Configure FMC with Ansible to Create FTD High Availability

Contents

Introduction

This document describes the steps to automate Firepower Management Center (FMC) to create Firepower Threat Defense (FTD) High Availability with Ansible.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Ansible
- Ubuntu Server
- Cisco Firepower Management Center (FMC) Virtual
- Cisco Firepower Threat Defense (FTD) Virtual

In the context of this laboratory situation, Ansible is deployed on Ubuntu.

It is essential to ensure that Ansible is successfully installed on any platform supported by Ansible for running the Ansible commands referenced in this article.

Components Used

The information in this document is based on these software and hardware versions:

- Ubuntu Server 22.04
- Ansible 2.10.8
- Python 3.10
- Cisco Firepower Threat Defense Virtual 7.4.1
- Cisco Firepower Management Center Virtual 7.4.1

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.

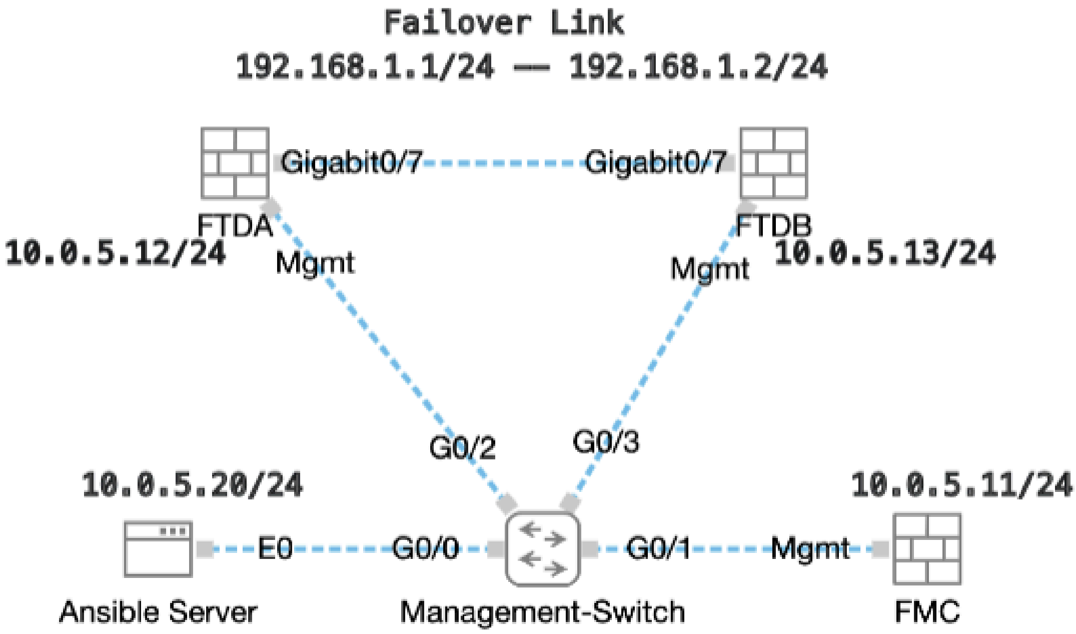
Background Information

Ansible is a highly versatile tool, demonstrating significant efficacy in managing network devices. Numerous methodologies can be employed to run automated tasks with Ansible. The method employed in this article serves as a reference for test purposes.

In this example, the FTD High Avaliability and standby IP address of it are created after runing the playbook example successfully.

Configure

Network Diagram



Topology

Configurations

Because Cisco does not support example scripts or customer-written scripts, we have some examples you can test depending on your needs.

It is essential to ensure that preliminary verificationhas been duly completed.

- Ansible server possesses internet connectivity.
- Ansible server is capable of successfully communicating with the FMC GUI Port (the default port for FMC GUI is 443).
- Two FTD devices are registered successfully to FMC.
- Primary FTD are configured with interface ip address.

Step 1. Connect to the CLI of the Ansible server via SSH or console.

Step 2. Run command ansible-galaxy collection install cisco.fmcansible in order to install Ansible collection of FMC on your Ansible server.

<#root>

```
cisco@inserthostname-here:~$

ansible-galaxy collection install cisco.fmcansible
```

Step 3. Run command mkdir /home/cisco/fmc_ansible in order to create a new folder to store the related files. In this example, the home directory is /home/cisco/, the new folder name is fmc_ansible.

<#root>

```
cisco@inserthostname-here:~$

mkdir /home/cisco/fmc_ansible
```

Step 4. Navigate to the folder /home/cisco/fmc_ansible, create inventory file. In this example, the inventory file name is inventory.ini.

<#root>

```
cisco@inserthostname-here:~$

 cd /home/cisco/fmc_ansible/


ccisco@inserthostname-here:~/fmc_ansible$

ls


inventory.ini
```

You can duplicate this content and paste it for utilization, altering the **bold** sections with the accurate parameters.

<#root>

```
[fmc]

10.0.5.11



[fmc:vars]
ansible_user=

cisco
```

```
ansible_password=

cisco


ansible_httpapi_port=443
ansible_httpapi_use_ssl=True
ansible_httpapi_validate_certs=False
network_type=HOST
ansible_network_os=cisco.fmcansible.fmc
```

Step 5. Navigate to the folder /home/cisco/fmc_ansible, create variable file for creating FTD HA. In this example, the variable file name is fmc-create-ftd-ha-vars.yml.

<#root>

```
cisco@inserthostname-here:~$

 cd /home/cisco/fmc_ansible/


ccisco@inserthostname-here:~/fmc_ansible$

ls


fmc-create-ftd-ha-vars.yml

inventory.ini
```

You can duplicate this content and paste it for utilization, altering the **bold** sections with the accurate parameters.

<#root>

```
user:
  domain: 'Global'
device_name:
  ftd1: '

FTDA

'
  ftd2: '

FTDB

'
ftd_ha:
  name: '

FTD_HA

'
  active_ip: '

192.168.1.1
```

```
'
  standby_ip: '
192.168.1.2
'
  key:
cisco
  mask24: '
255.255.255.0
'
```

Step 6. Navigate to the folder /home/cisco/fmc_ansible, create playbook file for creating FTD HA. In this example, the playbook file name is fmc-create-ftd-ha-playbook.yaml.

<#root>

```
cisco@inserthostname-here:~$
 cd /home/cisco/fmc_ansible/

ccisco@inserthostname-here:~/fmc_ansible$
ls


fmc-create-ftd-ha-playbook.yaml
fmc-create-ftd-ha-vars.yml inventory.ini
```

You can duplicate this content and paste it for utilization, altering the **bold** sections with the accurate parameters.

<#root>

```
---

- name: FMC Create FTD HA
  hosts: fmc
  connection: httpapi

  tasks:

    - name: Task01 - Get User Domain
      cisco.fmcansible.fmc_configuration:
        operation: getAllDomain
        filters:
          name: "{{
user.domain
 }}"
        register_as: domain
    - name: Task02 - Get FTD1
      cisco.fmcansible.fmc_configuration:
        operation: getAllDevice
        path_params:
          domainUUID: '{{ domain[0].uuid }}'
        filters:
          name: "{{

device_name.ftd1

 }}"
        register_as: ftd1_list

    - name: Task03 - Get FTD2
      cisco.fmcansible.fmc_configuration:
        operation: getAllDevice
        path_params:
          domainUUID: '{{ domain[0].uuid }}'
        filters:
          name: "{{

device_name.ftd2

 }}"
        register_as: ftd2_list

    - name: Task04 - Get Physical Interfaces
      cisco.fmcansible.fmc_configuration:
        operation: getAllFTDPhysicalInterface
        path_params:
          containerUUID: '{{ ftd1_list[0].id }}'
          domainUUID: '{{ domain[0].uuid }}'
        register_as: primary_physical_interfaces

    - name: Task05 - Configure FTD HA
      cisco.fmcansible.fmc_configuration:
        operation: "createFTDHADeviceContainer"
        data:
            primary: {'id': '{{ ftd1_list[0].id }}'}
            secondary: {'id': '{{ ftd2_list[0].id }}'}
            name: "{{

ftd_ha.name

 }}"
            type: "DeviceHAPair"
            ftdHABootstrap: {
              'isEncryptionEnabled': false,
              'encKeyGenerationScheme': 'CUSTOM',
              'sharedKey': "{{

ftd_ha.key

 }}",
              'useSameLinkForFailovers': true,
              'lanFailover': {
                'useIPv6Address': false,
                'subnetMask': "{{

ftd_ha.mask24

 }}",
                'interfaceObject': {
                  'id': '{{ primary_physical_interfaces[7].id }}',
                  'type': 'PhysicalInterface',
                  'name': 'GigabitEthernet0/7'
                  },
                'standbyIP': "{{

ftd_ha.standby_ip

 }}",
                'logicalName': 'LAN-INTERFACE',
                'activeIP': "{{

ftd_ha.active_ip

 }}"
              },
              'statefulFailover': {
                'useIPv6Address': false,
                'subnetMask': "{{

ftd_ha.mask24

 }}",
                'interfaceObject': {
                  'id': '{{ primary_physical_interfaces[7].id }}',
                  'type': 'PhysicalInterface',
                  'name': 'GigabitEthernet0/7'
                  },
                'standbyIP': "{{

ftd_ha.standby_ip

 }}",
                'logicalName': 'STATEFUL-INTERFACE',
                'activeIP': "{{

ftd_ha.active_ip

 }}"
              }
            }
        path_params:
            domainUUID: "{{ domain[0].uuid }}"

    - name: Task06 - Wait for FTD HA Ready
      ansible.builtin.wait_for:
        timeout: 360
      delegate_to: localhost

    - name: Task07 - Get FTD HA object
      cisco.fmcansible.fmc_configuration:
        operation: "getAllFTDHADeviceContainer"
        path_params:
          domainUUID: "{{ domain[0].uuid }}"
        query_params:
          expanded: true
        register_as: ftd_ha_container
      delay: 15

    - name: Task08 - Confirm Standby Ready Status
      cisco.fmcansible.fmc_configuration:
        operation: "getFTDHADeviceContainer"
        path_params:
          objectId: "{{ ftd_ha_container[0].id }}"
          domainUUID: "{{ domain[0].uuid }}"
        register_as: ha_status
      until: ha_status.metadata.secondaryStatus.currentStatus is match("Standby")
      retries: 1000
      delay: 2
```

Note: The names in bold in this example playbook serve as variables. The corresponding values for these variables are preserved within the variable file.

Step 7. Navigate to the folder **/home/cisco/fmc_ansible**, run command `ansible-playbook -i <inventory_name>.ini <playbook_name>.yaml -e@"<playbook_vars>.yml"`in order to play the ansible task.

In this example, the command is `ansible-playbook -i inventory.ini fmc-create-ftd-ha-playbook.yaml -e@"fmc-create-ftd-ha-vars.yml"` .

<#root>

```
cisco@inserthostname-here:~$

cd /home/cisco/fmc_ansible/
```

```
ccisco@inserthostname-here:~/fmc_ansible$

ls

fmc-create-ftd-ha-playbook.yaml fmc-create-ftd-ha-vars.yml inventory.ini

cisco@inserthostname-here:~/fmc_ansible$

ansible-playbook -i inventory.ini fmc-create-ftd-ha-playbook.yaml -e@"fmc-create-ftd-ha-vars.yml"


PLAY [FMC Create FTD HA] ******************************************************************************

TASK [Gathering Facts] ********************************************************************************
ok: [10.0.5.11]

TASK [Task01 - Get User Domain] ***********************************************************************
ok: [10.0.5.11]

TASK [Task02 - Get FTD1] ******************************************************************************
ok: [10.0.5.11]

TASK [Task03 - Get FTD2] ******************************************************************************
ok: [10.0.5.11]

TASK [Task04 - Get Physical Interfaces] ***************************************************************
ok: [10.0.5.11]

TASK [Task05 - Configure FTD HA] **********************************************************************
changed: [10.0.5.11]

TASK [Task06 - Wait for FTD HA Ready] *****************************************************************
ok: [10.0.5.11]

TASK [Task07 - Get FTD HA object] *********************************************************************
ok: [10.0.5.11]

TASK [Task08 - Confirm Standby Ready Status] **********************************************************
ok: [10.0.5.11]

PLAY RECAP ********************************************************************************************
10.0.5.11                  : ok=9    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
```

Step 8. Navigate to the folder /home/cisco/fmc_ansible, create variable file for updating FTD HA standby ip address. In this example, the variable file name is fmc-create-ftd-ha-standby-ip-vars.yml.

<#root>

```
cisco@inserthostname-here:~$

 cd /home/cisco/fmc_ansible/


ccisco@inserthostname-here:~/fmc_ansible$

ls


fmc-create-ftd-ha-playbook.yaml

fmc-create-ftd-ha-standby-ip-vars.yml
```

```
fmc-create-ftd-ha-vars.yml inventory.ini
```

You can duplicate this content and paste it for utilization, altering the **bold** sections with the accurate parameters.

<#root>

```
user:
  domain: 'Global'

ftd_data:
  outside_name: '

Outside

'
  inside_name: '

Inside

'
  outside_ip: '10.1.1.1'
  inside_ip: '10.1.2.1'
  mask24: '255.255.255.0'

ftd_ha:
  name: '

FTD_HA

'
  outside_standby: '

10.1.1.2

'
  inside_standby: '

10.1.2.2

'
```

Step 9. Navigate to the folder **/home/cisco/fmc_ansible**, create playbook file for updating FTD HA standby ip address. In this example, the playbook file name is fmc-create-ftd-ha-standby-ip-playbook.yaml.

<#root>

```
cisco@inserthostname-here:~$

 cd /home/cisco/fmc_ansible/


ccisco@inserthostname-here:~/fmc_ansible$

ls


fmc-create-ftd-ha-playbook.yaml
```

fmc-create-ftd-ha-standby-ip-playbook.yaml

```
 fmc-create-ftd-ha-standby-ip-vars.yml fmc-create-ftd-ha-vars.yml inventory.ini
```

You can duplicate this content and paste it for utilization, altering the **bold** sections with the accurate parameters.

<#root>

```
---

- name: FMC Update FTD HA Interface Standby IP
  hosts: fmc
  connection: httpapi

  tasks:

    - name: Task01 - Get User Domain
      cisco.fmcansible.fmc_configuration:
        operation: getAllDomain
        filters:
          name: "{{
```

user.domain

```
 }}"
        register_as: domain

    - name: Task02 - Get FTD HA Object
      cisco.fmcansible.fmc_configuration:
        operation: "getAllFTDHADeviceContainer"
        path_params:
          domainUUID: "{{ domain[0].uuid }}"
        query_params:
          expanded: true
        register_as: ftd_ha_container

    - name: Task03 - Get Outside Interface
      cisco.fmcansible.fmc_configuration:
        operation: "getAllFTDHAMonitoredInterfaces"
        path_params:
          containerUUID: "{{ ftd_ha_container[0].id }}"
          domainUUID: "{{ domain[0].uuid }}"
        filters:
            name: "{{
```

ftd_data.outside_name

```
 }}"
        register_as: outside_interface

    - name: Task04 - Get Inside Interface
      cisco.fmcansible.fmc_configuration:
        operation: "getAllFTDHAMonitoredInterfaces"
        path_params:
          containerUUID: "{{ ftd_ha_container[0].id }}"
          domainUUID: "{{ domain[0].uuid }}"
        filters:
            name: "{{
```

ftd_data.inside_name

```
 }}"
    register_as: inside_interface

  - name: Task05 - Configure Standby IP-Outside
   cisco.fmcansible.fmc_configuration:
    operation: "updateFTDHAMonitoredInterfaces"
    data:
     id: "{{ outside_interface[0].id }}"
     name: "{{ outside_interface[0].name }}"
     ipv4Configuration: {'standbyIPv4Address': "{{
```

ftd_ha.outside_standby

```
 }}"}
     monitorForFailures: true
    path_params:
     objectId: "{{ outside_interface[0].id }}"
     containerUUID: "{{ ftd_ha_container[0].id }}"
     domainUUID: "{{ domain[0].uuid }}"

  - name: Task06 - Config Standby IP-Inside
   cisco.fmcansible.fmc_configuration:
    operation: "updateFTDHAMonitoredInterfaces"
    data:
     id: "{{ inside_interface[0].id }}"
     name: "{{ inside_interface[0].name }}"
     ipv4Configuration: {'standbyIPv4Address': "{{
```

ftd_ha.inside_standby

```
 }}"}
     monitorForFailures: true
    path_params:
     objectId: "{{ inside_interface[0].id }}"
     containerUUID: "{{ ftd_ha_container[0].id }}"
     domainUUID: "{{ domain[0].uuid }}"

  - name: Task07 - Get Deployable Devices
   cisco.fmcansible.fmc_configuration:
    operation: getDeployableDevice
    path_params:
     domainUUID: '{{ domain[0].uuid }}'
    query_params:
     expanded: true
    register_as: deploy_devices

  - name: Task08 - Start Deployment
   cisco.fmcansible.fmc_configuration:
    operation: createDeploymentRequest
    data:
     version: '{{ deploy_devices[0].version }}'
     deviceList:
      - '{{ deploy_devices[0].device.id }}'
     forceDeploy: True
    path_params:
     domainUUID: '{{ domain[0].uuid }}'
    register_as: deployment_job

  - name: Task09 - Wait for Deployment Complete
   ansible.builtin.wait_for:
    timeout: 240
   delegate_to: localhost
```

```
- name: Task10 - Poll Deployment Status Until Deployment Successful
  cisco.fmcansible.fmc_configuration:
    operation: getDeploymentDetail
    path_params:
      containerUUID: '{{ deploy_devices[0].device.id }}'
      domainUUID: '{{ domain[0].uuid }}'
    register_as: deployment_status
  until: deployment_status[0].status is match("SUCCEEDED")
  retries: 1000
  delay: 3

- name: Task11 - Stop The Playbook If The Deployment Failed
  fail:
    msg: 'Deployment failed. Status: {{ deployment_status[0].status }}'
  when: deployment_status[0].status is not match("SUCCEEDED")
```

Note: The names in bold in this example playbook serve as variables. The corresponding values for these variables are preserved within the variable file.

Step 10. Navigate to the folder **/home/cisco/fmc_ansible**, run command `ansible-playbook -i <inventory_name>.ini <playbook_name>.yaml -e@"<playbook_vars>.yml"`in order to play the ansible task.

In this example, the command is `ansible-playbook -i inventory.ini fmc-create-ftd-ha-standby-ip-playbook.yaml -e@"fmc-create-ftd-ha-standby-ip-vars.yml"` .

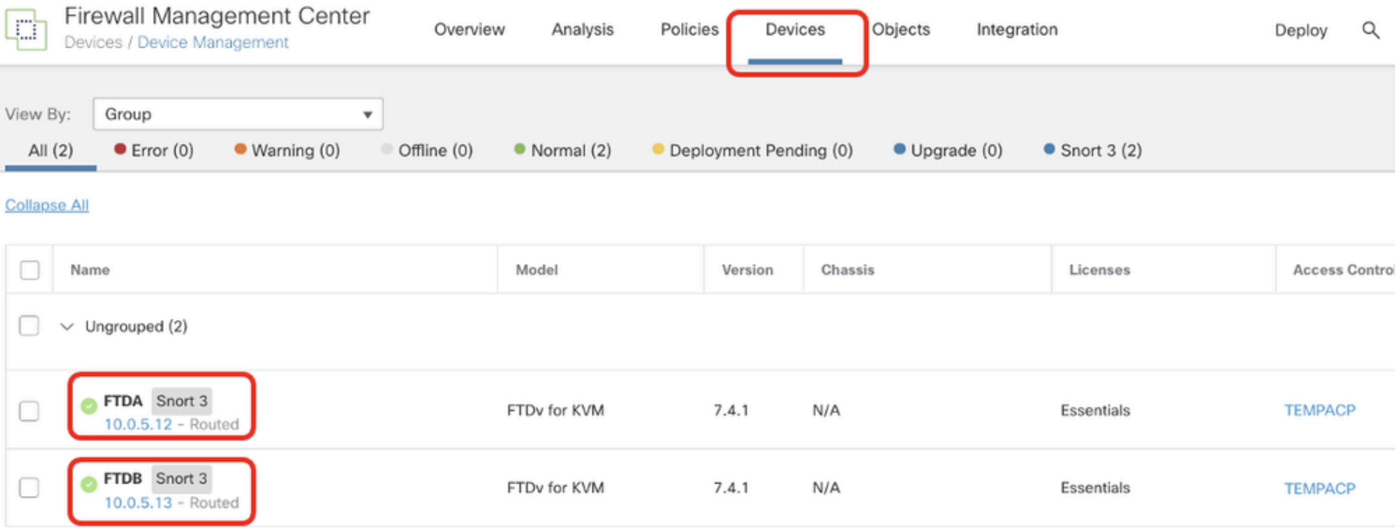
<#root>

```
cisco@inserthostname-here:~$

 cd /home/cisco/fmc_ansible/


ccisco@inserthostname-here:~/fmc_ansible$

ls


fmc-create-ftd-ha-playbook.yaml

fmc-create-ftd-ha-standby-ip-playbook.yaml



fmc-create-ftd-ha-standby-ip-vars.yml

 fmc-create-ftd-ha-vars.yml

inventory.ini



cisco@inserthostname-here:~/fmc_ansible$

ansible-playbook -i inventory.ini fmc-create-ftd-ha-standby-ip-playbook.yaml -e@"fmc-create-ftd-ha-stan

PLAY [FMC Update FTD HA Interface Standby IP] *********************************************************

TASK [Gathering Facts] ********************************************************************************
ok: [10.0.5.11]

TASK [Task01 - Get User Domain] ***********************************************************************
ok: [10.0.5.11]

TASK [Task02 - Get FTD HA Object] *********************************************************************
ok: [10.0.5.11]

TASK [Task03 - Get Outside Interface] *****************************************************************
ok: [10.0.5.11]

TASK [Task04 - Get Inside Interface] ******************************************************************
ok: [10.0.5.11]

TASK [Task05 - Configure Standby IP-Outside] **********************************************************
changed: [10.0.5.11]

TASK [Task06 - Config Standby IP-Inside] **************************************************************
changed: [10.0.5.11]

TASK [Task07 - Get Deployable Devices] ****************************************************************
ok: [10.0.5.11]
```

```
TASK [Task08 - Start Deployment] ************************************************************************
changed: [10.0.5.11]

TASK [Task09 - Wait for Deployment Complete] ************************************************************
ok: [10.0.5.11]

TASK [Task10 - Poll Deployment Status Until Deployment Successful] **************************************
ok: [10.0.5.11]

TASK [Task11 - Stop The Playbook If The Deployment Failed] **********************************************
skipping: [10.0.5.11]

PLAY RECAP **********************************************************************************************
10.0.5.11                  : ok=11   changed=3    unreachable=0    failed=0    skipped=1    rescued=0
```

Verify

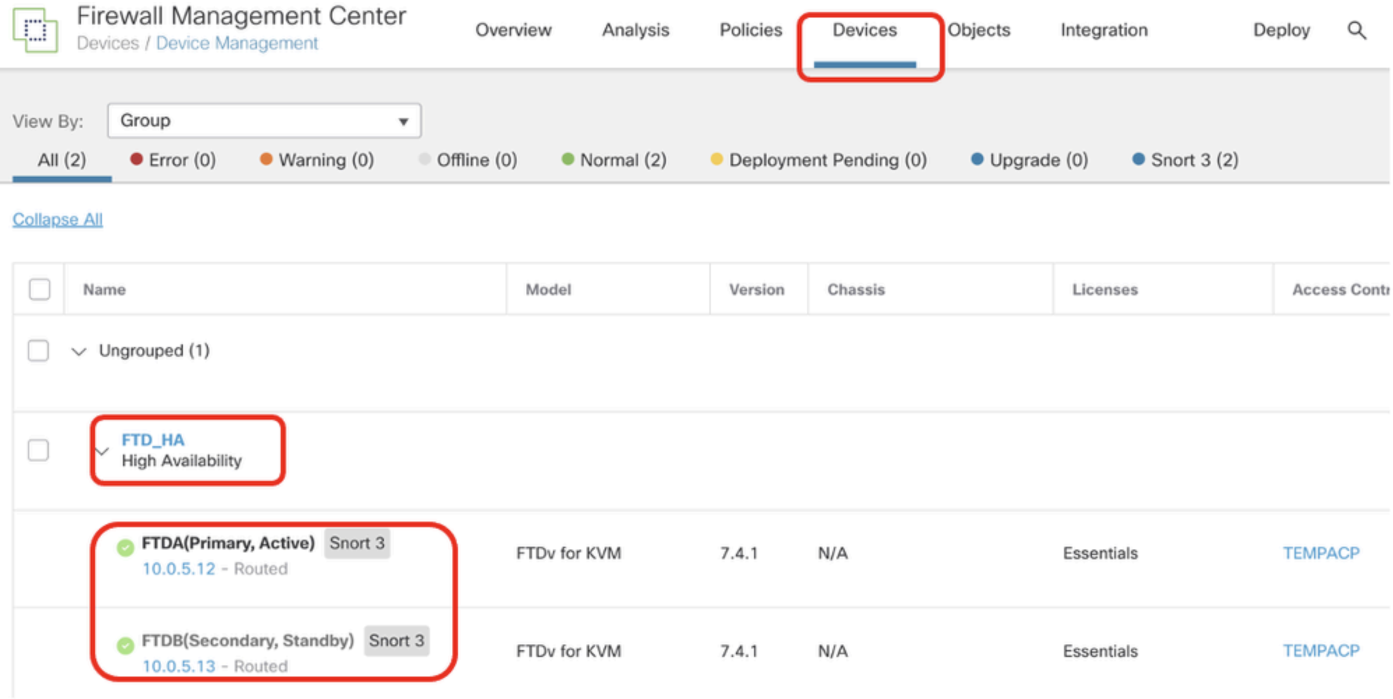
Before running the ansible task, log in FMC GUI. Navigate to **Devices** > **Device Management,** two FTD registered successfully on FMC with configured access control policy.



□	Name	Model	Version	Chassis	Licenses	Access Contro
□	Ungrouped (2)					
□	FTDA Snort 3 10.0.5.12 - Routed	FTDv for KVM	7.4.1	N/A	Essentials	TEMPACP
□	FTDB Snort 3 10.0.5.13 - Routed	FTDv for KVM	7.4.1	N/A	Essentials	TEMPACP

Before Running Ansible Task

After running the ansible task, log in FMC GUI. Navigate to **Devices** > **Device Management,** FTD HA is created successfully.



Firewall Management Center
Devices / Device Management

View By: Group

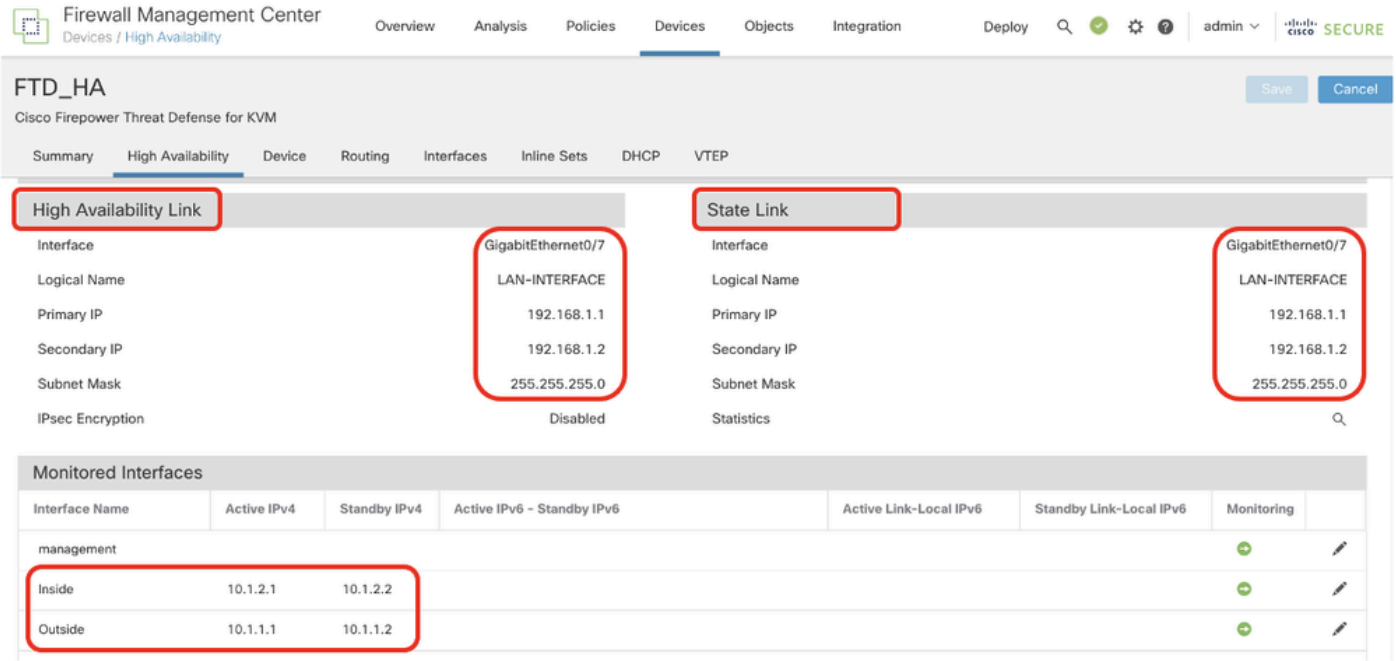
All (2) ● Error (0) ● Warning (0) ● Offline (0) ● Normal (2) ● Deployment Pending (0) ● Upgrade (0) ● Snort 3 (2)

Collapse All

Name	Model	Version	Chassis	Licenses	Access Contr
Ungrouped (1)					
FTD_HA High Availability					
FTDA(Primary, Active) Snort 3 10.0.5.12 - Routed	FTDv for KVM	7.4.1	N/A	Essentials	TEMPACP
FTDB(Secondary, Standby) Snort 3 10.0.5.13 - Routed	FTDv for KVM	7.4.1	N/A	Essentials	TEMPACP

After Running Ansible Task Successfully

Click **Edit** of FTD HA, failover ip address and interface standby ip address are configured successfully.



Firewall Management Center
Devices / High Availability

FTD_HA
Cisco Firepower Threat Defense for KVM

High Availability Link		State Link	
Interface	GigabitEthernet0/7	Interface	GigabitEthernet0/7
Logical Name	LAN-INTERFACE	Logical Name	LAN-INTERFACE
Primary IP	192.168.1.1	Primary IP	192.168.1.1
Secondary IP	192.168.1.2	Secondary IP	192.168.1.2
Subnet Mask	255.255.255.0	Subnet Mask	255.255.255.0
IPsec Encryption	Disabled	Statistics	

Monitored Interfaces

Interface Name	Active IPv4	Standby IPv4	Active IPv6 - Standby IPv6	Active Link-Local IPv6	Standby Link-Local IPv6	Monitoring
management						●
Inside	10.1.2.1	10.1.2.2				●
Outside	10.1.1.1	10.1.1.2				●

FTD High Availability Detail

Troubleshoot

This section provides information you can use to troubleshoot your configuration.

In order to see more logs of ansible playbook, you can run ansible playbook with –vvv.

<#root>

```
cisco@inserthostname-here:~/fmc_ansible$ ansible-playbook -i inventory.ini fmc-create-ftd-ha-standby-ip-playbook.yaml -e@"fmc-create-ftd-ha-standby-
-vvv
```

Related Information

Cisco Devnet FMC Ansible