Configure FMC with Ansible to Onboard FTD

Contents

Introduction

This document describes the steps to automate Firepower Threat Defense (FTD) registration to Firepower Management Center (FMC) 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.

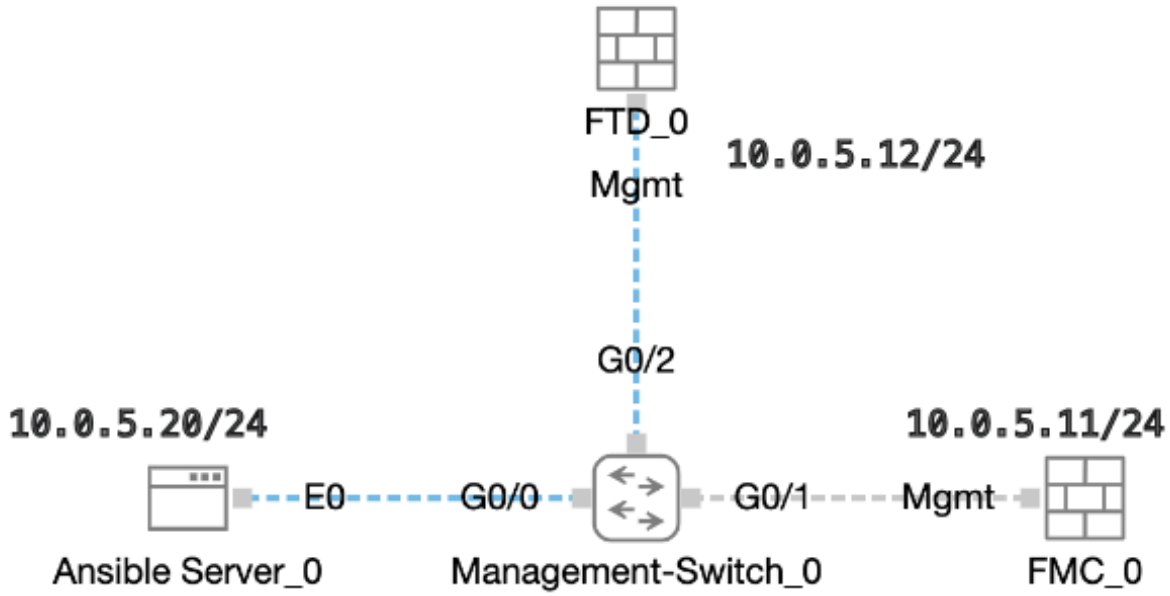
Background Information

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, after successfully onboarding the virtual FTD it is with base license, routed mode, feature tier FTDv30, and the access control policy which is with default permit action with log enabled sending to FMC.

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 verification has 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).
- The FTD is configured with correct manager ip address, register key and nat-id.
- The FMC is enabled with smart license successfully.

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 the following content and paste it for utilization, altering the **highlighted** 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. In this example, the variable file name is fmc-onboard-ftd-vars.yml.

```
<#root>
cisco@inserthostname-here:~$
 cd /home/cisco/fmc_ansible/


ccisco@inserthostname-here:~/fmc_ansible$
ls


fmc-onboard-ftd-vars.yml
inventory.ini
```

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

```
<#root>
user:
  domain: 'Global'
onboard:
  acp_name: '
TEMPACP
'
device_name:
  ftd1: '
FTDA
'
  ftd1_reg_key: '
cisco
'
  ftd1_nat_id: '
natcisco
```

```
'
mgmt:
  ftd1: '
10.0.5.12
'
```

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

<#root>

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

ccisco@inserthostname-here:~/fmc_ansible$
ls

fmc-onboard-ftd-playbook.yaml
fmc-onboard-ftd-vars.yml inventory.ini
```

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

<#root>

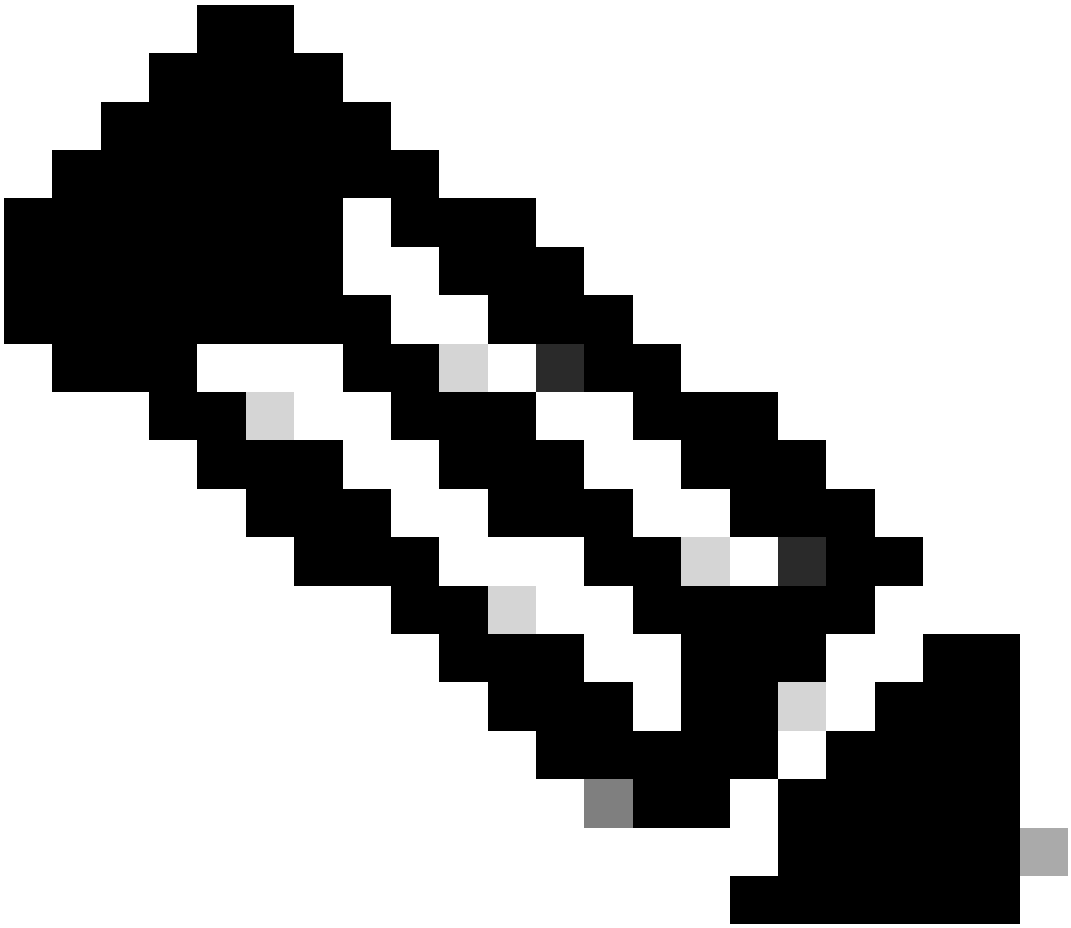
```
---

- name: FMC Onboard FTD
  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 - Create ACP TEMP_ACP
      cisco.fmcansible.fmc_configuration:
        operation: "createAccessPolicy"
        data:
           type: "AccessPolicy"
           name: "{{accesspolicy_name | default(
```

```
onboard.acp_name

) }}"
           defaultAction: {
             'action': 'PERMIT',
             'logEnd': True,
             'logBegin': False,
             'sendEventsToFMC': True
             }
        path_params:
          domainUUID: "{{ domain[0].uuid }}"

    - name: Task03 - Get Access Policy
      cisco.fmcansible.fmc_configuration:
        operation: getAllAccessPolicy
        path_params:
          domainUUID: "{{ domain[0].uuid }}"
        filters:
          name: "{{

onboard.acp_name

 }}"
        register_as: access_policy

    - name: Task04 - Add New FTD1
      cisco.fmcansible.fmc_configuration:
        operation: createMultipleDevice
        data:
          hostName: "{{ ftd_ip | default(item.key) }}"
          license_caps:
            - 'BASE'
          ftdMode: 'ROUTED'
          type: Device
          regKey: "{{ reg_key | default(

device_name.ftd1_reg_key

) }}"
          performanceTier: "FTDv30"
          name: "{{ ftd_name | default(item.value) }}"
          accessPolicy:
            id: '{{ access_policy[0].id }}'
            type: 'AccessPolicy'
          natID: "{{ nat_id | default(

device_name.ftd1_nat_id

) }}"
        path_params:
          domainUUID: '{{ domain[0].uuid }}'
      loop: "{{ ftd_ip_name | dict2items }}"
      vars:
        ftd_ip_name:
          "{{

mgmt.ftd1

}}": "{{

device_name.ftd1

}}"

    - name: Task05 - Wait For FTD Registration Completion
```

```
      ansible.builtin.wait_for:
        timeout: 120
      delegate_to: localhost


    - name: Task06 - Confirm FTD Init Deploy Complete
      cisco.fmcansible.fmc_configuration:
        operation: getAllDevice
        path_params:
          domainUUID: '{{ domain[0].uuid }}'
        query_params:
          expanded: true
        filters:
          name: "{{
```

device_name.ftd1

```
 }}"
        register_as: device_list
      until: device_list[0].deploymentStatus is match("DEPLOYED")
      retries: 1000
      delay: 3
```



Note: The names highlighted 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-onboard-ftd-playbook.yaml -e @"fmc-onboard-ftd-vars.yml" **.**

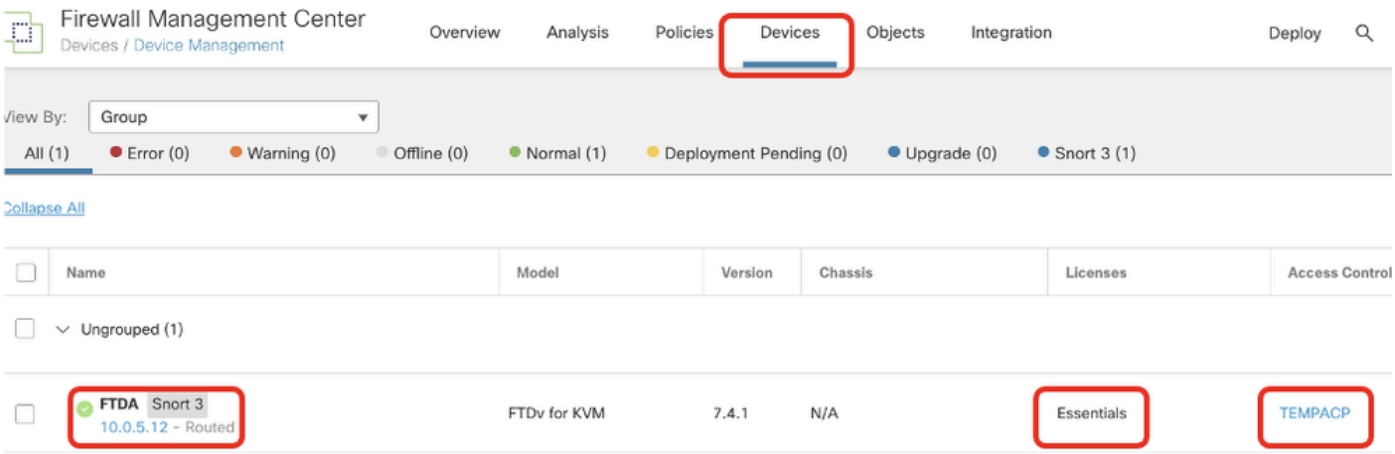
<#root>

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

cisco@inserthostname-here:~/fmc_ansible$
ls

fmc-onboard-ftd-playbook.yaml fmc-onboard-ftd-vars.yml inventory.ini
cisco@inserthostname-here:~/fmc_ansible$
ansible-playbook -i inventory.ini fmc-onboard-ftd-playbook.yaml -e @"fmc-onboard-ftd-vars.yml"

PLAY [FMC Onboard FTD] *********************************************************************************

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

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

TASK [Task02 - Create ACP TEMP_ACP] ********************************************************************
changed: [10.0.5.11]

TASK [Task03 - Get Access Policy] **********************************************************************
ok: [10.0.5.11]

TASK [Task04 - Add New FTD1] ***************************************************************************
changed: [10.0.5.11] => (item={'key': '10.0.5.12', 'value': 'FTDA'})

TASK [Task05 - Wait For FTD Registration Completion] ***************************************************
ok: [10.0.5.11]

TASK [Task06 - Confirm FTD Init Deploy Complete] *******************************************************
FAILED - RETRYING: Task06 - Confirm FTD Init Deploy Complete (1000 retries left).
FAILED - RETRYING: Task06 - Confirm FTD Init Deploy Complete (999 retries left).
FAILED - RETRYING: Task06 - Confirm FTD Init Deploy Complete (998 retries left).
FAILED - RETRYING: Task06 - Confirm FTD Init Deploy Complete (997 retries left).
FAILED - RETRYING: Task06 - Confirm FTD Init Deploy Complete (996 retries left).
ok: [10.0.5.11]

PLAY RECAP *********************************************************************************************
10.0.5.11    : ok=7  changed=2  unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

Verify

Use this section to confirm that your configuration works properly.

Log in FMC GUI. Navigate to **Devices > Device Management,** the FTD registered successfully on FMC with configured access control policy.



Device Management Page

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-onboard-ftd-playbook.yam
```

-vvv

Related Information

[Cisco Devnet FMC Ansible]