance

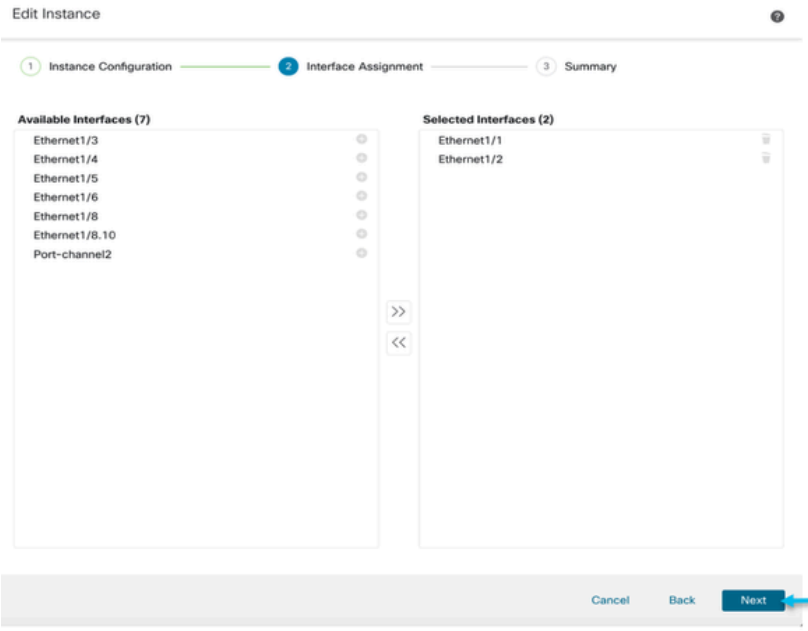
Click the pencil icon to edit an FTD instance:



Step 1. Edit FTD instance:



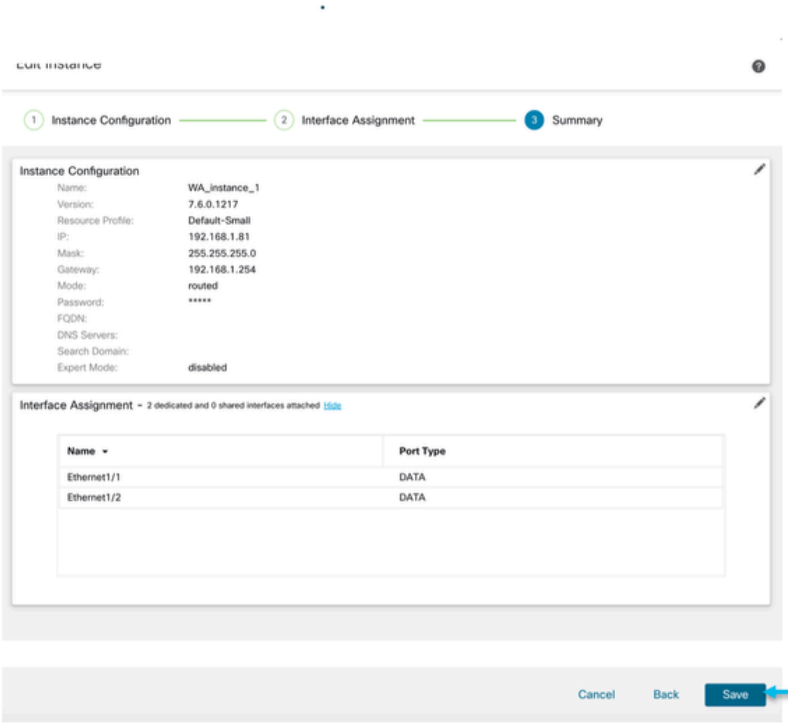
Step 2. Edit interface assignments for an instance:



The next step allows the user to modify interface assignments. User can add new interface or remove existing interfaces.

Click on the 'Next' button to view a summary of changes made to the instance

Step 3. Summary of edit instance:

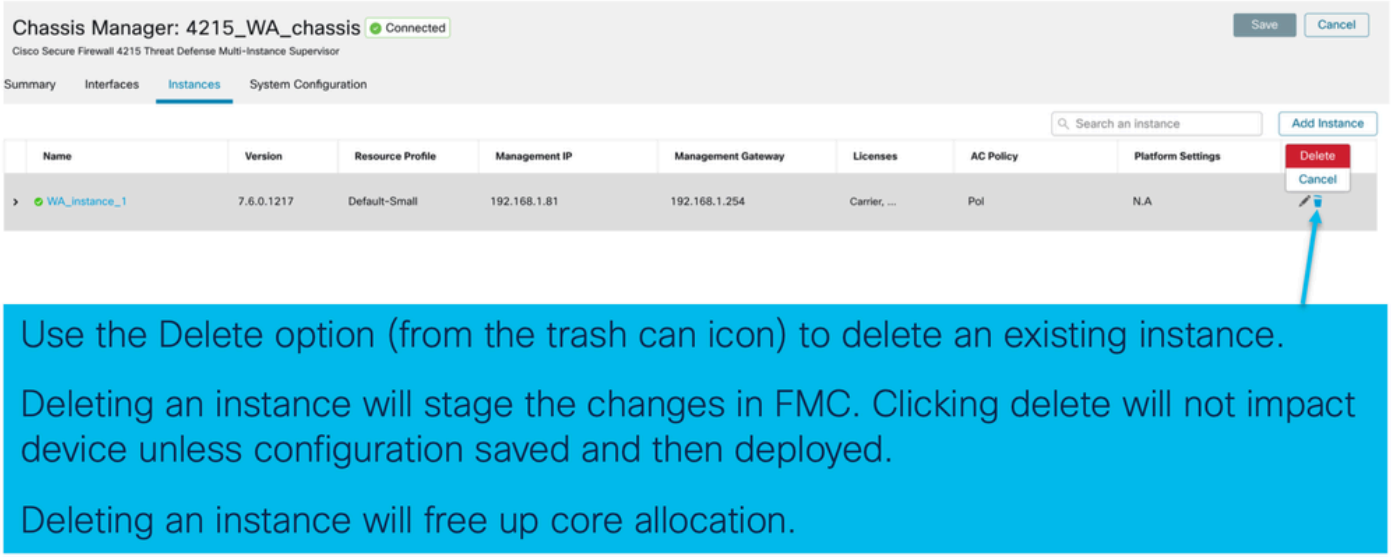


The last step of editing an instance is to view the summary of changes made to the instance.

Each tile has a pencil icon that navigates user to respective section of the edit steps.

Click the 'Save' button to stage the configuration changes in FMC. The user can review and deploy the changes at a later point in time.

Delete Instance



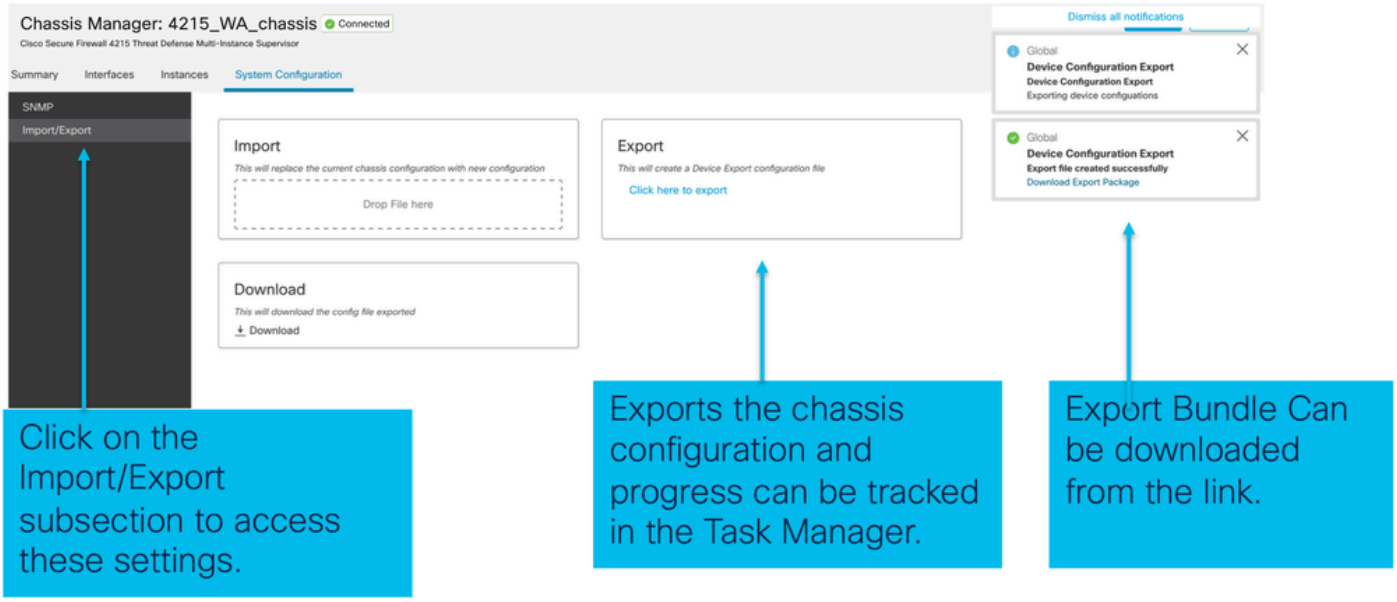
SNMP Configuration

Navigate to the system configuration tab for configuring SNMP:

Chassis Import / Export

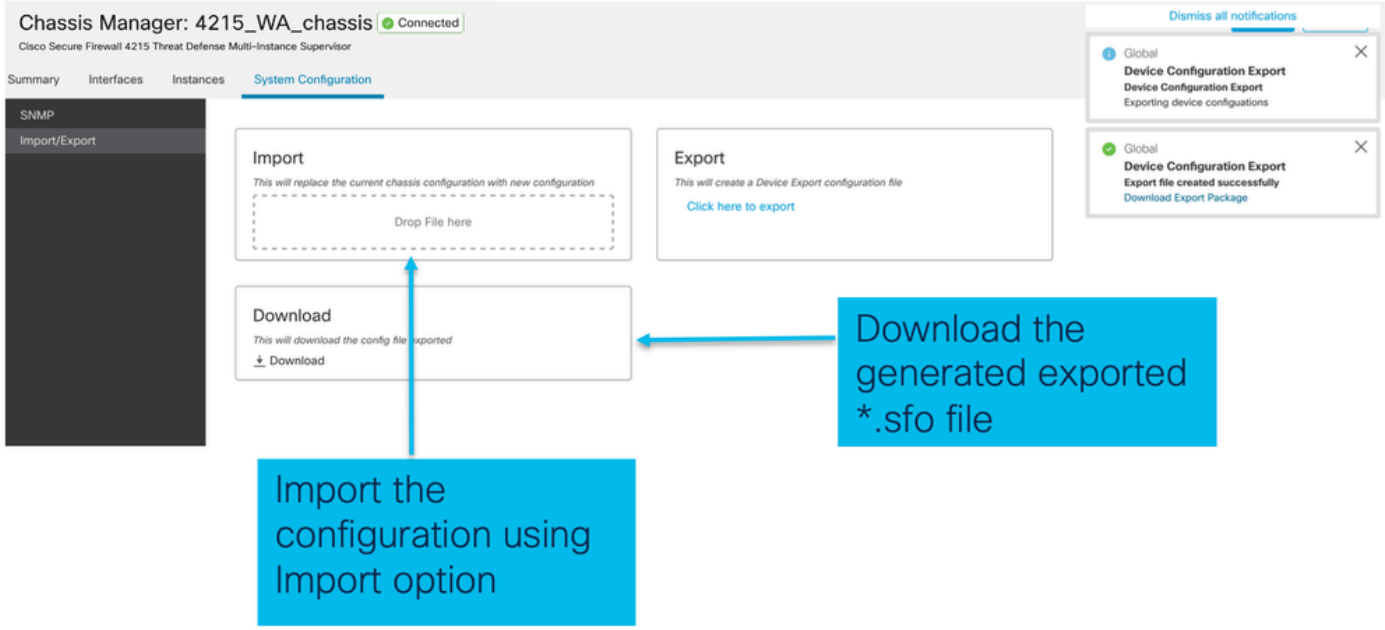
Export Configuration

Navigate to **Manage Chassis** > **System Configuration** > **Import/Export**:



Import Configuration

Navigate to **Manage Chassis** > **System Configuration** > **Import/Export**:



Things to Know about Chassis Import / Export

- All the existing configurations on the chassis is replaced by the configuration in the imported file.
- The platform software version where the config is imported must be same as exported version.
- The chassis where you are importing configuration must have same number of network modules installed when export was taken.
- The chassis where configuration is imported must have same application image installed for logical devices.
- Application-specific configuration settings are not exported. Only chassis configurations are exported.
- FTD Instance(s) back up needs to be taken separately.

Chassis Platform Settings Policy

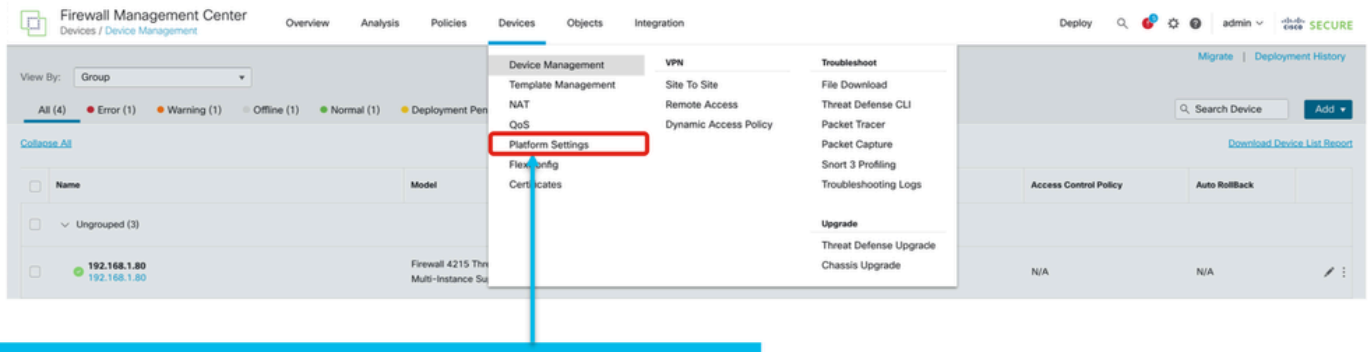
The chassis platform settings policy allows users to configure these platform specific configurations:

- Time Synchronization (NTP)
- DNS
- Syslog
- Time Zone
- User can create a new "Chassis Platform Setting" policy and assign it to multiple 4200 Series (MI mode) Chassis.

Tip: Chassis platform settings only apply to the chassis. If the user wants to apply platform settings to their instances, they can use a Threat Defense Platform Settings Policy.

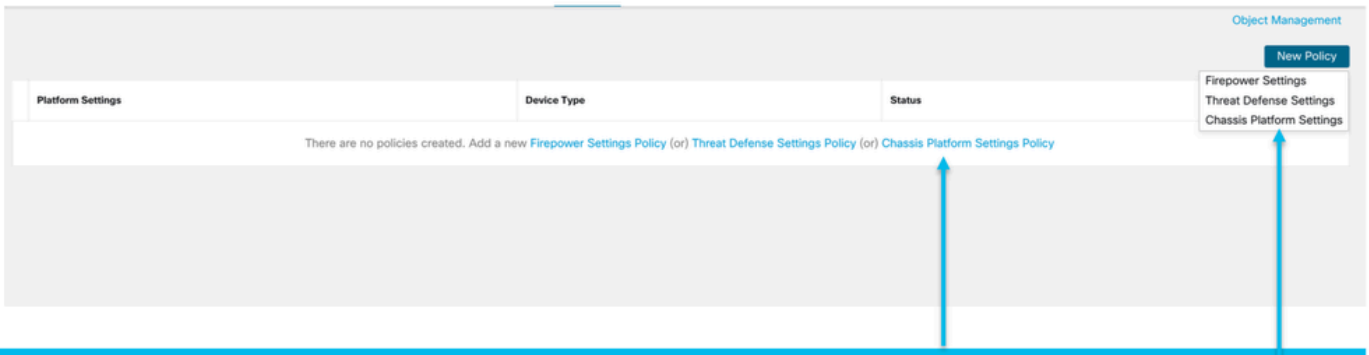
---

1. Navigate to chassis Platform Settings policy:



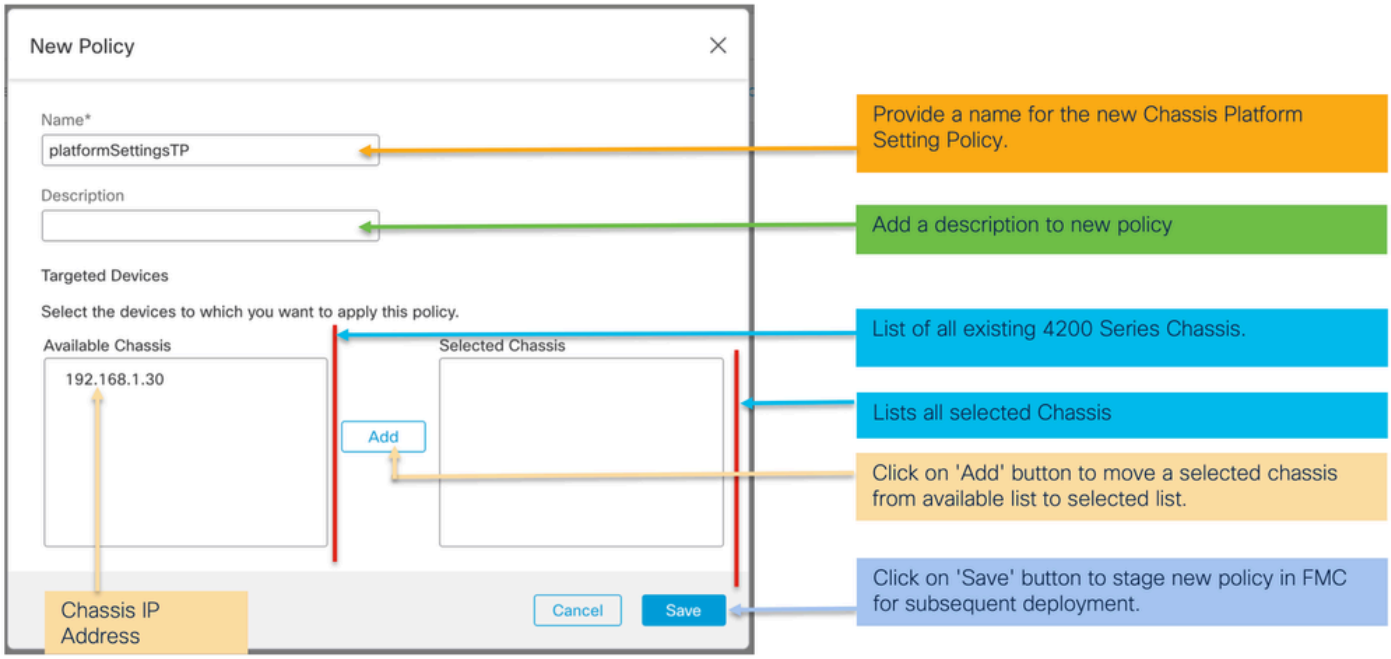
Head to the Platform Settings page to manage your Chassis Platform Settings.

2. Create Chassis Platform Settings:

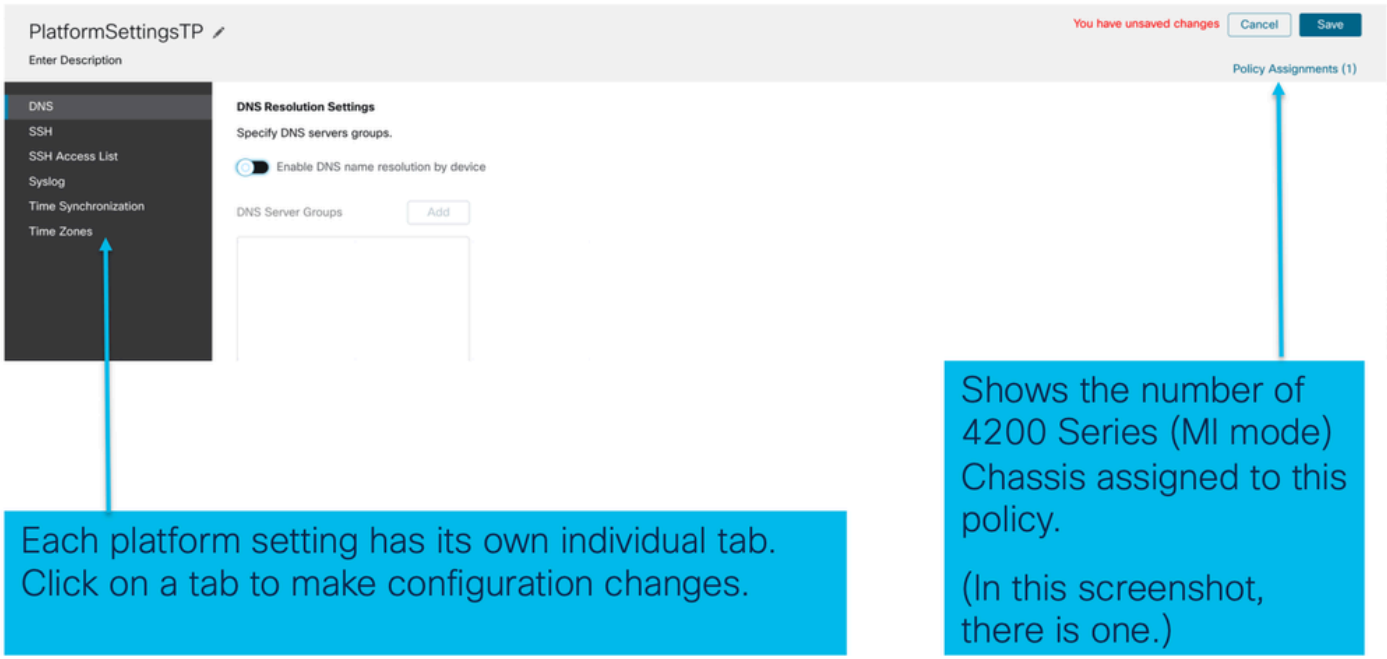


'Chassis Platform Settings' was added in 7.4.1.

- To create a new Chassis Platform Settings Policy click on 'Chassis Platform Settings' under 'New Policy' to launch new platform settings dialog.
- When there are no existing platform setting policies, you will see the 'Chassis Platform Settings Policy' link. This is your launch point to create.

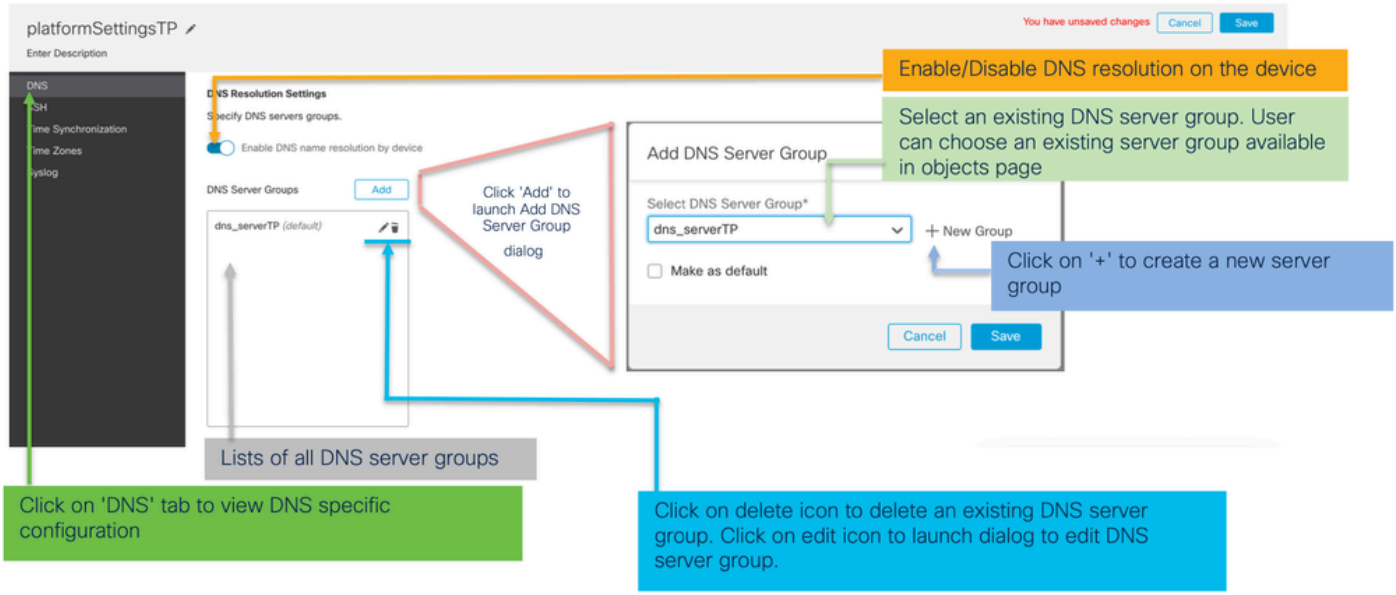


3. Chassis Platform Settings Policy Page:



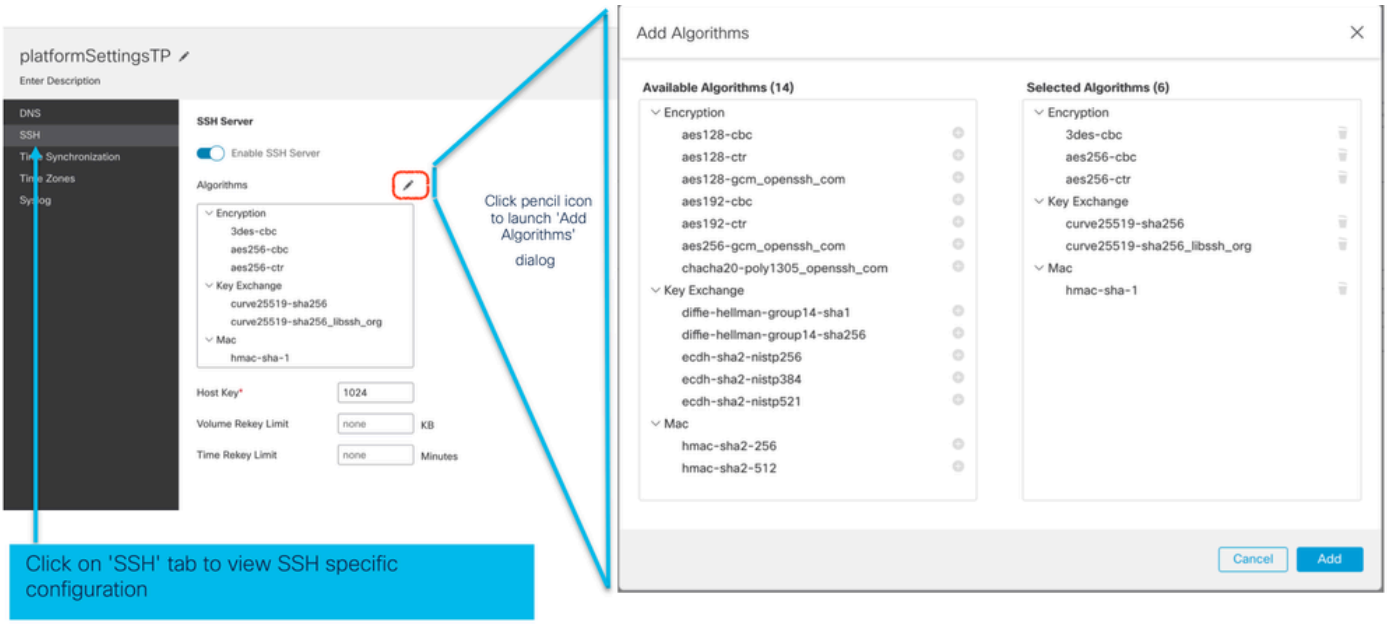
Chassis Platform Settings: DNS

Enable and Add DNS Server Groups under DNS section of Chassis Platform settings policy:

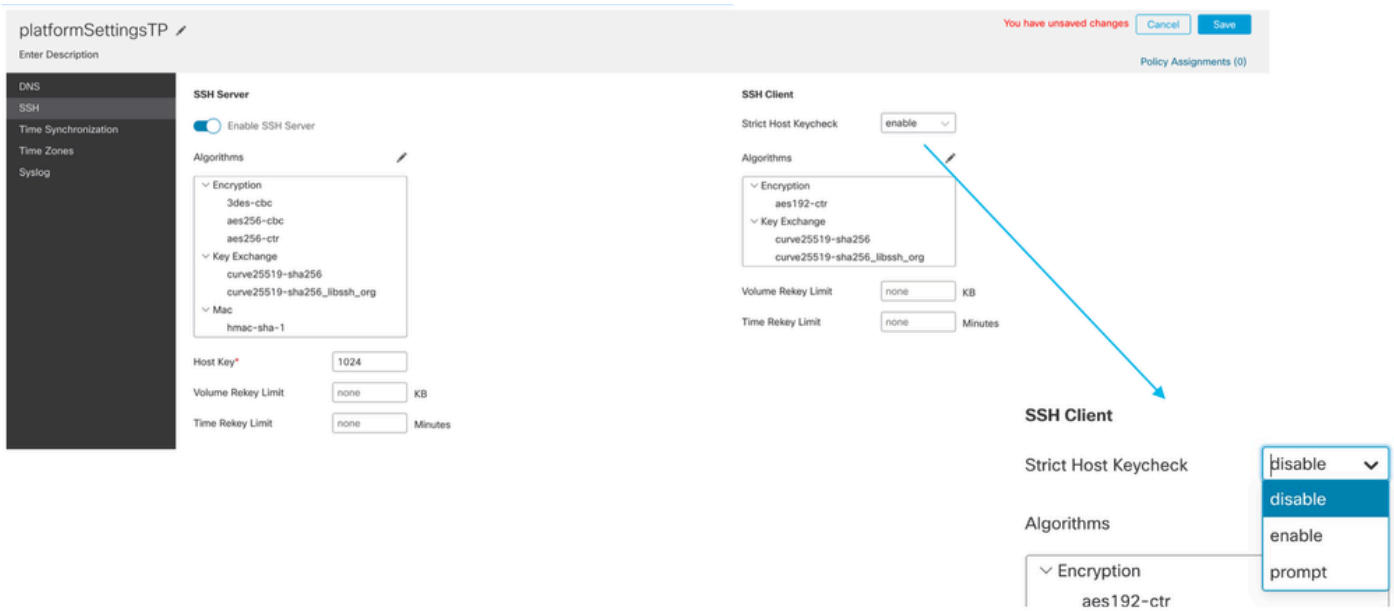


Chassis Platform Settings: SSH

- Enable and Add SSH Server under SSH section of Chassis Platform settings policy:



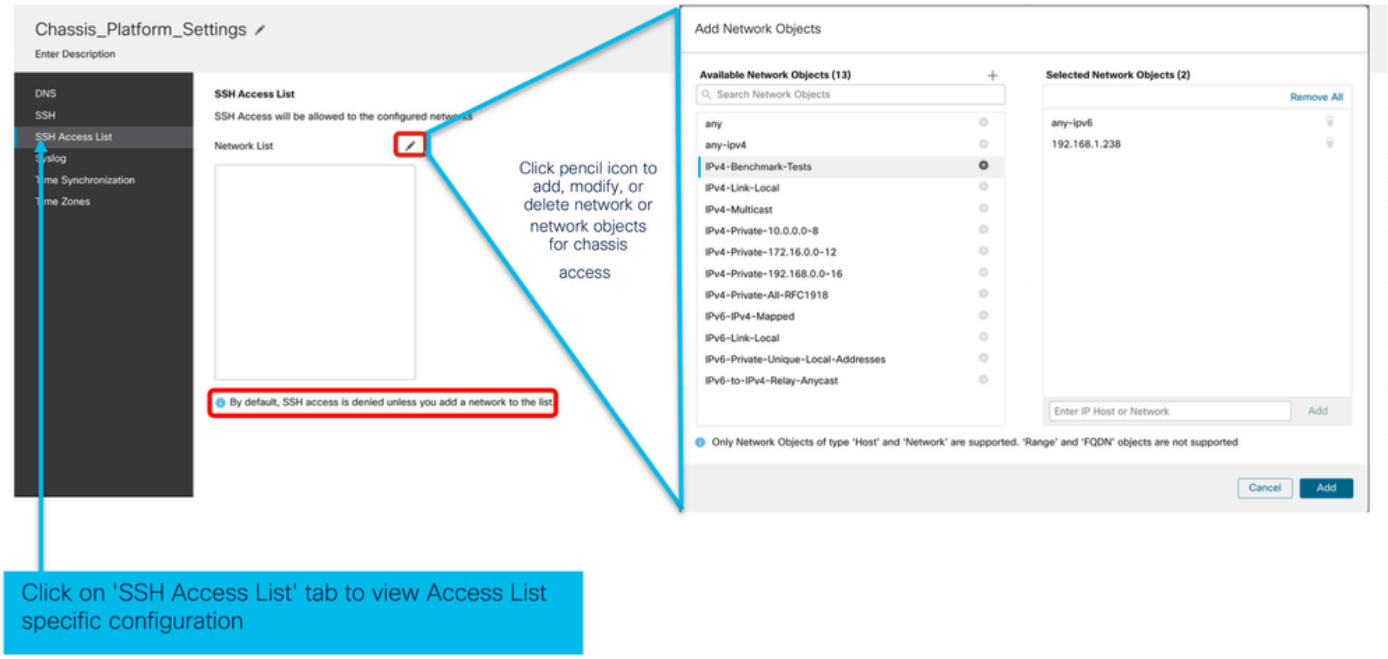
- Enable and Add SSH Client:



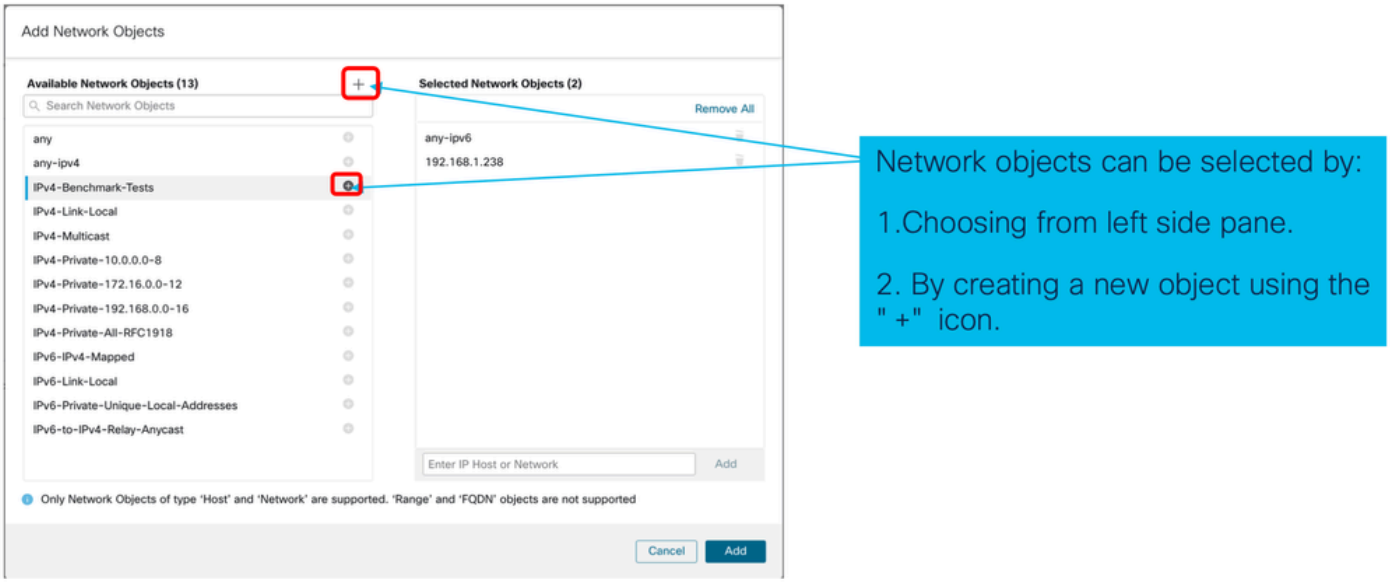
Chassis Platform Settings: SSH Access List

This tab shows up only after enabling SSH under SSH section of Chassis platform settings.

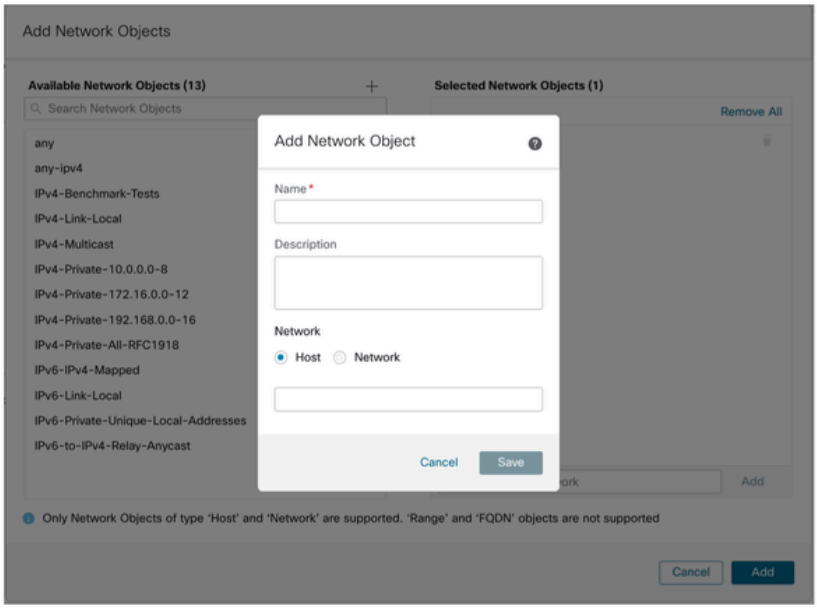
- Create SSH Access List:



- Add Network Objects for SSH access list:



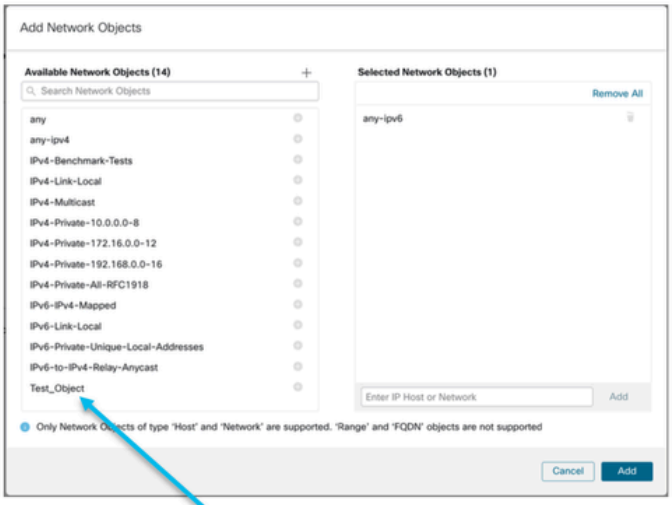
- Add a new Network Object:



Only Host and Network types are supported for chassis access list.

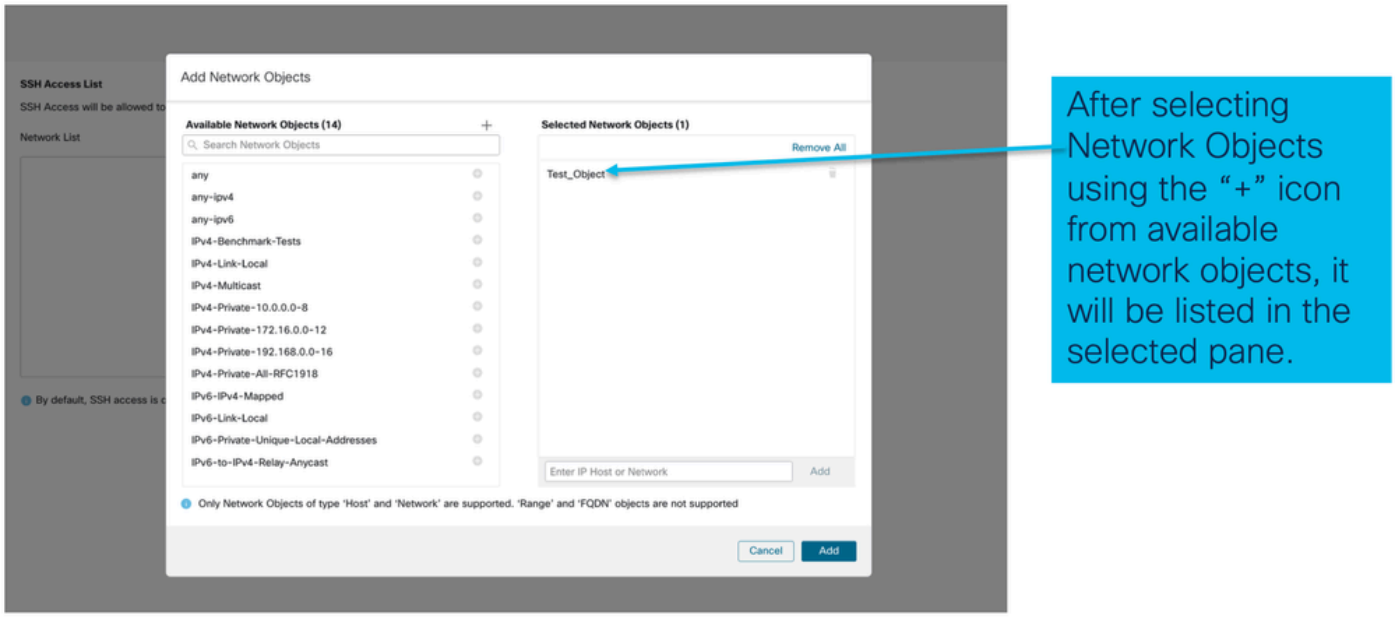
Range and FQDN are NOT allowed.

- View Network Object(s):

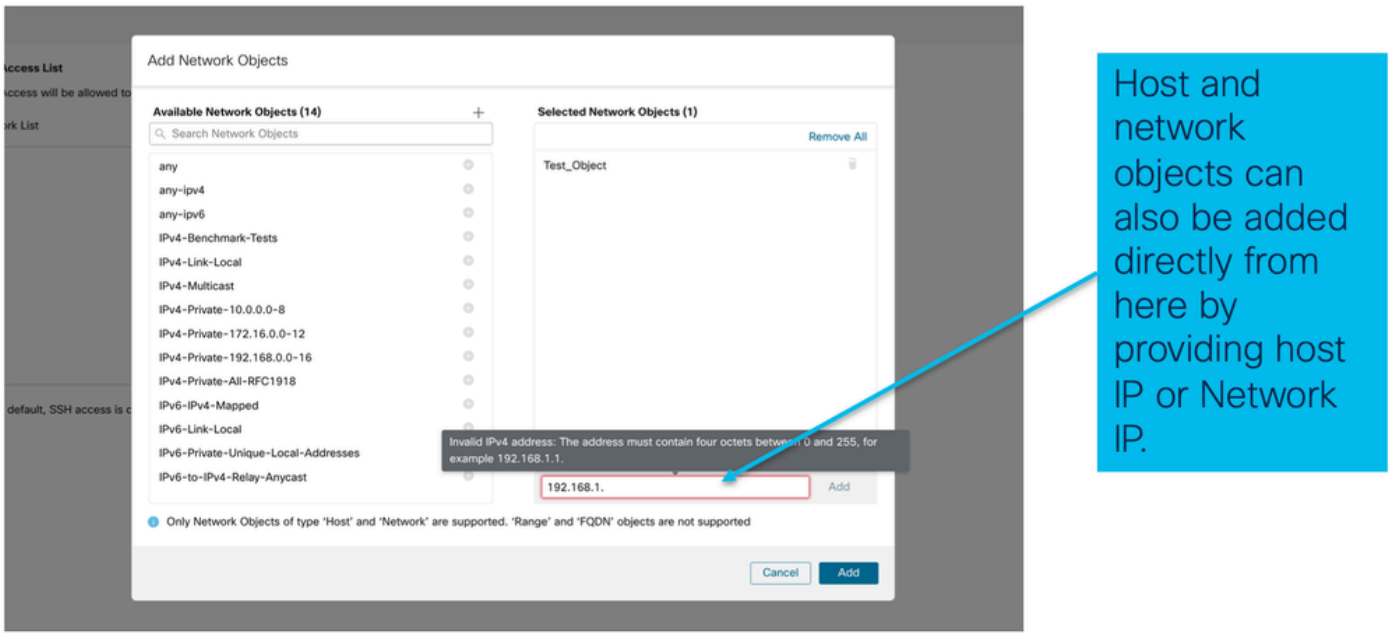


After creation of host object, it will be listed in the available network objects.

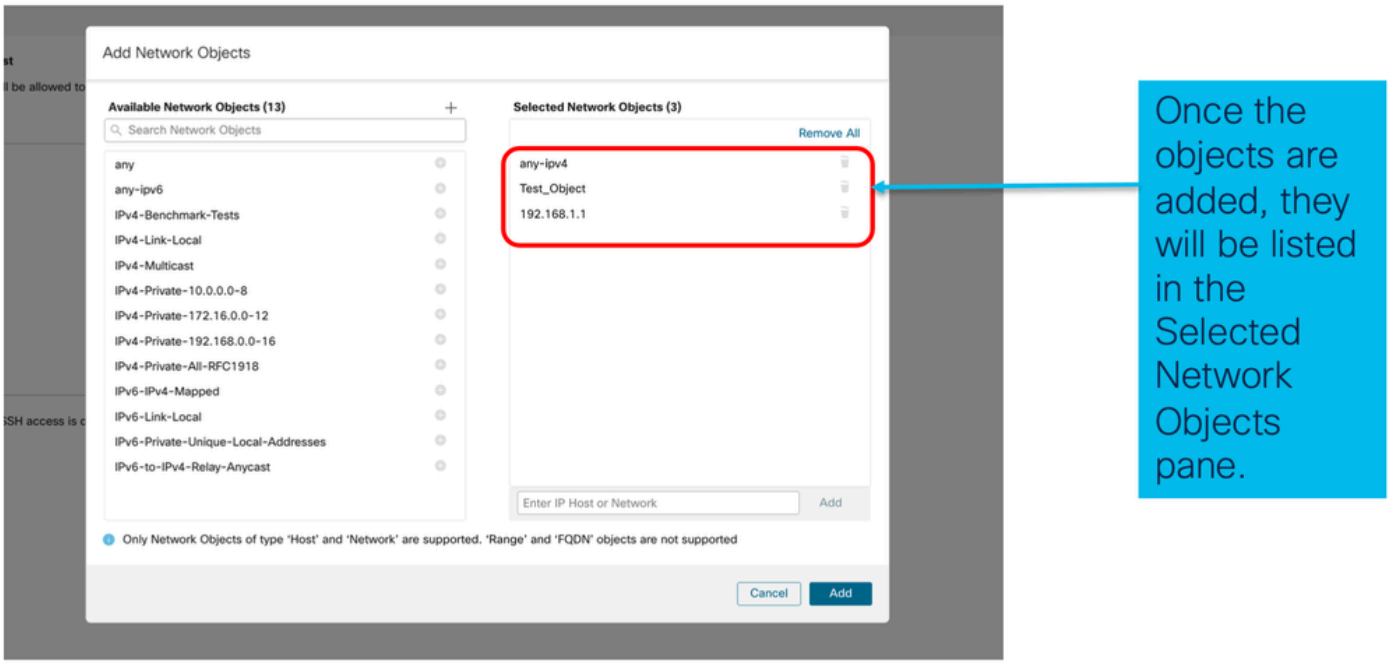
- Pick Network Object(s):



- Network Objects can be created as also shown in this image:



- View Added Network Objects:

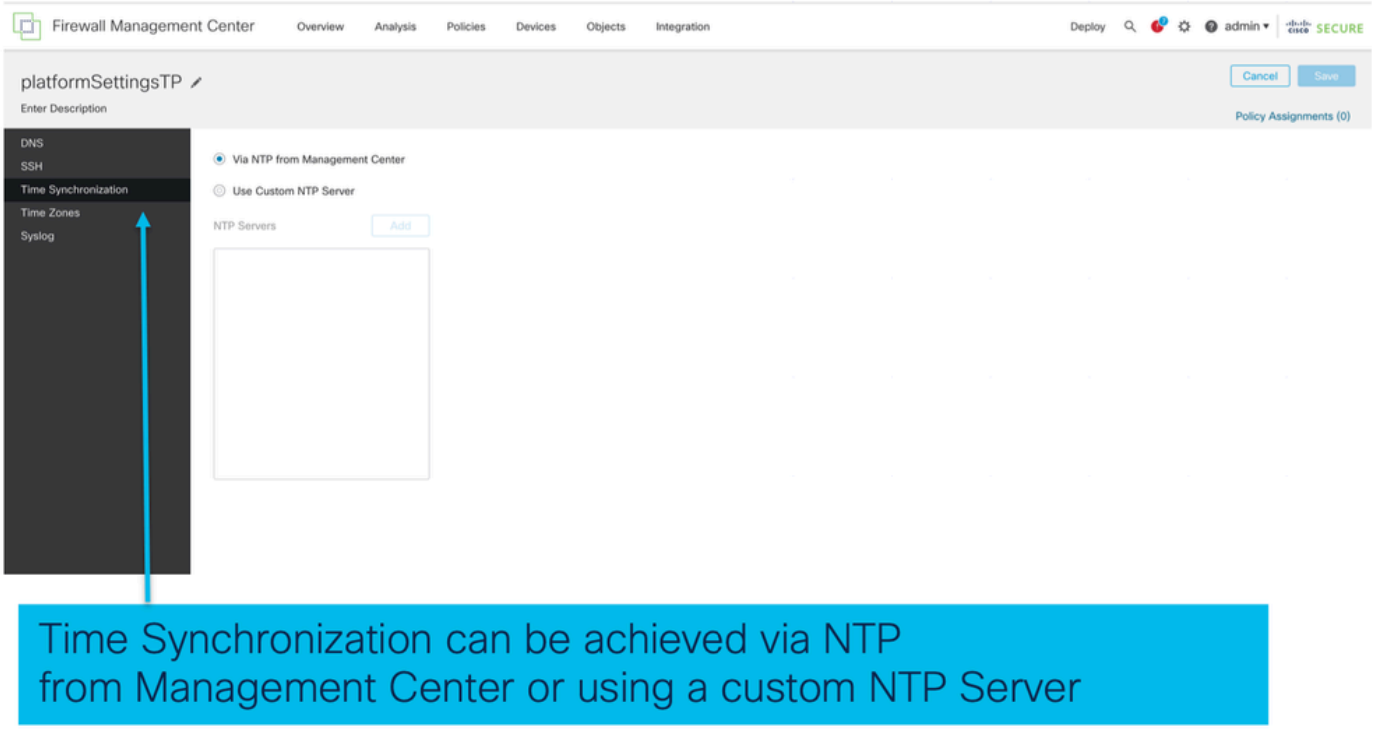


Chassis Platform Settings: Time Synchronization

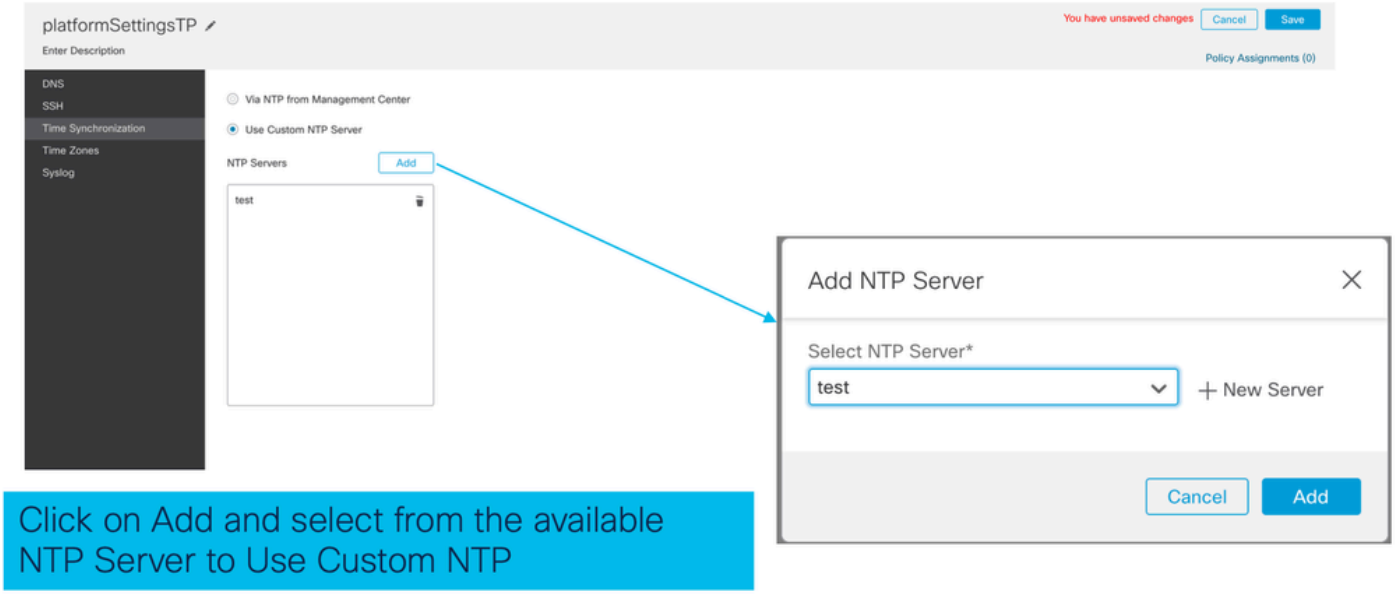
Time Synchronization can be done in two ways:

1. Via NTP from the Management Centre
2. On the custom NTP Server

From NTP from Management Center

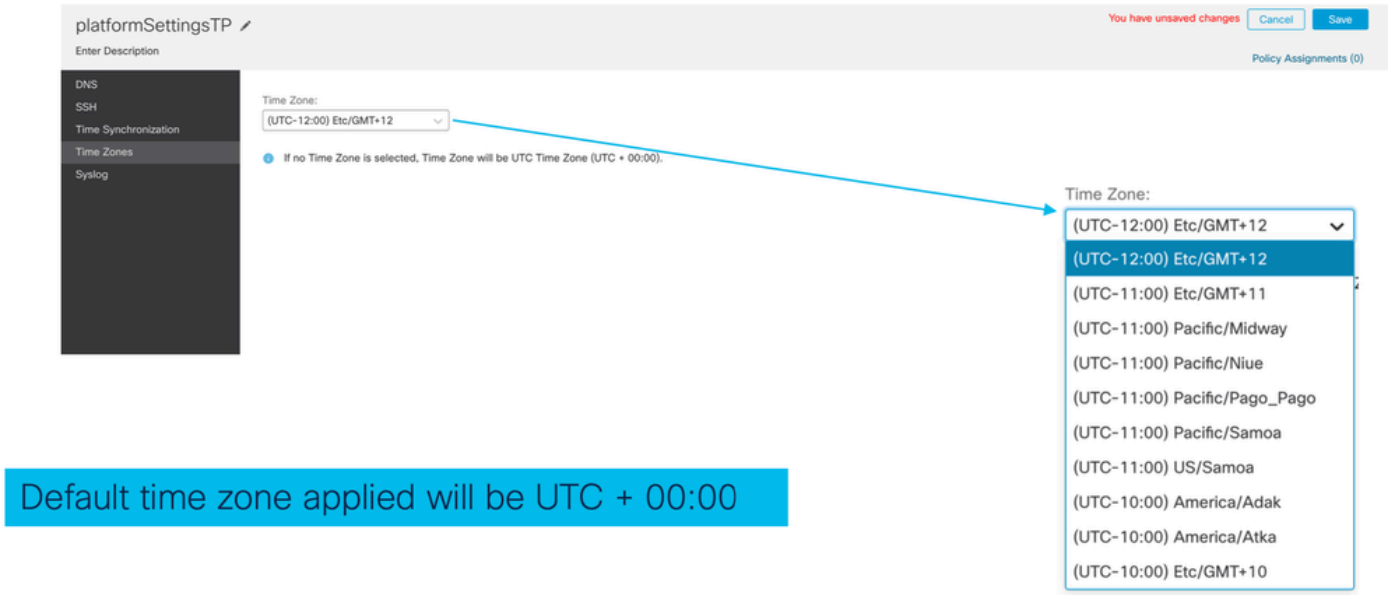


On the Custom NTP Server



Chassis Platform Settings: Time Zones

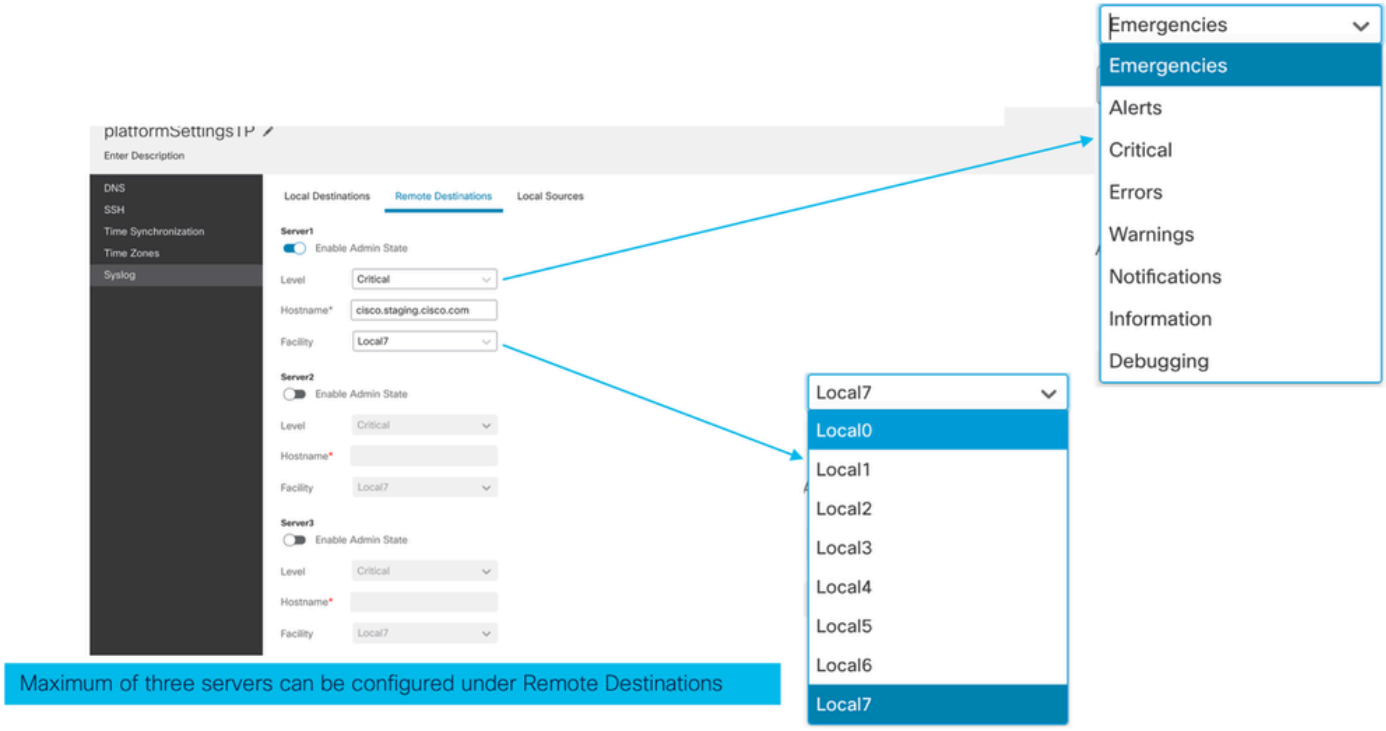
Set time zones:



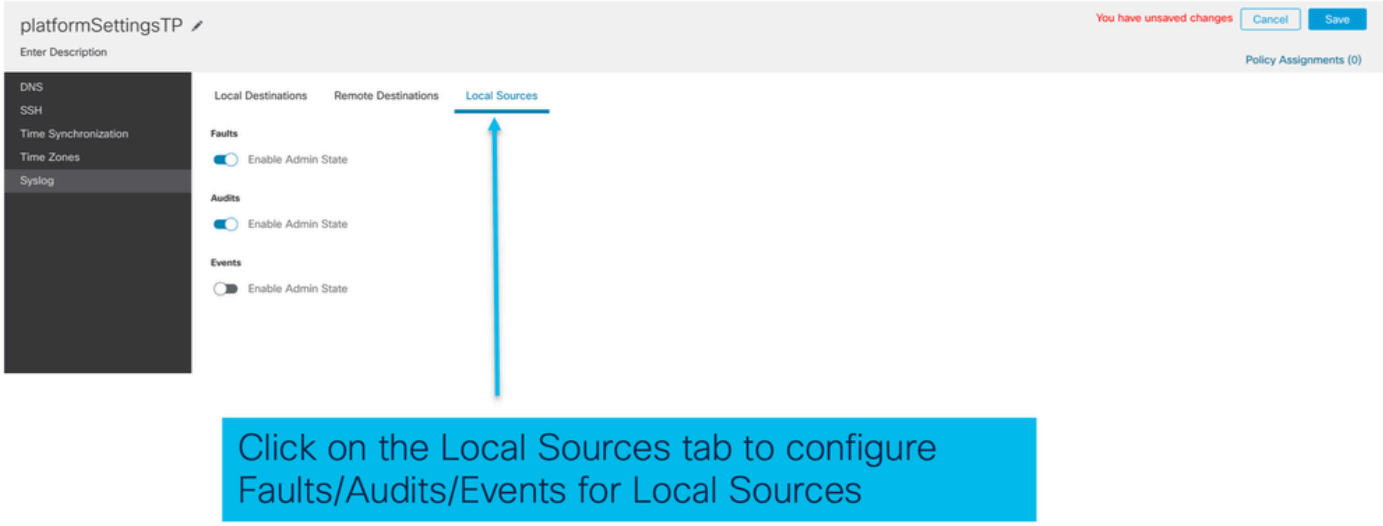
Chassis Platform Settings: Syslog

- Syslog Local Destinations tab:

- Syslog Remote Destinations tab:



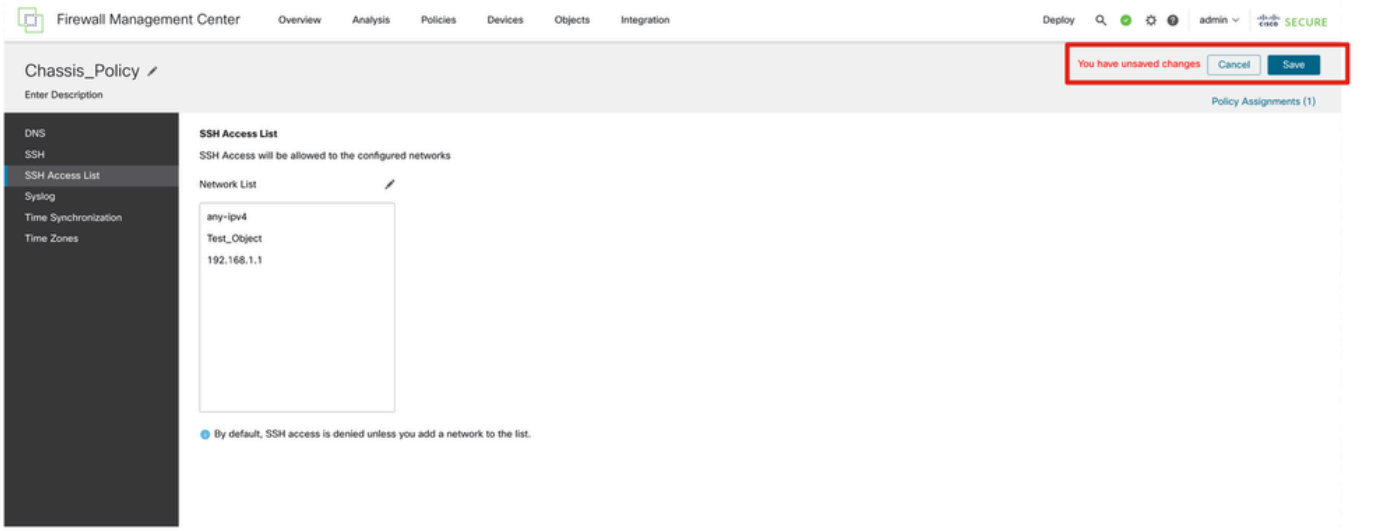
- Syslog Local Sources tab:



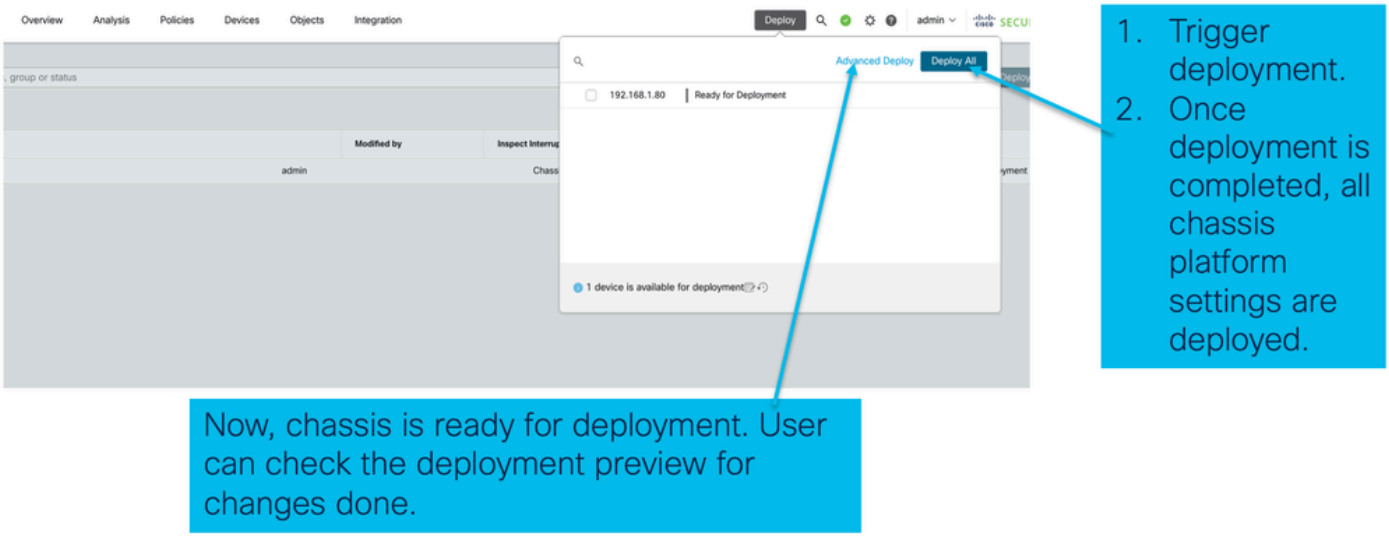
Click on the Local Sources tab to configure Faults/Audits/Events for Local Sources

Chassis Platform Settings: Save and Deploy

Save Chassis Platform Setting Changes, then deploy:

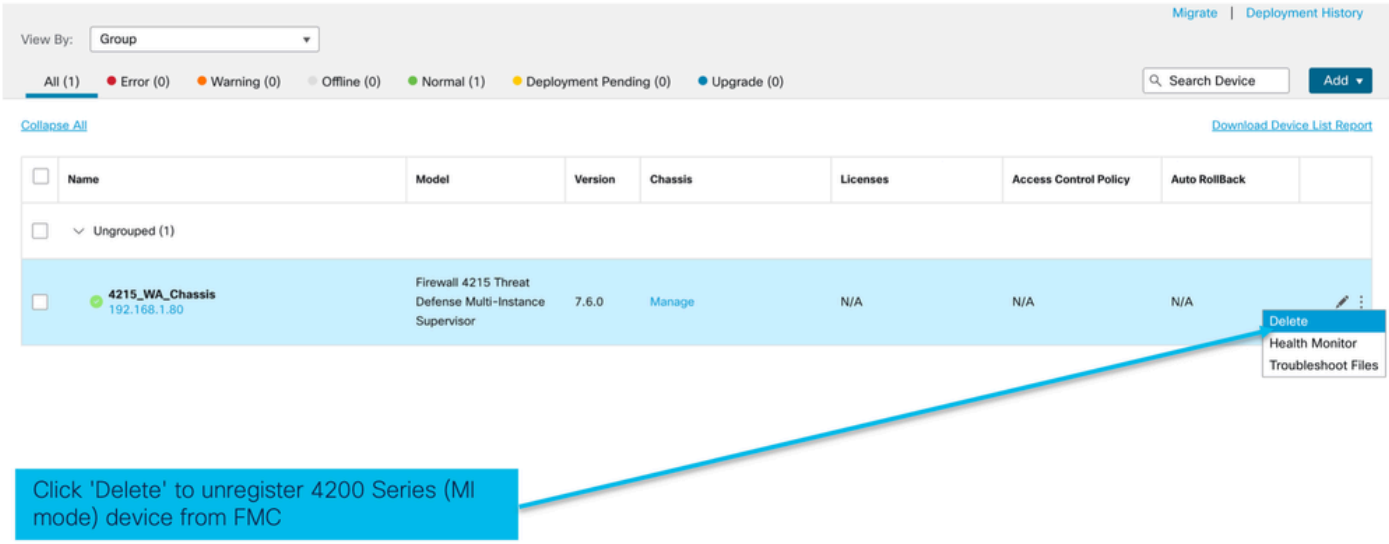


Now, save the changes which has all the platform settings. Chassis will go for pending deployment.



Unregistering Chassis

To unregister a chassis from FMC, navigate to **Devices** > **Device Management** > **delete**.



Convert from Multi-Instance to Native Mode

Currently, FMC only supports conversion from Native to Multi-Instance. Consequently, to convert a device back to Native mode, the user has to use the CLI.

Step 1: Unregister the Chassis from the FMC.

Step 2: Use this CLI command to convert 4200 Series device to native mode:

```
firepower-4215# scope system
firepower-4215 /system # set deploymode native
```

FMC Rest APIs

FMC Public REST APIs are available for all the operations supported from FMC.

REST APIs for Native to Multi- Instance Conversion

POST API to verify if native device is ready for Multi-Instance Conversion:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/operational/switchmodereadinesscheck

Sample POST request JSON:

```
{
  "devices": [
    {
      "id": "DeviceUUID",
      "type": "Device"
    }
  ],
  "conversionType": "NATIVE_TO_MULTI_INSTANCE"
}
```

POST API to trigger single native to Multi-Instance Conversion:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/operational/switchmode

Sample POST request JSON:

```
{
  "items": [
    {
```

```
      "id": "<Device_UUID>",
      "displayName": "Sample_Chassis_Name1"
    }
  ],
  "conversionType": "NATIVE_TO_MULTI_INSTANCE"
}
```

POST API to trigger bulk native to Multi-Instance Conversion:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/operational/switchmode

Sample POST request JSON:

```
{
  "items": [
    {
      "id": "<Device_UUID1>",
      "displayName": "Sample_Chassis_Name1"
    },
    {
      "id": "<Device_UUID2>",
      "displayName": "Sample_Chassis_Name2"
    }
  ],
  "conversionType": "NATIVE_TO_MULTI_INSTANCE"
}
```

REST APIs for Chassis Management

POST Add a Chassis to management center:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis

GET all Chassis:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/

GET a specific Chassis by uuid:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{objectId}

Delete a Chassis by uuid:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{objectId}

Sample POST request JSON:

```
{
  "type": "FMCManagedChassis",
  "chassisName": "CHASSIS123",
  "chassisHostName": "192.168.xx.74",
  "regKey": "*****"
```

```
}
```

REST APIs for Managing Netmods (Network Modules)

GET a Network Module by uuid:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/networkmodules/{object

GET ALL Network Modules:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/networkmodules/

PUT – Edit an existing Network Module by uuid :

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/networkmodules/{object

PUT – Retrieve Network module data from FXOS and update Management Center:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/operational/syncnetwork

Sample GET response


```
{
  "metadata": {
    "timestamp": 1688670821060,
    "domain": {
      "name": "Global",
      "id": "e276abec-e0f2-11e3-8169-************",
      "type": "Domain"
    }
  },
  "links": {
    "self": "https://u32c01p10-vrouter.cisco.com:32300/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169
  },
  "id": "0050568A-3F3F-0ed3-0000-************",
  "moduleState": "ENABLED",
  "type": "NetworkModule",
  "description": "Cisco FPR 8X1G 8X10G 1RU Module",
  "model": "FPR-3120",
  "operationState": "ok",
  "numOfPorts": 16,
  "slotId": "1",
  "vendor": "Cisco Systems, Inc.",
  "name": "Network Module 1"
}
```


REST APIs for Instance Management

POST Add a Chassis to management center:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/logicaldevices

GET all Chassis:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/logicaldevices

GET a specific Instance by uuid:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/logicaldevices/{objectId

PUT - Edit an Instance by uuid:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/logicaldevices/{objectId

Delete a Chassis by uuid:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/logicaldevices/{objectId

Sample PUT request:

```
{
  "name": "ftd1",
  "operationalState": "string",
  "deviceRegistration": {
    "licenseCaps": [
      "MALWARE",
      "URLFilter",
      "CARRIER",
      "PROTECT"
    ],
    "accessPolicy": {
      "name": "AC Policy name",
      "id": "<ac policy uuid>",
      "type": "AccessPolicy"
    },
    "deviceGroup": {
      "name": "DeviceGroup name",
      "id": "<device group uuid>",
      "type": "DeviceGroup"
    }
  },
  "managementBootstrap": {
    "ipv4": {
      "gateway": "192.168.xx.68",
      "ip": "192.168.xx.78",
      "mask": "255.255.255.0"
    },
    "adminState": "enable",
    "firepowerManagerIP": "192.168.xx.32",
    "permitExpertMode": "yes",
    "searchDomain": "string",
    "firewallMode": "Routed",
    "dnsServers": "192.168.xx.123",
    "natId": "natId",
    "registrationKey": "regKey",
    "adminPassword": "adminPwd",
    "fqdn": "fqdn"
  },
  "externalPortLink": [
    {
      "name": "Ethernet1/1",
      "id": "<interface uuid>",
      "type": "ChassisInterface"
```

```
    },
    {
      "name": "Ethernet2/2.1",
      "id": "<subInterface uuid>",
      "type": "ChassisInterface"
    }
  ],
  "type": "LogicalDevice"
}
```


REST APIs for SNMP Management

GET an SNMP Setting by uuid:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/snmpsettings/{objectId}

GET ALL SNMP Settings:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/snmpsettings/

PUT – Edit an existing Network Module by uuid:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/snmpsettings/{objectId}

Sample GET response:


```
{
  "snmpAdminInstance": {
    "id": "logicalDeviceUuid",
    "type": "LogicalDevice",
    "name": "ftd3"
  },
  "id": "snmpsettingsUUID2",
  "type": "SnmpSetting"
}
```


REST APIs to Fetch Summary

This list contains detailed information on the REST APIs for fetching the summary:

- Faults
- Instances
- Inventory
- Interfaces
- App Info

GET Faults Summary for a chassis:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/faultsummary

Sample Response:

```
{
  "links": {
    "self": "<fmc_url>/api/fmc_config/v1/domain/domainUUID/chassis/fmcmanagedchassis/containerUUID/faul
  },
  "items": [
    {
      "faultList": [
        {
          "id": 27429,
          "isAcknowledged": "no",
          "cause": "device-registration-pending",
          "gateway": "3::1",
          "ip": "3::2",
          "prefixLength": "33"
        }
      ],
      "managementPort": "Management1",
      "operationalState": "online",
      "adminState": "enabled",
      "deployType": "container"
    }
  ],
  "modifiedTime": "2022-07-05T06:39:25Z",
  "type": "InstanceSummary"
  ],
  "paging": {
    "offset": 0,
    "limit": 25,
    "count": 1,
    "pages": 1
  }
}
```


GET Instances Summary for a chassis:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/instancesummary

Sample Response:


```
{
  "links": {
    "self": "<fmc_url>/api/fmc_config/v1/domain/domainUUID/chassis/fmcmanagedchassis/containerUUID/inst
  },
  "items": [
    {
      "instanceList": [
        {
          "name": "ftdmi2",
          "startupVersion": "7.3.0.1402",
          "coresUsed": 6,
          "ipv4": {
            "gateway": "192.168.xx.68",
            "ip": "192.168.xx.78",
            "mask": "255.255.255.0"
          },
          "ipv6": {
            "gateway": "3::1",
            "ip": "3::2",
```

```
            "prefixLength": "33"
          },
          "managementPort": "Management1",
          "operationalState": "online",
          "adminState": "enabled",
          "deployType": "container"
        }
      ],
      "modifiedTime": "2022-07-05T06:39:25Z",
      "type": "InstanceSummary"
    }
  ],
  "paging": {
    "offset": 0,
    "limit": 25,
    "count": 1,
    "pages": 1
  }
}
```


GET Inventory Summary for a chassis:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/inventorysummary

Sample response:


```
{
  "links": {
    "self": "<fmc_url>/api/fmc_config/v1/domain/domainUUID/chassis/fmcmanagedchassis/containerUUID/inve
  },
  "items": [
    {
      "fanList": [
        {
          "operationalState": "operable",
          "operability": "operable",
          "power": "on",
          "thermalStatus": "ok",
          "module": 1,
          "tray": 1,
          "id": 1,
          "model": "N/A",
          "vendor": "N/A"
        },
        {
          "operationalState": "operable",
          "operability": "operable",
          "power": "on",
          "thermalStatus": "ok",
          "module": 1,
          "tray": 1,
          "id": 2,
          "model": "N/A",
          "vendor": "N/A"
        }
      ],
      "powerSupplyList": [
        {
```

```
          "id": 2,
          "operationalState": "operable",
          "operability": "operable",
          "serialNumber": "************",
          "thermalStatus": "ok",
          "model": "FPR2K-PWR-AC-400",
          "vendor": "Cisco Systems, Inc"
        }
      ],
      "processorList": [
        {
          "id": 1,
          "operationalState": "operable",
          "operability": "operable",
          "vendor": "AuthenticAMD",
          "model": "49 AMD EPYC 7282 16-Core Processor",
          "type": "CPU",
          "thermalStatus": "ok"
        }
      ],
      "securityModuleList": [
        {
          "id": 1,
          "operationalState": "ok",
          "operability": "operable",
          "serialNumber": "************",
          "vendor": "Cisco Systems, Inc",
          "model": "FPR-3120",
          "availableCores": 24,
          "totalCores": 32
        }
      ],
      "memoryList": [
        {
          "capacity": 65536,
          "id": 1,
          "array": 1,
          "bank": 0,
          "model": "HMAA8GR7AJR4N-XN",
          "operationalState": "operable",
          "operability": "operable",
          "performance": "ok",
          "power": "not-supported",
          "serialNumber": "********",
          "thermalStatus": "ok",
          "vendor": "Hynix"
        }
      ],
      "model": "FPR-3120",
      "availableCores": 24,
      "totalCores": 32
    }
  ],
  "paging": {
    "offset": 0,
    "limit": 25,
    "count": 1,
    "pages": 1
  }
}
```

GET Interface Summary for a chassis:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/interfacesummary

Sample Response:


```
{
  "links": {
    "self": "<fmc_url>/api/fmc_config/v1/domain/domainUUID/chassis/fmcmanagedchassis/containerUUID/inte
  },
  "items": [
    {
      "interfaceList": [
        {
          "name": "Ethernet1/8",
          "operationalState": "up",
          "adminState": "disabled",
          "portType": "data",
          "operationalSpeed": "10mbps",
          "adminSpeed": "1gbps",
          "adminDuplex": "fullDuplex",
          "autoNegotiation": "yes",
          "mediaType": "rj45",
          "type": "PhysicalInterface"
        },
        {
          "name": "Ethernet1/7",
          "operationalState": "up",
          "adminState": "disabled",
          "portType": "data",
          "operationalSpeed": "1gbps",
          "adminSpeed": "1gbps",
          "adminDuplex": "fullDuplex",
          "autoNegotiation": "yes",
          "mediaType": "rj45",
          "type": "PhysicalInterface"
        },
        {
          "name": "Ethernet1/6",
          "operationalState": "up",
          "adminState": "disabled",
          "portType": "data",
          "operationalSpeed": "1gbps",
          "adminSpeed": "1gbps",
          "adminDuplex": "fullDuplex",
          "autoNegotiation": "yes",
          "mediaType": "rj45",
          "type": "PhysicalInterface"
        },
        {
          "name": "Ethernet1/3",
          "operationalState": "up",
          "adminState": "disabled",
          "portType": "data",
          "operationalSpeed": "1gbps",
          "adminSpeed": "1gbps",
          "adminDuplex": "fullDuplex",
          "autoNegotiation": "yes",
          "mediaType": "rj45",
          "type": "PhysicalInterface"
```

```
        },
        {
          "name": "Ethernet1/2",
          "operationalState": "up",
          "adminState": "enabled",
          "portType": "data",
          "operationalSpeed": "1gbps",
          "adminSpeed": "1gbps",
          "adminDuplex": "fullDuplex",
          "autoNegotiation": "yes",
          "mediaType": "rj45",
          "type": "PhysicalInterface"
        },
        {
          "name": "Ethernet1/1",
          "operationalState": "up",
          "adminState": "enabled",
          "portType": "data",
          "operationalSpeed": "1gbps",
          "adminSpeed": "1gbps",
          "adminDuplex": "fullDuplex",
          "autoNegotiation": "yes",
          "mediaType": "rj45",
          "type": "PhysicalInterface"
        },
        {
          "name": "Port-channel48",
          "operationalState": "up",
          "adminState": "enabled",
          "portType": "data",
          "operationalSpeed": "1gbps",
          "adminSpeed": "1gbps",
          "adminDuplex": "fullDuplex",
          "autoNegotiation": "yes",
          "mediaType": "rj45",
          "type": "EtherChannelInterface"
        }
      ],
      "modifiedTime": "2022-07-05T06:39:25Z",
      "type": "InterfaceSummary"
    }
  ],
  "paging": {
    "offset": 0,
    "limit": 25,
    "count": 1,
    "pages": 1
  }
}
```


GET App Info for a chassis:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/inventorysummary

Sample Response:


```
{
```

```
  "links": {
    "self": "<fmc_url>/api/fmc_config/v1/domain/domainUUID/chassis/fmcmanagedchassis/containerUUID/appi
  },
  "items": [
    {
      "appVersion": "7.4.0.1024",
      "type": "AppInfo"
    },
    {
      "appVersion": "7.4.0.1075",
      "type": "AppInfo"
    }
  ],
  "paging": {
    "offset": 0,
    "limit": 25,
    "count": 1,
    "pages": 1
  }
}
```


REST APIs for Interface Management

This section has detailed information on the REST APIs for interface config management:

- URLs to be used for interface config modifications
- URLs to be used for Break/Join of interfaces
- URLs to be used for Sync Device configurations

Update Physical Interface

To support update of physical interfaces, these URLs have been introduced.

GET all physical interfaces:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/physicalinterfaces

GET a specific physical interface by interface uuid:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/physicalinterfaces/{interfaceUUID}

Update interface by interface uuid:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/physicalinterfaces/{interfaceUUID}

Physical Interface model looks like this:


```
{
  "metadata": {
    "supportedSpeed": "TEN_GBPS,ONE_GBPS,TWENTY_FIVE_GBPS,DETECT_SFP",
    "mediaType": "sfp",
    "sfpType": "none",
    "isBreakoutCapable": false,
```

```
    "isSplitInterface": false,
    "timestamp": 1692344434067,
    "domain": {
      "name": "Global",
      "id": "e276abec-e0f2-11e3-8169-**********",
      "type": "Domain"
    }
  },
  "type": "PhysicalInterface",
  "name": "Ethernet2/2",
  "portType": "DATA",
  "adminState": "DISABLED",
  "hardware": {
    "flowControlSend": "OFF",
    "fecMode": "AUTO",
    "autoNegState": true,
    "speed": "DETECT_SFP",
    "duplex": "FULL"
  },
  "LLDP": {
    "transmit": false,
    "receive": false
  },
  "id": "************************************"
}
```


Configure Sub-Interfaces

To support management of sub-interfaces, these URLs have been introduced.

GET all sub interfaces:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/subinterfaces

GET a specific sub interface by interface uuid:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/subinterfaces/{interfaceU

POST a new sub interface:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/subinterfaces

UPDATE interface by interface uuid :

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/subinterfaces/{interfaceU

DELETE a sub interface by interface uuid:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/subinterfaces/{interfaceU

Sub-interface model looks like this:


```
{
  "metadata": {
    "isBreakoutCapable": false,
    "isSplitInterface": false,
```

```
    "timestamp": 1692536476265,
    "domain": {
      "name": "Global",
      "id": "e276abec-e0f2-11e3-8169-**********",
      "type": "Domain"
    }
  },
  "type": "SubInterface",
  "name": "Ethernet1/3.3",
  "portType": "DATA",
  "subIntfId": 3,
  "parentInterface": {
    "type": "PhysicalInterface",
    "id": "00505686-9A51-0ed3-0000-**********",
    "name": "Ethernet1/3"
  },
  "vlanId": 3,
  "id": "************************************"
}
```


Configure EtherChannel Interfaces

To support management of etherchannel EtherChannel interfaces, these URLs have been introduced.

GET all etherchannel interfaces:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/etherchannelinterfaces/{

GET a specific etherchannel interface by interface uuid:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/etherchannelinterfaces/{

POST a new etherchannel interface:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/etherchannelinterfaces

UPDATE interface by interface uuid :

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/etherchannelinterfaces/{

DELETE a etherchannel interface by interface uuid:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/etherchannelinterfaces/{

EtherChannel Interface model looks like this:


```
{
  "metadata": {
    "supportedSpeed": "HUNDRED_MBPS,TEN_MBPS,ONE_GBPS",
    "timestamp": 1692536640172,
    "domain": {
      "name": "Global",
      "id": "e276abec-e0f2-11e3-8169-**********",
      "type": "Domain"
    }
  },
```

```
  "type": "EtherChannelInterface",
  "name": "Port-channel45",
  "portType": "DATA",
  "etherChannelId": 45,
  "selectedInterfaces": [
    {
      "type": "PhysicalInterface",
      "id": "00505686-9A51-0ed3-0000-**********",
      "name": "Ethernet1/4"
    },
    {
      "type": "PhysicalInterface",
      "id": "00505686-9A51-0ed3-0000-**********",
      "name": "Ethernet1/5"
    }
  ],
  "lacpMode": "ON",
  "lacpRate": "FAST",
  "adminState": "DISABLED",
  "hardware": {
    "flowControlSend": "OFF",
    "autoNegState": true,
    "speed": "ONE_GBPS",
    "duplex": "FULL"
  },
  "LLDP": {
    "transmit": true,
    "receive": true
  },
  "id": "00505686-9A51-0ed3-0000-**********"
}
```


REST APIs Break/Join Interfaces

To support the Breakout/Join of interfaces in 4200 Series, these URLs can be used:

GET:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/chassisinterfaces/{interfa

Evaluates the feasibility of break/join for an interface

POST:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/operational/breakoutinte

Breaks an interface

POST:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/operational/joininterface
Joins a set of broken interfaces

REST Flow for Interface Break

1. Find FMC managed chassis device (4200) using the fmcmanagedchassis endpoint.

GET /api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis

Returns the list of FMC managed chassis devices along with Multi Instance devices with the details like id, name, model of each device. Choose the ”MULTIINSTANCE” devices.

Sample Response:


```
{
  "id": "fcaa9ca4-85e5-4bb0-b049-**********",
  "type": "FMCManagedChassis",
  "chassisName": "192.168.0.75",
  "chassisMode": "MULTIINSTANCE",
  "links": {
    "self": "https://u32c01p06-vrouter.cisco.com:22512/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169
  }
}
```


2. Check if the interface is breakout capable using interfaces/physicalinterfaces endpoint.

Breakout is possible only if “isBreakoutCapable” is true and mediaType is QSFP.

GET /api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/interfaces

Sample Response:


```
{
  "metadata": {
    "supportedSpeed": "FORTY_GBPS,DETECT_SFP",  >>>>>>>>>
    "mediaType": "qsfp",                        >>>>>>>>>
    "sfpType": "none",
    "isBreakoutCapable": true,                  >>>>>>>>>
    "breakoutFactor": "4",                      >>>>>>>>>
    "isSplitInterface": false,
    "timestamp": 1692344434067,
    "domain": {
      "name": "Global",
      "id": "e276abec-e0f2-11e3-8169-**********",
      "type": "Domain"
    }
  },
  "type": "PhysicalInterface",
  "name": "Ethernet2/4",
  "portType": "DATA",
  "adminState": "DISABLED",
  "hardware": {
    "flowControlSend": "OFF",
    "fecMode": "AUTO",
    "autoNegState": true,
    "speed": "DETECT_SFP",
    "duplex": "FULL"
  },
  "LLDP": {
    "transmit": false,
    "receive": false
  },
```

```
  "id": "00505686-9A51-0ed3-0000-**********"
}
```

3. On the interface, evaluate feasibility of break operation using evaluateoperation endpoint.

GET
/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/chassisinterfaces/{interfa

If there are no warnings/errors in the response, user can perform break operation.

Sample Response:

```
  {
        "operationType": "BREAKOUT",
        "readinessState": "READY",
        "links": {
          "self": "https://u32c01p06-
vrouter.cisco.com:22542/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-
6d9ed49b625f/chassis/fmcmanagedchassis/19d967e6-ef81-4f2e-b311-
85ff6cef6d3f/chassisinterfaces/00505686-662F-0ed3-0000-
004294969274/evaluateoperation/00505686-662F-0ed3-0000-004294969274"
        },
        "type": "ChassisInterface",
        "id": "00505686-662F-0ed3-0000-004294969274"
      }
```

If there are errors in the response, user is not allowed to perform break operation:

```
{
  "operationType": "BREAKOUT",
  "interfaceUsages": [
    {
      "conflictType": "Interface usage on instance(s)",
      "severity": "ERROR",              >>>>>>>>>
      "description": "Interface Ethernet2/4 can not be split. Remove it from instances [FTD1] and try a
    }
  ],
  "readinessState": "NOT_READY",        >>>>>>>>>
  "links": {
    "self": "https://u32c01p06-vrouter.cisco.com:22542/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169
  },
  "type": "ChassisInterface",
  "id": "00505686-662F-0ed3-0000-**********"
}
```

4. If the interface is breakout capable, and the readiness state is “READY”, break the interface using breakoutinterfaces endpoint.

POST
/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/operational/breakoutinte

Request:

```
{
  "targetInterfaces": [
    {
      "id": "***************ed3-0000-004294969276",
      "metadata": {
        "type": "PhysicalInterface"
      }
    }
  ],
  "type": "BreakoutInterface"
}
```


Response:


```
{
  "id": "4294969716",
  "type": "TaskStatus",
  "links": {
    "self": "https://u32c01p06-vrouter.cisco.com:22542/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169
  },
  "taskType": "DEVICE_DEPLOYMENT",
  "message": "Deployment status for ************************************: SUCCEEDED",
  "status": "Interface notification received"
}
```


5. Track the task completion using the task id in break response. Set Task status to “Interface Notification received.”

GET /api/fmc_config/v1/domain/{domainUUID}/job/taskstatuses/{objectId}


```
{
  "metadata": {
    "task": {
      "id": "4294969699",
      "links": {
        "self": "https://u32c01p06-vrouter.cisco.com:22542/api/fmc_config/v1/domain/e276abec-e0f2-11e3-
      }
    }
  },
  "targetInterfaces": [
    {
      "id": "00505686-662F-0ed3-0000-**********",
      "type": "PhysicalInterface"
    }
  ],
  "type": "BreakoutInterface"
}

{
  "id": "4294969716",
  "type": "TaskStatus",
  "links": {
    "self": "https://u32c01p06-vrouter.cisco.com:22542/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169
```

```
  },
  "taskType": "DEVICE_DEPLOYMENT",
  "message": "Deployment status for ************************************: SUCCEEDED",
  "status": "Interface notification received"
}
```


6. Get the interfaces changes using chassisinterfaceevents endpoint.

GET /api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/chassisinterfaceevents

Sample Response:


```
[
  {
    "change": "Interface is deleted",
    "type": "PhysicalInterface",
    "state": "DISASSOCIATED",
    "name": "Ethernet2/3"
  },
  {
    "change": "Interface is associated",
    "type": "PhysicalInterface",
    "state": "ASSOCIATED",
    "name": "Ethernet2/3/2"
  },
  {
    "change": "Interface is associated",
    "type": "PhysicalInterface",
    "state": "ASSOCIATED",
    "name": "Ethernet2/3/3"
  },
  {
    "change": "Interface is associated",
    "type": "PhysicalInterface",
    "state": "ASSOCIATED",
    "name": "Ethernet2/3/4"
  }
]
```


7. If interface notification is not received, do sync device using chassisinterfaceevents endpoint and check that there are pending changes.

POST /api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{containerUUID}/chassisinterfaceevents

Request:


```
{
  "action": "SYNC_WITH_DEVICE"
}
```

Response:


```
{
  "action": "SYNC_WITH_DEVICE",
  "hasPendingChanges": true
}
```


8. Once the notification is received, accept the changes using chassisinterfaceevents endpoint.

POST /api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{containerUUID}/chassisinterfaceevents

Request:


```
{
   "action":"ACCEPT_CHANGES"
}
```


9. Get all the chassis interfaces and find the split(broken) interfaces using interfaces endpoint.

GET /api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/interfaces

One 40G interface, say eth2/2, is split into 4x10G interfaces – eth2/2/1, eth2/2/2, eth2/2/3 and eth2/2/4

REST Flow for Interface Join

1. Check if the interface is broken using interfaces/physicalinterfaces endpoint.

Join operation is possible only if “isSplitInterface” is true and mediaType is SFP

GET /api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/interfaces


```
{
  "metadata": {
    "supportedSpeed": "TEN_GBPS,DETECT_SFP",
    "mediaType": "sfp",
    "sfpType": "none",
    "isBreakoutCapable": false,
    "breakoutFactor": "4",
    "isSplitInterface": true,
    "timestamp": 1692541554935,
    "domain": {
      "name": "Global",
      "id": "e276abec-e0f2-11e3-8169-**********",
      "type": "Domain"
    }
  },
  "type": "PhysicalInterface",
  "name": "Ethernet2/3/4",
  "portType": "DATA",
```

```
  "adminState": "DISABLED",
  "LLDP": {
    "transmit": false,
    "receive": false
  },
  "hardware": {
    "flowControlSend": "OFF",
    "speed": "DETECT_SFP",
    "duplex": "FULL",
    "fecMode": "AUTO",
    "autoNegState": true
  },
  "id": "00505686-662F-0ed3-0001-**********"
}
```


2. Evaluate feasibility of Join operation using evaluateoperation endpoint on one of the four split interfaces.

GET /api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/chassisinterfaces/{interfaceUUID}/evaluateoperation

- If there are no warnings/errors in the response, user can perform Join operation.


```
{
  "operationType": "JOIN",
  "readinessState": "READY",
  "links": {
    "self": "https://u32c01p06-vrouter.cisco.com:22542/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169
  },
  "type": "ChassisInterface",
  "id": "00505686-662F-0ed*******************"
}
```


- If there are errors in the response, the user is not allowed to perform join operation.


```
{
  "operationType": "JOIN",
  "interfaceUsages": [
    {
      "conflictType": "Interface used in EtherChannel Configuration",
      "severity": "ERROR",
      "description": "Interface (Ethernet2/3/4) referred to in Ether Channel Interface (Port-channel32)
    }
  ],
  "readinessState": "NOT_READY",
  "links": {
    "self": "https://u32c01p06-vrouter.cisco.com:22542/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169
  },
  "type": "ChassisInterface",
  "id": "00505686-662F-0ed*******************"
}
```


3. If the interface is broken, and the readiness state is “READY”, join the interface using joininterfaces

endpoint. Interface_uuid can be id of any of 4 broken interfaces.

POST/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/operational/joinint

Request:


```
{
  "targetInterfaces": [
    {
      "id": "***************ed3-0001-692539698200",
      "type": "PhysicalInterface"
    }
  ],
  "type": "JoinInterface"
}
```


Response:


```
{
  "metadata": {
    "task": {
      "id": "4294970217",
      "links": {
        "self": "<FMC_IP>/api/fmc_config/v1/domain/e27"***************-8169-6d9ed49b625f/job/taskstatus
      }
    }
  },
  "targetInterfaces": [
    {
      "id": "***************ed3-0001-692539698200",
      "type": "PhysicalInterface"
    },
    {
      "id": "***************ed3-0001-692539698201",
      "type": "PhysicalInterface"
    },
    {
      "id": "***************ed3-0001-692539698202",
      "type": "PhysicalInterface"
    },
    {
      "id": "***************ed3-0001-692539698203",
      "type": "PhysicalInterface"
    }
  ],
  "type": "JoinInterface"
}
```


4. Track the task completion using the task id in join response. Set Task status to “Interface Notification received.”

GET /api/fmc_config/v1/domain/{domainUUID}/job/taskstatuses/{objectId}

Response:


```
{
  "id": "4294970237",
  "type": "TaskStatus",
  "links": {
    "self": "https://u32c01p06-vrouter.cisco.com:22542/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169
  },
  "taskType": "SSP_EPM_OIR",
  "message": "Deployment status for 19d967e6-xxxx-xxxx-xxxx-85ff6cef6d3f: SUCCEEDED",
  "status": "Interface notification received"
}
```


5. Get the interfaces changes using chassisinterfaceevents endpoint.

GET
/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{containerUUID}/chassisinterfaceevents

Response:


```
[
  {
    "change": "Interface is associated",
    "type": "PhysicalInterface",
    "state": "ASSOCIATED",
    "name": "Ethernet2/3"
  },
  {
    "change": "Interface is deleted",
    "type": "PhysicalInterface",
    "state": "DISASSOCIATED",
    "name": "Ethernet2/3/1"
  },
  {
    "change": "Interface is deleted",
    "type": "PhysicalInterface",
    "state": "DISASSOCIATED",
    "name": "Ethernet2/3/2"
  },
  {
    "change": "Interface is deleted",
    "type": "PhysicalInterface",
    "state": "DISASSOCIATED",
    "name": "Ethernet2/3/3"
  },
  {
    "change": "Interface is deleted",
    "type": "PhysicalInterface",
    "state": "DISASSOCIATED",
    "name": "Ethernet2/3/4"
  }
]
```

6. If interface notification is not received, do sync device using chassisinterfaceevents endpoint and check that there are pending changes.

POST /api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{containerUUID}/chassisinterfaceevents

Request:

```
{
   "action":"SYNC_WITH_DEVICE"
}
```

Response:

```
{
   "action":"SYNC_WITH_DEVICE",
   "hasPendingChanges":true
}
```

7. Once the notification is received, accept the changes using chassisinterfaceevents endpoint.

POST /api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{containerUUID}/chassisinterfaceevents

Request:

```
{
   "action":"ACCEPT_CHANGES"
}
```

8. Get all the chassis interfaces and find the joined interfaces as well as the other interfaces using interfaces endpoint.

GET /api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/interfaces

Say Join was initiated on 10G interface say eth2/2/1, then a 40G interface eth2/2 is available in the response.

Sync Device REST APIs

To support the Sync of Network Module as well as Interfaces, these URLs have been introduced.

POST:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/chassisinterfaceevents

With Payload

{"action": "SYNC_WITH_DEVICE”} - > Triggers the Sync

{"action": "ACCEPT_CHANGES”} - > Accept the Changes

GET:

/api/fmc_config/v1/domain/{domainUUID}/chassis/fmcmanagedchassis/{containerUUID}/chassisinterface events

Lists the generated changed events

Troubleshooting / Diagnostics

FXOS Logging

If registration fails, these FXOS CLIs can be used to check if sftunnel, sfipproxy processes are up.

```
firepower# connect local-mgmt
firepower-4215(local-mgmt)# show processes | include sftunnel grep: (standard input): binary file matche
3323 root 20 0 80328 2024 1544 S 0.0 0.0 0:11.53 /opt/cisco/sftunnel/sfipproxy -d –f /etc/sf/sfipproxy.
22066 root 20 0 376880 7140 5944 S 0.0 0.0 0:41.18 /opt/cisco/sftunnel/sftunnel -d -f /etc/sf/sftunnel.
```

If using the terminal console for the CLI, ensure the output of show processes is not truncated by setting the terminal width to an appropriate value using this CLI shown:

```
firepower-4215(local-mgmt)# terminal width 100
```

If the SFTunnel process is up and running, yet registration is failing, these commands can be used to find any potential reason for failure.

Introduced new CLI in FXOS from connect local-mgmt to view syslog messages in /opt/cisco/platform/logs/sfmessages

```
firepower# connect local-mgmt
firepower(local-mgmt)# tail-mgmt-log sfmessages
<snip>
Dec 9 18:31:17 firepower Ipc [30483]: add ep: 1,0x5613aa0e2fe8 total = 1
Dec 9 18:31:17 firepower Ipc [30483]: add ep: 1,0x5613aa0ec528 total = 2
Dec 9 18:31:17 firepower Ipc [30483]: add ep: 1,0x5613aa0f5ea8 total = 3
Dec 9 18:31:18 firepower SF-IMS[12621]: [12625] sftunneld:SYNC_PROC [INFO] Change in directory /var/sf/
```

FMC Logging

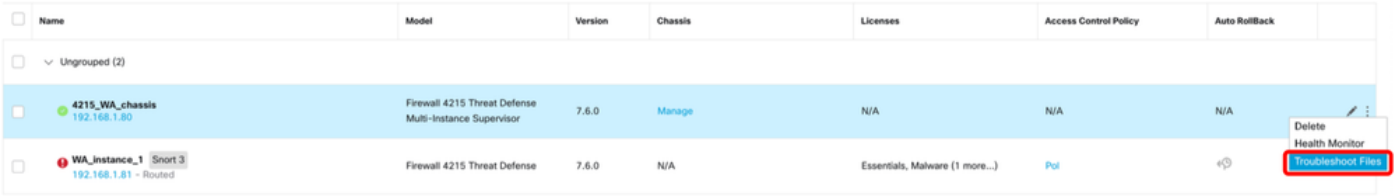
- If device registration fails, find usmsharedsvcs.log and vmssharedsvcs.log at this location and look for the string "CHASSIS DISCOVERY” or “NATIVE_TO_MULTI_INSTANCE" to find the potential cause of failure.

    - Also, look in /var/log/action_queue.log and /var/sf/messages for SFTunnel issues.
    - /var/opt/CSCOpx/MDC/log/operation/usmsharedsvcs.log
      /var/opt/CSCOpx/MDC/log/operation/vmssharedsvcs.log
- If chassis auto-registration fails, find *usmsharedsvcs.log* and *vmssharedsvcs.log* and look for the string ”CHASSIS DISCOVERY" and “NATIVE_TO_MULTI_INSTANCE” to find the potential cause of failure.
- If instance auto-registration fails, find *usmsharedsvcs.log* and *vmssharedsvcs.log* and look for the string "MI_FTD_INSTANCE_AUTO_REGISTRATION" to find the potential cause of failure.
- If there is a deployment failure on the device, navigate to **Deploy -> Deployment History -> Click on the failed deployment -> Open Transcript**. This file contains the reason for failure.

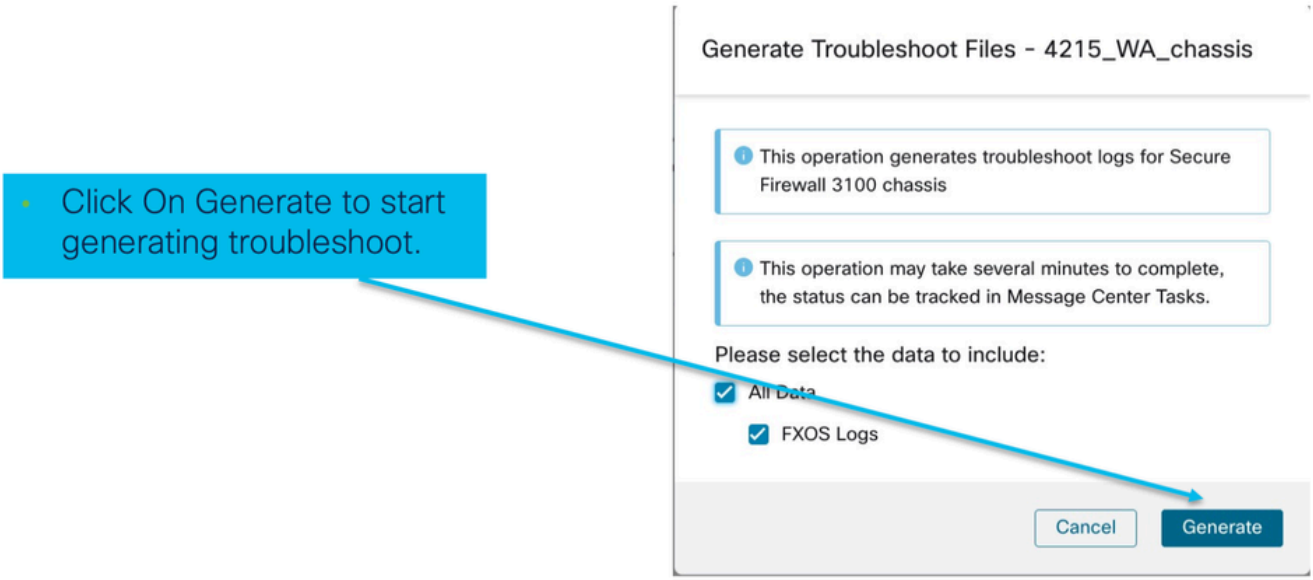
Chassis Troubleshoot

FMC supports generation of chassis troubleshoot (FPRM) from the device management page.

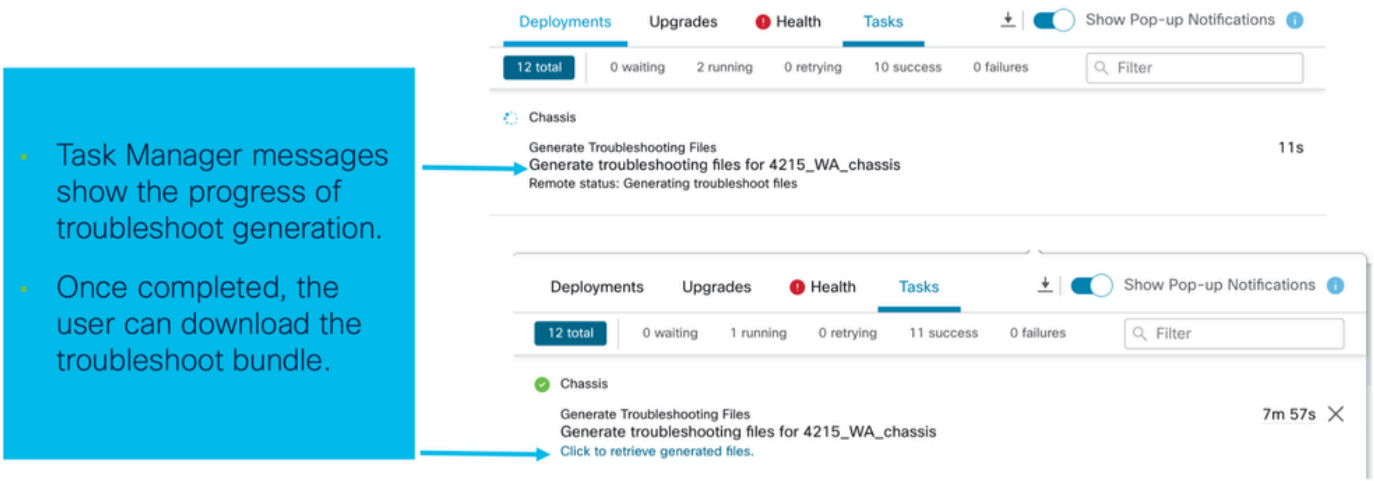
- Like FTD device, there is a troubleshoot option available for chassis device which generates chassis troubleshoot and allows user to download the troubleshoot bundle from FMC.
- This collects the "show tech-support form" bundle from the chassis:



Chassis troubleshooting options and generating:



Chassis troubleshooting progress and download:



Sample Problems with Troubleshooting Walkthroughs

Auto-Registration of Chassis Failure in FMC

Problem: Auto Registration of Chassis is failing in FMC.

Expected Result:

- Once Conversion starts from FMC, It is expected to be unregistered and auto-registered in FMC.

Actual Result:

- Chassis auto-registration failed

Troubleshooting the Problem

1. Check conversion:

- Ensure the conversion has been triggered on FMC.
- Log into the device and check if the device has been converted to container mode.
- Run the commands to see if the device has been converted:

```
firepower# scope sys
firepower /system # show
Systems:
Name Mode Deploy Mode System IP Address System IPv6 Address
---------- ----------- ----------- ----------------- -------------------
firepower Stand Alone Container 192.168.xx.xx ::
```

2. Check device manager:

- Check if the device manager has been set properly:

```
firepower# show device-manager
    Device manager:
```

```
Name: manager
Hostname: 10.10.xx.xx
NAT id: 3ab4bb1a-d723-11ee-a694-89055xxxxxxx
Registration Status: Completed
Error Msg:
```

3. Logs to check:

3.1. Navigate to /var/opt/CSCOpx/MDC/log/operation/vmssharedsvcs.log and /var/opt/CSCOpx/MDC/log/operation/usmsharedsvcs.log

3.2. Search for the keywords ”NATIVE_TO_MI_CONVERSION" and ”CHASSIS DISCOVERY” in the files to find the reason for failure.

Auto-Registration of Instance in FMC

Problem: Auto Registration of Instance is failing in FMC.

Expected Result:

- Once Instance is provisioned from FMC, It is expected to be auto registered in FMC

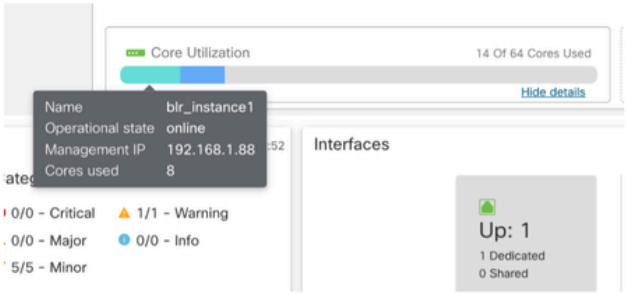
Actual Result:

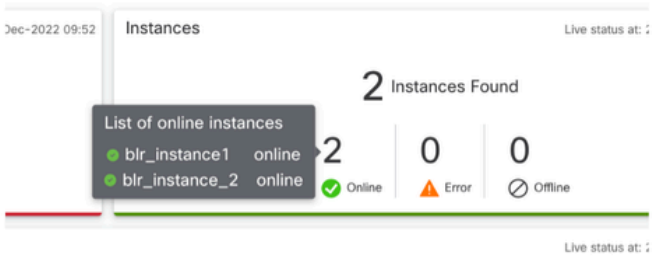
- Instance auto registration failed

Troubleshooting the Problem

- Ensure deployment was triggered after instance creation.
  - If deploy is not done, ensure to deploy the changes to the device.
  - If there is a failure in deployment, proceed to **Deployment History -> Click on Transcript**. Check the reason for failure, fix and retry the deployment.

- Ensure the instance is installed, and its operational state is online. You can use the summary page of chassis to check the status of Instance provisioning.





- Check SFTunnel is up and running on the Instance FTD using this command:

```
ps -ef | grep -i "sftunnel”
```

- If SFTunnel is not running, try to execute a restart command:

```
pmtool restartById sftunnel
```

- Navigate to /var/opt/CSCOpx/MDC/log/operation/vmssharedsvcs.log and /var/opt/CSCOpx/MDC/log/operation/usmsharedsvcs.log
- Search for the keyword "MI_FTD_INSTANCE_AUTO_REGISTRATION" in the file to find the reason for failure.

Native Device Registration in FMC

Problem: Native Device Registration is failing in FMC after converting the device back to native mode

- In case the user converts the Chassis(MI Mode) back to native mode but forgets to delete the Chassis from the FMC, the device goes offline on the FMC.
- If the user tries to re-register this native device back to the FMC, the registration fails.

Troubleshooting the Problem

- Make sure the Chassis Entry has been deleted from the FMC before converting the device back to native mode.
- Once the entry is deleted, try re-registering the native device to FMC.

Useful References

- Information about shared interfaces:

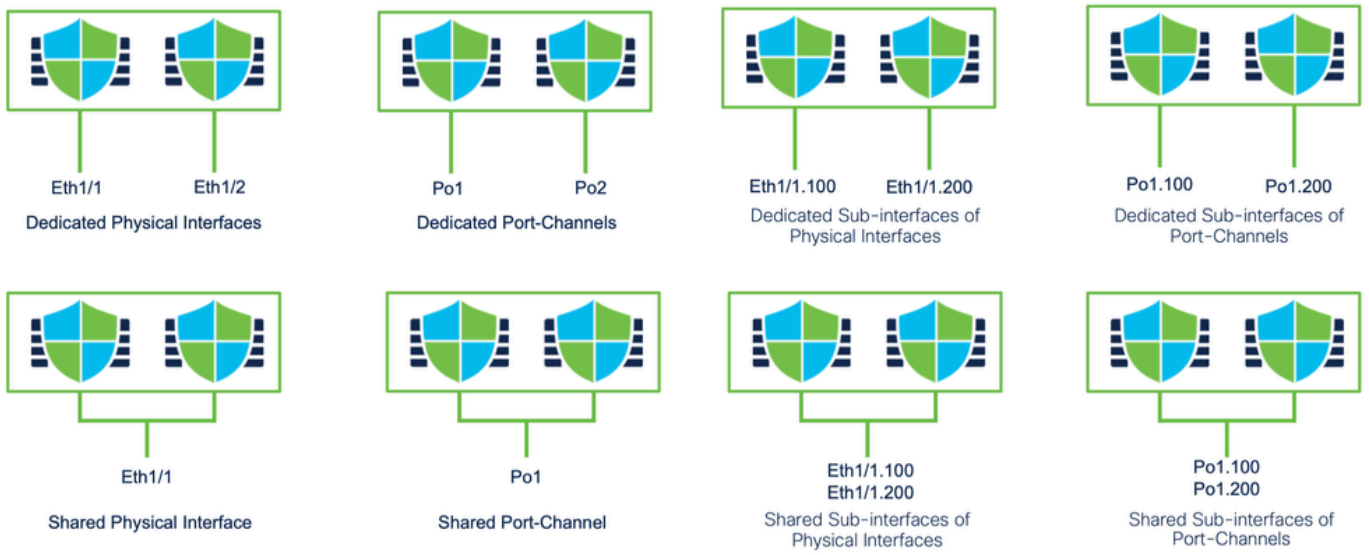
https://www.cisco.com/c/en/us/td/docs/security/secure-firewall/threat-defense/use-case/multi-instance-sec-fw/multi-instance-sec-fw.html#shared-interface-scalability-WGUIEF

- 3100 Multi-Instance page on the Cisco support site:

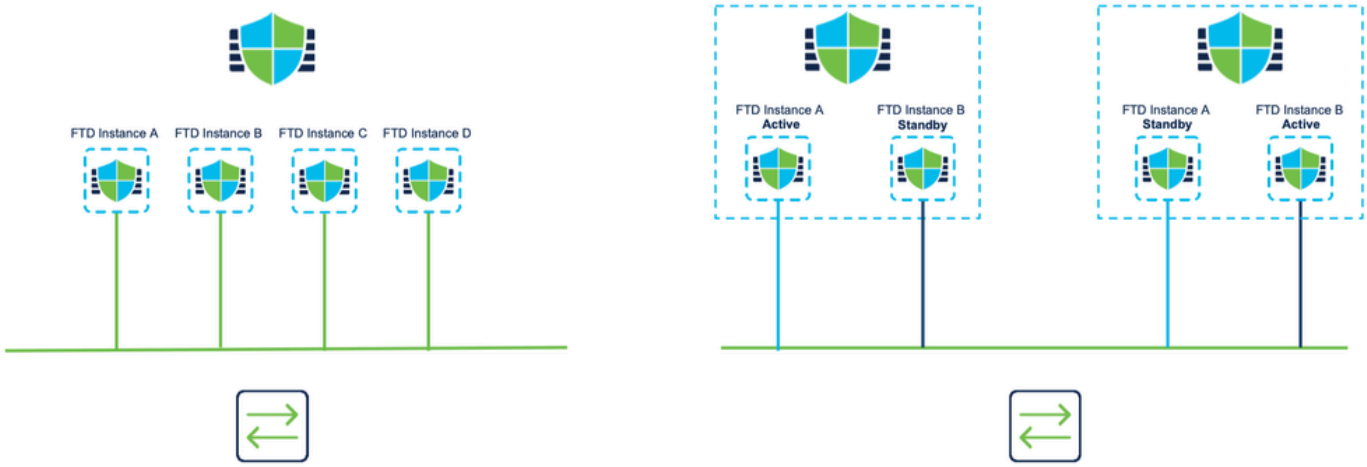
https://www.cisco.com/c/en/us/td/docs/security/secure-firewall/threat-defense/use-case/multi-instance-sec-fw/multi-instance-sec-fw.html

Interface Options and High Availability

Interface Options



Standalone or High-Availability



Leveraging the Dual Management Interfaces

- Like the 4200 in native mode, the two physical management ports are provided to support interface redundancy for management traffic, or to support separate interfaces for management and eventing.
  - The 9300 and 4100 devices, as well as the 4200 Series, have dual management interfaces. The second management interface, Management 1/2, is intended for you to use for events.
- In multi-instance (aka “container”) mode, you can configure this interface at the Threat Defense CLI in each instance. Assign an IP address on the same network for each instance.
- When in container mode, each FTD instance has both Management 1/1 and Management 1/2 interfaces automatically assigned to it.
  - The second management interface is disabled by default.
  - You cannot configure Management1/2 using FMC; you have to configure it through the FTD CLISH (on the 9300/4100, which. by contrast, is done in the FXOS CLI). Use this command with the desired IP address type, address, subnet, and static route:

```
configure network ipv4 manual 192.168.0.xx 255.255.255.0 192.168.0.1 management1
```