

The background features a vibrant, abstract design with overlapping, wavy bands of color in shades of blue, green, yellow, orange, and red. On the right side, a bright white light source radiates outwards, creating a starburst effect with sharp, colorful rays that fan across the entire image.

cisco *Live!*

Let's go

#CiscoLiveAPJC



The bridge to possible

Automation in Cisco Secure using FMC Ansible collection

Sameer Pratap Singh
Technical Marketing Engineer
DEVNET-2373

CISCO *Live!*

#CiscoLiveAPJC

Cisco Webex App

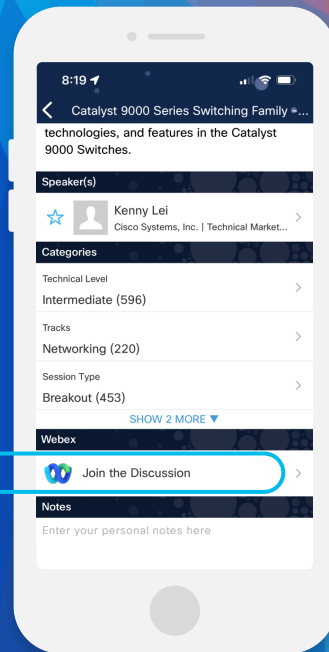
Questions?

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 December 22, 2023.



<https://ciscolive.ciscoevents.com/ciscolivebot/#DEVNET-2373>

HashiCorp 2023 State of Cloud Strategy Survey

56%

Boosted cloud spending in the last year, despite macroeconomic uncertainty

92%

Of high-cloud-maturity organizations say multi-cloud is working, or is expected to within a year

53%

Of high-maturity organizations are using multi-cloud to save money

#1

Rank of security as a multi-cloud success factor

#1

Rank of skills shortages as a multi-cloud barrier

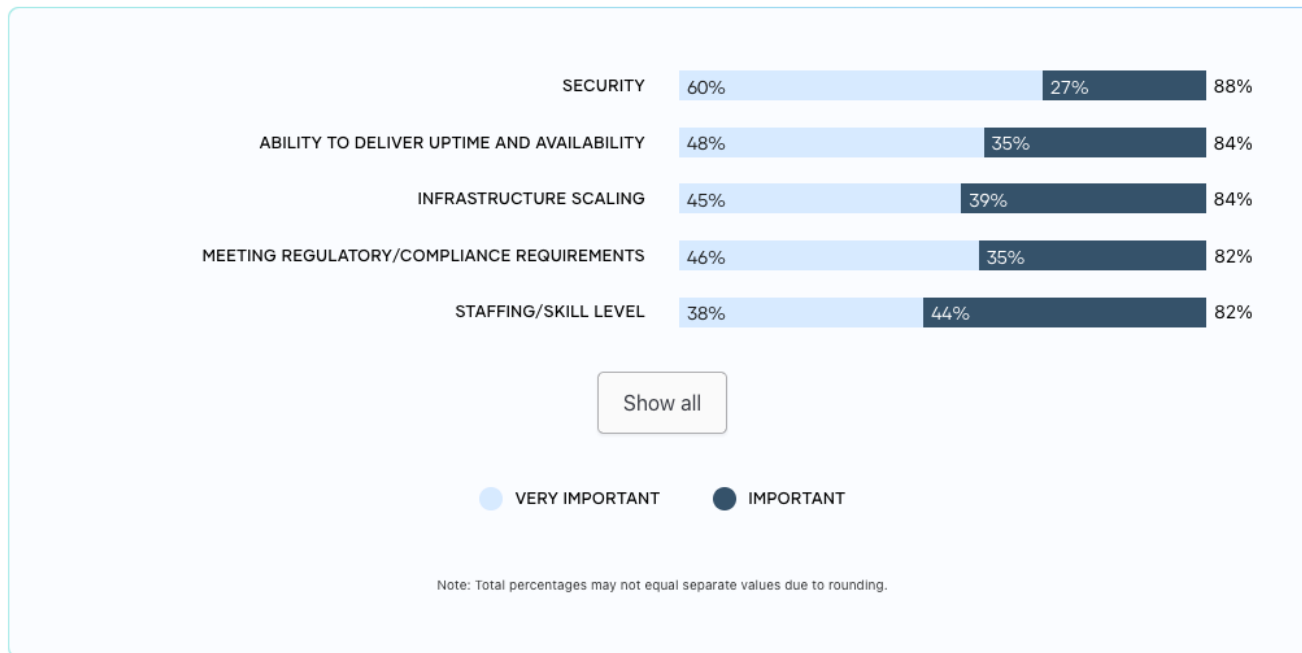
74%

Of high-maturity companies say multi-cloud helps them attract, motivate, and retain talent

92%

Of organizations are adopting, standardizing, or scaling platform teams

Factors in determining the success of your organization's multi-cloud strategy?



*“You’re either the one that creates
the automation or you are getting
automated”*

Tom Preston Werner

About the Speaker

Sameer Pratap Singh

- Degree in Electronics and Communication Engineering
- Security Solutions Consulting Engineer till 2021
- Technical Marketing Engineer – Network Security
- CCIE Security
- Interested in everything automation



Agenda

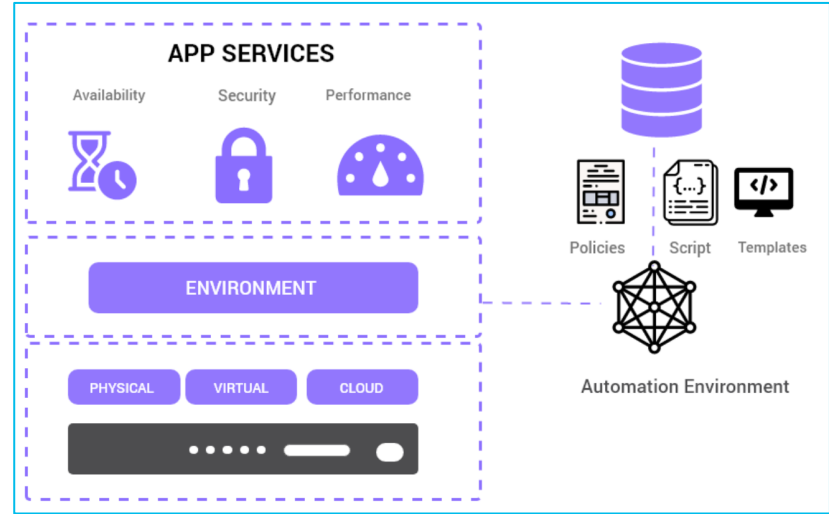
- Introduction
- Infrastructure as Code
- Ansible
- FMC REST API
- FMC Ansible Collection
- Demo
- Conclusion

Introduction

This session will walk you through how Ansible can be used as an infrastructure as a code tool to achieve automation and orchestration in your Cisco Secure Firewall deployment, management and operation.

Infrastructure as Code

- Process of managing and provisioning the complete IT infrastructure using machine-readable definition files
- Software engineering approach toward operations
- Ideal Goal – Help in automating the complete Infrastructure by using programming scripts



Core Principles

- Infrastructure as Code



- Automation is the way to go
- Spend time solving new problems
- Increase value of business in the time spent



- Failure is Normal
- Studying causes and handling exceptions instead of <<Hotfixes>>



- Continuity at all stages
- Testing, Integration, Deployment, Monitoring and Feedback

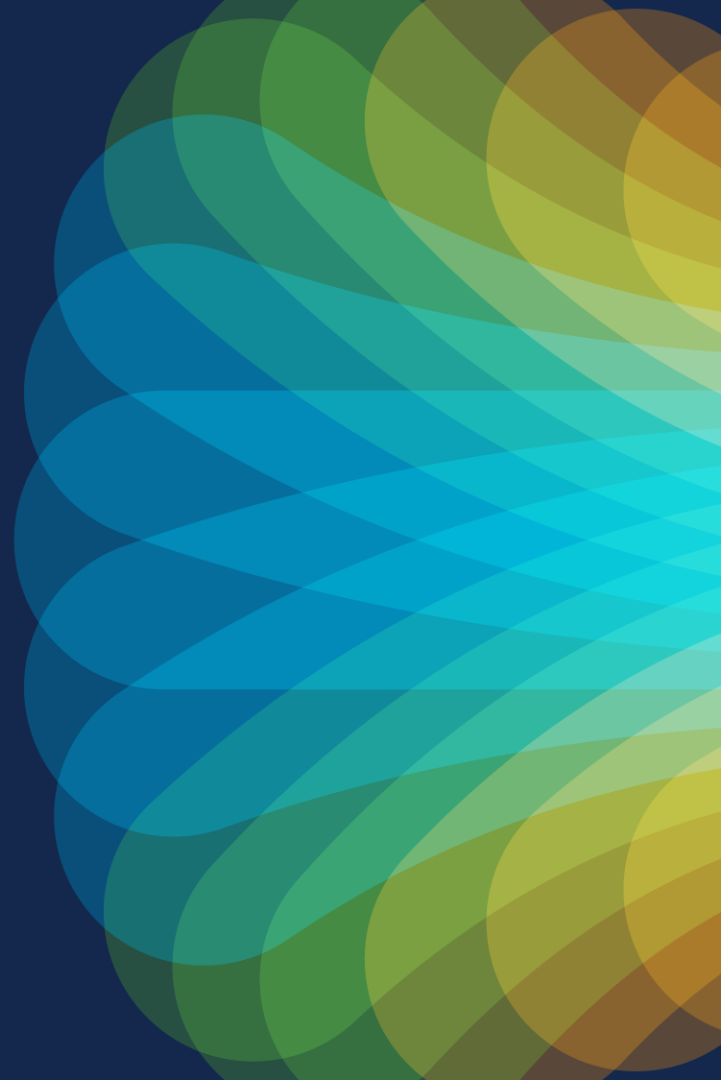
Leverages tools such as

- Version control
- CI/CD Pipeline
- Automated Testing

Examples

- Ansible
- Terraform
- AWS CloudFormation
- Azure Resource Manager

Ansible



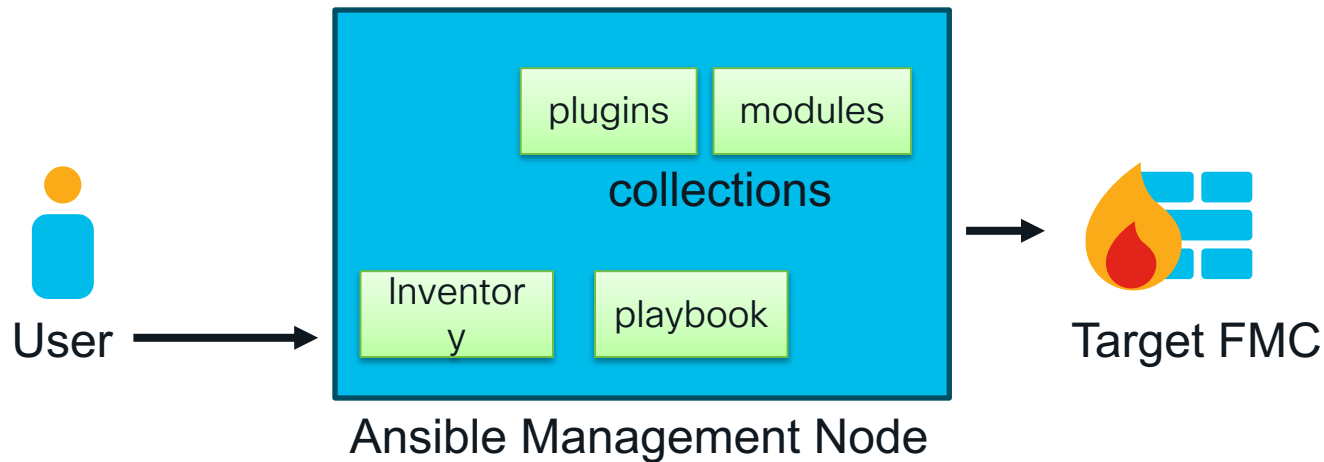
Ansible

- Automation, Configuration and Orchestration Tool
- Open Source
- Agentless Push model
- Idempotency
- REST API interaction

Ansible Capabilities

- Infrastructure Provisioning
- Configuration Management
- Application Deployment
- Security and Compliance
- Orchestration
- Cloud provisioning

Ansible Architecture





Ansible playbook

A blueprint of automation tasks written in human-readable YAML format and executed on a set, group, or classification of hosts, which together make up an Ansible inventory.



Task

consists of a module and its parameters. Ansible modules are pre-defined units of code that perform specific actions on the target



Inventory

hosts and groups of hosts upon which commands, modules and tasks in a playbook operate.

Ansible Installation

Create a virtualenv

```
$ python -m virtualenv venv
```

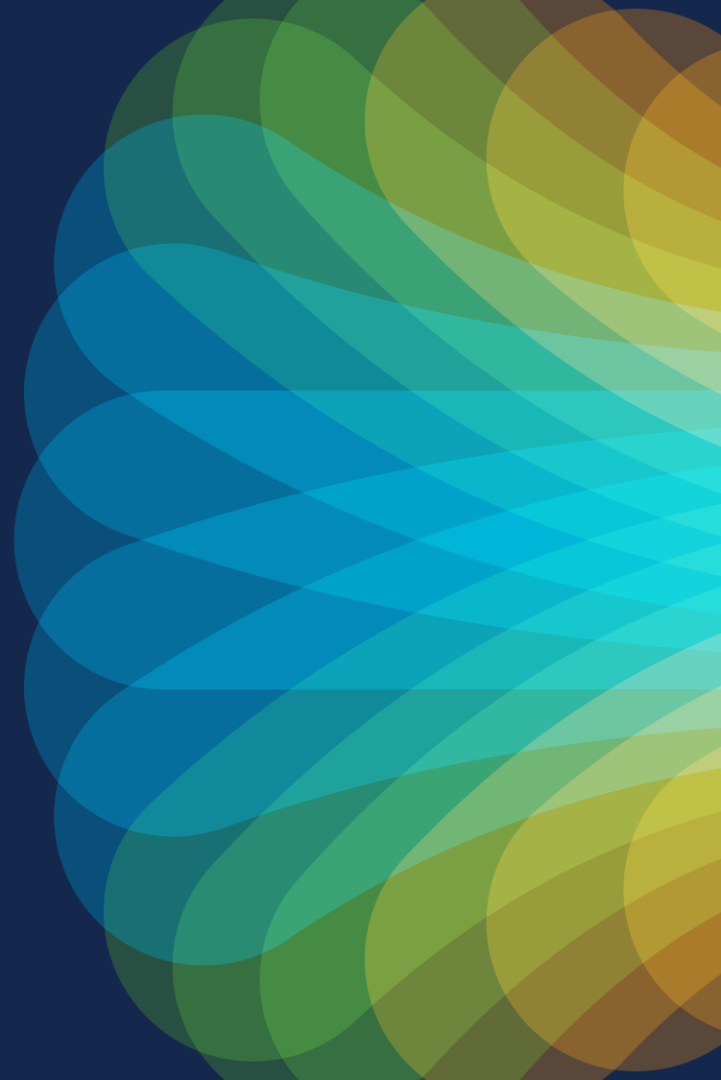
Activate the virtual environment

```
$ source venv/bin/activate
```

Install Ansible

```
$ pip install ansible
```

FMC REST API



FMC REST API

- The REST API is an application programming interface (API), based on “**RESTful**” principles, which you can quickly enable on FMC and use with a REST client.
- Use standard **HTTP methods** to access current configuration information, and issue additional configuration parameters.



CREATE



UPDATE



READ



DELETE

Enabling REST API on FMC

- Navigate to **System>Configuration>REST API Preferences>Enable REST API**
- Check the "Enable REST API" checkbox.
- Click "**Save**". A "Save Successful" dialog will display when the REST API is enabled.

Swagger example GUI on FMC

POST /api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies

Retrieves, deletes, creates, or modifies the access control policy associated with the specified ID. Also, retrieves list of all access control policies. *Check the response section for applicable examples (if any).*

ParametersTry it out

Name	Description
ticket-id string (header)	The ticket ID corresponding to which the configuration changes are done.
	<input type="text" value="ticket-id - The ticket ID corresponding to which"/>
domainUUID * required string (path)	Domain UUID
	<input type="text" value="e276abec-e0f2-11e3-8169-6d9ed49b625f"/>

Request body requiredapplication/json ▾

Input representation of access control policy.

Examples:

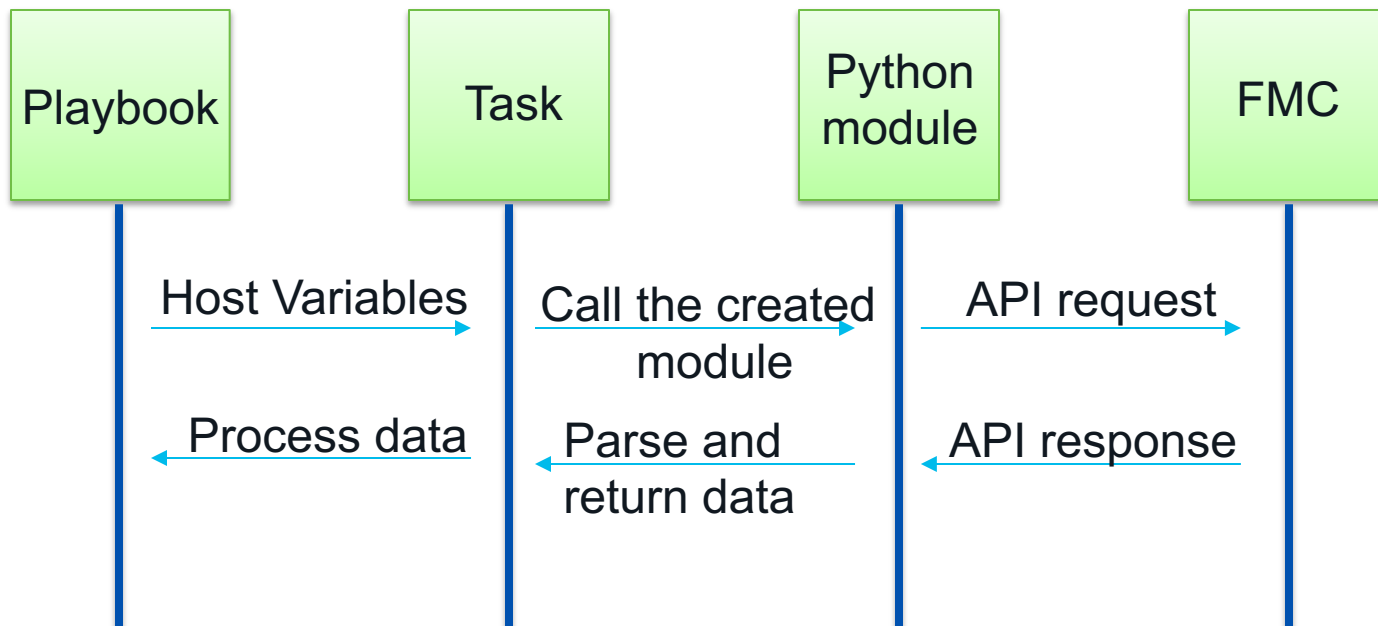
Example Value | **Schema**

```
{
  "type": "AccessPolicy",
  "name": "AccessPolicy1",
  "defaultAction": {
    "action": "BLOCK"
  }
}
```

FMC Ansible Collection

Cisco Secure Firewall Management Center (FMC) Ansible Collection

- This Ansible collection automates configuration management and execution of operational tasks on Cisco Secure Firewall Threat Defence (FTD) being managed by a Firewall Management Center (FMC) using Ansible.
- **fmc_Configuration** module – this module will be used to configure and manage FTD devices that an FMC is managing. FMC REST API will be used to communicate with the device by sending HTTPS calls formatted according to the REST API specification.



Installation Guide

Collections are a distribution format for Ansible content that can include **playbooks, roles, modules, and plugins**.

Install the FMC Ansible collection using the following command

```
ansible-galaxy collection install cisco.fmcansible
```

Creating Inventory

Ansible inventory contains information about systems where the playbooks should be run.

The default location for inventory is **/etc/ansible/hosts**

The inventory file requires:

Hostname or IP Address of the FMC

Username for FMC

Password for the given user

Operations

operations are similar to CRUD functions and can be divided into the following groups:

- **Get** - fetches an object by its ID
- **GetAll** - fetches a list of objects matching given criteria
- **Create** - creates a new object
- **Update** - updates an existing object
- **Delete** - deletes an existing object by its ID
- **Upsert** - creates a new object if it does not exist, or updates it when the object already exists

Parameters

Parameter	Required	Type	Description
operation	True	string	The name of the operation to execute. Commonly, the operation starts with 'create', 'update', 'get', 'upsert' or 'delete' verbs, but can have an arbitrary name too.
data	False	dict	Corresponds to the body part in HTTP request and is mandatory for create, update and upsert operations.
query_params	False	dict	Corresponds to the query string in HTTP request
path_params	False	dict	Corresponds to URL parameters in HTTP request
register_as	False	string	A name for the fact that is registered with the response from the server.
filters	False	dict	A map with filter criteria for upsert and getAll operations.

Server Response

- The module registers server response as **ansible_facts** to be used in the playbook as regular variables.
- fact names are constructed as:
`{OBJECT_TYPE} _{LOWERCASE_OBJECT_NAME}`
- If you want to change the default naming convention, add a **register_as** parameter with the desired fact name to the play.

Task Idempotency

- The result of running it once is exactly the same as the result of running it multiple times.
- FMC Ansible checks whether the desired final state is already achieved.
 - If yes, no actions are executed, and the operation finishes showing that the state has not changed.
 - If no, the change is executed

Examples

Create a Network Object

```
- name: Create Network Object
  cisco.fmcansible.fmc_configuration:
    operation: createMultipleNetworkObject
    path_params:
      domainUUID: '{{ domain[0].uuid }}'
    data:
      name: test-network4
      value: 3.3.3.0/24
      type: networkobject
```

Create a Network Object Group

```
- name: Create Network Object
  cisco.fmcansible.fmc_configuration:
    operation: createMultipleNetworkObject
    path_params:
      domainUUID: '{{ domain[0].uuid }}'
    data:
      name: test-network
      value: 3.3.3.0/24
      type: networkobject
      register_as: test_network

- name: Create Network Object Group
  cisco.fmcansible.fmc_configuration:
    operation: createMultipleNetworkGroup
    path_params:
      domainUUID: '{{ domain[0].uuid }}'
    data:
      objects:
        - name: '{{ test_network.name }}'
          id: '{{ test_network.id }}'
```

Examples

Get a Network Object from FMC

```
- name: Get a specific Network Object
  cisco.fmcansible.fmc_configuration:
    operation: getAllNetworkObject
    path_params:
      domainUUID: '{{ domain[0].uuid }}'
    query_params:
      filter: "nameOrValue:test-network4"
```

Delete Network Object

```
- name: Delete created Network Object
  cisco.fmcansible.fmc_configuration:
    operation: deleteNetworkObject
    path_params:
      domainUUID: '{{ domain[0].uuid }}'
      objectId: "{{ test_network.id }}"
```

Examples

```
- name: Get Sec Zone
  cisco.fmcansible.fmc_configuration:
    operation: getAllSecurityZoneObject
    path_params:
      domainUUID: '{{ domain[0].uuid }}'
    register_as: getZones
```

```
- name: Setup Outside Interface
  cisco.fmcansible.fmc_configuration:
    operation: updateFTDPhysicalInterface
    data:
      ifname: Outside
      ipv4:
        static:
          address: "{{ outside_ip | default('198.18.133.81') }}"
          netmask: "{{ outside_netmask | default('18') }}"
        securityZone: "{{ (getZones | selectattr('name', '==', 'OutZone'))[0] }}"
      MTU: 1500
      enabled: True
      mode: NONE
      type: physicalinterface
      name: GigabitEthernet0/0
    path_params:
      domainUUID: '{{ domain[0].uuid }}'
      containerUUID: '{{ device_list[0].id }}'
      objectId: '{{ physical_interfaces[0].id }}'
```



Demo

day01-config.txt

main.tf.txt

response.json

test.yml

test2.yml

...

fmc_configuration > ! test2.yml

```
1 - hosts: all
2   connection: httpapi
3   tasks:
4     - name: Get Domain
5       cisco.fmcansible.fmc_configuration:
6         operation: getAllDomain
7         register_as: domain
8
9     - name: create auxiliary network object
10      cisco.fmcansible.fmc_configuration:
11        operation: createMultipleNetworkObject
12        path_params:
13          domainUUID: '{{ domain[0].uuid }}'
14        data:
15          name: "ansible-fmc-network2"
16          value: "192.22.22.0/24"
17          type: "networkobject"
18          register_as: net1
19
20     - name: Get Access Policy
21       cisco.fmcansible.fmc_configuration:
22         operation: getAllAccessPolicy
23         path_params:
24           domainUUID: '{{ domain[0].uuid }}'
25         filters:
26           name: NGFW-Access-Policy27
27         register_as: accesspolicy
```

Ln 15, Col 38 Spaces: 2 UTF-8 LF YAML



Did you know?

You can have a
one-on-one session with
a technical expert!

Visit Meet the Expert in The HUB
to meet, greet, whiteboard & gain
insights about your unique questions
with the best of the best.



Meet the Expert Opening Hours:

Tuesday	3:00pm – 7:00pm
Wednesday	11:15am – 7:00pm
Thursday	9:30am – 4:00pm
Friday	10:30am – 1:30pm

Session Surveys

We would love to know your feedback on this session!

- Complete a minimum of four session surveys and the overall event surveys to claim a Cisco Live T-Shirt



Continue your education



- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Expert 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



The bridge to possible

Thank you

CISCO *Live!*

#CiscoLiveAPJC

The background is a vibrant, abstract graphic featuring a spectrum of colors from red and orange on the left to blue and green on the right. The colors are arranged in a series of overlapping, wavy bands that create a sense of movement and depth. A bright, white, starburst-like light source is positioned on the right side, from which rays of light emanate across the entire scene, enhancing the dynamic and energetic feel of the design.

cisco *Live!*

Let's go

#CiscoLiveAPJC