Configuration de FMC avec Ansible pour mettre à jour l'interface FTD IP

Table des matières

Introduction

Conditions préalables

- Exigences
- Composants utilisés

Informations générales

Configurer

- Diagramme du réseau
- Configurations

Vérifier

Dépannage

Informations connexes

Introduction

Ce document décrit les étapes d'automatisation de Firepower Management Center (FMC) pour configurer l'interface IP Firepower Threat Defense (FTD) avec Ansible.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Ansible
- Serveur Ubuntu
- Cisco Firepower Management Center (FMC) virtuel
- Cisco Firepower Threat Defense (FTD) virtuel

Dans le cadre de cette situation de laboratoire, Ansible est déployé sur Ubuntu.

Il est essentiel de s'assurer que Ansible est correctement installé sur toute plate-forme prise en charge par Ansible pour exécuter les commandes Ansible mentionnées dans cet article.

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Serveur Ubuntu 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. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

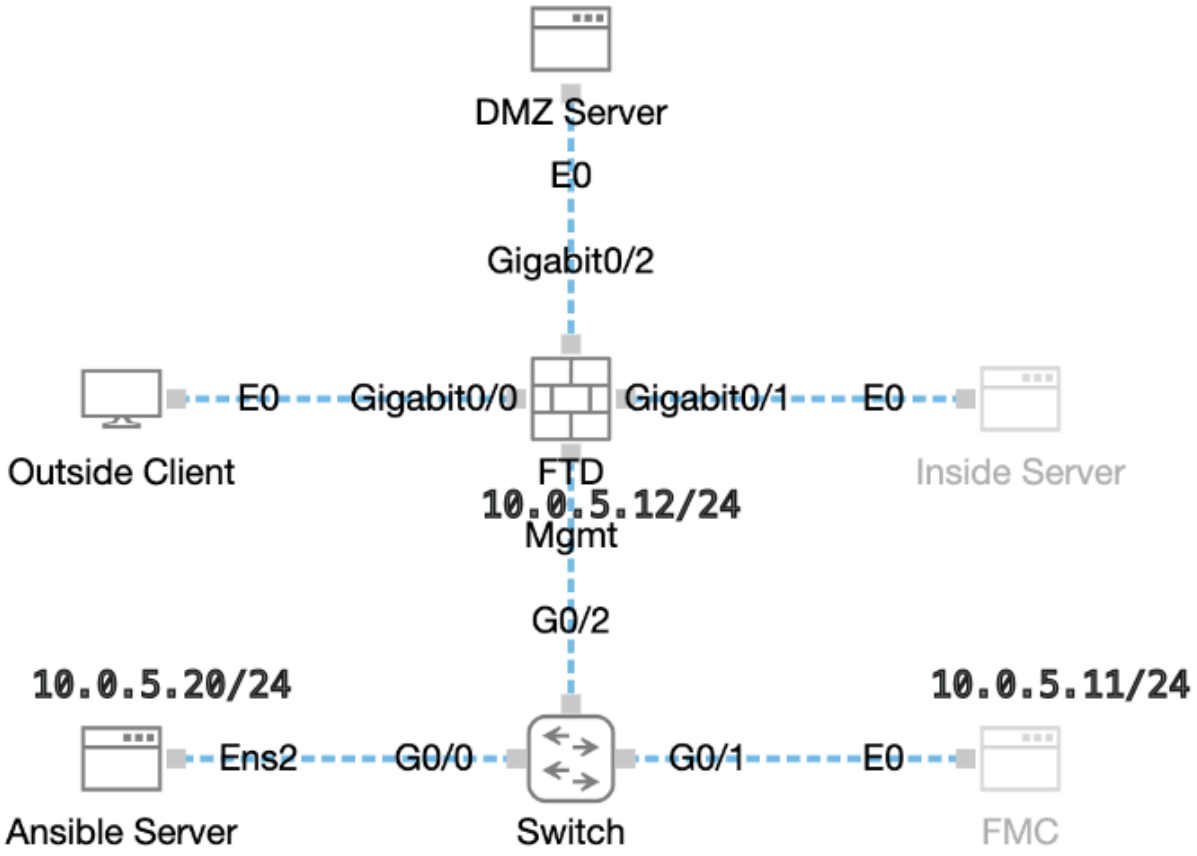
Informations générales

Ansible est un outil très polyvalent, qui démontre une efficacité significative dans la gestion des périphériques réseau. De nombreuses méthodologies peuvent être utilisées pour exécuter des tâches automatisées avec Ansible. La méthode utilisée dans cet article sert de référence aux fins de l'essai.

Dans cet exemple, l'adresse IP de l'interface, le masque et le nom de l'interface sont mis à jour en FTD après l'exécution réussie de l'exemple de guide.

Configurer

Diagramme du réseau



Topologie

Configurations

Étant donné que Cisco ne prend pas en charge les scripts d'exemple ou les scripts écrits par le client, nous avons quelques exemples que vous pouvez tester en fonction de vos besoins.

Il est essentiel de veiller à ce que la vérification préliminaire ait été dûment menée à bien.

- Le serveur Ansible possède une connectivité Internet.
- Le serveur Ansible est capable de communiquer avec le port de l'interface graphique FMC (le port par défaut de l'interface graphique FMC est 443).
- Le FTD est correctement enregistré auprès de FMC.

Étape 1. Connectez-vous à la CLI du serveur Ansible via SSH ou la console.

Étape 2. Exécutez la commandeansible-galaxy collection install cisco.fmcansible afin d'installer la collection Ansible de FMC sur votre serveur Ansible.

<#root>

cisco@inserthostname-here:~$

ansible-galaxy collection install cisco.fmcansible

Étape 3. Exécutez mkdir /home/cisco/fmc_ansiblela commande afin de créer un nouveau dossier pour stocker les fichiers associés. Dans cet exemple, le répertoire de base est /home/cisco/, le nouveau nom de dossier est fmc_ansible.

<#root>

cisco@inserthostname-here:~$

mkdir /home/cisco/fmc_ansible

Étape 4. Accédez au dossier **/home/cisco/fmc_ansible**, create inventory file. Dans cet exemple, le nom du fichier d'inventaire est inventory.ini.

<#root>

cisco@inserthostname-here:~$

cd /home/cisco/fmc_ansible/

ccisco@inserthostname-here:~/fmc_ansible$

ls

inventory.ini

Vous pouvez dupliquer ce contenu et le coller pour l'utiliser, en modifiant les sections **mises en surbrillance** avec les paramètres précis.

<#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

Étape 5. Accédez au dossier **/home/cisco/fmc_ansible**, create variable file. Dans cet exemple, le nom de fichier variable est fmc-configure-interface-vars.yml.

<#root>

cisco@inserthostname-here:~$

cd /home/cisco/fmc_ansible/

ccisco@inserthostname-here:~/fmc_ansible$

ls

fmc-configure-interface-vars.yml

inventory.ini

Vous pouvez dupliquer ce contenu et le coller pour l'utiliser, en modifiant les sections **mises en surbrillance** avec les paramètres précis.

<#root>

user: domain: 'Global' onboard: acp_name: 'TEMPACP' device_name: ftd1: 'FTDA' ftd_data: outside_name: '

Outside

' inside_name: '

Inside

' dmz_name: '

DMZ

' outside_ip: '

10.1.1.1

' inside_ip: '

10.1.2.1

' dmz_ip: '

10.1.3.1

' mask24: '

255.255.255.0

'

Étape 6. Accédez au dossier **/home/cisco/fmc_ansible**, créez un fichier de manuel. Dans cet exemple, le nom du fichier du guide est fmc-configure-interface-playbook.yaml.

<#root>

cisco@inserthostname-here:~$

cd /home/cisco/fmc_ansible/

ccisco@inserthostname-here:~/fmc_ansible$

ls

fmc-configure-interface-playbook.yaml

fmc-configure-interface-vars.yml inventory.ini

Vous pouvez dupliquer ce contenu et le coller pour l'utiliser, en modifiant les sections **mises en surbrillance** avec les paramètres précis.

<#root>

--- - name: Update FTD Interface IP Address hosts: fmc connection: httpapi tasks: - name: Task01 - Get User Domain cisco.fmcansible.fmc_configu

user.domain

}}" register_as: domain - name: Task02 - Get Devices cisco.fmcansible.fmc_configuration: operation: ge

device_name.ftd1

}}" register_as: device_list - name: Task03 - Get Physical Interfaces cisco.fmcansible.fmc_configurati

ftd_data.outside_name

}}" ipv4: static: address: "{{ Outside_ip | default(

ftd_data.outside_ip

) }}" netmask: "{{ Outside_netmask | default(

ftd_data.mask24

) }}" MTU: 1500 enabled: True mode: NONE type: physicalinterface name:

GigabitEthernet0/0

path_params: domainUUID: '{{ domain[0].uuid }}' containerUUID: '{{ device_list[0].id }}' objectId: '{{

ftd_data.inside_name

}}" ipv4: static: address: "{{ Inside_ip | default(

ftd_data.inside_ip)

}}" netmask: "{{ Inside_netmask | default(

ftd_data.mask24

) }}" MTU: 1500 enabled: True mode: NONE type: physicalinterface name:

GigabitEthernet0/1

path_params: domainUUID: '{{ domain[0].uuid }}' containerUUID: '{{ device_list[0].id }}' objectId: '{{

ftd_data.dmz_name

}}" ipv4: static: address: "{{ DMZ_ip | default(

ftd_data.dmz_ip

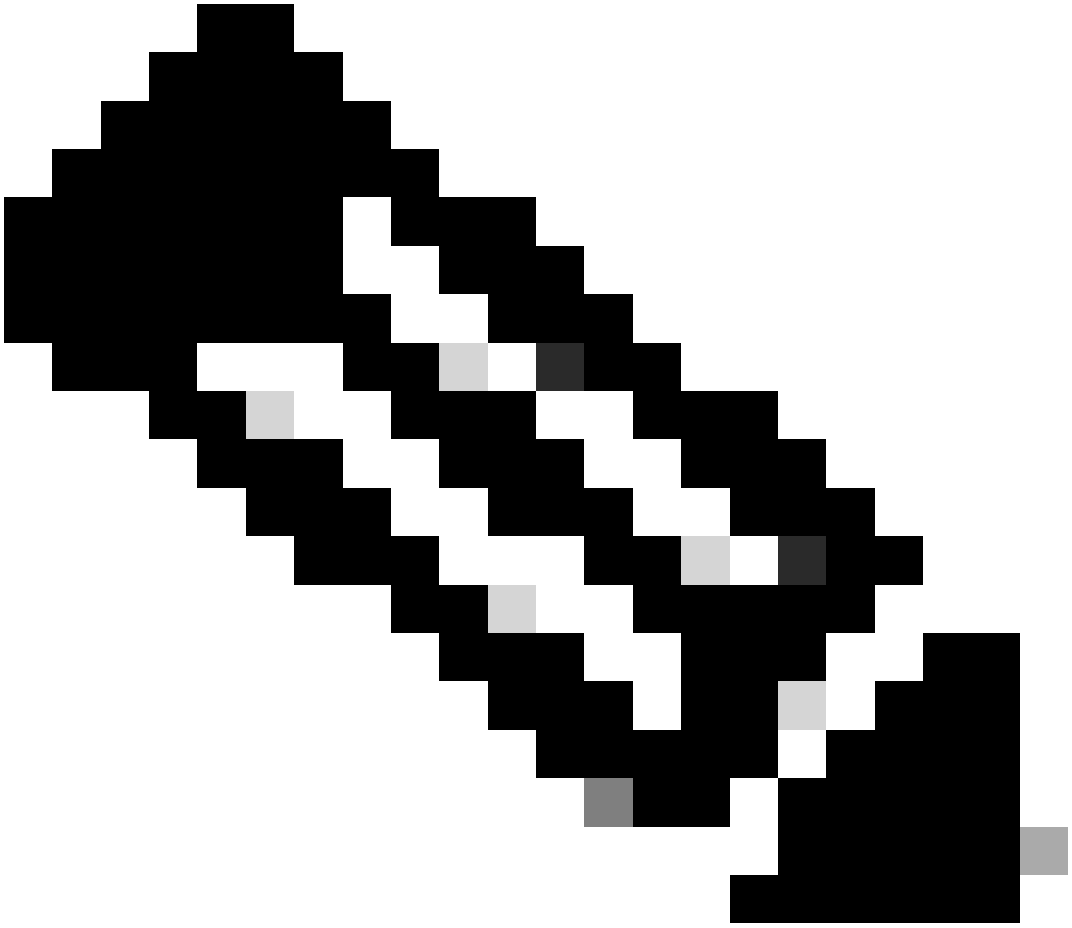
) }}" netmask: "{{ DMZ_netmask | default(

ftd_data.mask24

) }}" MTU: 1500 enabled: True mode: NONE type: physicalinterface name:

GigabitEthernet0/2

path_params: domainUUID: '{{ domain[0].uuid }}' containerUUID: '{{ device_list[0].id }}' objectId: '{{



Remarque : les noms mis en surbrillance dans cet exemple de guide de vente servent de variables. Les valeurs correspondantes de ces variables sont conservées dans le fichier de variables.

Étape 7. Accédez au dossier **/home/cisco/fmc_ansible**, run command ansible-playbook -i <inventory_name>.ini <playbook_name>.yaml -e@"<playbook_vars>.yml"afin de lire la tâche ansible.

Dans cet exemple, la commande est ansible-playbook -i inventory.ini fmc-configure-interface-playbook.yaml -e@"fmc-configure-interface-vars.yml" .

<#root>

cisco@inserthostname-here:~$

cd /home/cisco/fmc_ansible/

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

ls

```
fmc-configure-interface-playbook.yaml fmc-configure-interface-vars.yml inventory.ini

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

ansible-playbook -i inventory.ini fmc-configure-interface-playbook.yaml -e@"fmc-configure-interface-vars

```
PLAY [Update FTD Interface IP Address] ***************************************************************

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

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

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

TASK [Task03 - Get Physical Interfaces] **************************************************************
ok: [10.0.5.11]

TASK [Task04 - Setup Outside Interface with static IP] ***********************************************
changed: [10.0.5.11]

TASK [Task05 - Setup Inside Interface with static IP] ************************************************
changed: [10.0.5.11]

TASK [Task06 - Setup DMZ Interface with static] ******************************************************
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 [Wait for Deployment Complete] ******************************************************************
ok: [10.0.5.11]

TASK [Task09 - Poll Deployment Status Until Deployment Successful] ***********************************
ok: [10.0.5.11]

TASK [Task10 - Stop The Playbook If The Deployment Failed] *******************************************
skipping: [10.0.5.11]

PLAY RECAP *******************************************************************************************
10.0.5.11 : ok=11 changed=4 unreachable=0 failed=0 skipped=1 rescued=0 ignored=0
```

Vérifier

Référez-vous à cette section pour vous assurer du bon fonctionnement de votre configuration.

Connectez-vous à la CLI du FTD via SSH ou la console et exécutez les commandes show interface ip brief et show running-config interface GigabitEthernet 0/X .

Le nom de l'interface, l'adresse IP et le masque sont correctement configurés.

<#root>

> show interface ip brief
Interface IP-Address OK? Method Status Protocol

GigabitEthernet0/0 10.1.1.1

YES manual

up up

GigabitEthernet0/1 10.1.2.1

YES manual

up up

GigabitEthernet0/2 10.1.3.1

YES manual

up up

>

show running-config interface GigabitEthernet 0/0

!
interface GigabitEthernet0/0
nameif

Outside

cts manual
propagate sgt preserve-untag
policy static sgt disabled trusted
security-level 0

ip address 10.1.1.1 255.255.255.0

>

show running-config interface GigabitEthernet 0/1

!

```
interface GigabitEthernet0/1
nameif
```

Inside

```
cts manual
propagate sgt preserve-untag
policy static sgt disabled trusted
security-level 0
```

ip address 10.1.2.1 255.255.255.0

```
>
```

show running-config interface GigabitEthernet 0/2

```
!
interface GigabitEthernet0/2
nameif
```

DMZ

```
cts manual
propagate sgt preserve-untag
policy static sgt disabled trusted
security-level 0
ip address 10.1.3.1 255.255.255.0
```

Dépannage

Cette section fournit des informations que vous pouvez utiliser pour dépanner votre configuration.

Afin de voir plus de journaux du playbook ansible, vous pouvez exécuter le playbook ansible avec -vvv

cisco@inserthostname-here:~/fmc_ansible$ ansible-playbook -i inventory.ini fmc-configure-interface-playbook.yaml -e@"fmc-configure-interface-vars.yn

Informations connexes

[Cisco Devnet FMC Ansible]

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.