

Push Objects in Bulk to FMC à l'aide de REST-API

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Limites](#)

[Informations générales](#)

[Configuration](#)

[Vérification](#)

[Dépannage](#)

Introduction

Ce document décrit comment un administrateur de l'interface de programmation d'applications (API) peut envoyer des objets réseau, port et URL en masse vers Firepower Management Center (FMC).

Conditions préalables

Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Compréhension des différents appels de l'API REST. ([Que sont les API REST ?](#))
- Examen du [Guide de démarrage rapide de l'API FMC](#)
- Examen des [objets réutilisables FMC](#)
- Connaissances de base de la bibliothèque de requêtes Python

Components Used

- Firepower Management Center qui prend en charge les API REST (version 6.1 ou ultérieure) avec l'API REST activée
- Interactions avec l'API REST à l'aide de Python.

Limites

- FMC n'accepte pas que le nom de l'objet soit supérieur à 64 caractères.
- Le nom de l'objet ne doit pas comporter d'espace au début du nom de l'objet et de point-virgule à la fin.
- La charge utile ne peut pas contenir plus de 1 000 entrées dans une seule transmission en

masse.

- La taille de la charge utile ne peut pas être supérieure à 2 Mo dans une seule transmission en masse.

Informations générales

Les API REST sont de plus en plus populaires en raison de l'approche programmable légère que les administrateurs réseau peuvent utiliser pour configurer et gérer leurs réseaux. FMC prend en charge la configuration et la gestion à l'aide de n'importe quel client REST et aussi à l'aide de l'explorateur d'API intégré.

L'exemple de ce document prend un fichier CSV en entrée et pousse les objets vers FMC via l'interface API REST. Le document ne couvre que la diffusion en masse du réseau hôte et une logique similaire peut être étendue aux autres objets. Un exemple de code est joint au document pour les objets URL et Port.

Voici la référence de l'API pour le POST sur les hôtes réseau qui sont utilisés, comme l'illustre l'image :

POST /api/fmc_config/v1/domain/{domainUUID}/object/hosts

Retrieves, deletes, creates, or modifies the host object associated with the specified ID. If no ID is specified for a GET, retrieves list of all host objects. *Check the response section for applicable examples (if any).*

Parameters Try it out

Name	Description
body required object (body)	Input representation of host object. Parameter content type application/json
bulk boolean (query)	Enables bulk create for host objects. --
domainUUID required string (path)	Domain UUID e276abec-e0f2-11e3-8169-6d9ed49b625f

Responses Response content type application/json

Code	Description
201	Created Example Value Model Request example 1 : POST /fmc_config/v1/domain/domainUUID/object/hosts (POST to create a host object) <pre>{ "name": "TestHost", "type": "Host", "value": "10.5.3.20", "description": "Test Description" }</pre> Request example 2 : POST /fmc_config/v1/domain/domainUUID/object/hosts?bulk=true (Bulk POST operation for Host object) <pre>[{ "name": "host1", "type": "Host", "value": "10.5.3.20", "description": "Test Description" }, { "name": "Host2", "type": "Host", "value": "1.2.3.4", "description": "Host object 2" }]</pre>

Configuration

Étape 1. Activez l'API REST et générez un jeton d'authentification. Pour obtenir des exemples et des étapes de configuration détaillées, reportez-vous à la section [Generate Authentication Token on FMC](#).

```
import requests import csv import json from requests.auth import HTTPBasicAuth from getpass import getpass address = input("Enter IP Address of the FMC: ") username = input ("Enter Username: ") password = getpass("Enter Password: ") api_uri = "/api/fmc_platform/v1/auth/generatetoken" url = "https://" + address + api_uri response = requests.request("POST", url, verify=False, auth=HTTPBasicAuth(username, password)) accesstoken = response.headers["X-auth-access-token"] refreshtoken = response.headers["X-auth-refresh-token"] DOMAIN_UUID = response.headers["DOMAIN_UUID"]
```

Étape 2. Convertissez le fichier CSV fourni à un dictionnaire à utiliser comme charge utile JSON pour la demande. Un exemple de fichier CSV pour chaque type d'objet est joint au document.

	A	B	C	D
1	name	description	type	value
2	Host-1	Host-1	Host	10.10.10.10
3	Host-2	Host-2	Host	10.10.10.1
4	Network-1	Network-1	Network	10.10.9.0/24
5	Host-3	Host-3	Host	10.10.10.2
6	Range-1	Rannge-1	Range	10.20.20.1-10.20.20.20
7				

```
csvFilePath = input("Please enter the CSV Filepath (For eg. : path/to/file/objects.csv) :") host = [] with open(csvFilePath, encoding='utf-8-sig') as csvf: csvReader = csv.DictReader(csvf) for rows in csvReader: if rows['type'] == "Host": host.append(rows) host_payload = json.dumps(host)
```

À ce stade, host_payload ressemble à l'image suivante :

```
[{ "name": "Host-1", "description": "Host-1", "type": "Host", "value": "10.10.10.10" }, { "name": "Host-2", "description": "Host-2", "type": "Host", "value": "10.10.10.1" }, { "name": "Host-3", "description": "Host-3", "type": "Host", "value": "10.10.10.2" } ]
```

Étape 3. Créez la demande à partir de l'entrée reçue des étapes précédentes et envoyez la demande si la charge utile n'est pas vide.

```
host_api_uri = "/api/fmc_config/v1/domain/" + DOMAIN_UUID + "/object/hosts?bulk =true" host_url = "https://" + address + host_api_uri headers = { 'Content-Type': 'application/json', 'x-auth-access-token': accesstoken } if host != []: response = requests.request("POST", host_url, headers=headers, data = host_payload, verify = False) else : print("Please Validate that the CSV file provided is correct or at correct location")
```

Vérification

- Imprimez le code d'état de la réponse pour vérifier si la demande a réussi ou échoué, comme indiqué ici.

```
if response.status_code == 201 or response.status_code == 202: print("Host Objects successfully pushed") else: print("Host Object creation failed")
```

- Connectez-vous à FMC Naviguez jusqu'à **Object > Object Management > Network** et vérifiez les objets hôtes, comme illustré dans l'image :

Name	Value	Type
Host-1	10.10.10.10	Host
Host-2	10.10.10.1	Host
Host-3	10.10.10.2	Host

Dépannage

- Lors de l'utilisation du client REST, il se peut que des erreurs liées au problème de certificat SSL se produisent en raison d'un certificat auto-signé. Vous pouvez désactiver cette validation en fonction du client que vous utilisez.
- Les jetons d'authentification de l'API REST FMC sont valides pendant 30 minutes et peuvent être actualisés jusqu'à trois fois.
- L'erreur liée à la demande peut être extraite du corps de la réponse. Il peut être collecté en tant que fichier journal pour faciliter le dépannage.

```
logfile = "requestlog.txt" log = open(logfile,"w+") log.write(response.text) log.close
```

- Toutes les requêtes REST sont connectées à ces deux fichiers journaux sur FMC. Recherchez votre URL (ex. .../object/hosts) avec le bon fonctionnement (Si vous recherchez une erreur pour l'opération GET, assurez-vous que votre journal démarre quelque chose comme GET ...objet/hôtes)

```
tail -f /var/opt/CSCOpX/MDC/tomcat/logs/stdout.logs tail -f /var/opt/CSCOpX/MDC/log/operation/usmshredsvcs.log
```