



Automating Cisco APIC Configuration with Postman: A Practical Guide

Amin H. Younessi
Reonel Oy

Tampere, Finland June 17, 2025

About me

- A brief background about me
- Cisco NetAcad Trainer
- MikroTik Certified Trainer and consultant from 2016
- Ubiquiti Certified Trainer from 2017 to 2025

Presentation Topics

- Introduction to ACI
- Understanding the basic of ACI
- Automation in ACI
- Object structure model in ACI
- Practical demonstration

Introduction to Cisco ACI

The screenshot displays the Cisco APIC web interface in a browser window. The address bar shows the URL `https://sandboxapicdc.cisco.com/#ca.1[root]eqptQuickstart`. The interface features a top navigation bar with tabs for System, Tenants, Fabric (selected), Virtual Networking, Admin, Operations, Apps, and Integrations. Below this is a sub-navigation bar with links for Inventory, Fabric Policies, and Access Policies. The left sidebar, titled 'Inventory', contains a tree view with categories like Quick Start, Topology, Pod 1 (expanded), and various protocols (BGP, COOP, IPv4, IPv6, ISIS, LLDP, OSPF, TWAMP). The main content area is titled 'Welcome to Inventory' and includes a brief description of the Inventory menu's purpose. Below this, a 'What's Next?' section offers two buttons: 'Add Remote Leaf' (RL) and 'Add Pod' (P). A 'Related Links' section at the bottom provides links to documentation such as 'Fabric Discovery Process', 'Fabric Initialization and Node Registration', 'ACI Cluster and Node Registration Troubleshooting', 'Cisco ACI Multipod White Paper', and 'Cisco ACI Remote Leaf White Paper'.

APIC (sandboxapicdc.cisco.com) X

← → ↻ 🔒 https://sandboxapicdc.cisco.com/#ca.1[root]eqptQuickstart ☆ 📧 🔔 ⚙️ ? 🖨️

admin 🔍 🗣️ 🔔 ⚙️ ? 🖨️

System Tenants **Fabric** Virtual Networking Admin Operations Apps Integrations

Inventory | Fabric Policies | Access Policies

Inventory 🔍 📄 🔄

- > Quick Start
- > Topology
- ✓ Pod 1
 - > LEAF-101 (Node-101)
 - > LEAF-102 (Node-102)
 - ✓ SPINE-1 (Node-103)
 - > Chassis
 - > Interfaces
 - > Control Plane Statistics
 - > Protocols
 - > BGP
 - > COOP
 - > IPv4
 - > IPv6
 - > ISIS
 - > LLDP
 - > OSPF
 - > TWAMP
 - > Processes
 - SMU patch version
 - > Span Sessions
 - File System Partitions
 - > Rules
 - > VRF Contexts
- Pod Fabric Setup Policy
- Fabric Membership
- Disabled Interfaces and Decommissioned Switches

Welcome to Inventory

The Inventory menu lets you browse fabric assets (leaf and spine switches, APIC controllers, view interface status, etc.). If you have not already done so, now is a good time to register discovered switches

What's Next?

RL Add Remote Leaf

P Add Pod

Related Links

- Fabric Discovery Process
- Fabric Initialization and Node Registration
- ACI Cluster and Node Registration Troubleshooting
- Cisco ACI Multipod White Paper
- Cisco ACI Remote Leaf White Paper

Introduction to Cisco ACI

- What is SDN?
- Cisco ACI Can be integrate with the following solutions:
 - Kubernetes
 - Rancher
 - OpenStack
 - OpenShift
 - Microsoft
 - Red Hat
 - VMware & VMware SDN

Introduction to Cisco ACI



APIC

System

Tenants

Fabric

Virtual Networking

Admin

Operations

Apps

Integrations

Kubernetes

| Rancher RKE

| OpenShift

| OpenStack

| Microsoft

| Red Hat

| VMware

| VMware SDN

Domains

Understanding the basic of Cisco ACI

- Cisco ACI and the APIC
- APIC REST API
 - It allows us to communicate with the APIC using standard web protocols, primarily HTTP.
 - Sending requests to specific web addresses or endpoints leads to perform actions.
 - These actions typically defined by HTTP methods like **GET** (Retrieve information), **POST** (To create new objects), **PUT** (to modify existing objects) and **DELETE** (to remove objects).
 - The APIC API primarily uses JavaScript Object Notation (JSON) as the data format for both responses (Requests and Receiving).
- Postman overview

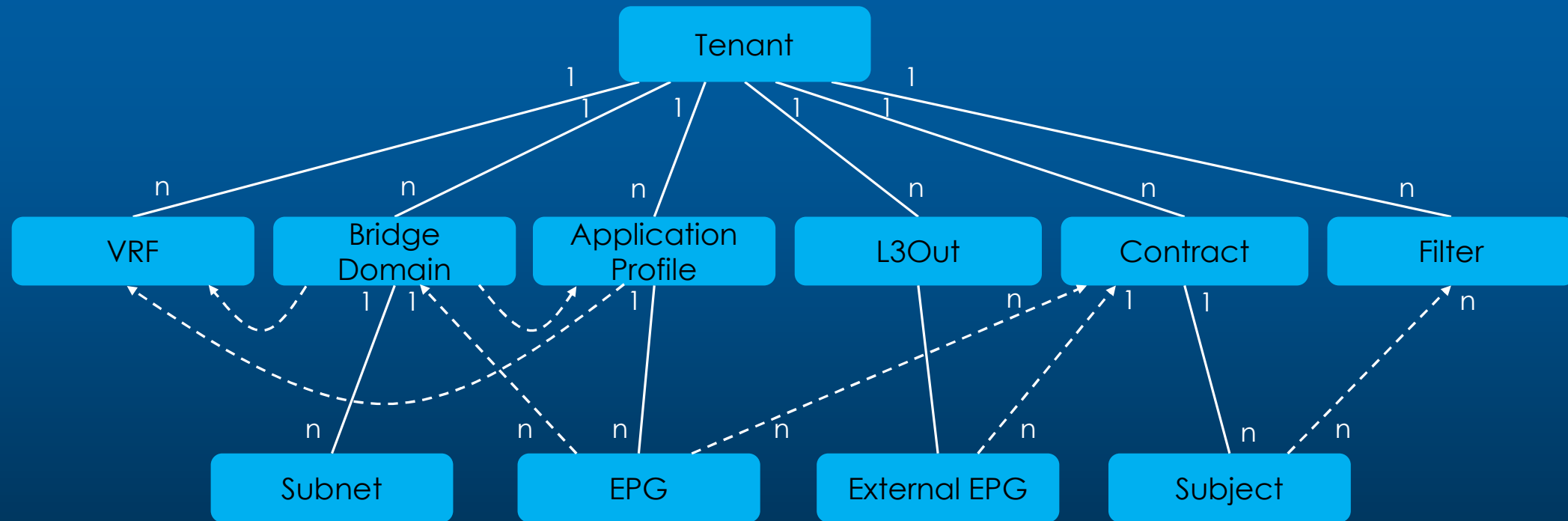
Automation might be a requirement in ACI

- Simplified configuration management.
- Reducing human errors.
- Increasing network reliability.
- Central network management and control.
- Enforcing policies consistently.
- Compatibility with multi-vendor environment.
- Rapid application deployment and many more.

Vital things about Cisco ACI for automation

- ACI object structure model and it's REST API
- Automation tools can be used with ACI
 - REST API
 - Python SDK (Cobra SDK)
 - acitoolkit
 - Ansible
 - Terraform
- Necessary data for configuration

Cisco ACI Object structure model

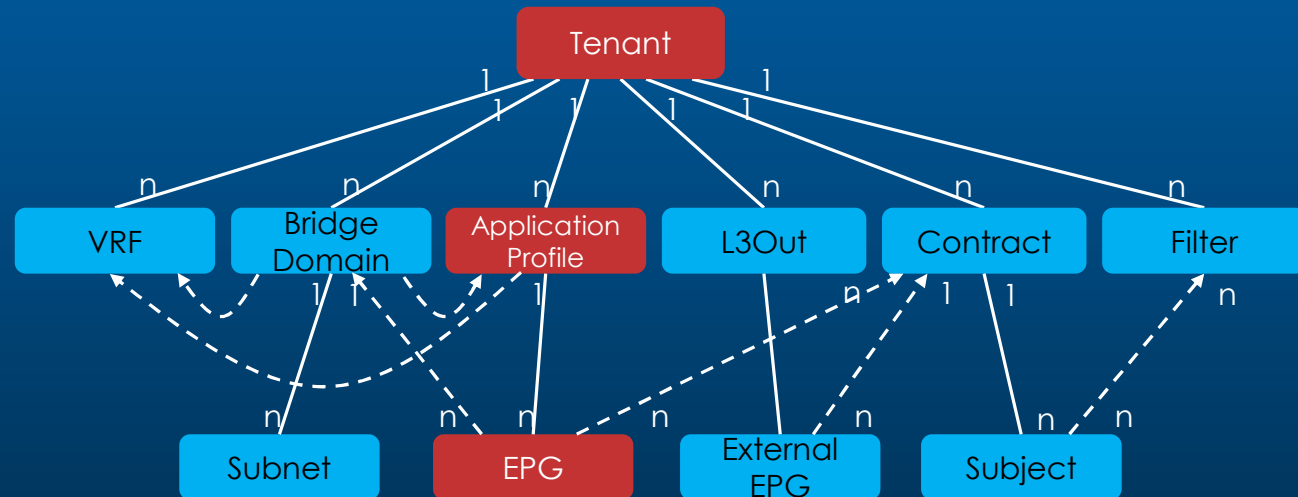
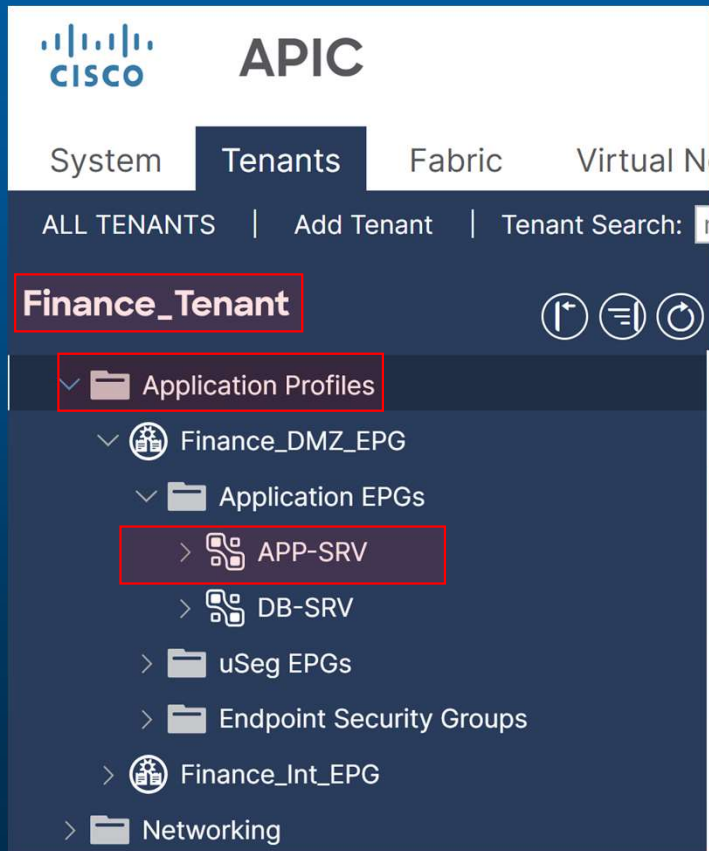


Description:

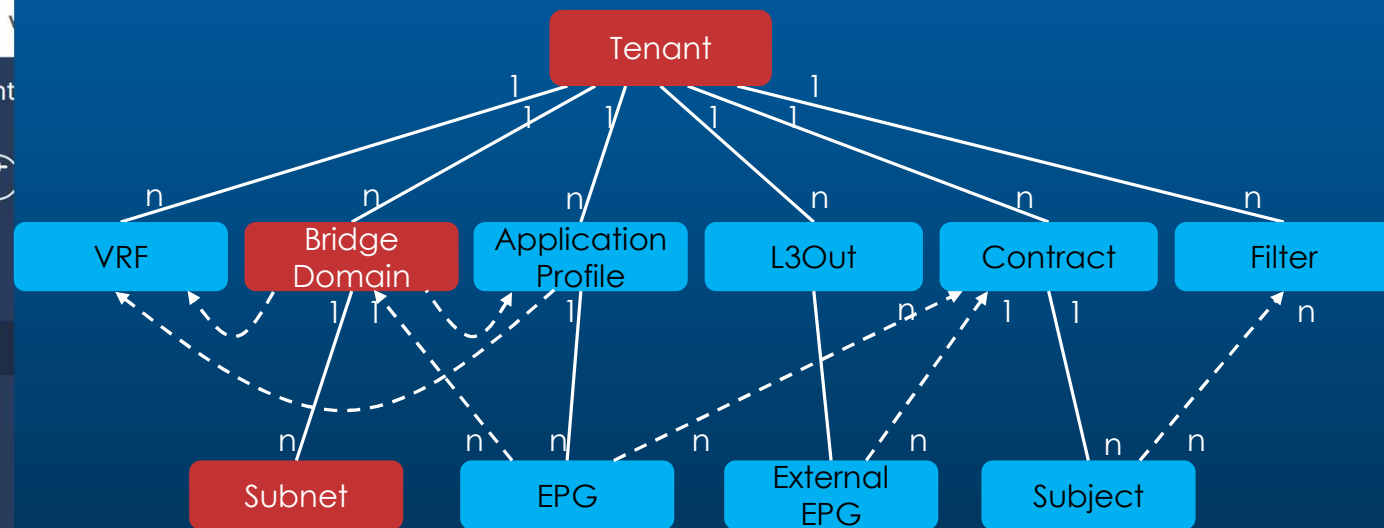
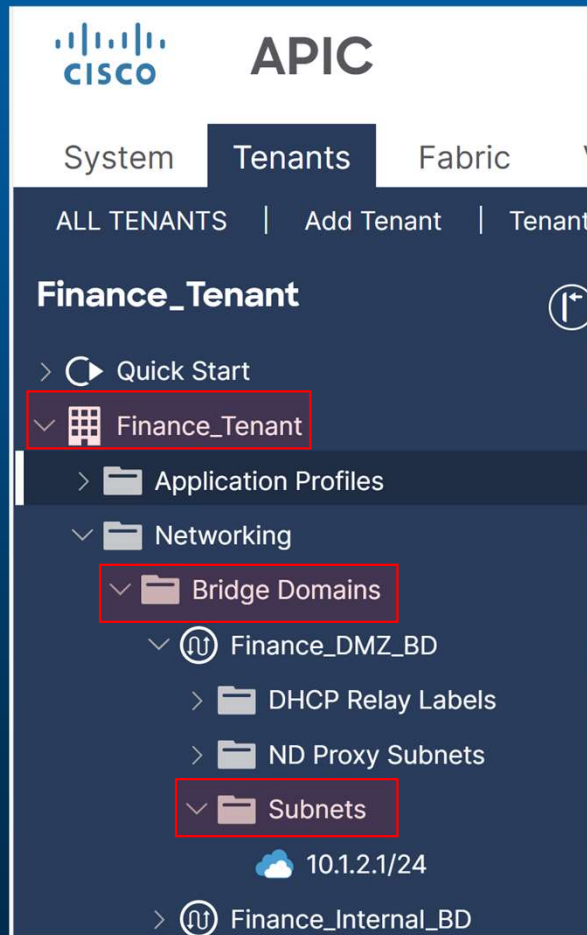
1:n means one to many, n:n indicates many to many.

———— Hierarchical structure
-----> Referential

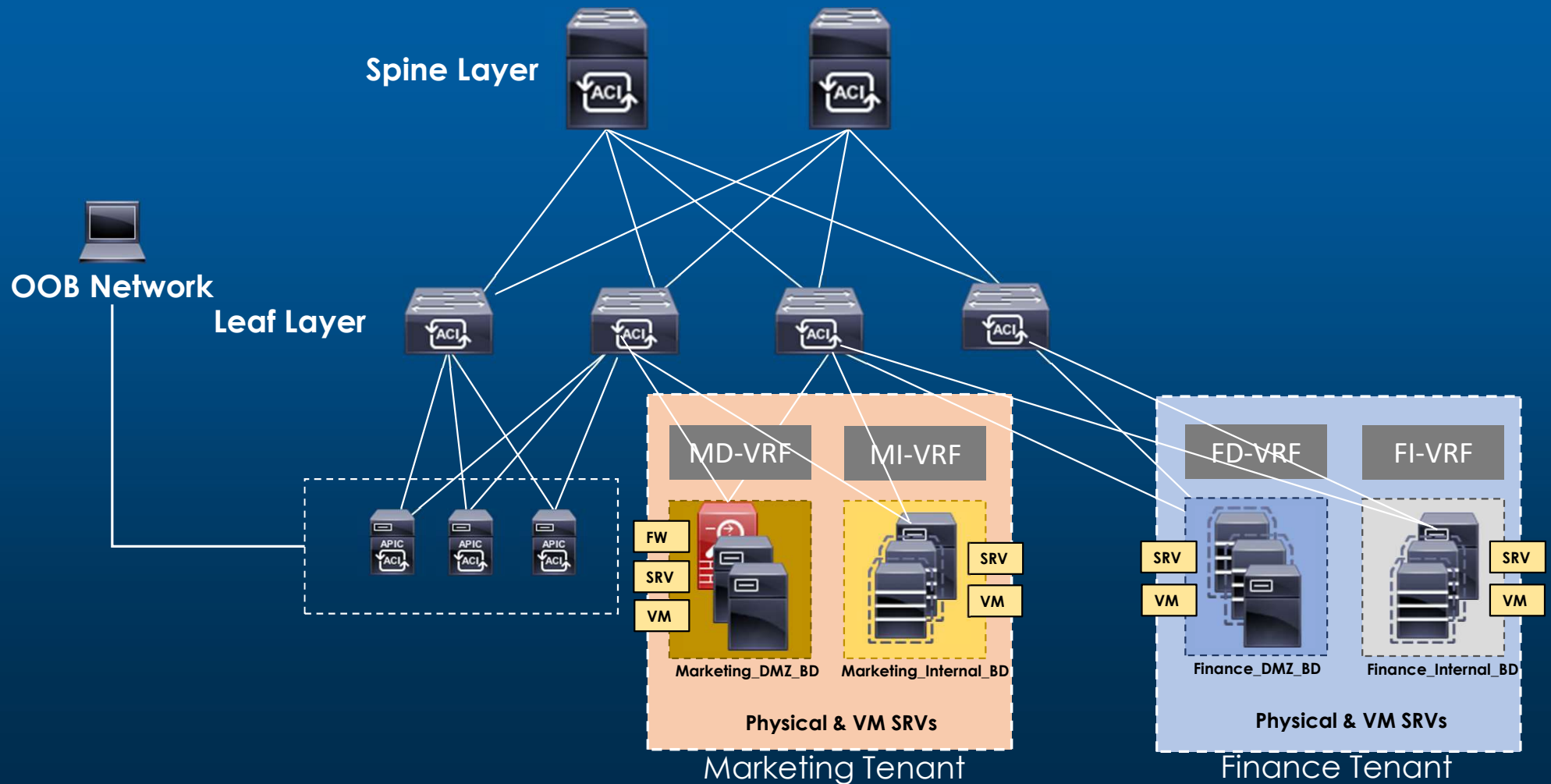
Cisco ACI Object structure model



Cisco ACI Object structure model



Practical demonstration



Using ACI Inspector to create an API calls

The screenshot displays the Cisco APIC (Application Policy Infrastructure Controller) interface. The top navigation bar includes tabs for System, Tenants, Fabric, Virtual Networking, Admin, Operations, Apps, and Integrations. A dropdown menu is open, showing options: ACI Fabric Setup, Show API Inspector, Start Remote Logging, Object Store Browser, Show Debug Info, and Config Sync Issues. The 'Show API Inspector' option is highlighted.

Below the navigation bar, the 'System Health' section shows a 'Warning' status. To the right, the 'Fault Counts by Domain' table is visible, showing counts for various domains like SYSTEM WIDE, Access, External, Framework, Infra, Management, Security, Tenant, and Apps.

In the foreground, a browser window (Microsoft Edge) is open, displaying an error message: 'An error occurred. Sorry, the page you are looking for is currently unavailable. Please try again later. If you are the system administrator of this resource then you should check the error log for details. Faithfully yours, nginx.' The browser's developer console shows the following log entries:

```
</style>
</head>
<body>
<h1>An error occurred.</h1>
<p>Sorry, the page you are looking for is currently unavailable.<br/>
Please try again later.</p>
<p>If you are the system administrator of this resource then you should check
the error log for details.</p>
<p><em>Faithfully yours, nginx.</em></p>
</body>
</html>

timestamp: 01:21:03 DEBUG
method: GET
url: https://sandboxapicdc.cisco.com/api/node/mo/topology/HDfabricOverallHealth5min-0.js
response: {"totalCount":"1","imdata":[{"fabricOverallHealthHist5min":{"attributes":{"chi
timestamp: 01:21:13 DEBUG
method: GET
url: https://sandboxapicdc.cisco.com/api/node/mo/topology/HDfabricOverallHealth5min-0.js
response: {"totalCount":"1","imdata":[{"fabricOverallHealthHist5min":{"attributes":{"chi
```

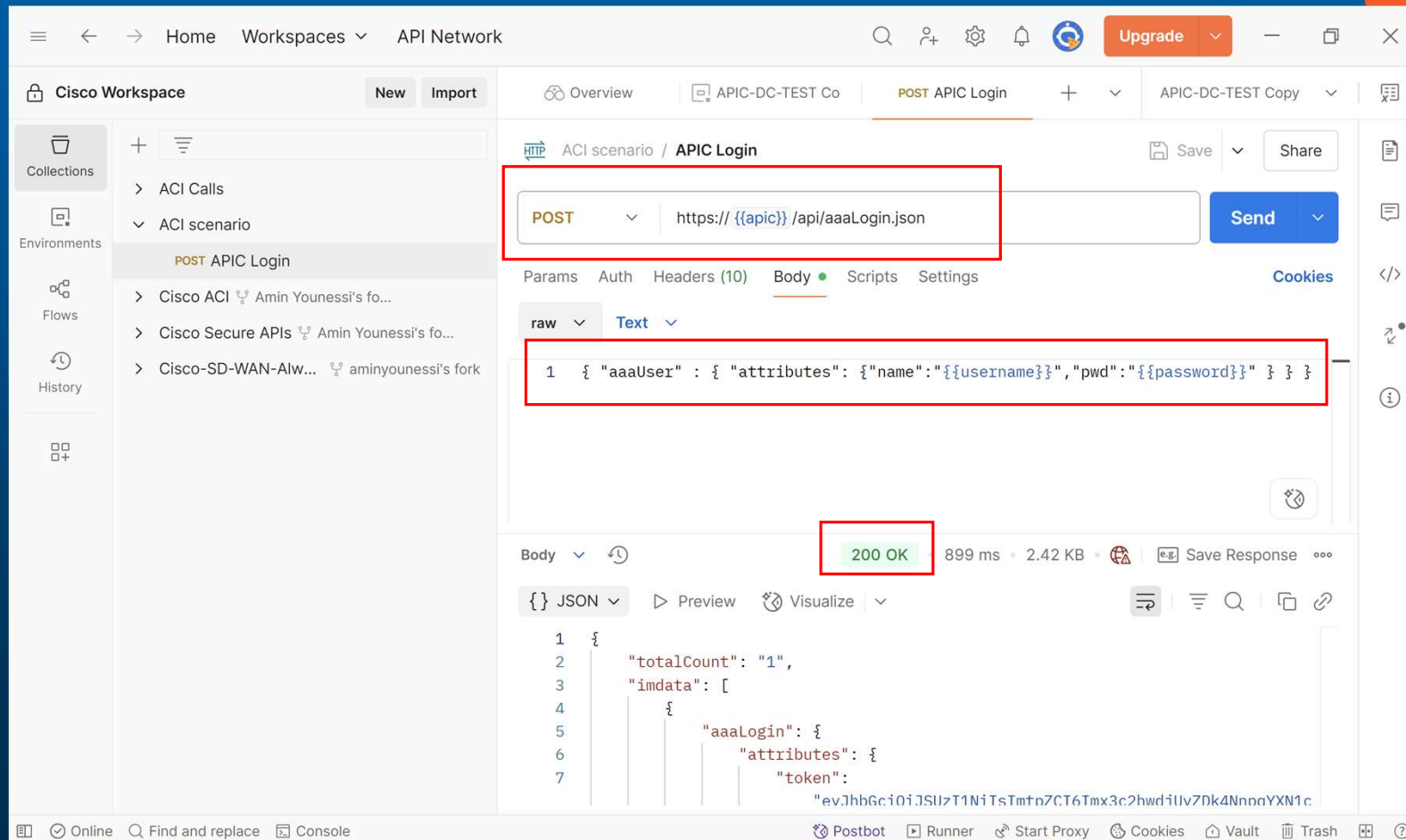
The browser's address bar shows the URL: <https://sandboxapicdc.cisco.com/#> and the current system time is 2025-06-13T22:18 UTC+00:00.

Creating and environment with desired variables

The screenshot displays the Postman interface. On the left sidebar, the 'Environments' tab is selected. Under 'Globals', the 'APIC-DC-TEST Copy' environment is highlighted. The main panel shows the 'APIC-DC-TEST Copy' environment with a table of variables. The table has columns for 'Variable', 'Type', 'Initial value', and 'Current value'. The variables listed are: apic, username, password, TENANT, VRF, BD, subnet, Application Profile, and EPG. Each variable has a checkbox in the first column, which is checked for all. The 'password' variable is of type 'secret' and has masked values. The 'Add new variable' button is at the bottom of the table.

	Variable	Type	Initial value	Current value
<input checked="" type="checkbox"/>	apic	default	131.226.217.153	131.226.217.153
<input checked="" type="checkbox"/>	username	default	admin	admin
<input checked="" type="checkbox"/>	password	secret
<input checked="" type="checkbox"/>	TENANT	default	Finance_Tenant	Finance_Tenant
<input checked="" type="checkbox"/>	VRF	default	VRFFinance	FI-VRF
<input checked="" type="checkbox"/>	BD	default	Finance_DMZ_BD	Finance_DMZ_BD
<input checked="" type="checkbox"/>	subnet	default	10.1.1.1/24	10.1.1.1/24
<input checked="" type="checkbox"/>	Application Profile	default	Finance_INT_EPG	Finance_INT_EPG
<input checked="" type="checkbox"/>	EPG	default	WEB-SRV	WEB-SRV
	Add new variable			

Login to the APIC by Postman



The image shows the Postman application interface with a POST request configured for the APIC login endpoint. The request is highlighted with a red box, showing the method 'POST' and the URL 'https://{{apic}}/api/aaaLogin.json'. The body is in raw text format, also highlighted with a red box, containing a JSON payload for login. The response is a 200 OK status, highlighted with a red box, with a response time of 899 ms and a size of 2.42 KB. The response body is shown in JSON format, containing login details like token and attributes.

Request:

```
POST https://{{apic}}/api/aaaLogin.json
```

Body (raw text):

```
1 { "aaaUser" : { "attributes": { "name": "{{username}}", "pwd": "{{password}}" } } }
```

Response:

```
200 OK 899 ms • 2.42 KB
```

Response Body (JSON):

```
1 {
2   "totalCount": "1",
3   "imdata": [
4     {
5       "aaaLogin": {
6         "attributes": {
7           "token":
            "ev.JhbGc:10i.J5U7T1N1TsTmtn7CT6Tmx3c2hwdiIv7Dk4NnnYXN1c
```

Creating a Tenant with REST API

The screenshot shows a REST client interface with the following elements:

- Navigation Bar:** Includes tabs for 'Overview', 'APIC-DC', 'POST APIC L', 'POST Creati', and 'POST Creati'. The 'POST Creati' tab is active.
- URL Bar:** Displays the URL 'https://sandboxapicdc.cisco.com/api/node/mo/uni/tn-Finance_Tenant.json'. The entire URL is highlighted with a red box.
- Method and URL:** The method is 'POST'. The URL is 'https://{{apic}}/api/node/mo/uni/tn-{{TENANT}}.json'. The placeholders '{{apic}}' and '{{TENANT}}' are highlighted with red boxes.
- Buttons:** 'Send' and 'Share' buttons are visible.
- Params Tab:** The 'Body' tab is selected. The 'Text' format is chosen.
- Body Content:** The JSON payload is:

```
1 payload{"fvTenant":{"attributes":{"dn":"uni/tn-Finance_Tenant",  
  "name":"Finance_Tenant", "rn":"tn-Finance_Tenant" "status":"created"},  
  "children":[]}}response: {"totalCount":"0","imdata":[]}}
```
- Replace Dialog:** A 'Replace All (Ctrl+Alt+Enter)' dialog is open, showing 'Finance_Tenant' in the search field and '{{TENANT}}' in the replace field. A red arrow points to the 'Replace All' button.

Overview

APIC-DC-TE

POST APIC Logi

POST Creating T

APIC-DC-TEST Copy

ACI scenario / Creating Tenant

POST

https://{{apic}}/api/node/mo/uni/tn-{{TENANT}}.json

Send

Params

Auth

Headers (10)

Body

Scripts

Settings

Cookies

raw

Text

1

payload{"fvTenant":{"attributes":{"dn":"uni/tn-{{TENANT}}","name":"{{TENANT}}","rn":"tn-{{TENANT}}","status":"created"},"children":[]}}response:{"totalCount":"0","imdata":[]}}

Body

200 OK

281 ms

637 B

Save Response

JSON

Preview

Visualize

1

{

2

"totalCount": "0",

3

"imdata": []

4

}

CISCO

APIC

System

Tenants

Fabric

Virtual

ALL TENANTS

Add Tenant

Tenant Search:

All Tenants

Name	Alias	D
common		
DMZ-Test		
FIB_DC		
Finance_Tenant		
infra		

Creating a VRF

HTTP ACI scenario / Creating VRF

Save

Share

POST

https://{{apic}}/api/node/mo/uni/tn-{{TENANT}}/ctx-{{VRF}}.json

Send

Params Auth Headers (10) Body Scripts Settings

Cookies

raw

Text

```
1 payload{"fvCtx":{"attributes":{"dn":"uni/tn-{{TENANT}}/ctx-{{VRF}}","name":"{{VRF}}","rn":"ctx-{{VRF}}","status":"created","children":[]}}
2 response: {"totalCount":"0","imdata":[]}
```

Body

200 OK

929 ms

637 B



Save Response

{ } JSON

Preview

Visualize



```
1 {
2   "totalCount": "0",
3   "imdata": []
4 }
```

System

Tenants

Fabric

ALL TENANTS | Add Tenant | Tenant

Finance_Tenant

> Quick Start

> Finance_Tenant

> Application Profiles

> Networking

> Bridge Domains

> VRFs

> FI-VRF

> L2Outs

Creating a Bridge Domain & Subnet

ACI scenario / Creating Bridge Domain & Subnet

Save Share

POST https://{{apic}}/api/node/mo/uni/tn-{{TENANT}}/BD-{{BD}}.json Send

Params Authorization Headers (10) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL Text

```
1 payload{"fvBD":{"attributes":{"dn":"uni/tn-{{TENANT}}/BD-{{BD}}","mac":"00:22:BD:F8:19:FF","arpFlood":"true","name":"{{BD}}","rn":"BD-{{BD}}","status":"created"},"children":[{"fvSubnet":{"attributes":{"dn":"uni/tn-{{TENANT}}/BD-{{BD}}/subnet-{{subnet}}","ctrl":"","ip":"{{subnet}}","rn":"subnet-{{subnet}}","status":"created"},"children":[]},"fvRsCtx":{"attributes":{"tnFvCtxName":"FI-VRF","status":"created,modified"},"children":[]}}]}
2 response: {"totalCount":"0","imdata":[]}
```

200 OK • 337 ms • 637 B • Save Response

JSON Preview Visualize

```
1 {
2   "totalCount": "0",
3   "imdata": []
4 }
```

System Tenants Fabric Virtual

ALL TENANTS | Add Tenant | Tenant Search

Finance_Tenant

Quick Start

Finance_Tenant

Application Profiles

Networking

Bridge Domains

Finance_DMZ_BD

DHCP Relay Labels

ND Proxy Subnets

Subnets

10.1.1.1/24

VRFs

Creating an Application Profile & EPG

ACI scenario / Creating Application Profile

POST https://{{apic}}/api/node/mo/uni/tn-{{TENANT}}/ap-{{Application Profile}}.json

Params Authorization Headers (10) Body Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL Text

```
1 payload{"fvAp":{"attributes":{"dn":"uni/tn-{{TENANT}}/ap-{{Application Profile}}","name":"{{Application Profile}}","rn":"ap-{{Application Profile}}","status":"created"},"children":[{"fvAEPg":{"attributes":{"dn":"uni/tn-{{TENANT}}/ap-{{Application Profile}}/epg-{{EPG}}","name":"{{EPG}}","rn":"epg-{{EPG}}","status":"created"},"children":[{"fvRsBd":{"attributes":{"tnFvBDName":"{{BD}}","status":"created","modified":"","children":[]}}}]}}]}
```

```
2 response: {"totalCount":"0","imdata":[]}
```

Body Cookies (1) Headers (14) Test Results

200 OK • 279 ms • 637 B

JSON Preview Visualize

```
1 {
2   "totalCount": "0",
3   "imdata": []
4 }
```

System Tenants Fabric

ALL TENANTS | Add Tenant | Tenant

Finance_Tenant

Quick Start

Finance_Tenant

Application Profiles

Finance_INT_EPG

Application EPGs

WEB-SRV

uSeg EPGs

Endpoint Security Groups

Networking

Lets make a .CSV file

Untitled spreadsheet ☆ 📁 ☁

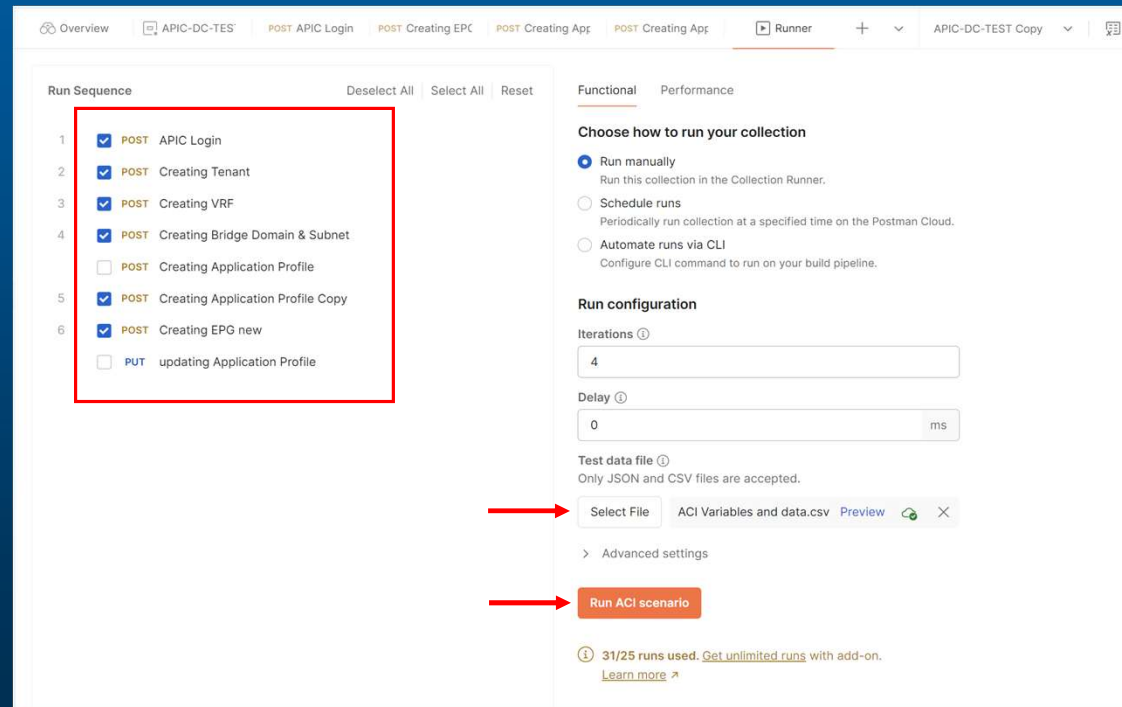
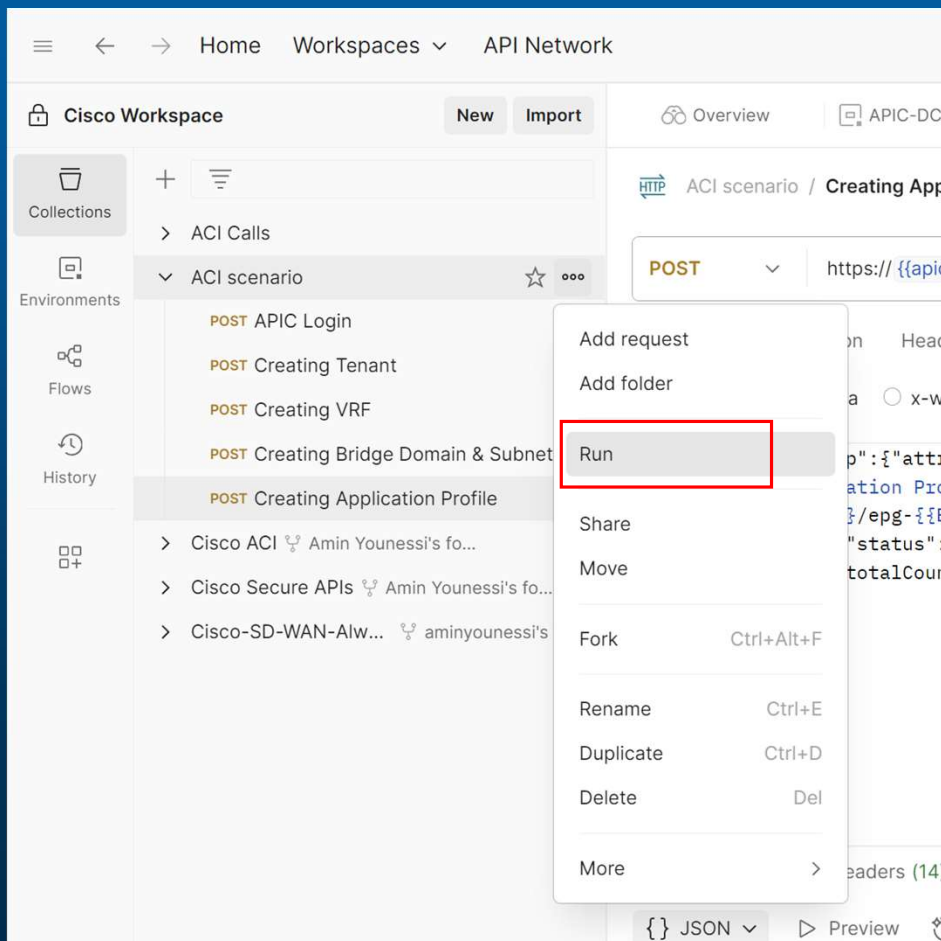
File Edit View Insert Format Data Tools Extensions Help

🔍 ↶ ↷ 🖨 📋 100% ▾ | £ % .0 ← .00 → 123 | Defaul... ▾ | - 10 + | **B** *I* ⌂ A

A1 ▾ | *fx* TENANT

	A	B	C	D	E	F
1	TENANT	BD	subnet	VRF	Application Profile	EPG
2	Marketing_Tenant	Marketing_DMZ_BD	10.1.1.1/24	MD-VRF	Marketing_Int_EPG	WEB_APP
3	Marketing_Tenant	Marketing_Internal_BD	10.1.2.1/24	MI-VRF	Marketing_Int_EPG	DB_APP
4	Finance_Tenant	Finance_DMZ_BD	10.2.1.1/24	FD-VRF	Finance_Int_EPG	NG_FW
5	Finance_Tenant	Finance_Internal_BD	10.2.2.1/24	FI-VRF	Finance_Int_EPG	File_SRV
6						
7						

Add the .CSV file into the Runner



Run results

ACI scenario - Run results

[Run Again](#)[Automate Run](#) [+ New Run](#) [Export Results](#)

Ran today at 06:52:55 · [View all runs](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	APIC-DC-TEST Copy	4	9s 676ms	20	302 ms

[All Tests](#) [Passed \(4\)](#) [Failed \(16\)](#) [Skipped \(0\)](#)

[View Summary](#)

Iteration 1

POST APIC Login

https://131.226.217.153/api/aaaLogin.json

No tests found

POST Creating Tenant

https://131.226.217.153/api/node/mo/uni/tn-Marketing_Tenant.json

No tests found

POST Creating VRF

https://131.226.217.153/api/node/mo/uni/tn-Marketing_Tenant/ctx-MD-VRF.json

No tests found

POST Creating Bridge Domain & Subnet

https://131.226.217.153/api/node/mo/uni/tn-Marketing_Tenant/BD-Marketing_DMZ_BD.json

FAIL

Response status code is 400 | AssertionError: expected 200 to deeply equal 400

FAIL

Response time is less than 200ms | AssertionError: expected 247 to be below 200

PASS

Response has required fields: totalCount and imdata

FAIL

Validate the error object structure in imdata | AssertionError: expected [] to have a length above +0 but got +0

FAIL

Error code is a non-empty string | AssertionError: expected [] not to be empty

POST Creating Application Profile Copy

https://131.226.217.153/api/node/mo/uni/tn-Marketing_Tenant/ap-Finance_INT_EPG.json

200

942 ms

2.457 KB

200

352 ms

637 B

200

258 ms

637 B

200

247 ms

637 B

200

317 ms

637 B

Run results

ACI scenario - Run results

Run AgainAutomate Run+ New RunExport Results

Ran today at 06:52:55 · [View all runs](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	APIC-DC-TEST Copy	4	9s 676ms	20	302 ms

All TestsPassed (4)Failed (16)Skipped (0)[View Summary](#)

PASSresponse has required fields: totalCount and imdata

FAILValidate the error object structure in imdata | AssertionError: expected [] to ...

FAILError code is a non-empty string | AssertionError: expected [] not to be emp...

POSTCreating Application Profile Copy
https://131.226.217.153/api/node/mo/uni/tn-Finance_Tenant/ap-Finance_INT_EPG.json200

No tests found

POSTCreating EPG new
https://131.226.217.153/api/node/mo/uni/tn-Finance_Tenant/ap-Finance_INT_EPG/epg-N...200

No tests found

Iteration 4

POSTAPIC Login
https://131.226.217.153/api/aaaLogin.json200

No tests found

POSTCreating Tenant
https://131.226.217.153/api/node/mo/uni/tn-Finance_Tenant.json400

No tests found

POSTCreating VRF
https://131.226.217.153/api/node/mo/uni/tn-Finance_Tenant/ctx-FI-VRF.json200

No tests found

1POSTACI scenario / Creating Tenant

2ResponseHeadersRequest400 · 233 ms · 536 B

3

4Pretty

1{
2 "totalCount": "1",
3 "imdata": [
4 {
5 "error": {
6 "attributes": {
7 "code": "103",
8 "text": "Cannot create Tenant
 (fvTenant); object uni/
 tn-Finance_Tenant already exists."
 }
6 }
5 }
4]
3 }
2 }
1 }
0 }

PostbotRunnerStart ProxyCookiesVaultTrash

Verify the configuration from GUI

The screenshot displays the Cisco APIC (Application Policy Infrastructure Controller) GUI. The top navigation bar includes the Cisco logo, the APIC title, and a user profile 'admin' with various status icons. Below this is a secondary navigation bar with tabs for System, Tenants, Fabric, Virtual Networking, Admin, Operations, Apps, and Integrations. The 'Tenants' tab is active, showing a list of tenants: ALL TENANTS, Add Tenant, and a search bar. The 'Finance_Tenant' is selected, and its configuration is displayed in the main area.

The left sidebar shows the 'Finance_Tenant' configuration tree. The 'Application Profiles' section is expanded, showing 'Finance_INT_EPG' and its sub-items 'Application EPGs' (File_SRV, NG_FW) and 'uSeg EPGs'. The 'Networking' section is also expanded, showing 'Bridge Domains' (Finance_DMZ_BD, Finance_Internal_BD) and their sub-items 'DHCP Relay Labels', 'ND Proxy Subnets', and 'Subnets'. The 'VRFs' section is expanded, showing 'FD-VRF' and 'FI-VRF'.

The main content area shows the 'Tenant - Finance_Tenant' configuration page. The 'Summary' tab is active, displaying a dashboard with six summary cards:

- Application EPGs:** 2 Total, 1 warning icon, 100% health.
- Endpoint Security Groups:** 0 Total.
- Bridge Domains:** 2 Total, 0 warning icon, 100% health.
- VRFs:** 2 Total, 0 warning icon, 100% health.
- L2Outs:** 0 Total.
- L3Outs:** 0 Total.

Additional cards for 'SR-MPLS L3Outs' and 'Contracts' are visible at the bottom.

Verify the configuration from GUI

The screenshot displays the Cisco APIC (Application Policy Infrastructure Controller) GUI. The top navigation bar includes the Cisco logo, the APIC title, and a user profile 'admin' with various system icons. Below this, a secondary navigation bar lists tabs: System, Tenants (selected), Fabric, Virtual Networking, Admin, Operations, Apps, and Integrations. A search bar for tenants is present, with 'Marketing_Tenant' selected from a list that also includes 'common', 'infra', 'sb', and 'mgmt'.

The left sidebar, titled 'Marketing_Tenant', shows a hierarchical tree structure. The 'Marketing_Tenant' folder is expanded, revealing sub-folders: 'Application Profiles' (containing 'Finance_INT_EPG' with its own sub-folders 'Application EPGs' and 'uSeg EPGs'), 'Networking' (containing 'Bridge Domains' and 'Marketing_Internal_BD'), and 'VRFs' (containing 'MD-VRF' and 'MI-VRF'). The 'Marketing_Internal_BD' folder is further expanded, showing 'DHCP Relay Labels', 'ND Proxy Subnets', and 'Subnets' with the IP address '10.1.2.1/24' listed under 'Subnets'.

The main content area is titled 'Tenant - Marketing_Tenant' and features a 'Summary' tab. It displays six summary cards for the tenant's configuration:

- Application EPGs:** 2 Total. Status: 2 warnings, 100% health.
- Endpoint Security Groups:** 0 Total.
- Bridge Domains:** 2 Total. Status: 2 warnings, 100% health.
- VRFs:** 2 Total.
- L2Outs:** 0 Total.
- L3Outs:** 0 Total.

Verify from Postman by using GET method

The screenshot displays the Postman application interface. The top navigation bar includes 'Home', 'Workspaces', and 'API Network'. The left sidebar shows a collection named 'Cisco Workspace' with a sub-collection 'ACI scenario'. The main workspace shows a 'GET' request to the URL `https://{{apic}}/api/node/mo/uni/tn-{{TENANT}}.json?query-target=subtree&rsp-subtree=full`. The request is highlighted with a red box. The response is a JSON object with a status of '200 OK'. The JSON body is displayed in the 'Body' tab, showing a hierarchical structure with 'totalCount', 'imdata', and 'fvTenant' details.

Request Details:

- Method: GET
- URL: `https://{{apic}}/api/node/mo/uni/tn-{{TENANT}}.json?query-target=subtree&rsp-subtree=full`

Response Details:

- Status: 200 OK
- Time: 1.22 s
- Size: 9.3 KB

JSON Response Body:

```
1 {
2   "totalCount": "59",
3   "imdata": [
4     {
5       "fvTenant": {
6         "attributes": {
7           "annotation": "",
8           "childAction": "",
9           "descr": "",
10          "dn": "uni/tn-Finance_Tenant",
11          "extMngdBy": "",
12          "lcOwn": "local",
13          "modTs": "2025-06-12T15:51:17.801+00:00",
14          "monPolDn": "uni/tn-common/monepg-default",
15          "name": "Finance_Tenant",
16          "nameAlias": "",
17          "ownerKey": "",
18          "ownerTag": "",
19          "status": "",
20          "uid": "15374",
21          "userdom": ":all:"
22        },
23        "children": [
24          {
25            "vnsSvcCont": {
```



Thank you