cisco Live!

Let's go

The bridge to possible

FirePower Threat Defense Automation using API's

Rafael Leiva-Ochoa

BRKCRT-3001

Cisco Webex App

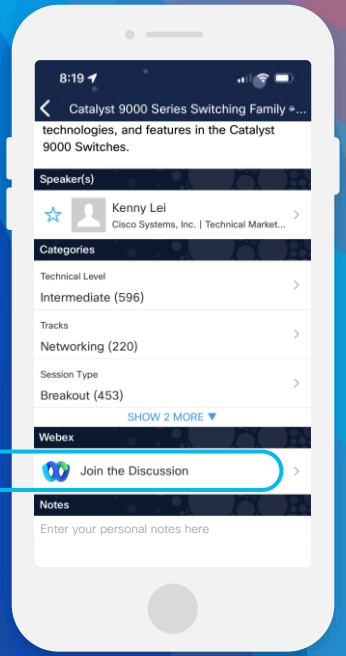
Questions?

Use Cisco Webex App to chat
with the speaker after the session

How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until June 9, 2023.



https://ciscolive.ciscoevents.com/ciscolivebot/#BRKCRT-3001

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Agenda

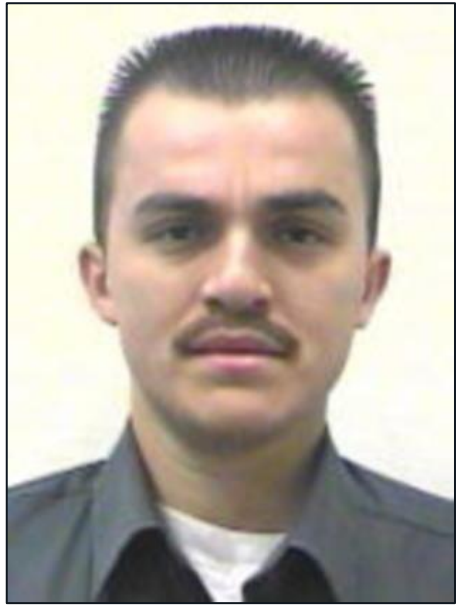
- Introduction
- Overview on FMC REST API
  - API Explorer
  - Postman
  - Python
- FMC API Operations
- Creating FMC Objects
- Creating FMC Access Control Policy
- Managing the FirePower Device Configuration
- FMC API Troubleshooting

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Introduction

Introduction

- Rafael Leiva-Ochoa
- @Cisco since Oct 2000
- Works in the CX Training Group (Part of Learning@Cisco)
- Delivers courses on Security to Global TAC Centers and Customers
- CCIE 19322 Security since 2007 (16 years!!)



CISCO CERTIFIED CCIE SECURITY

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

What is the Goal?

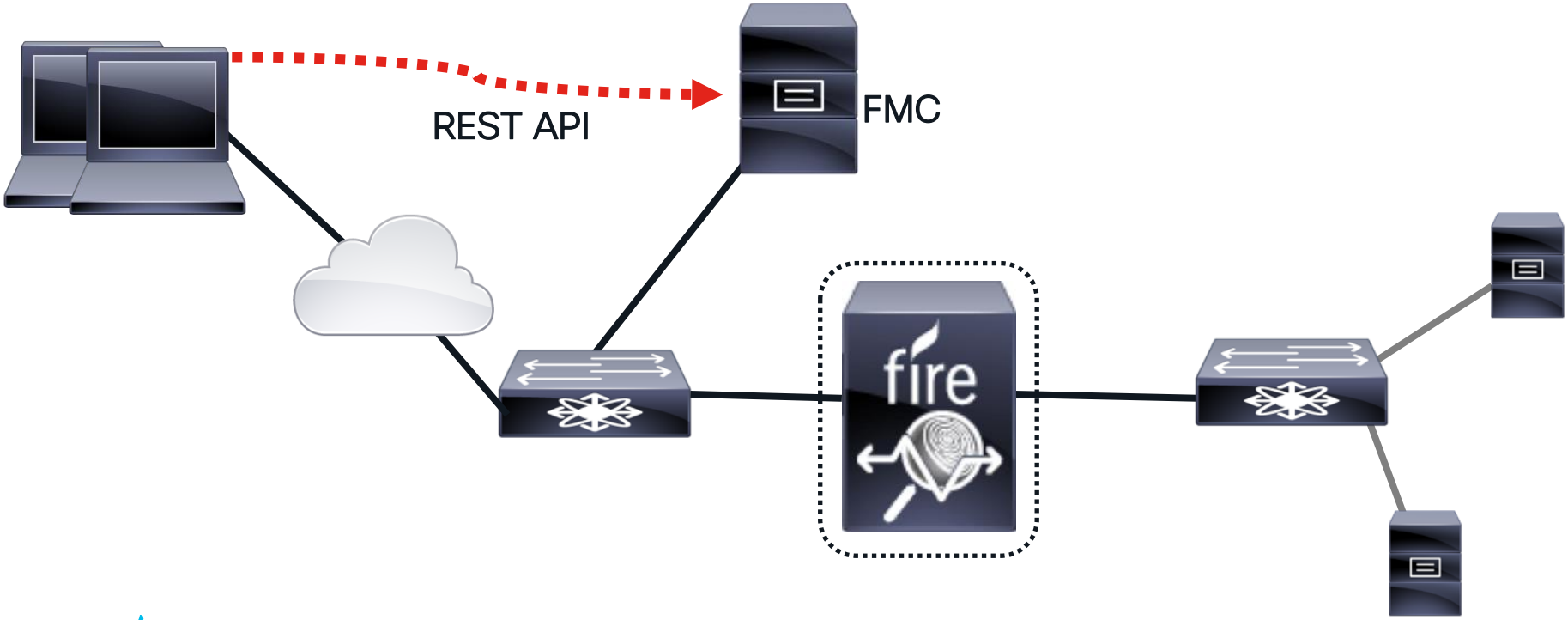
- To give the administrator/operator a better understating of the FMC REST API, and its capabilities. Learning REST API should be easy not hard, and that is the goal of this breakout.

- **Note:** The vast majority of the examples will be in Python, since it offers an array of features, and customization options.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Overview on FMC REST API

REST API Overview



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

REST API Overview (Cont.)

- With the Cisco Firepower Management Center REST API, you now have a lightweight, easy-to-use option for managing and configuring the Firepower Threat Defense. (Virtual or Physical)
- The REST API is an application programming interface (API), based on “RESTful” principles, which you can quickly enable on any Firepower Management Center running version 6.1 or higher, and use with a REST client.
- REST APIs use the same methods as HTTP to send requests or get responses from the FirePower Management Center like **GET, POST, PUT,** and **DELETE.**

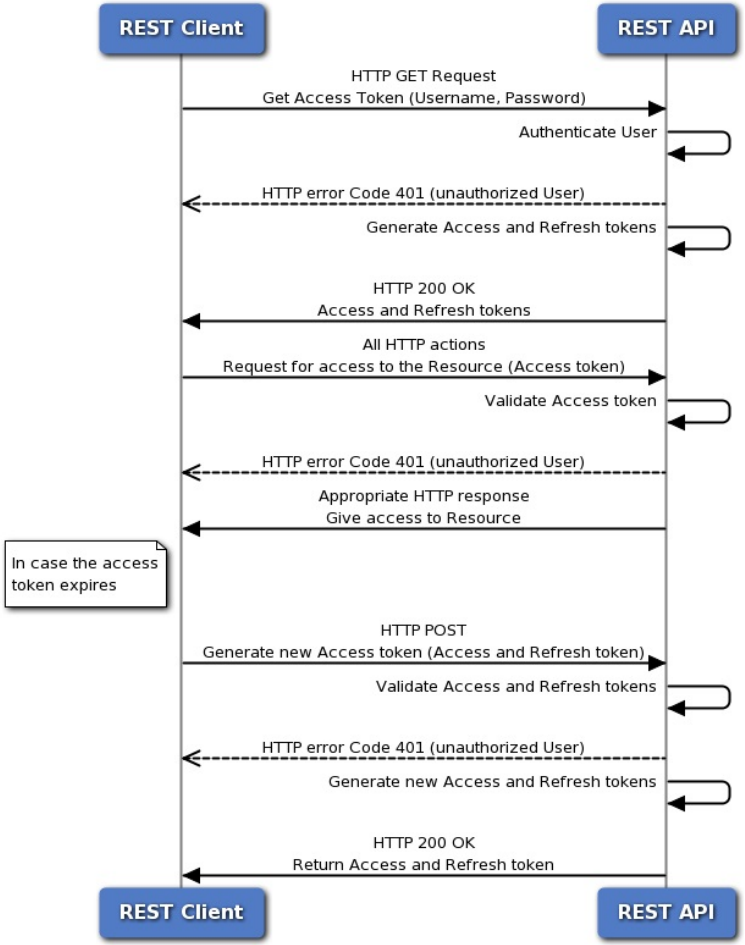
© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

REST API Explorer Access Token

- Tokens are used to access the HTTP service for a limited time period without the need for the username and password with every request
- In order eliminate the need for authenticating with your username and password with each request, you replace user credentials with a uniquely generated access token
- Tokens are only good for 30 minutes and can refresh up to three times.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

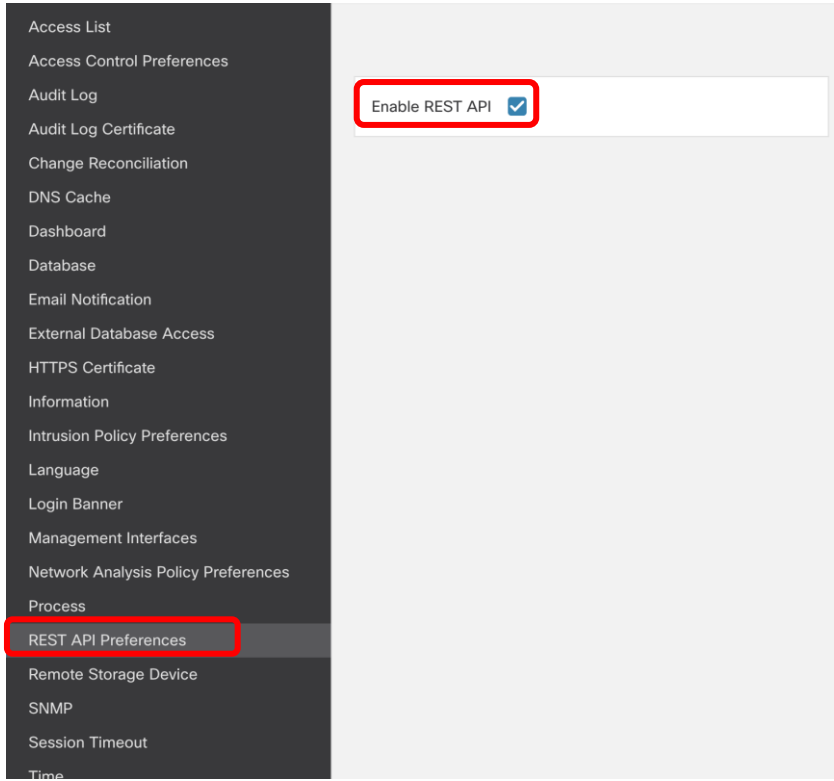
Token-Based Authentication



FMC

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Enable REST API on FMC

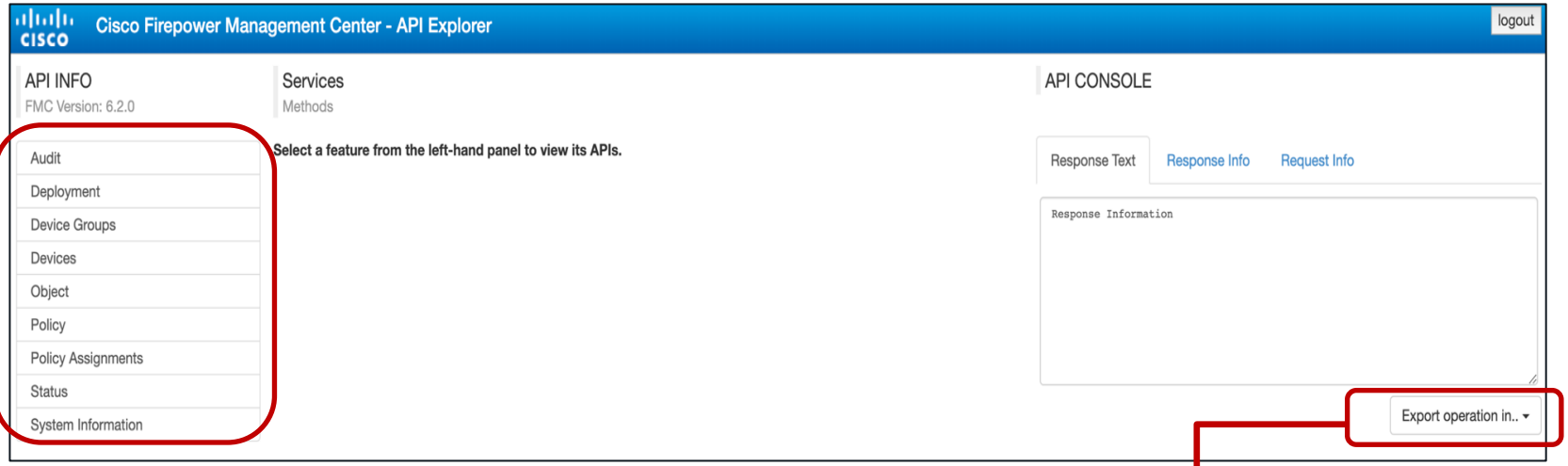


- System > Configuration
- The REST API is enabled by default
- Base URL:

https://<FMC_FQDN>/api/api-explorer

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Access API thru API-Explorer – Older Interface

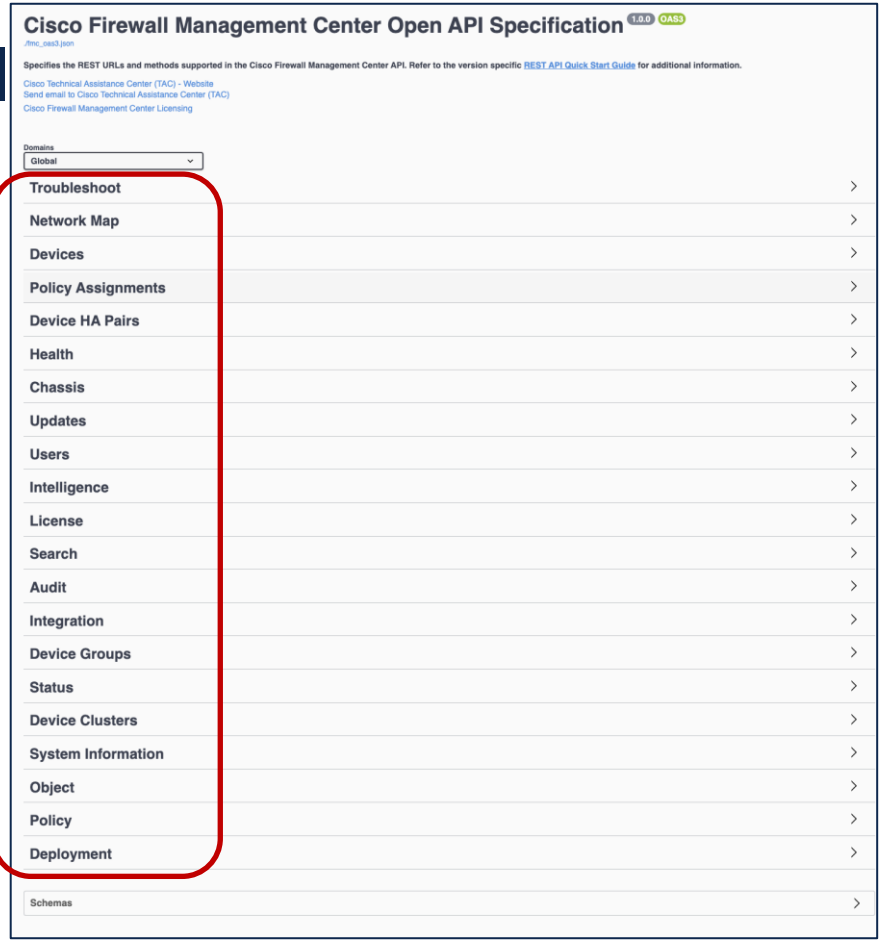


- Less Object Support
- Non-Standard Code Generation (Limited to Python, and Perl)
- No domain selection

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Access API thru API-Expl

- Interface changes during 6.4.0 release
- Simplified design
- More objects in the REST API selection
- Sample Code with CodeGen
  - Using Open API Spec
  - HTML, Perl, Python, and Java
- Troubleshooting Objects
  - Packet Tracer
  - PCAP file support
- Domain selection supported on interface



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Access API thru API-Explorer (Cont.)

GET /api/fmc_platform/v1/info/domain

API Operation for Domains.

Parameters

Try it out

Name	Description
offset integer($int32) (query)	Index of first item to return. offset - Index of first item to return.
limit integer($int32) (query)	Number of items to return. limit - Number of items to return.
expanded boolean (query)	If set to true, the GET response displays a list of objects with additional attributes. --

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Access API thru API-Explorer (Cont.)

Parameters

Cancel

Name	Description
offset integer($int32) (query)	Index of first item to return. offset - Index of first item to return.
limit integer($int32) (query)	Number of items to return. limit - Number of items to return.
expanded boolean (query)	If set to true, the GET response displays a list of objects with additional attributes. --

Execute

Responses

Code	Description	Links
200	OK Media type: application/json Controls Accept header. Example Value | Schema	No links
default	Error Media type: application/json Example Value | Schema	No links

```
{
  "links": {
    "parent": "string",
    "self": "string"
  },
  "paging": {
    "pages": 0,
    "offset": 0,
    "limit": 0,
    "count": 0
  },
  "items": {
    "name": "string",
    "links": {
      "parent": "string",
      "self": "string"
    },
    "id": "string",
    "type": "string",
    "uuid": "string"
  }
}
```

```
{
  "severity": "ERROR",
  "messages": [
    {
      "severity": "ERROR",
      "code": "string",
      "description": "string",
      "errorCode": "string",
      "bulkPayloadIndex": "string",
      "location": "string",
      "details": "string"
    }
  ],
  "category": "FRAMEWORK"
}
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Postman REST API Client – Token Generation

POST

https://192.168.1.208/api/fmc_platform/v1/auth/generatetoken

Send

Save

Params | Authorization | Headers (9) | Body | Pre-request Script | Tests | Settings | Cookies | Code

TYPE

Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about authorization

Preview Request

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables

Username: admin

Password: ••••••••

Show Password

Sending request...

Cancel

- Generate Token for REST API Session
- An FMC user that is not being used is recommended.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Postman REST API Client - Token Generation (Cont.)

Body | Cookies | Headers (16) | Test Results | Status: 204 No Content | Time:

KEY	VALUE
Date	Thu, 12 Mar 2020 17:11:01 GMT
Server	Apache
Cache-Control	no-cache, no-store, must-revalidate, max-age=0
Accept-Ranges	bytes
Vary	Accept-Charset,Accept-Encoding,Accept-Language,Accept
X-auth-access-token	242504ad-034f-4a26-b777-44638d4110e3
X-auth-refresh-token	6a5102ef-0798-4482-a8fd-80f17997a4fd

- Headers will contain the X-auth-access-token, this will be good for 30mins, and the X-auth-refresh-token for up to 3 refreshes.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Postman REST API Client – Using Token

Untitled Request

Comments (0)

GET

https://192.168.1.208/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/applicationcategories

Send

Save

Params

Authorization

Headers (9)

Body

TYPE

No Auth

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Headers

6 hidden

	KEY	VALUE
☑	X-auth-access-token	55b39460-f170-4bce-bf9a-25236142613b
☑	X-auth-refresh-token	8be68c6b-7040-4414-82e6-f697e8896117
	Key	Value

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Postman REST API Client - Using Token (Cont.)

Body | Cookies | Headers (11) | Test Results | Status: 200 OK | Time: 1694ms | Size: 1.11 KB | Save Response

Pretty | Raw | Preview | Visualize | JSON


```
1  {
2      "links": {
3          "self": "https://192.168.1.208/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/applicationcategories?offset=0&limit=25"
4      },
5      "items": [
6          {
7              "name": "Active Directory",
8              "id": "123",
9              "links": {
10                 "self": "https://192.168.1.208/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/applicationcategories/123"
11             },
12             "type": "ApplicationCategory",
13             "links": {
14                 "self": "https://192.168.1.208/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/applicationcategories/123"
15             }
16         },
17         {
18             "name": "ad portal",
19             "id": "118",
20             "links": {
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Python REST API Client – Token Generation

```python
import requests
from requests.auth import HTTPBasicAuth

FMC_HOST = 'myfmc.domain.com'
USERNAME = 'admin'
PASSWORD = 'mypassword'

basic = HTTPBasicAuth(USERNAME, PASSWORD)

payload={}

headers = {
'Content-Type': 'application/json'
}

url = f"https://{FMC_HOST}/api/fmc_platform/v1/auth/generatetoken"

response = requests.request("POST", url, headers=headers, data=payload, auth=basic, verify=False)

if response.status_code == 204:
    print()
    print("Token")
    print(response.headers["X-auth-access-token"])
    print()
    print("Refresh Token")
    print(response.headers["X-auth-refresh-token"])
    print()
    print("Domain UUID")
    print(response.headers["DOMAIN_UUID"])
    print()
else:
    print(f'Error: {response.status_code} - {response.text}')
```

Required to bypass Self Signed Cert security(Not Production)

```
/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages/urllib3/connectionpool.py:1013:
InsecureRequestWarning: Unverified HTTPS request is being made to host myfmc.domain.com'. Adding certificate verification is strongly advised. See: https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
  warnings.warn(
```

Token
a7f05c85-3de0-4e1c-942a-4a23a1573447

Refresh Token
031e661b-f211-4ad6-8ed6-e59947f976a2

Domain UUID
e276abec-e0f2-11e3-8169-6d9ed49b625f

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Python REST API Client – Using Token (Cont.)

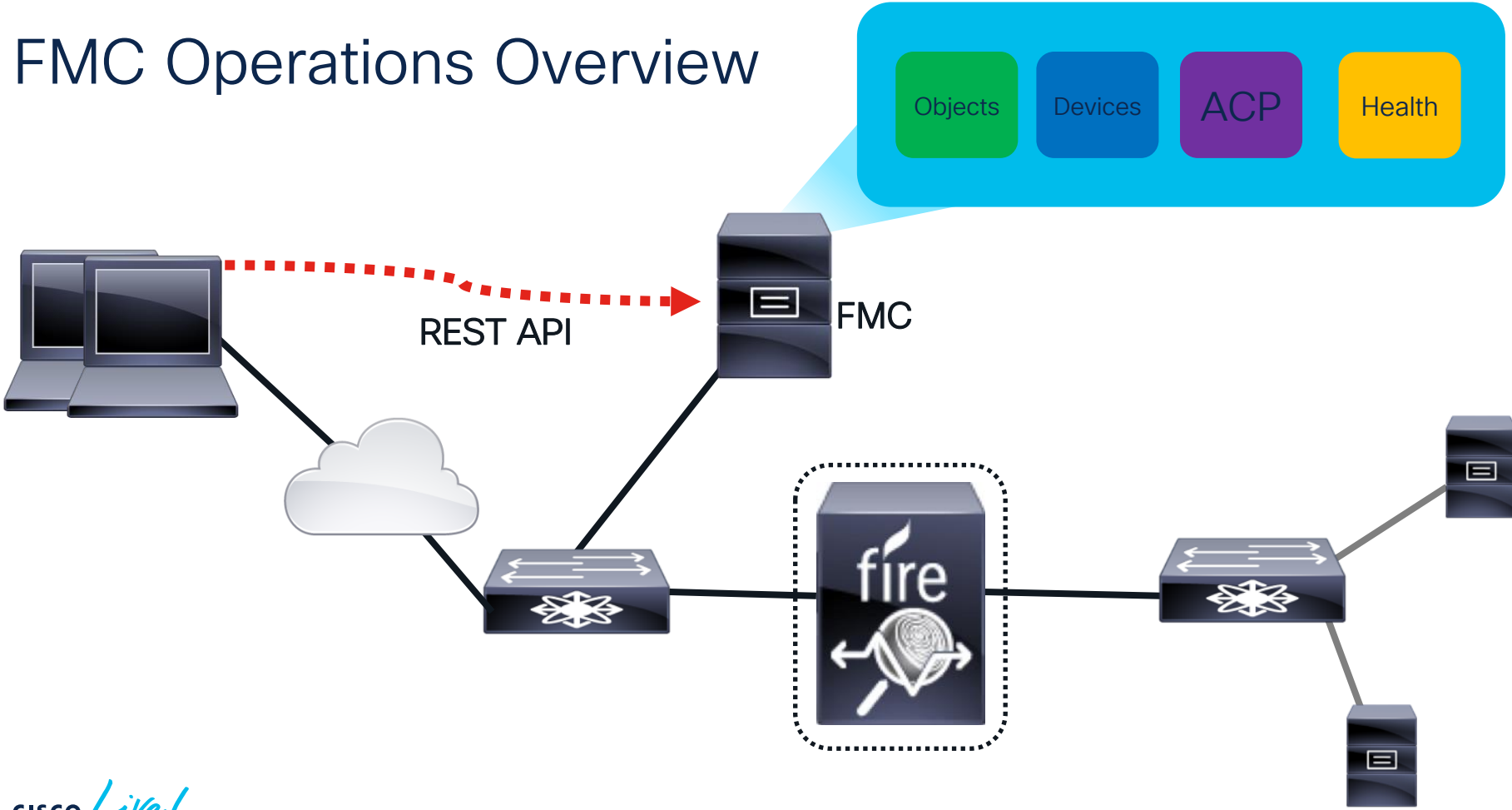
```python
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = '2c6d8b0d-1bed-4cae-b82a-26ed7eb17e44'
refresh_token = 'ced91e88-8b5a-41d6-9c30-26d301aca8eb!'

payload={}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_platform/v1/info/domain"

response = requests.request("GET", url, headers=headers, data=payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```

```
{
    "links": {
        "self":
"https://myfmc.domain.com/api/fmc_platform/v1/info/domain?offset=0&limit=25"
    },
    "items": [
        {
            "uuid": "e276abec-e0f2-11e3-8169-6d9ed49b625f",
            "name": "Global",
            "type": "Domain"
        }
    ],
    "paging": {
        "offset": 0,
        "limit": 25,
        "count": 1,
        "pages": 1
    }
}
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

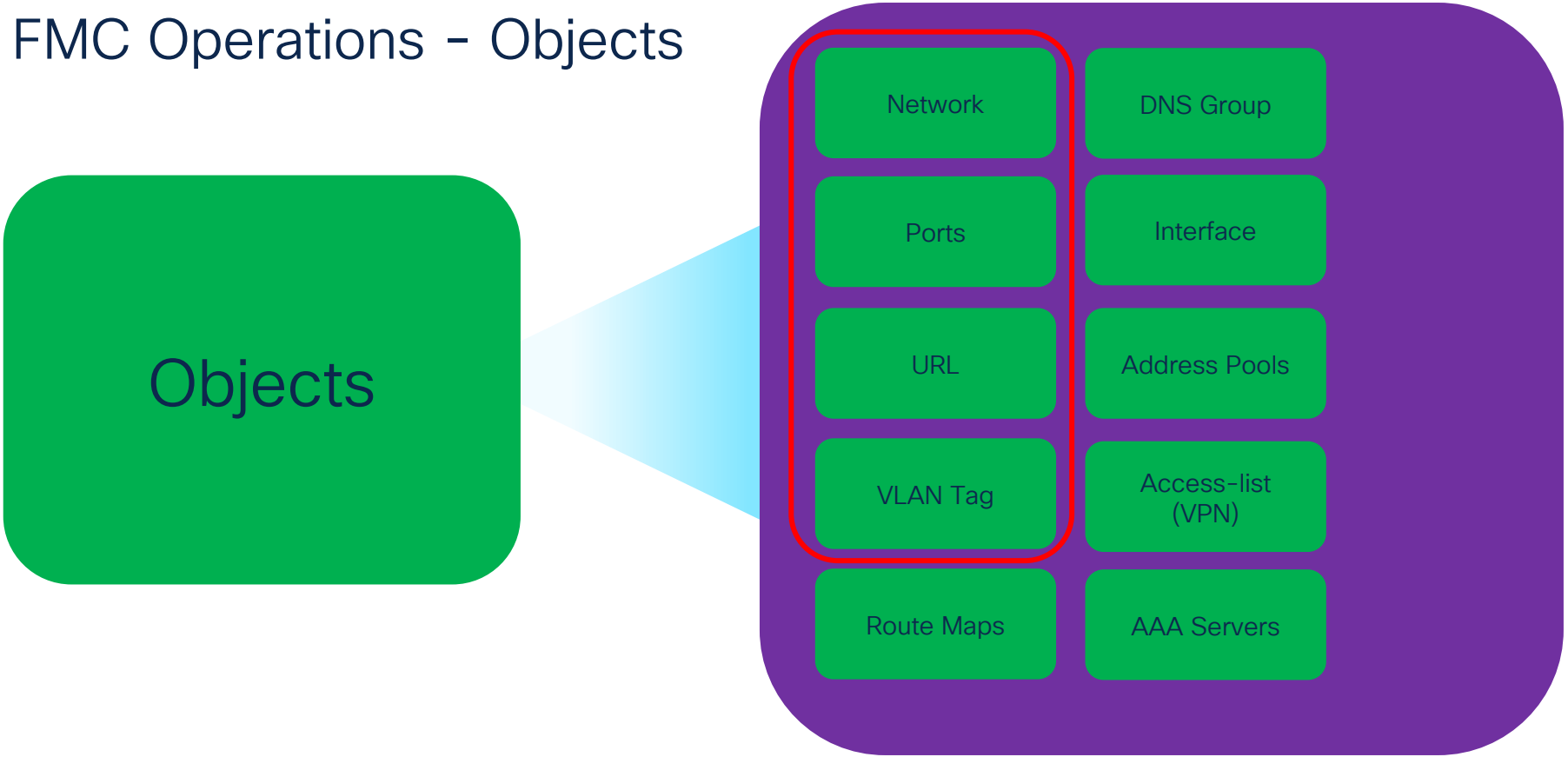
FMC Operations Using API’s

FMC Operations Overview



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

FMC Operations - Objects



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

FMC Operations – Output Objects with Python

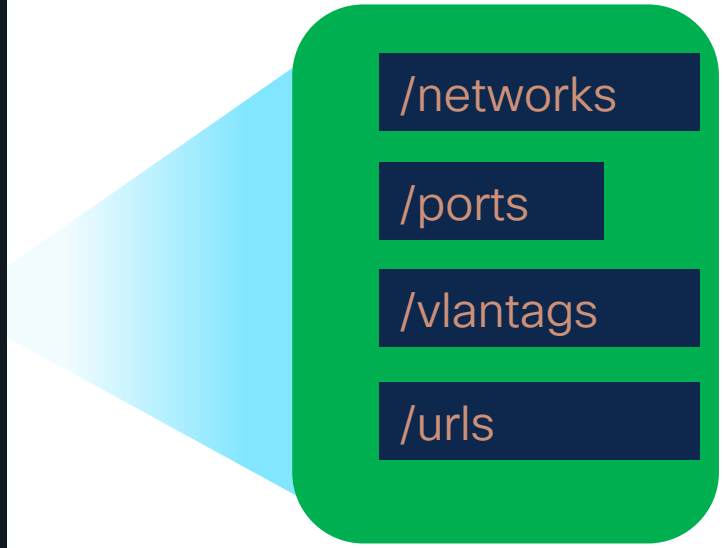
```
/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages/urllib3/connectionpool.py:1013: InsecureRequestWarning: Unverified HTTPS request is being made to host myfmc.domain.com'. Adding certificate verification is strongly advised. See: https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
  warnings.warn(

Token
a7f05c85-3de0-4e1c-942a-4a23a1573447

Refresh Token
031e661b-f211-4ad6-8ed6-[illegible]59947f976a2

Domain UUID
e276abec-e0f2-11e3-8169-6d9ed49b625f
```

```
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'

payload={}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/object/networks"

response = requests.request("GET", url, headers=headers, data=payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

FMC Operations – Output Objects with Python (Cont.)

```
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'

payload={}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url =
f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/objects/networks"

response = requests.request("GET", url, headers=headers, data=payload,
verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)\
```



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

FMC Operations – Output Objects with Python (Cont.)


```
{
   "links": {
      "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/networks?offset=0&limit=25"
   },
   "items": [
{
         "links": {
            "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/networks/95916354-5aa1-4057-8eea-b42a5a207abc",
            "parent": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/networkaddresses"
         },
         "type": "Network",
         "name": "IPv4-Private-10.0.0.0-8",
         "id": "95916354-5aa1-4057-8eea-b42a5a207abc"
      },
      {
         "links": {
            "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/networks/b7a78a7d-20c5-47b2-b02f-86b4360112ac",
            "parent": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/networkaddresses"
         },
         "type": "Network",
         "name": "IPv4-Private-172.16.0.0-12",
         "id": "b7a78a7d-20c5-47b2-b02f-86b4360112ac"
      },
      {
         "links": {
            "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/networks/1dcefdd8-07f7-438a-9221-97d63710614e",
            "parent": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/networkaddresses"
         },
         "type": "Network",
         "name": "IPv4-Private-192.168.0.0-16",
         "id": "1dcefdd8-07f7-438a-9221-97d63710614e"
      },<omitted...>
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

FMC Operations – Output Objects with Python (Cont.)

```
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'

object_id = '95916354-5aa1-4057-8eea-b42a5a207abc'

payload={}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/object/networks/{object_id}"

response = requests.request("GET", url, headers=headers, data=payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

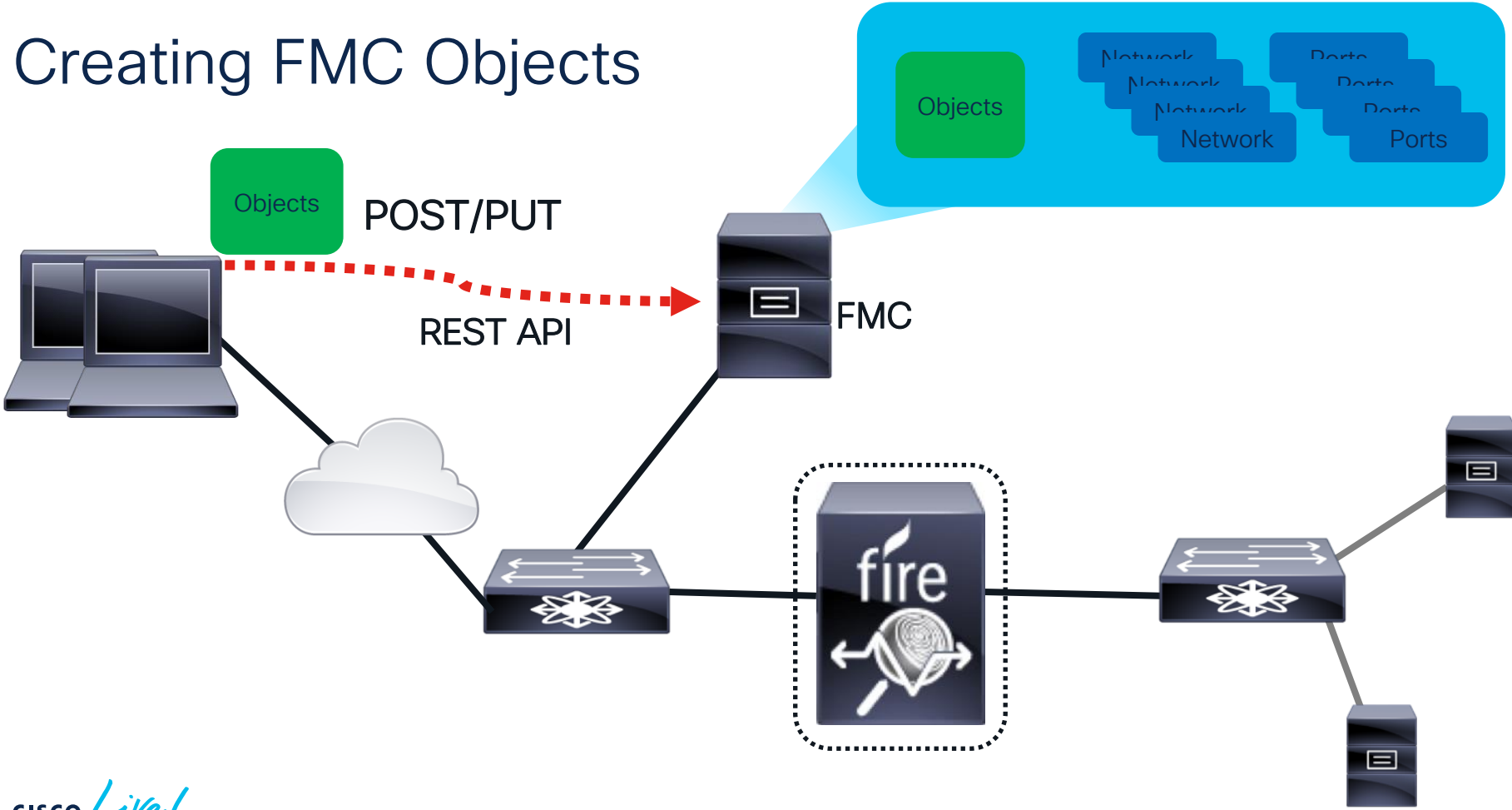
FMC Operations – Output Objects with Python (Cont.)


```
{
    "links": {
        "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/networks/95916354-5aa1-4057-8eea-b42a5a207abc",
        "parent": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/networkaddresses"
    },
    "type": "Network",
    "value": "10.0.0.0/8",
    "overridable": false,
    "description": " ",
    "name": "IPv4-Private-10.0.0.0-8",
    "id": "95916354-5aa1-4057-8eea-b42a5a207abc",
    "metadata": {
        "readOnly": {
            "state": true,
            "reason": "SYSTEM"
        },
        "wildcard": false,
        "timestamp": 1633315418693,
        "lastUser": {
            "name": "admin"
        },
        "domain": {
            "name": "Global",
            "id": "e276abec-e0f2-11e3-8169-6d9ed49b625f",
            "type": "Domain"
        },
        "ipType": "V_4",
        "parentType": "NetworkAddress"
    }
}
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

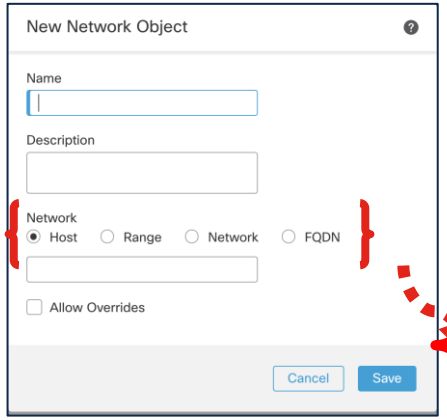
Creating FMC Objects

Creating FMC Objects



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Network Objects Types



Network

/networks

/fqdns

/hosts

/ranges

https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/object/{OBJECT}

POST/PUT

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Object Creation Options

Single Data: Only one object entry

- FMC does not accept the name of the object to be greater than 64 characters.
- Object name should not have space at the beginning of the object name and semicolon at the end.
- The payload cannot contain more than 1 entry in a single Push

Bulk Data: Many new objects at a time

- FMC does not accept the name of the object to be greater than 64 characters.
- Object name should not have space at the beginning of the object name and semicolon at the end.
- The payload cannot contain more than 1,000 entries in a single Bulk Push.
- Payload size cannot be more than 2MB in a single Bulk Push.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Object Schema

POST `/api/fmc_config/v1/domain/{domainUUID}/object/networks`

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

Parameters

Name	Description
bulk `boolean` *(query)*	Enables bulk create for network objects. --
domainUUID * required `string` *(path)*	Domain UUID e276abec-e0f2-11e3-8169-6d9ed49b625f

Request body required — application

The input network object model.

Examples:

Example 1 : POST /fmc_config/v1/domain/DomainUUID/object/networks?bulk=true (Bulk POST operation on network objects.)

Example Value | Schema

```
[
  {
    "name": "net1",
    "value": "1.0.0.0/24",
```

Cisco Firewall Management Center Open API Specification 1.0.0 OAS3

/fmc_oas3.json

Specifies the REST URLs and methods supported in the Cisco Firewall Management Center API. Refer to the version specific REST API Quick Start Guide for additional information.

Cisco Technical Assistance Center (TAC) - Website
Send email to Cisco Technical Assistance Center (TAC)
Cisco Firewall Management Center Licensing

Domains
Global

- Troubleshoot
- Network Map
- Devices
- Policy Assignments
- Device HA Pairs
- Health
- Chassis
- Updates
- Users
- Intelligence
- License
- Search
- Audit
- Integration
- Device Groups
- Status
- Device Clusters
- System Information
- Object
- Policy
- Deployment

Schemas

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Object Schema (Cont.)

NetworkObject ⌄ {

description:	A model representing Network object definition.
metadata	**Metadata** > {...}
overridable	boolean Boolean indicating whether object values can be overridden.
name	string User assigned resource name.
description	string User provided resource description.
links	**Links** > {...}
overrides	**IOverride** > {...}
id	string Unique identifier representing response object.
type	string Type associated with resource: NetworkObject.
value	string Actual value of the network.
version	string Version number of the response object.
overrideTargetId	string Unique identifier of domain or device when override assigned to child domain. Used as path parameter to GET override details for a specific object on a specific target (device or domain).

}

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Object Schema (Cont.)

Example Value | **Schema**

NetworkObject ⌄ {

`description:`

A model representing Network object definition.

`metadata` **Metadata** ⌄ {

`description:`

Defines read only details about the object - whether it is system defined, last user who modified the object etc.

`lastUser` **MetadataUser** > {...}

`domain` **Domain** > {...}

`readOnly` **ReadOnly** > {...}

`timestamp` `integer($int32)`

The last updated timestamp.

}

`overridable` `boolean`

Boolean indicating whether object values can be overridden.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Single Object Creation

```
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'

payload={
'name': 'Data_Center_Network_1',
'value': '192.168.67.0/24',
'description': '192.168.67.0/24 Data Center Network'
}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/obje

response = requests.request("POST", url, headers=headers, json=payload,

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```

NetworkObject ⌄ {

description:	A model representing Network object definition.
metadata	**Metadata** › {...}
overridable	boolean Boolean indicating whether object values can be overridden.
name	string User assigned resource name.
description	string User provided resource description.
links	**Links** › {...}
overrides	**IOverride** › {...}
id	string Unique identifier representing response object.
type	string Type associated with resource: NetworkObject.
value	string Actual value of the network.
version	string Version number of the response object.
overrideTargetId	string Unique identifier of domain or device when override assigned to child domain. Used as path parameter to GET override details for a specific object on a specific target (device or domain).

}

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Single Object Creation

```
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'

payload={
'name': 'Data_Center_Network_1',
'value': '192.168.67.0/24',
'description': '192.168.67.0/24 Data Center Network'
}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/object/networks"

response = requests.request("POST", url, headers=headers, json=payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```

data=payload

```
warnings.warn(
{
    "error": {
        "category": "OTHER",
        "messages": [
            {
                "description": "Unprocessable Entity"
            }
        ],
        "severity": "ERROR"
    }
}
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Single Object Creation Verification


```
{
    "links": {
        "self": "https://myfmc.domain.com/api/fmc_config/v1/dom
6d9ed49b625f/object/networks/000C2908-741C-0ed3-0000-
        "parent": "https://myfmc.domain.com/api/fmc_config/v1/d
6d9ed49b625f/object/networkaddresses"
    },
    "type": "Network",
    "value": "192.168.67.0/24",
    "overridable": false,
    "description": "192.168.67.0/24 Data Center Network",
    "id": "000C2908-741C-0ed3-0000-197568495776",
    "name": "Data_Center_Network_1",
    "metadata": {
        "wildcard": false,
        "timestamp": 0,
        "lastUser": {
            "name": "admin"
        },
        "domain": {
            "name": "Global",
            "id": "e276abec-e0f2-11e3-8169-6d9ed49b625f",
            "type": "Domain"
        },
        "ipType": "V_4",
        "parentType": "NetworkAddress"
    }
}
```


Name	Value	Type	Override
any	0.0.0.0/0 ::/0	Group	
any-ipv4	0.0.0.0/0	Network	
any-ipv6	::/0	Host	
Data_Center_Network_1	192.168.67.0/24	Network	
DNS-Master	192.168.1.26	Host	
GRE-Router1	192.168.40.40	Host	
GRE-Router2	192.168.40.70	Host	
IPv4-Benchmark-Tests	198.18.0.0/15	Network	
IPv4-Link-Local	169.254.0.0/16	Network	
IPv4-Multicast	224.0.0.0/4	Network	
IPv4-Private-10.0.0.0-8	10.0.0.0/8	Network	

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Single Object Update

```
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'

obj_id = '000C2908-741C-0ed3-0000-197568495776'

payload={
'name': 'Data_Center_Network_0',
'value': '192.168.70.0/24'
'id': f'{obj_id}'
}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/object/networks/{obj_id}"

response = requests.request("PUT", url, headers=headers, json=payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```

Caveats: The name and the id are required in the payload.
(Regardless if you are changing the name)
The id is required in the PUT request.

Single Object Update Verification


```
{
    "links": {
        "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-
e0f2-11e3-8169-6d9ed49b625f/object/networks/000C2908-741C-0ed3-0000-
197568495776",
        "parent": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-
e0f2-11e3-8169-6d9ed49b625f/object/networkaddresses"
    },
    "type": "Network",
    "value": "192.168.67.0/24",
    "overridable": false,
    "description": "192.168.67.0/24 Data Center Network",
    "id": "000C2908-741C-0ed3-0000-197568495776",
    "name": "Data_Center_Network_1",
    "metadata": {
        "wildcard": false,
        "timestamp": 0,
        "lastUser": {
            "name": "admin"
        },
        "domain": {
            "name": "Global",
            "id": "e276abec-e0f2-11e3-8169-6d9ed49b625f",
            "type": "Domain"
        },
        "ipType": "V_4",
        "parentType": "NetworkAddress"
    }
}
```


Before


```
{
    "links": {
        "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-
8169-6d9ed49b625f/object/networks/000C2908-741C-0ed3-0000-197568495814",
        "parent": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-
8169-6d9ed49b625f/object/networkaddresses"
    },
    "type": "Network",
    "value": "192.168.70.0/24",
    "overridable": false,
    "id": "000C2908-741C-0ed3-0000-197568495814",
    "name": "Data_Center_Network_0",
    "metadata": {
        "wildcard": false,
        "timestamp": 0,
        "lastUser": {
            "name": "admin"
        },
        "domain": {
            "name": "Global",
            "id": "e276abec-e0f2-11e3-8169-6d9ed49b625f",
            "type": "Domain"
        },
        "ipType": "V_4",
        "parentType": "NetworkAddress"
    }
}
```


After

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Single Object Update Creation Verification (Cont.)

Name	Value	Type	Override	
any	0.0.0.0/0 ::/0	Group		
any-ipv4	0.0.0.0/0	Network		
any-ipv6	::/0	Host		
Data_Center_Network_0	192.168.70.0/24	Network		

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Single Object Removal

```
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'

obj_id = '000C2908-741C-0ed3-0000-197568495814'

payload={
}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/object/networks/{obj_id}"

response = requests.request("DELETE", url, headers=headers, json=payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```

Caveats: Only the object **id** is required; no payload is needed.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Single Object Removal Verification


```
{
    "links": {
        "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/networks/000C2908-741C-0ed3-0000-197568495814",
        "parent": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/networkaddresses"
    },
    "type": "Network",
    "value": "192.168.70.0/24",
    "overridable": false,
    "description": " ",
    "id": "000C2908-741C-0ed3-0000-197568495814",
    "name": "Data_Center_Network_0",
    "metadata": {
        "wildcard": false,
        "timestamp": 1677181653883,
        "lastUser": {
            "name": "admin"
        },
        "domain": {
            "name": "Global",
            "id": "e276abec-e0f2-11e3-8169-6d9ed49b625f",
            "type": "Domain"
        },
        "ipType": "V_4",
        "parentType": "NetworkAddress"
    }
}
```


Name	Value	Type	Override
any	0.0.0.0/0 ::/0	Group	
any-ipv4	0.0.0.0/0	Network	
any-ipv6	::/0	Host	
DNS-Master	192.168.1.26	Host	
GRE-Router1	192.168.40.40	Host	
GRE-Router2	192.168.40.70	Host	
IPv4-Benchmark-Tests	198.18.0.0/15	Network	
IPv4-Link-Local	169.254.0.0/16	Network	

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Bulk Object Creation

List with dictionary elements

```
payload=[

{"name": "Data_Center_0",
"value": "192.168.69.0/24",
"overridable": 'false',
"description": "Network obj 1",
"type": "Network"
},

{"name": "Data_Center_1",
"value": "192.168.70.0/24",
"overridable": 'false',
"description": "Network obj 2",
"type": "Network"
}
]
```

Bulk option required in the URL

?bulk=true

```
FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'

payload=[

{"name": "Data_Center_0",
"value": "192.168.69.0/24",
"overridable": 'false',
"description": "Network obj 1",
"type": "Network"
},

{"name": "Data_Center_1",
"value": "192.168.70.0/24",
"overridable": 'false',
"description": "Network obj 2",
"type": "Network"
}
]

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/object/networks?bulk=true"

response = requests.request("POST", url, headers=headers, json=payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Bulk Object Creation Verification


```
{
"links": {
            "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/networks/000C2908-741C-0ed3-0000-197568495981",
            "parent": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/networkaddresses"
        },
        "type": "Network",
        "value": "192.168.69.0/24",
        "overridable": false,
        "description": "Network obj 1",
        "id": "000C2908-741C-0ed3-0000-197568495981",
        "name": "Data_Center_0",
        "metadata": {
            "wildcard": false,
            "timestamp": 0,
            "lastUser": {
                "name": "admin"
            },
            "domain": {
                "name": "Global",
                "id": "e276abec-e0f2-11e3-8169-6d9ed49b625f",
                "type": "Domain"
            },
            "ipType": "V_4",
            "parentType": "NetworkAddress"
        }
    },

}
```

```
{
        "links": {
            "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/networks/000C2908-741C-0ed3-0000-197568495982",
            "parent": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/object/networkaddresses"
        },
        "type": "Network",
        "value": "192.168.70.0/24",
        "overridable": false,
        "description": "Network obj 2",
        "id": "000C2908-741C-0ed3-0000-197568495982",
        "name": "Data_Center_1",
        "metadata": {
            "wildcard": false,
            "timestamp": 0,
            "lastUser": {
                "name": "admin"
            },
            "domain": {
                "name": "Global",
                "id": "e276abec-e0f2-11e3-8169-6d9ed49b625f",
                "type": "Domain"
            },
            "ipType": "V_4",
            "parentType": "NetworkAddress"
        }
    }

]
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Bulk Object Creation using CSV file

```
name,description,type,value
Data_Center_0,Data Center 0,Network,192.168.61.0/24
Data_Center_1,Data Center 1,Network,192.168.62.0/24
Data_Center_2,Data Center 2,Network,192.168.63.0/24
Data_Center_3,Data Center 3,Network,192.168.64.0/24
```

- Network
- Host
- FQDN
- Range

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Bulk Object Creation using CSV file (Cont.)

```
CSV_File_Path = "/Users/tom/Desktop/FMC Python Code/Network_Objects.csv"

host = []

with open(CSV_File_Path, encoding='utf-8-sig') as csv_file:
    csvReader = csv.DictReader(csv_file)
    for rows in csvReader:
        if rows['type'] == "Network":
            host.append(rows)

host_payload = json.dumps(host)

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/object/networks?bulk=true"

response = requests.request("POST", url, headers=headers, data=host_payload, verify=False)
```

```
json=host_payload
```

Caveats: Since we are outputting it to json using the json.dumps(), the json= parameter will not work, since the object is already in json format. This will now require the use of the data= parameter.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Bulk Object Creation using CSV file (Cont.)

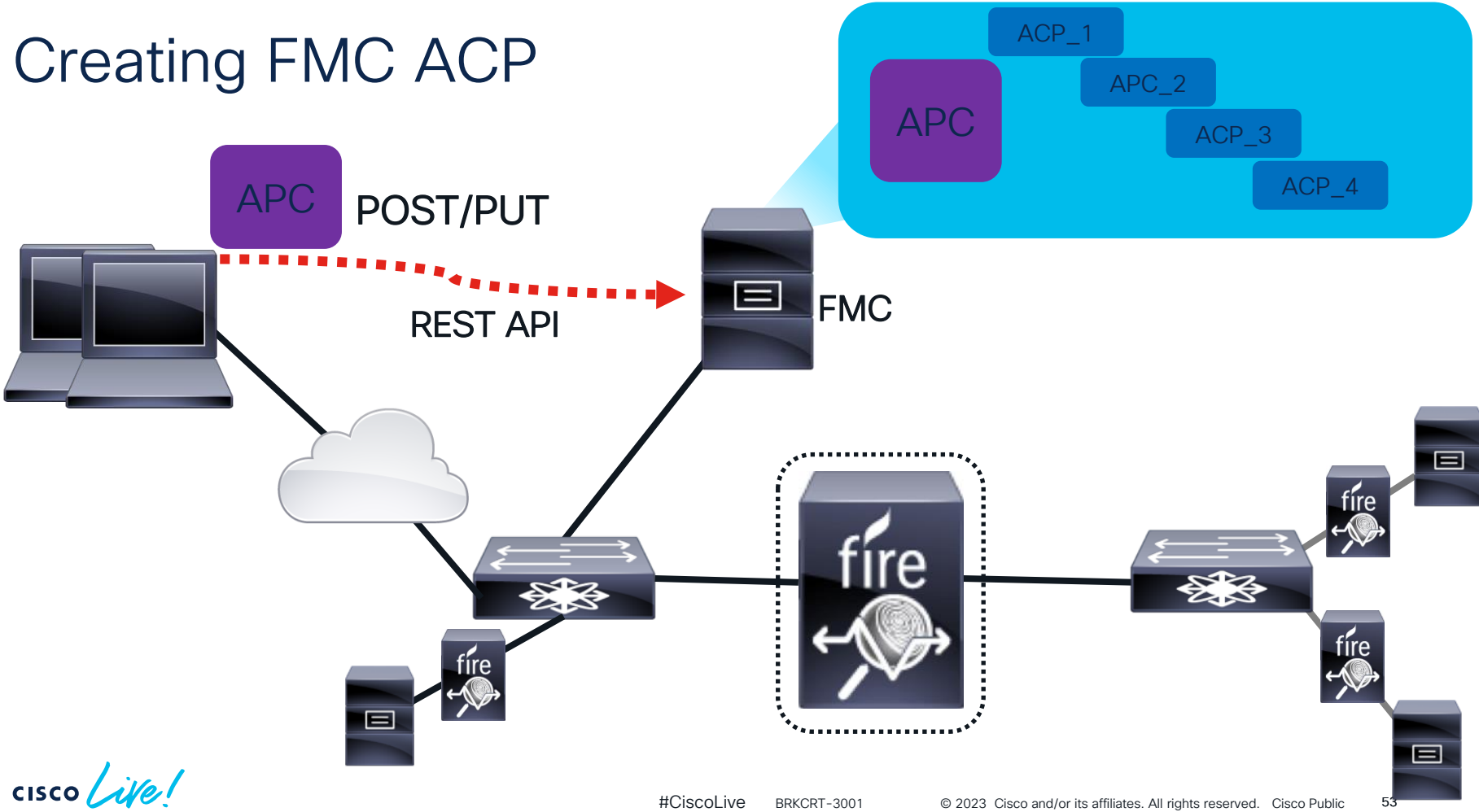
- CSV files provides a better way to store your objects and re-use them.
- The code is cleaner and much less painful to change.

```
import requests
import json
import csv

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'

CSV_File_Path = "/Users/rleivaoc/Desktop/FMC Python Code/Network_Objects.csv"

host = []

with open(CSV_File_Path, encoding='utf-8-sig') as csv_file:
    csvReader = csv.DictReader(csv_file)
    for rows in csvReader:
        if rows['type'] == "Network":
            host.append(rows)

host_payload = json.dumps(host)

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/object/networks?bulk=true"

response = requests.request("POST", url, headers=headers, data=host_payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Creating FMC Access Control Policy

Creating FMC ACP



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Access Control Policy

https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/policy/accesspolicies

Access Control Policy	Domain
APC_Main	Global
New_Policy-ACP	Global
Sub-ACP-Main	Global

BRKCRT-3001

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Access Control Policy (Cont.)

```
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'

payload={
}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/policy/accesspolicies"

response = requests.request("GET", url, headers=headers, json=payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```

?expanded=true

Displays a list of objects with additional attributes

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Access Control Policy (Cont.)


```
"items": [
    {
        "type": "AccessPolicy",
        "links": {
            "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-
11e3-8169-6d9ed49b625f/policy/accesspolicies/000C2908-741C-0ed3-0000-
004294967299"
        },
        "name": "APC_Main",
        "id": "000C2908-741C-0ed3-0000-004294967299"
    },
    {
        "type": "AccessPolicy",
        "links": {
            "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-
11e3-8169-6d9ed49b625f/policy/accesspolicies/000C2908-741C-0ed3-0000-
154618824509"
        },
        "name": "New_Policy-ACP",
        "id": "000C2908-741C-0ed3-0000-154618824509"
    },
    {
        "type": "AccessPolicy",
        "links": {
            "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-
11e3-8169-6d9ed49b625f/policy/accesspolicies/000C2908-741C-0ed3-0000-
154618823282"
        },
        "name": "Sub-ACP-Main",
        "id": "000C2908-741C-0ed3-0000-154618823282"
    }
  ],
}
```


Access Control Policy	Domain
APC_Main	Global
New_Policy-ACP	Global
Sub-ACP-Main	Global

"id": "000C2908-741C-0ed3-0000-004294967299"

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Access Control Policy (Cont.)

```
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'
acp_UUID = '000C2908-741C-0ed3-0000-004294967299'

payload={
}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/policy/accesspolicies/{acp_UUID}"

response = requests.request("GET", url, headers=headers, json=payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Access Control Policy (Cont.)


```
{
    "metadata": {
        "inherit": false,
        "lockingStatus": {
            "status": "UNLOCKED"
        },
        "domain": {
            "name": "Global",
            "id": "e276abec-e0f2-11e3-8169-6d9ed49b625f",
            "type": "Domain"
        }
    },
    "type": "AccessPolicy",
    "links": {
        "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-
8169-6d9ed49b625f/policy/accesspolicies/000C2908-741C-0ed3-0000-004294967299"
    },
    "rules": {
        "refType": "list",
        "type": "AccessRule",
        "links": {
            "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-
8169-6d9ed49b625f/policy/accesspolicies/000C2908-741C-0ed3-0000-
004294967299/accessrules"
        }
    },
    "securityIntelligence": {
        "id": "000C2908-741C-0ed3-0000-004294967343",
        "type": "SecurityIntelligencePolicy",
        "links": {
            "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-
8169-6d9ed49b625f/policy/accesspolicies/000C2908-741C-0ed3-0000-
004294967299/securityintelligencepolicies/000C2908-741C-0ed3-0000-004294967343"
```

```
    }
    },
    "name": "APC_Main",
    "defaultAction": {
        "action": "PERMIT",
        "type": "AccessPolicyDefaultAction",
        "enableSyslog": false,
        "intrusionPolicy": {
            "name": "Balanced Security and Connectivity",
            "id": "6c66b83c-bc23-55b6-879d-c4d847443503",
            "type": "IntrusionPolicy",
            "inspectionMode": "DETECTION"
        },
        "logBegin": false,
        "logEnd": false,
        "variableSet": {
            "name": "Default-Set",
            "id": "76fa83ea-c972-11e2-8be8-8e45bb1343c0",
            "type": "VariableSet"
        },
        "sendEventsToFMC": false,
        "id": "000C2908-741C-0ed3-0000-000268434432"
    },
    "prefilterPolicySetting": {
        "name": "Default Prefilter Policy",
        "id": "4897c8f4-e211-4661-b0a4-25b0826cded9",
        "type": "PrefilterPolicy"
    },
    "id": "000C2908-741C-0ed3-0000-004294967299"
}
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Access Control Policy – Display Rules in ACP

```
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'
acp_UUID = '000C2908-741C-0ed3-0000-004294967299'

payload={
}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/policy/accesspolicies/{acp_UUID}/accessrules"

response = requests.request("GET", url, headers=headers, json=payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Access Control Policy – Display Rules in ACP (Cont.)


```
{
    "links": {
        "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-
e0f2-11e3-8169-6d9ed49b625f/policy/accesspolicies/000C2908-741C-
0ed3-0000-004294967299/accessrules?offset=0&limit=25"
    },
    "items": [
        {
            "links": {
                "self":
"https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-
8169-6d9ed49b625f/policy/accesspolicies/000C2908-741C-0ed3-0000-
004294967299/accessrules/000C2908-741C-0ed3-0000-000268439553"
            },
            "name": "Server",
            "type": "AccessRule",
            "id": "000C2908-741C-0ed3-0000-000268439553"
        },
        {
            "links": {
                "self":
"https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-
8169-6d9ed49b625f/policy/accesspolicies/000C2908-741C-0ed3-0000-
004294967299/accessrules/000C2908-741C-0ed3-0000-000268439554"
            },
            "name": "FTP",
            "type": "AccessRule",
            "id": "000C2908-741C-0ed3-0000-000268439554"
        },
        {
```

```
{
            "links": {
                "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-
e0f2-11e3-8169-6d9ed49b625f/policy/accesspolicies/000C2908-741C-0ed3-
0000-004294967299/accessrules/000C2908-741C-0ed3-0000-000268439555"
            },
            "name": "SIP",
            "type": "AccessRule",
            "id": "000C2908-741C-0ed3-0000-000268439555"
        },
        {
            "links": {
                "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-
e0f2-11e3-8169-6d9ed49b625f/policy/accesspolicies/000C2908-741C-0ed3-
0000-004294967299/accessrules/000C2908-741C-0ed3-0000-000268439557"
            },
            "name": "Allow_Access",
            "type": "AccessRule",
            "id": "000C2908-741C-0ed3-0000-000268439557"
        },
        {
            "links": {
                "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-
e0f2-11e3-8169-6d9ed49b625f/policy/accesspolicies/000C2908-741C-0ed3-
0000-004294967299/accessrules/000C2908-741C-0ed3-0000-000268439558"
            },
            "name": "test",
            "type": "AccessRule",
            "id": "000C2908-741C-0ed3-0000-000268439558"
        }
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Access Control Policy – Display Rules in ACP (Cont.)

```
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'
acp_UUID = '000C2908-741C-0ed3-0000-004294967299'
rule_id = '000C2908-741C-0ed3-0000-000268439553'

payload={
}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/policy/accesspolicies/{acp_UUID}/accessrules/{rule_id}"

response = requests.request("GET", url, headers=headers, json=payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Access Control Policy – Display Rule Info


```
"metadata": {
    "section": "Mandatory",
    "category": "Services",
    "accessPolicy": {
        "type": "AccessPolicy",
        "name": "APC_Main",
        "defaultAction": {
            "type": "AccessPolicyDefaultAction",
            "enableSyslog": false,
            "logBegin": false,
            "logEnd": false,
            "sendEventsToFMC": false
        },
        "id": "000C2908-741C-0ed3-0000-004294967299"
    },
    "timestamp": 1666902565563,
    "domain": {
        "name": "Global",
        "id": "e276abec-e0f2-11e3-8169-6d9ed49b625f",
        "type": "Domain"
    }
},
```

```
"links": {
    "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/policy/accesspolicies/000C2908-741C-0ed3-0000-004294967299/accessrules/000C2908-741C-0ed3-0000-000268439553"
},
    ""action": "ALLOW",
    "name": "Server",
    "type": "AccessRule",
    "enabled": true,
    "id": "000C2908-741C-0ed3-0000-000268439553",
    "enableSyslog": false,
    "vlanTags": {},
    "applications": {
        "applications": [
            {
                "id": "676",
                "type": "Application",
                "name": "HTTP"
            }
        ]
    },
    "sourceNetworks": {
        "objects": [
            {
                "type": "Network",
                "overridable": false,
                "name": "IPv4-Private-10.0.0.0-8",
                "id": "95916354-5aa1-4057-8eea-b42a5a207abc"
            }
        ]
    },
    "destinationNetworks": {
        "objects": [
            {
                "type": "Network",
                "overridable": false,
                "name": "IPv4-Private-192.168.0.0-16",
                "id": "1dcefdd8-07f7-438a-9221-97d63710614e"
            }
        ]
    },   "logFiles": false,
    "sourceDynamicObjects": {},
    "destinationDynamicObjects": {},
    "sendEventsToFMC": false
}
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Access Control Policy Schema

```
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'

payload={
"type": "AccessPolicy",
"name": "My_New_ACP_Policy",
"defaultAction": {
"action": "BLOCK"
}
}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/policy/ac

response = requests.request("POST", url, headers=headers, json=payload, verif

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```

AccessPolicy ⌄ {

metadata	**Metadata** › {...}
defaultAction*	**IAccessPolicyDefaultAction** › {...}
name	string User-specified name of the access control policy.
identityPolicySetting	**IACPolicyIdentityPolicySettings** › {...}
description	string Description of Access Policy
links	**Links** › {...}
rules	› {...}
securityIntelligence	**IReference** › {...}
prefilterPolicySetting	**IAccessPolicyPrefilterPolicySettingModel** › {...}
id	string Unique identifier (UUID) representing the access control policy.
type	string Type of the access control policy; this value is always AccessPolicy
version	string Version number of the response object.

}

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Create new Access Control Policy

```
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'

payload={
"type": "AccessPolicy",
"name": "My_New_ACP_Policy",
"defaultAction": {
"action": "BLOCK"
}
}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/policy/accesspolicies"

response = requests.request("POST", url, headers=headers, json=payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```

Access Control Policy	Domain
APC_Main	Global
New_Policy-ACP	Global
Sub-ACP-Main	Global
My_New_ACP_Policy	Global

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Create new Access Control Policy (Cont.)


```
{
    "metadata": {
        "inherit": false,
        "lockingStatus": {
            "status": "UNLOCKED"
        }
        "domain": {
            "name": "Global",
            "id": "e276abec-e0f2-11e3-8169-6d9ed49b625f",
            "type": "Domain"
        }
    },
    "type": "AccessPolicy",
    "links": {
        "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-
8169-6d9ed49b625f/policy/accesspolicies/000C2908-741C-0ed3-0000-210453400484"
    },
    "rules": {
        "refType": "list",
        "type": "AccessRule",
        "links": {
            "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-
8169-6d9ed49b625f/policy/accesspolicies/000C2908-741C-0ed3-0000-
210453400484/accessrules"
        }
    },
    "name": "My_New_ACP_Policy",
    "id": "000C2908-741C-0ed3-0000-210453400484"
}
```


Access Control Policy	Domain
APC_Main	Global
New_Policy-ACP	Global
Sub-ACP-Main	Global
My_New_ACP_Policy	Global

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Access Rule Schema

```
AccessRule ⌄ {
  description:
                    Represents Access Rule contained within an Access Policy.

  metadata          AccessRuleMetadata > {...}
  snmpConfig        ISNMPConfig > {...}
  timeRangeObjects  > [...]
  sourceNetworks    INetworkObjectsContainer > {...}
  syslogSeverity    string

                    Specifies the Override Severity if alerts are being sent to default syslog configuration. One of: ALERT | CRIT | DEBUG |
                    EMERG | ERR | INFO | NOTICE | WARNING.

                    Enum:
                    > Array [ 8 ]
  sourceZones       ISourceZoneContainer > {...}
  destinationZones  ISecurityZoneContainer > {...}
  destinationDynamicObjects IDynamicObjectsContainer > {...}
  description       string

                    User provided resource description.

  originalSourceNetworks INetworkObjectsContainer > {...}
  enableSyslog      boolean

                    Boolean indicating whether the alerts associated with the access rule are sent to default syslog configuration in Access
                    Control Logging.

  type              string

                    Type of the response object. This value is always AccessRule.

  safeSearch        ISafeSearch > {...}
  enabled           boolean

                    Boolean indicating whether the access rule is in effect (true) or not (false). Default is true.

  destSGTTypes      > [...]
  syslogConfig      ISyslogConfig > {...}
  urls              IUrlObjectsContainer > {...}
  endPointDeviceTypes > [...]
  destinationNetworks INetworkObjectsContainer > {...}
  youTube           IYouTube > {...}
```

```
                    > Array [ 7 ]
  links             ILinks > {...}
  networkAccessDeviceIPs INetworkObjectsContainer > {...}
  id                string

                    Unique identifier (UUID) for the access rule.

  sourceSecurityGroupTags ISecurityGroupTagContainer > {...}
  destinationSecurityGroupTags ISecurityGroupTagContainer > {...}
  logEnd            boolean

                    Boolean indicating whether the device will log events at the end of the connection. Default is false. If 'MONITOR' action
                    is selected for access rule, logEnd will always be true.

  logBegin          boolean

                    Boolean indicating whether the device will log events at the beginning of the connection. Default is false. If 'MONITOR'
                    action is selected for access rule, logBegin will always be false.

  sendEventsToFMC   boolean

                    Boolean indicating whether the device will send events to the Firepower Management Center event viewer. Default is
                    false. If 'MONITOR' action is selected for access rule, sendEventsToFMC will always be true.

  destinationPorts  IPortObjectsContainer > {...}
  sourcePorts       IPortObjectsContainer > {...}
  ipsPolicy         IIntrusionPolicyModel > {...}
  variableSet       IVariableSet > {...}
  version           string

                    Version number of the response object.

  users             IUsersContainer > {...}
  logFiles          boolean

                    Boolean indicating whether the device will log file events. Default is false.

  commentHistoryList > [...]
  filePolicy        IFilePolicy > {...}
  name              string

                    User-specified name of the access rule.

  sourceDynamicObjects IDynamicObjectsContainer > {...}
  vlanTags          IVLanTagsContainer > {...}
  applications      IApplicationsContainer > {...}
}
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Create New Access Control Policy Rule (Cont.)

```
payload={
"action": "ALLOW",
"enabled": "true",
"type": "AccessRule",
"name": "My_New_Rule",
"sendEventsToFMC": "false",
"logFiles": "false",
"logBegin": "false",
"logEnd": "false",
"sourceNetworks": {
"objects": [
{
"type": "Network",
"overridable": "false",
"name": "IPv4-Private-10.0.0.0-8",
"id": "95916354-5aa1-4057-8eea-b42a5a207abc"
}
]
},
"destinationNetworks": {
"objects": [
{
"type": "Network",
"overridable": "false",
"name": "IPv4-Private-192.168.0.0-16",
"id": "1dcefdd8-07f7-438a-9221-97d63710614e"
}
]
}
}
```

Variables

```
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'
acp_UUID = '000C2908-741C-0ed3-0000-210453400484'
```

URL Format

```
"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/policy/accesspolicies/{acp_UUID}/accessrules"
```

Domain UUID

ACP UUID

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Create New Access Control Policy Rule (Cont.)

```
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'
acp_UUID = '000C2908-741C-0ed3-0000-210453400484'

payload={<ACP_Schema>}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/policy/accesspolicies/{acp_UUID}/accessrules"

response = requests.request("POST", url, headers=headers, json=payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

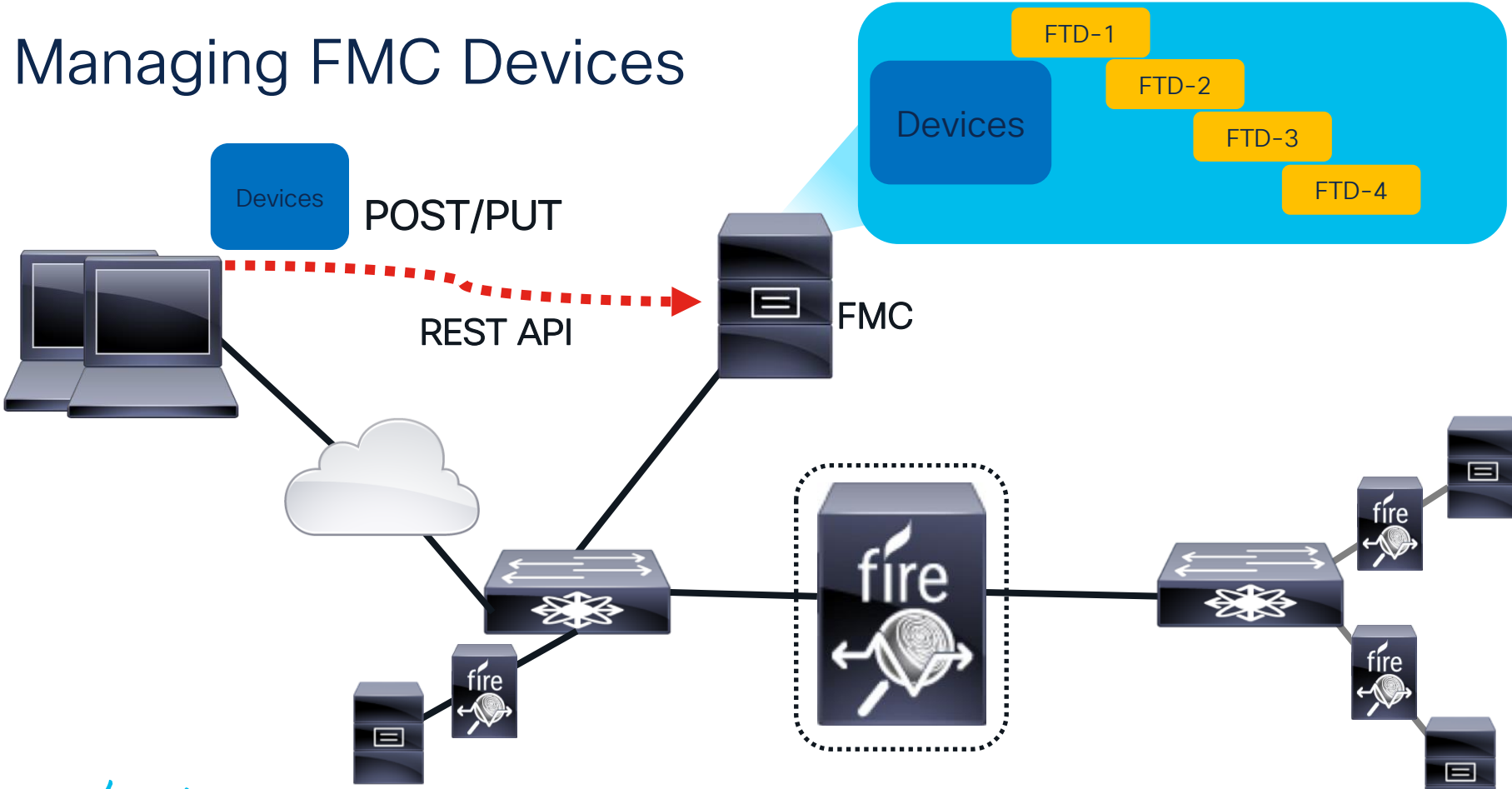
Create New Access Control Policy Rule (Cont.)


```
{
    "metadata": {
        "ruleIndex": 1,
        "section": "Default",
        "category": "--Undefined--",
        "accessPolicy": {
            "type": "AccessPolicy",
            "name": "My_New_ACP_Policy",
            "defaultAction": {
                "type": "AccessPolicyDefaultAction",
                "enableSyslog": false,
                "logBegin": false,
                "logEnd": false,
                "sendEventsToFMC": false
            },
            "id": "000C2908-741C-0ed3-0000-210453400484"
        }
    },
    "links": {
        "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-
11e3-8169-6d9ed49b625f/policy/accesspolicies/000C2908-741C-0ed3-0000-
210453400484/accessrules/000C2908-741C-0ed3-0000-000268441601"
    },
    "action": "ALLOW",
    "name": "My_New_Rule",
    "type": "AccessRule",
    "enabled": true,
    "id": "000C2908-741C-0ed3-0000-000268441601",
    "enableSyslog": false,
    "vlanTags": {},
    "sourceNetworks": {
        "objects": [
            {
                "type": "Network",
                "overridable": false,
                "name": "IPv4-Private-10.0.0.0-8",
                "id": "95916354-5aa1-4057-8eea-b42a5a207abc"
            }
        ]
    },
    "destinationNetworks": {
        "objects": [
            {
                "type": "Network",
                "overridable": false,
                "name": "IPv4-Private-192.168.0.0-16",
                "id": "1dcefdd8-07f7-438a-9221-97d63710614e"
            }
        ]
    },
    "logBegin": false,
    "logEnd": false,
    "variableSet": {
        "name": "Default-Set",
        "id": "76fa83ea-c972-11e2-8be8-8e45bb1343c0",
        "type": "VariableSet"
    },
    "logFiles": false,
    "sourceDynamicObjects": {},
    "destinationDynamicObjects": {},
    "sendEventsToFMC": false
}
```


 © 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Managing the FirePower Device Configuration

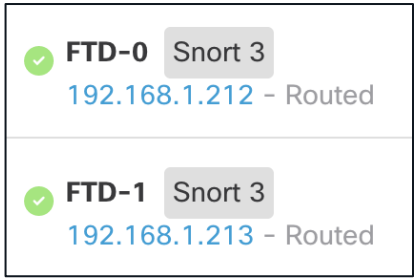
Managing FMC Devices

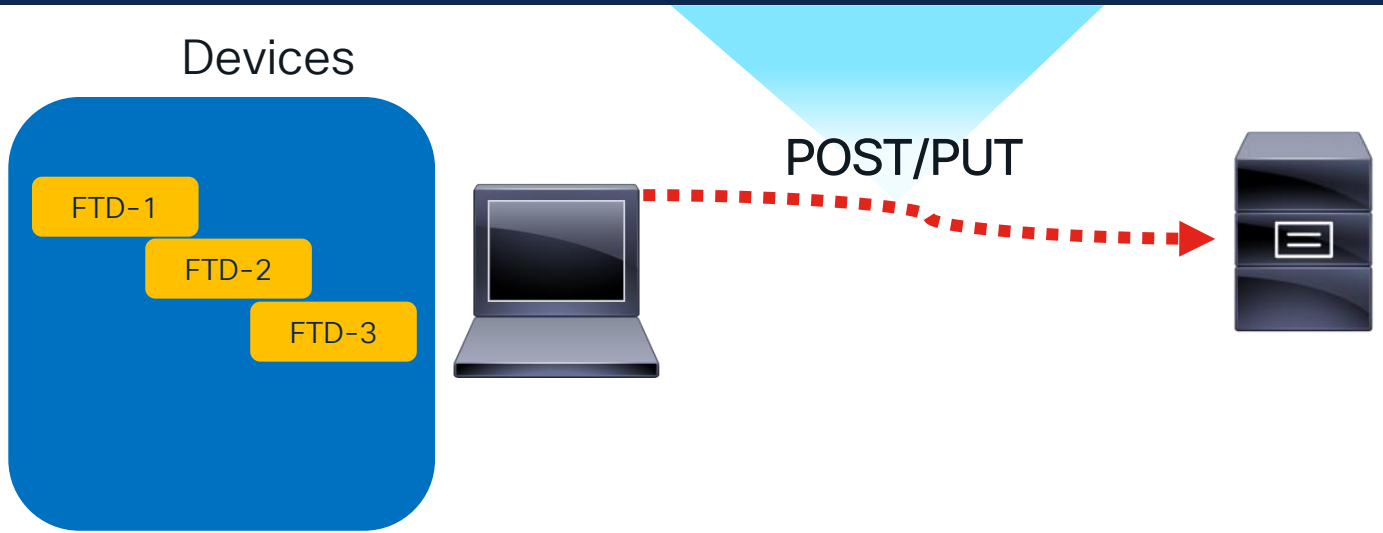


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

FMC Devices

https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/devices/devicerecords





BRKCRT-3001

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

FMC Devices (Cont.)

```
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'

payload={
}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/devices/devicerecords"

response = requests.request("GET", url, headers=headers, json=payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```

?expanded=true

Displays a list of objects with additional attributes

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

FMC Devices (Cont.)


```
{
  "links": {
    "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/devices/devicerecords?offset=0&limit=25"
  },
  "items": [
    {
      "id": "a579d810-2d41-11ec-b1a7-f97833419e16",
      "type": "Device",
      "links": {
        "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/devices/devicerecords/a579d810-2d41-11ec-b1a7-f97833419e16"
      }
      "name": "FTD-0"
    },
    {
      "id": "cd62875a-2d41-11ec-a171-ac03309085cd",
      "type": "Device",
      "links": {
        "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/devices/devicerecords/cd62875a-2d41-11ec-a171-ac03309085cd"
      }
      "name": "FTD-1"
    }
  ],
  "paging": {
    "offset": 0,
    "limit": 25,
    "count": 2,
    "pages": 1
  }
}
```

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

FMC Devices (Cont.)

```
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'
device_UUID = 'a579d810-2d41-11ec-b1a7-f97833419e16'

payload={
}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/devices/devicerecords/{device_UUID}"

response = requests.request("GET", url, headers=headers, json=payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

FMC Devices (Cont.)


```
{
  "deploymentStatus": "DEPLOYMENT_PENDING",
  "id": "a579d810-2d41-11ec-b1a7-f97833419e16",
  "type": "Device",
  "links": {
    "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-
8169-6d9ed49b625f/devices/devicerecords/a579d810-2d41-11ec-b1a7-f97833419e16"
  },
  "name": "FTD-0",
  "description": "NOT SUPPORTED",
  "model": "Cisco Firepower Threat Defense for VMware",
  "modelId": "A",
  "modelNumber": "75",
  "modelType": "Sensor",
  "healthStatus": "green",
  "healthMessage": "1.9% of 54.0M used",
  "sw_version": "7.2.0.1",
  "healthPolicy": {
    "id": "80372d7e-2d45-11ec-b6a6-4a4dae0c354f",
    "type": "HealthPolicy",
    "name": "Initial_Health_Policy 2021-10-14 23:21:46"
  },
  "accessPolicy": {
    "name": "APC_Main",
    "id": "000c2908-741c-0ed3-0000-004294967299",
    "type": "AccessPolicy"
  },
  "hostName": "192.168.1.212",
  "license_caps": [
    "APEX",
    "BASE",
    "MALWARE",
    "PLUS",
    "THREAT",
    "URLFilter"
  ],
  "performanceTier": "FTDv30",
  "keepLocalEvents": false,
  "prohibitPacketTransfer": true,
  "isConnected": true,
  "ftdMode": "ROUTED",
  "analyticsOnly": false,
  "metadata": {
    "readOnly": {
      "state": false
    },
    "inventoryData": {
      "cpuCores": "1 CPU (8 cores)",
      "cpuType": "CPU Xeon E5 series 2000 MHz",
      "memoryInMB": "16384 MB RAM"
    },
    "deviceSerialNumber": "9ARUN7UC0XF",
    "domain": {
      "name": "Global",
      "id": "e276abec-e0f2-11e3-8169-6d9ed49b625f",
      "type": "Domain"
    },
    "isMultiInstance": false,
    "snortVersion": "3.1.21.1-126",
    "vdbVersion": "Build 353 - 2022-03-07 22:13:19.0",
    "lspVersion": "lsp-rel-20220511-1540",
    "clusterBootstrapSupported": true
  },
  "snortEngine": "SNORT3"
}
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

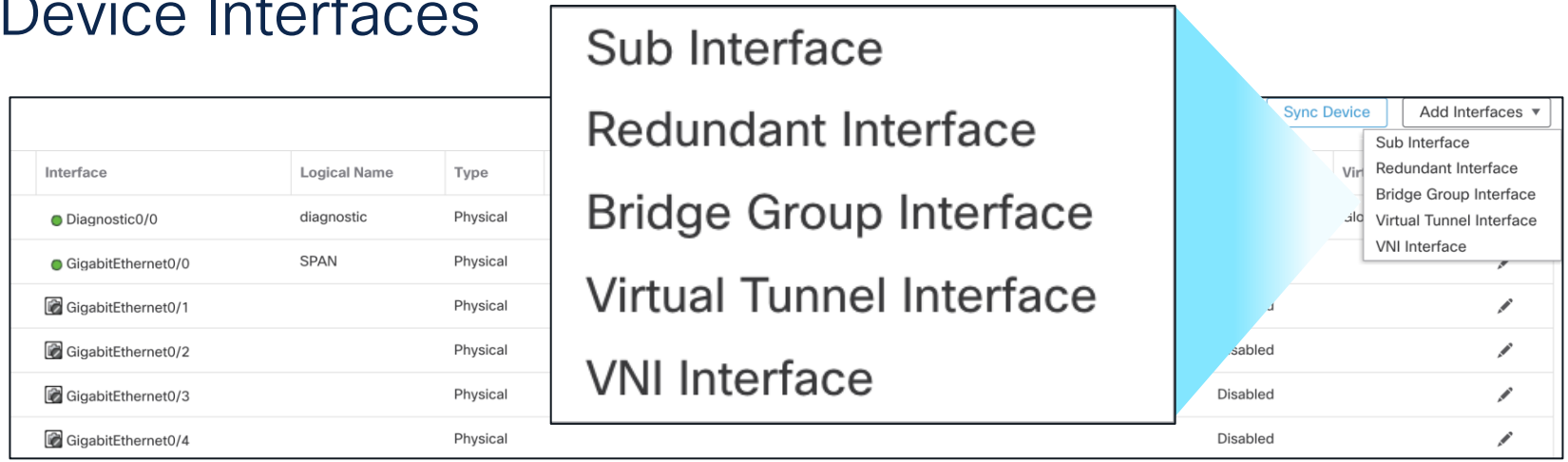
Device Interfaces

Interface	Logical Name	Type	Security Zones
Diagnostic0/0	diagnostic	Physical	
GigabitEthernet0/0	SPAN	Physical	SPAN
GigabitEthernet0/1		Physical	
GigabitEthernet0/2		Physical	
GigabitEthernet0/3		Physical	
GigabitEthernet0/4		Physical	

https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/devices/devicerecords/{device_UUID}/physicalinterfaces

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Device Interfaces



- Depending on the platform, not all interface types are supported.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Device Interface Types

- Sub Interface
- Redundant Interface
- Bridge Group Interface
- Virtual Tunnel Interface
- VNI Interface

```
"https://{FMC_HOST}/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{device_UUID}/subinterfaces
"https://{FMC_HOST}/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{device_UUID}/redundantinterfaces
"https://{FMC_HOST}/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{device_UUID}/bridgegroupinterfaces
"https://{FMC_HOST}/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{device_UUID}/virtualtunnelinterfaces
"https://{FMC_HOST}/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{device_UUID}/vniinterfaces
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Device Physical Interface

```
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'
device_UUID = 'a579d810-2d41-11ec-b1a7-f97833419e16'

payload={
}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/devices/devicerecords/{device_UUID}/physicalinterfaces"

response = requests.request("GET", url, headers=headers, json=payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Device Physical Interface (Cont.)


```
{
   "links": {
      "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-
11e3-8169-6d9ed49b625f/devices/devicerecords/a579d810-2d41-11ec-b1a7-
f97833419e16/physicalinterfaces?offset=0&limit=25"
   },
   "items": [
      {
         "links": {
            "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-
e0f2-11e3-8169-6d9ed49b625f/devices/devicerecords/a579d810-2d41-11ec-
b1a7-f97833419e16/physicalinterfaces/000C2908-741C-0ed3-0000-
004294968282"
         }
         "type": "PhysicalInterface",
         "name": "GigabitEthernet0/0",
         "id": "000C2908-741C-0ed3-0000-004294968282"
      },
      {
         "links": {
            "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-
e0f2-11e3-8169-6d9ed49b625f/devices/devicerecords/a579d810-2d41-11ec-
b1a7-f97833419e16/physicalinterfaces/000C2908-741C-0ed3-0000-
004294968283"
         }
         "type": "PhysicalInterface",
         "name": "GigabitEthernet0/1",
         "id": "000C2908-741C-0ed3-0000-004294968283"
      },
```

```
{
         "links": {
            "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-
11e3-8169-6d9ed49b625f/devices/devicerecords/a579d810-2d41-11ec-b1a7-
f97833419e16/physicalinterfaces/000C2908-741C-0ed3-0000-004294968284"
         }
         "type": "PhysicalInterface",
         "name": "GigabitEthernet0/2",
         "id": "000C2908-741C-0ed3-0000-004294968284"
      },
      {
         "links": {
            "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-
11e3-8169-6d9ed49b625f/devices/devicerecords/a579d810-2d41-11ec-b1a7-
f97833419e16/physicalinterfaces/000C2908-741C-0ed3-0000-004294968285"
         }
         "type": "PhysicalInterface",
         "name": "GigabitEthernet0/3",
         "id": "000C2908-741C-0ed3-0000-004294968285"
      },
      {
         "links": {
            "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-
11e3-8169-6d9ed49b625f/devices/devicerecords/a579d810-2d41-11ec-b1a7-
f97833419e16/physicalinterfaces/000C2908-741C-0ed3-0000-004294968286"
         }
         "type": "PhysicalInterface",
         "name": "GigabitEthernet0/4",
         "id": "000C2908-741C-0ed3-0000-004294968286"
      },
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

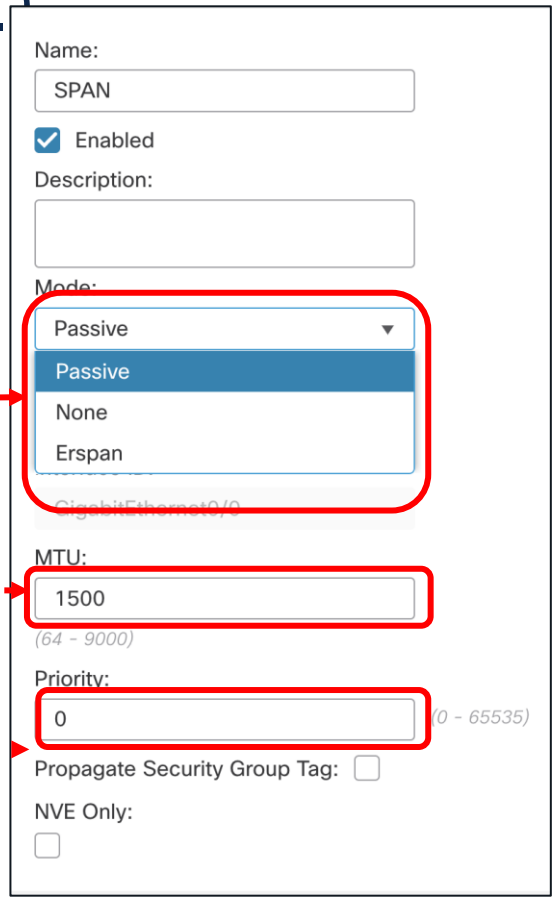
Device Physical Interface

```
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'
device_UUID = 'a579d810-2d41-11ec-b1a7-f97833419e16'
interface_UUID = '000C2908-741C-0ed3-0000-004294968282'

payload={
}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/devices/devicerecords/{device_UUID}/physicalinterfaces/{interface_UUID}"

response = requests.request("GET", url, headers=headers, json=payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Device Physical Interface (Cont.)


```
"metadata": {
    "timestamp": 1634254517646,
    "domain": {
        "name": "Global",
        "id": "e276abec-e0f2-11e3-8169-6d9ed49b625f",
        "type": "Domain"
    }
    "state": "COMMITTED"
},
"links": {
    "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-
8169-6d9ed49b625f/devices/devicerecords/a579d810-2d41-11ec-b1a7-
f97833419e16/physicalinterfaces/000C2908-741C-0ed3-0000-004294968282"
},
"type": "PhysicalInterface",
"MTU": 1500,
"enabled": true,
"name": "GigabitEthernet0/0",
"id": "000C2908-741C-0ed3-0000-004294968282",
"mode": "PASSIVE",
"hardware": {
    "speed": "AUTO",
    "duplex": "AUTO"
},
"nveOnly": false,
"priority": 0,
"managementOnly": false,
"ifname": "SPAN",
"enableSGTPropagate": false,
"securityZone": {
    "id": "57c95964-2d47-11ec-acad-9e7a12b7a7c6",
    "type": "SecurityZone"
}
}
```

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Device Deployment Status

Deployment Pending

FTD-1

ACP_1

Obj_1

Device	Modified by	Inspect Interruption	Type
FTD-1	admin		FTD
Access Control Group Access Control Policy: Sub-ACP-Main	admin		
FTD-0	admin		FTD
Access Control Group Access Control Policy: APC_Main	admin		

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Device Deployment Status (Cont.)

```
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'

payload={
}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/deployment/deployabledevices"

response = requests.request("GET", url, headers=headers, json=payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```

?expanded=true

Displays a list of objects with additional attributes

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Device Deployment Status (Cont.)


```
{
    "links": {
        "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/deployment/deployabledevices?offset=0&limit=25"
    },
    "items": [
        {
            "version": "1678387531086",
            "name": "FTD-0",
            "type": "DeployableDevice"
        },
        {
            "version": "1678387531086",
            "name": "FTD-1",
            "type": "DeployableDevice"
        }
    ],
    "paging": {
        "offset": 0,
        "limit": 25,
        "count": 2,
        "pages": 1
    }
}
```


?expanded=true


```
{
            "upToDate": true,
            "policy": {
                "id": "000C2908-741C-0ed3-0000-146028888644",
                "type": "PG.PLATFORM.NgfwNetworkVirtualizationEndPoint"
            },
            "versionTimeStamp": 1660680123303,
            "trafficInterrupting": false
        },
        {
            "upToDate": false,
            "policy": {
                "name": "Sub-ACP-Main",
                "id": "000C2908-741C-0ed3-0000-154618823282",
                "type": "PG.FIREWALL.NGFWAccessControlPolicy"
            },
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Deploy Changes Schema

```
payload={
 "type": "DeploymentRequest",
 “version”: “<version>”,
 "forceDeploy": false,
 "ignoreWarning": true,
 "deviceList": [
  " a579d810-2d41-11ec-b1a7-f97833419e16”,
  ” cd62875a-2d41-11ec-a171-ac03309085cd”
 ],
 "deploymentNote": "yournotescomehere”
}
```

```
{
        "id": "a579d810-2d41-11ec-b1a7-f97833419e16",
        "type": "Device",
        "links": {
            "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-
8169-6d9ed49b625f/devices/devicerecords/a579d810-2d41-11ec-b1a7-f97833419e16"
        },
        "name": "FTD-0"
    },
    {
        "id": "cd62875a-2d41-11ec-a171-ac03309085cd",
        "type": "Device",
        "links": {
            "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-11e3-
8169-6d9ed49b625f/devices/devicerecords/cd62875a-2d41-11ec-a171-ac03309085cd"
        },
        "name": "FTD-1"
    }
```

DeploymentRequest ⌄ {

description:	Represents the deployment request object which will be used for policy deployment of a device.
metadata	**DeploymentRequestMetadata** › {...}
forceDeploy	boolean Boolean indicating the deployment is force deploy or not. Value set to `true` only if device is not out-of-date and user wants to force a deployment. This should not be used unless needed. Default is `false` .
selectedPoliciesforDevices	› [...]
name	string String represents the name to the device.
deviceList	› [...]
ignoreWarning	boolean Boolean indicating whether the warnings needs to be ignored during the prevalidation of deployment job. Default is `false` .
description	string User provided description of the object.
links	**ILinks** › {...}
id	string Id created as part of the deployment.
type	string Type of the Request object. This value is always DeploymentRequest.
version	string Integer indicating the version to which the deployment should be done.
deploymentNote	string User provided job note with max of 512 characters

}

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Deploy Changes NOW!

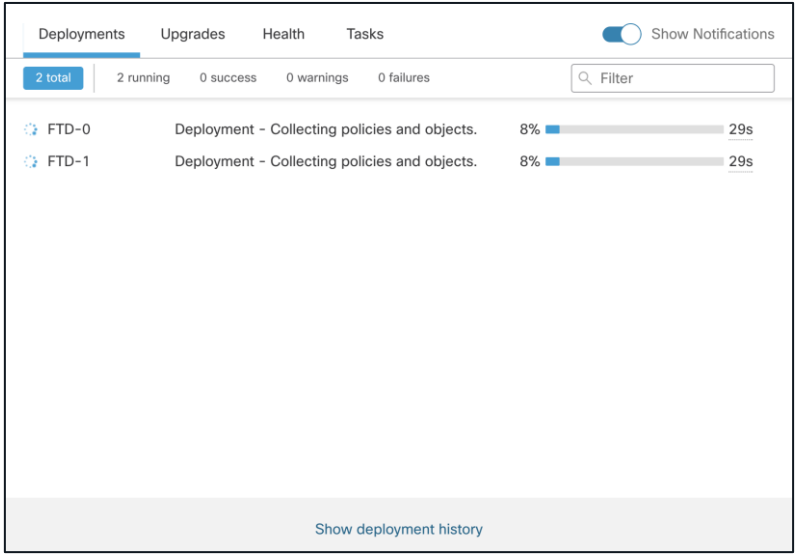
```
import requests
import json

FMC_HOST = 'myfmc.domain.com'
token = <token>
refresh_token = <r_token>
domain_UUID = 'e276abec-e0f2-11e3-8169-6d9ed49b625f'

payload={<Deployment_Schema>}

headers = {
'Content-Type': 'application/json',
'X-auth-access-token': f'{token}',
'X-auth-refresh-token': f'{refresh_token}'
}

url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/deployment/deploymentrequests"

response = requests.request("POST", url, headers=headers, json=payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```

- Once the POST request is sent, the deployment will start.
- Note: ALL PENDING deployment will be sent
- If you want to deploy a subset of polices, then use the: `selectedPoliciesforDevices` schema

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Deploy Changes NOW! (Cont.)

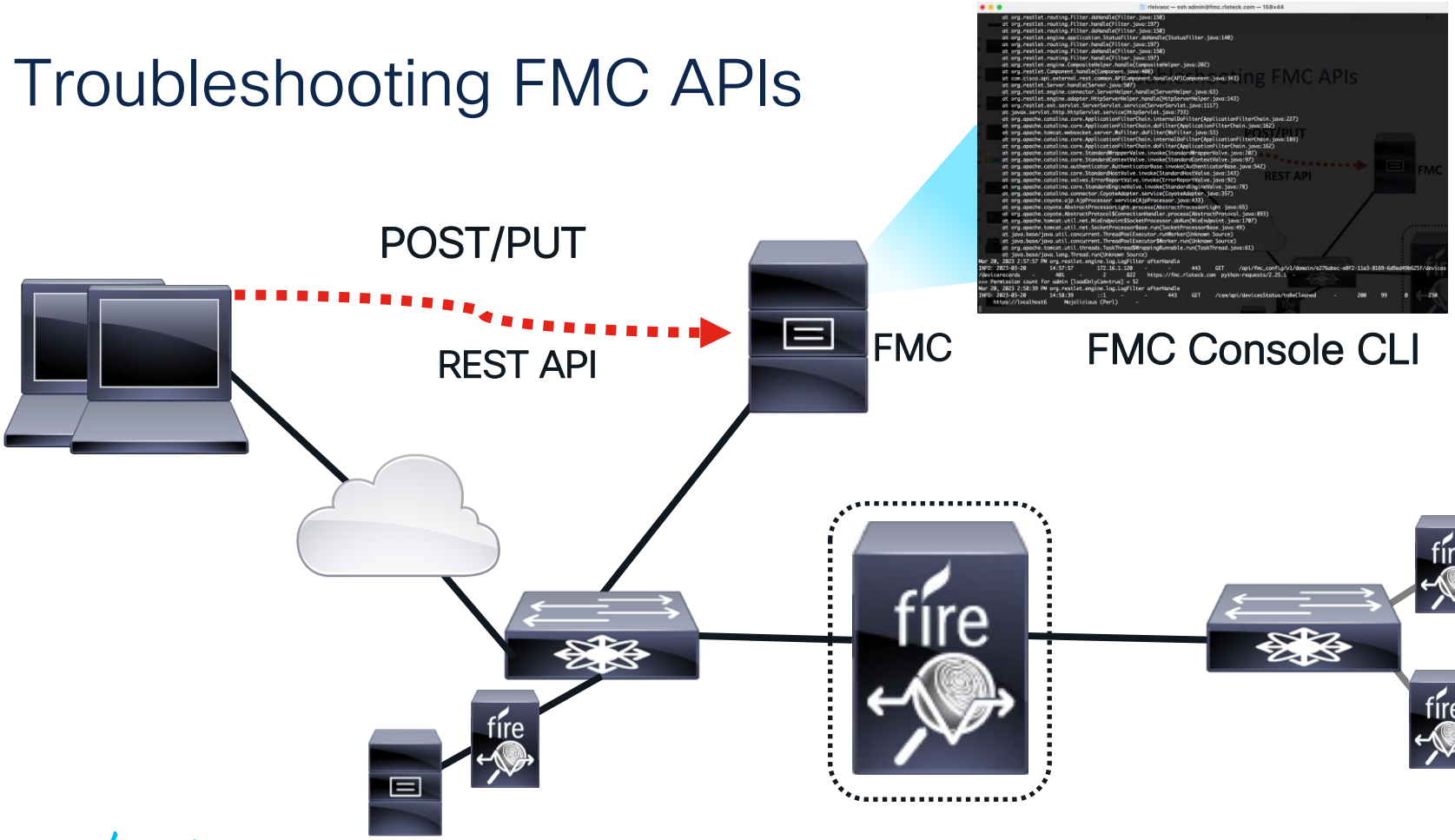

```
{
  "version": "1678387531086",
  "metadata": {
    "task": {
      "id": "214748365047",
      "links": {
        "self": "https://myfmc.domain.com/api/fmc_config/v1/domain/e276abec-e0f2-
11e3-8169-6d9ed49b625f/job/taskstatuses/214748365047"
      }
    }
  }
  "deviceList": [
    "a579d810-2d41-11ec-b1a7-f97833419e16",
    "cd62875a-2d41-11ec-a171-ac03309085cd"
  ],
  "forceDeploy": false,
  "ignoreWarning": true,
  "deploymentNote": "Admin made changes to the ACP",
  "type": "DeploymentRequest"
}
```

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

FMC API Troubleshooting

Troubleshooting FMC APIs

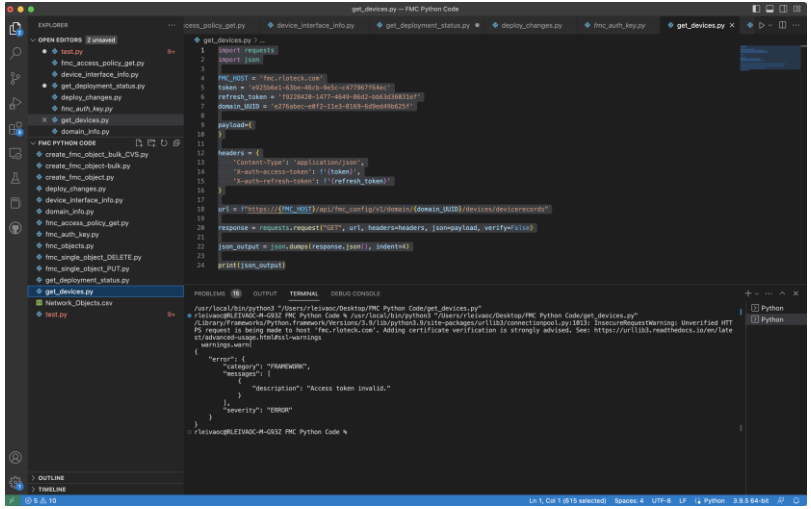


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

FMC Console CLI, or Python IDE Console?

```
rleivaoc — ssh admin@fmc.rloteck.com — 146×42
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APIIdentityPlatformApplication@7738b226 to URI: /api/fmi
_platform
Mar 20, 2023 2:57:56 PM org.apache.catalina.core.ApplicationContext log
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APIUIPlatformApplication@a353dff to URI: /api/ui_platfor
m
Mar 20, 2023 2:57:56 PM org.apache.catalina.core.ApplicationContext log
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APIPlatformApplication@7215172a to URI: /api/fmc_platfor
m
Mar 20, 2023 2:57:56 PM org.apache.catalina.core.ApplicationContext log
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APITroubleshootApplication@5f76d1a0 to URI: /api/fmc_tro
ubleshoot
Mar 20, 2023 2:57:56 PM org.apache.catalina.core.ApplicationContext log
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APICDOServicesApplication@3507531f to URI: /api/cdo_serv
ices
Mar 20, 2023 2:57:56 PM org.apache.catalina.core.ApplicationContext log
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APIUserIdentityApplication@59f1a96c to URI: /api/identit
y
Mar 20, 2023 2:57:56 PM org.apache.catalina.core.ApplicationContext log
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APILamplighterUIApplication@c421318 to URI: /api/tid_ui
Mar 20, 2023 2:57:56 PM org.apache.catalina.core.ApplicationContext log
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APILamplighterApplication@5b218570 to URI: /api/fmc_tid
Mar 20, 2023 2:57:56 PM org.apache.catalina.core.ApplicationContext log
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APINetmapApplication@736f4c27 to URI: /api/fmc_netmap
Mar 20, 2023 2:57:56 PM org.apache.catalina.core.ApplicationContext log
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APIConfigApplication@30dd9615 to URI: /api/fmc_config
Mar 20, 2023 2:57:56 PM org.apache.catalina.core.ApplicationContext log
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APIExplorerApplication@22887352 to URI: /api/api-explore
r
[ajp-nio-127.0.0.1-9009-exec-2] ERROR com.cisco.api.external.rest.common.filters.AuthenticationFilter - Access Token invalid.
[ajp-nio-127.0.0.1-9009-exec-2] ERROR com.cisco.api.external.rest.common.filters.AuthenticationFilter - Access token invalid.
APIException:Access token invalid.
        at com.cisco.api.external.rest.common.filters.AuthenticationFilter.authenticateAccessToken(AuthenticationFilter.java:383)
        at com.cisco.api.external.rest.common.filters.AuthenticationFilter.beforeHandle(AuthenticationFilter.java:150)
        at org.restlet.routing.Filter.handle(Filter.java:195)
        at org.restlet.routing.Filter.doHandle(Filter.java:150)
        at org.restlet.routing.Filter.handle(Filter.java:197)
        at org.restlet.routing.Filter.doHandle(Filter.java:150)
        at com.cisco.api.external.rest.common.filter.AuditFilter.doHandle(AuditFilter.java:133)
        at org.restlet.routing.Filter.handle(Filter.java:197)
        at org.restlet.routing.Filter.doHandle(Filter.java:150)
        at org.restlet.routing.Filter.handle(Filter.java:197)
        at org.restlet.routing.Filter.doHandle(Filter.java:150)
```

FMC Console CLI



Python IDE Console

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Pros and Cons: FMC Console CLI, or Python IDE Console

FMC Console CLI:

- Need to be in **expert** mode. The API engine provides stdout.logs on the /var/opt/CSCOpx/MDC/tomcat/logs/ directory.
- FMC API logging uses java.util.logging framework, which can be very verbose.

Python IDE Console:

- Provides standard error messaging.
- Can be used to store error logging using the write() method.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

FMC API Troubleshooting

The most common problems with APIs on the FMC are the following:

- Token Expired
- Certificate problems
- API URL Structure
- Formatting Issues

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Token Expired

```
  warnings.warn(
{
    "error": {
        "category": "FRAMEWORK",
        "messages": [
            {
                "description": "Access token invalid."
            }
        ],
        "severity": "ERROR"
    }
}
rleivaoc@RLEIVAOC-M-G93Z FMC Python Code %
```

Python IDE Output

```
rleivaoc — ssh admin@fmc.rloteck.com — 160×54
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APIIdentityPlatformApplication@7738b226 to URI: /api/fmi_platform
Mar 20, 2023 2:57:56 PM org.apache.catalina.core.ApplicationContext log
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APIUIPlatformApplication@a353dff to URI: /api/ui_platform
Mar 20, 2023 2:57:56 PM org.apache.catalina.core.ApplicationContext log
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APIPlatformApplication@7215172a to URI: /api/fmc_platform
Mar 20, 2023 2:57:56 PM org.apache.catalina.core.ApplicationContext log
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APITroubleshootApplication@5f76d1a0 to URI: /api/fmc_troubleshoot
Mar 20, 2023 2:57:56 PM org.apache.catalina.core.ApplicationContext log
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APICDOServicesApplication@3507531f to URI: /api/cdo_services
Mar 20, 2023 2:57:56 PM org.apache.catalina.core.ApplicationContext log
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APIUserIdentityApplication@59f1a96c to URI: /api/identity
Mar 20, 2023 2:57:56 PM org.apache.catalina.core.ApplicationContext log
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APILamplighterUIApplication@c421318 to URI: /api/tid_ui
Mar 20, 2023 2:57:56 PM org.apache.catalina.core.ApplicationContext log
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APILamplighterApplication@5b218570 to URI: /api/fmc_tid
Mar 20, 2023 2:57:56 PM org.apache.catalina.core.ApplicationContext log
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APINetmapApplication@736f4c27 to URI: /api/fmc_netmap
Mar 20, 2023 2:57:56 PM org.apache.catalina.core.ApplicationContext log
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APIConfigApplication@30dd9615 to URI: /api/fmc_config
Mar 20, 2023 2:57:56 PM org.apache.catalina.core.ApplicationContext log
INFO: USMAPIServlet: [Restlet] Attaching restlet: com.cisco.api.external.rest.application.APIExplorerApplication@22887352 to URI: /api/api-explorer
[ajp-nio-127.0.0.1-9009-exec-2] ERROR com.cisco.api.external.rest.common.filters.AuthenticationFilter - Access Token invalid.
[ajp-nio-127.0.0.1-9009-exec-2] ERROR com.cisco.api.external.rest.common.filters.AuthenticationFilter - Access token invalid.
APIException:Access token invalid.
        at com.cisco.api.external.rest.common.filters.AuthenticationFilter.doHandle(AuthenticationFilter.java:150)
        at com.cisco.api.external.rest.common.filter.AuditFilter.doHandle(AuditFilter.java:133)
        at org.restlet.routing.Filter.doHandle(Filter.java:150)
        at org.restlet.routing.Filter.handle(Filter.java:197)
        at org.restlet.routing.Router.doHandle(Router.java:422)
        at org.restlet.routing.Router.handle(Router.java:641)
        at org.restlet.routing.Filter.doHandle(Filter.java:150)
        at org.restlet.routing.Filter.handle(Filter.java:197)
```

[ajp-nio-127.0.0.1-9009-exec-2] ERROR com.cisco.api.external.rest.common.filters.AuthenticationFilter - Access Token invalid.
[ajp-nio-127.0.0.1-9009-exec-2] ERROR com.cisco.api.external.rest.common.filters.AuthenticationFilter - Access token invalid.
APIException:Access token invalid.

FMC Console CLI

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Certificate problems

```
Traceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages/requests/adapters.py", line 439, in send
    resp = conn.urlopen(
  File "/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages/urllib3/connectionpool.py", line 755, in urlopen
    retries = retries.increment(
  File "/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages/urllib3/util/retry.py", line 574, in increment
    raise MaxRetryError(_pool, url, error or ResponseError(cause))
urllib3.exceptions.MaxRetryError: HTTPSConnectionPool(host='myfmc.domain.com', port=443): Max retries exceeded with url: /api/fmc_platform/v1/auth/generatetoken
(Caused by SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate (_ssl.c:1129)')))

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/Users/rleivaoc/Desktop/FMC Python Code/fmc_auth_key.py", line 34, in <module>
    response = requests.request("POST", url, headers=headers, data=payload, auth=basic)
  File "/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages/requests/api.py", line 61, in request
    return session.request(method=method, url=url, **kwargs)
  File "/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages/requests/sessions.py", line 542, in request
    resp = self.send(prep, **send_kwargs)
  File "/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages/requests/sessions.py", line 655, in send
    r = adapter.send(request, **kwargs)
  File "/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages/requests/adapters.py", line 514, in send
    raise SSLError(e, request=request)
requests.exceptions.SSLError: HTTPSConnectionPool(host='myfmc.domain.com', port=443): Max retries exceeded with url: /api/fmc_platform/v1/auth/generatetoken
(Caused by SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate (_ssl.c:1129)')))
rleivaoc@RLEIVAOC-M-G93Z FMC Python Code %
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Certificate problems (Cont.)

In a test environment in order to resolve the certificate self-sign errors, you need to add the following **verify** parameter on your response.

```
response = requests.request("POST", url, headers=headers, data=payload, auth=basic, verify=False)
```

In a production environment you need to provide the path where your Trusted CA's are located on the **verify** parameter.

```
response = requests.request("POST", url, headers=headers, data=payload, auth=basic, verify="/path/trustedca")
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

API URL Structure

Devices

GET	/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{objectId}
PUT	/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{objectId}
DELETE	/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{objectId}
GET	/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords
POST	/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords
GET	/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{containerUUID}/fpphysicalinterfaces/{objectId}
PUT	/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{containerUUID}/fpphysicalinterfaces/{objectId}
GET	/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{containerUUID}/fplogicalinterfaces/{objectId}
PUT	/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{containerUUID}/fplogicalinterfaces/{objectId}
DELETE	/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{containerUUID}/fplogicalinterfaces/{objectId}
GET	/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{containerUUID}/fplogicalinterfaces
POST	/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{containerUUID}/fplogicalinterfaces
GET	/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{containerUUID}/inlinesets/{objectId}
PUT	/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{containerUUID}/inlinesets/{objectId}

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

API URL Structure (Cont.)

PUT `/api/fmc_config/v1/domain/{domainUUID}/devices/devicerecords/{objectId}`

Retrieves or modifies the device record associated with the specified ID. Registers or unregisters a device. If no ID is specified for a GET, retrieves list of all device records. ***Check the response section for applicable examples (if any).***

Parameters — **Try it out**

Name	Description
objectId * required string *(path)*	Identifier for a device. objectId - Identifier for a device.
domainUUID * required string *(path)*	Domain UUID e276abec-e0f2-11e3-8169-6d9ed49b625f

Request body required — **application/json**

Input representation of device.

Examples:

Example 1 : PUT /fmc_config/v1/domain/DomainUUID/devices/devicerecords/device_uuid (Test PUT of network object to update re

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

API URL Structure (Cont.)

Input representation of device.

Examples:

Example 1 : PUT /fmc_config/v1/domain/DomainUUID/devices/devicerecords/device_uuid (Test PUT of network obj

```
{
  "id": "device_uuid",
  "name": "nachos_updated",
  "type": "Device",
  "hostName": "10.20.30.40",
  "license_caps": [
    "PROTECT",
    "MALWARE"
  ],
  "performanceTier": "FTDv30",
  "prohibitPacketTransfer": true
}
```

Example payload using supported schema values.

Execute

Curl

```
curl -X 'PUT' \
  'https://Fmc_fqdn/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/devices/devicerecords/test' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -H 'X-auth-access-token: 05267a00-c035-4d33-81d6-866bde648da7' \
  -d '{
  "id": "device_uuid",
  "name": "nachos_updated",
  "type": "Device",
  "hostName": "10.20.30.40",
  "license_caps": [
    "PROTECT",
    "MALWARE"
  ],
  "performanceTier": "FTDv30",
  "prohibitPacketTransfer": true
}'
```

Request URL

```
https://Fmc_fqdn/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/devices/devicerecords/test
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Formatting Issues

```
url = f"https://{FMC_HOST}/api/fmc_config/v1/domain/{domain_UUID}/policy/accesspolicies/{acp_UUID}/accessrules/{rule_id}"

response = requests.request("GET", url, headers=headers, json=payload, verify=False)

json_output = json.dumps(response.json(), indent=4)

print(json_output)
```

data=payload

json=payload

- All payload needs to be in **json** format. The use of the json parameter is needed when the data is not already formatted in json.
- If the date is already formatted in json, then the **data** parameter can be used.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Fill out your session surveys!

Attendees who fill out a minimum of four session surveys and the overall event survey will get **Cisco Live-branded socks** (while supplies last)!

Attendees will also earn 100 points in the **Cisco Live Challenge** for every survey completed.

These points help you get on the leaderboard and increase your chances of winning daily and grand prizes

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Continue your education

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Thank you

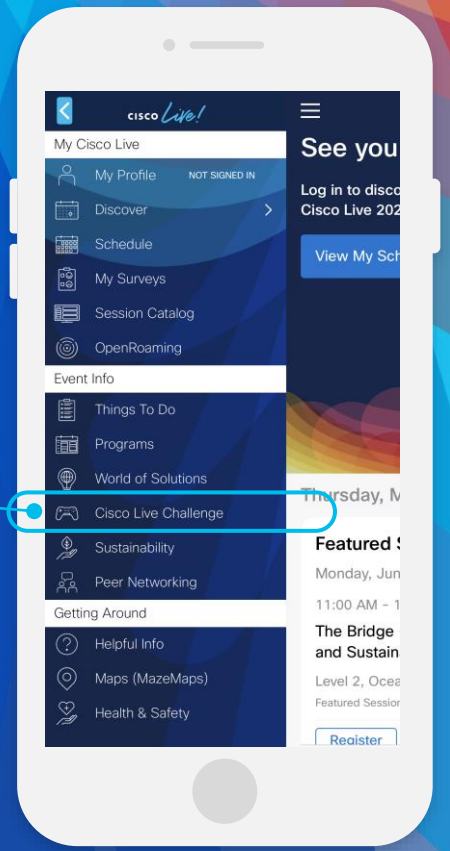
Cisco Live
Challenge

Gamify your Cisco Live experience!

Get points for attending this session!

How:

1. Open the Cisco Events App.
2. Click on 'Cisco Live Challenge' in the side menu.
3. Click on View Your Badges at the top.
4. Click the + at the bottom of the screen and scan the QR code:



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

cisco Live!

Let's go