Cisco Catalyst Center Monitoring and Automation API Laboratory v1.0

Héctor Morales, PhD
hemorale@cisco.com
Cisco Systems Inc.

Executive Summary

This documents describes the version 1.0 of the Catalyst Center Monitoring and Automation Lab using Catalyst Center APIs. This lab is based on the Catalyst Center API workshop v3.0 (Morales 2025*a*), (Morales 2025*b*). This lab uses the standard Cisco Enterprise Networks Hardware Sandbox v4.1 available in Cisco dCloud (requires scheduling) and all the code is documented on my GitHub by accessing `https://github.com/hemorale/CatC-API-Lab`. The main goal of this lab is to show the basics of Catalyst Center automation with a basic Campus network. The document is divided in 3 main sections. Chapter 1 describes the Lab and the tools to work with the different assets, both physical and virtual. Chapter 2 talks about the basics of Catalyst Center APIs, using postman collections. Chapter 3 shows how to use Intent APIs interactively using Jupyter notebooks. Each of these Chapters are fully documented on the GitHub and this document provides the step-by-step to complete this lab. If you found any error or you have any issue or mistake, please send me an email to hemorale@cisco.com.

1 Lab Description

This initial section provides the description of the lab and the basics on how to use it. Figure 1 shows the way the lab is accessed. The lab requires access to the dCloud VPN (available in your session description) by using Cisco Anyconnect Client.

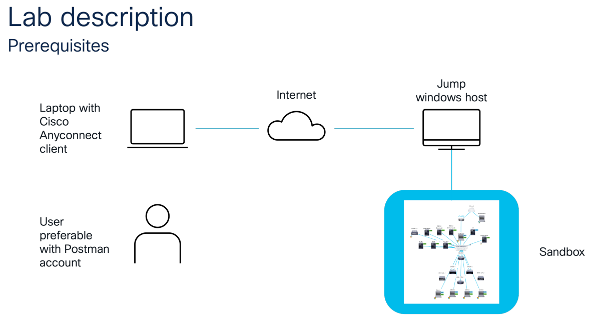

Lab description
Prerequisites
Laptop with Cisco Anyconnect client
Internet
Jump windows host
User preferable with Postman account
Sandbox

Figure 1: Lab Description

Figure 2 shows the logical topology on how the elements are connected. As could be seen, there is a mix of physical and virtual machines.

Here is the list of Virtual Machines:

- DNA Center 2.3.5.5.
- Identity Services Engine (ISE) 3.2 Patch 5 – (Deployed).

- Identity Services Engine (ISE) 3.2 – (Undeployed).
- Secure Network Analytics (Stealthwatch) 7.4.2.
- FlowCollector 7.4.2.
- Cisco Prime Infrastructure 3.10.4.
- Wireless LAN Controllers – C9800 running IOS-XE 17.12.3 code.
- Windows 10 Jump Host – Contains links to common URLs needed to view and configure the environment. Can also be used to TFTP files to/from the hardware devices. Can be used to pull files from Box.
- Ubuntu 20.04.3 – Can be used as a script server and for programing tools.
- Windows Server 2019 – Can be configured to provide identity, DHCP, DNS, etc.
- Windows 10 Clients – Used to simulate network clients participating in the environment. Can be used to test segmentation, host onboarding, policy implementation, etc.

Here is the list of the hardware devices:

- ISR 4331 Router - 17.09.02a IOS-XE Code.
- Catalyst 9300 Switches - 17.09.02 IOS-XE Code with Embedded Wireless Controller (EWC) and ThousandEyes Enterprise Agent.
- 9117 Access Point.
- 4800 Access Points.
- Silex Controllers (2 Wired NICs)

The physical topology as indicated in Figure 3, describes the way hardware is connected as well as the IP addresses. Once you are connected via the VPN, you are in the green IP addressing area. From there you can jump to any device.

To access the environment, you have to use the Cisco Anyconnect®client, using the host destination, user and password provided in your session details (ask your instructor in case you don't have your session details). See figure 4 for anyconnect details.

The way to connect to your environment is as follows:

1. Connect to the demo with Cisco AnyConnect®VPN. Find your AnyConnect information in the Session Details for your dCloud session.
2. Use your local browser to connect to URLs as outlined in the table 1 in the Virtual Machines section below.
3. Use your local RDP client to connect to RDP-capable Virtual Machines as outlined in Virtual Machines section that follows. The recommended RDP client is the Windows App, that can be downloaded directly from here `https://apps.microsoft.com/detail/9n1f85v9t8bn?hl=en-US&gl=US`

Table 1 shows the IP addressing and credentials for each of the devices available on the sandbox.

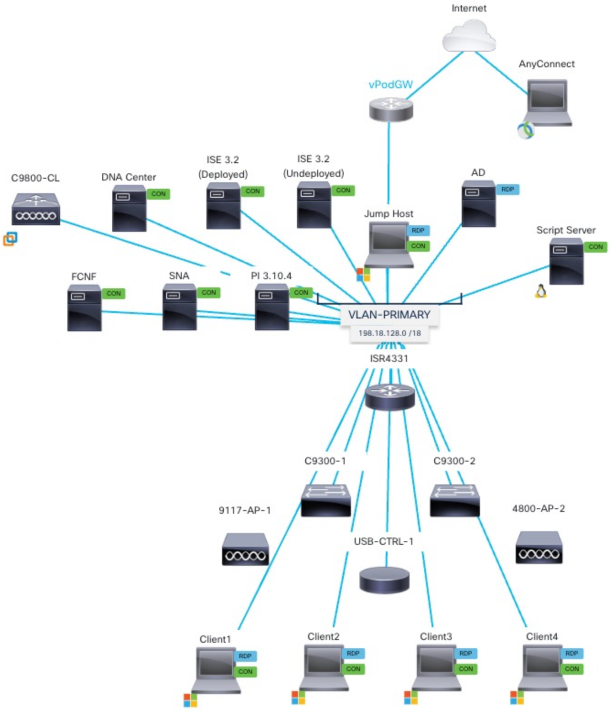


Figure 2: Logical Topology

1.1 Get familiar to use the lab

The easiest and more convenient access to the lab resources is by using the Jump Host, which is a windows virtual machine that provides access to all devices, both virtual and physical from the host. To use the Jump Host, follow these steps:

1. Connect to the dCloud environment using anyconnect client, as shown in Figure 5:

2. Open the Windows App and enter the credentials for the jump host, as shown in Figure 6:

3. Once on the Windows Jump host server, you will see the applications you will use in this lab: Google Chrome for the web-based applications such as Catalyst Center, mRemoteMG, for accessing the hardware console via SSH and Postman for the APIs testing. Figure 7:

4. Inside the Windows jump host, click on the Google Chrome application on the left. You will see different applications bookmarked on the top. For the purpose of this lab you will use the Web HW Console and Catalyst Center mostly, but if there is time, you will use the WLC as well. You will need to open a new tab to access individual web UIs.

1.2 Brief Catalyst Center tour

Catalyst Center (CatC) is the management platform for Catalyst-based campus deployments. CatC is designed to simplify network operations and enable IT admins to deliver business outcomes, smarter and faster at scale. CatC provides network visibility, observability and insights. It also provides a full set of APIs that allows to automate the network using Infrastructure as Code, with higher levels of security and at the same time, provide telemetry-based interpretations using Machine Learning models, based on 40 years of expertise on enterprise networking.

CatC has 4 pillars:

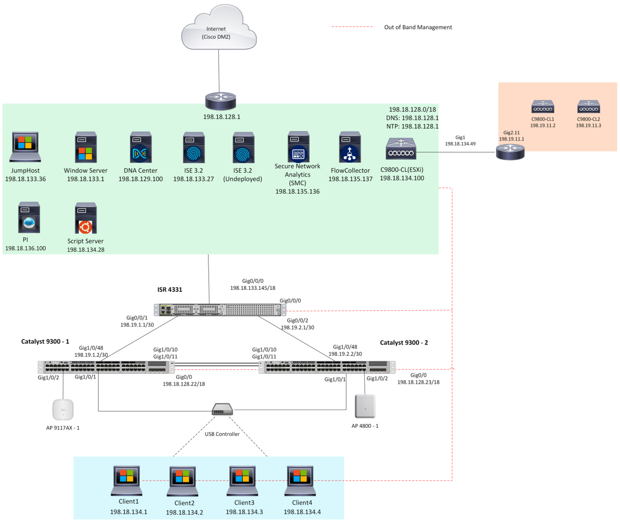
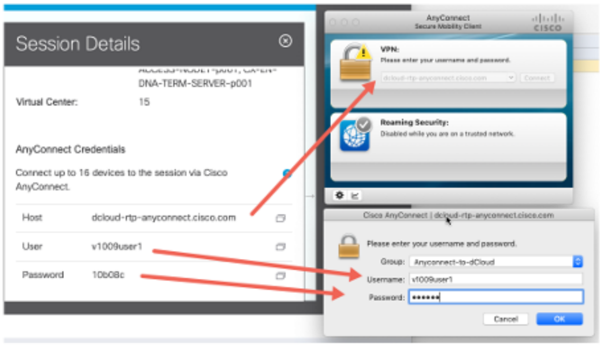


Figure 3: Physical Topology



Session Details

ACCESS-NODE1-p001, CA-EN-
DNA-TERM-SERVER-p001

Virtual Center: 15

AnyConnect Credentials

Connect up to 16 devices to the session via Cisco AnyConnect.

Host	dcloud-rtp-anyconnect.cisco.com
User	v1009user1
Password	10b08c

AnyConnect
Secure Mobility Client

VPN:
Please enter your username and password.

dcloud-rtp-anyconnect.cisco.com

Roaming Security:
Disabled while you are on a trusted network.

Cisco AnyConnect | dcloud-rtp-anyconnect.cisco.com

Please enter your username and password.

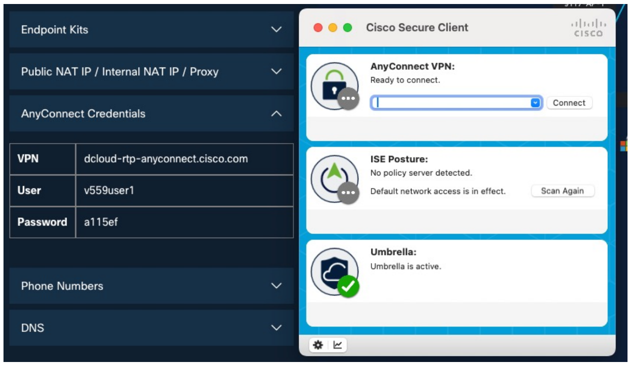
Group: Anyconnect-to-dCloud
Username: v1009user1
Password: ••••••

Figure 4: Accessing your environment using Cisco Anyconnect®

- **NetOps:** Increases network management scale, providing customers with business resiliency, continuity, and quick time to value. CatC NetOps enables customers to achieve Compliance of network with configuration policies and it also provides automation to simplify the creation and maintenance of customers networks.
- **SecOps:** Improves security at the network level. AI-driven security is used to classify endpoints and enforce security policies for a complete zero trust workplace solution. Automates end point visibility, classification, and grouping.
- **DevOps:** Improves service delivery using Infrastructure as Code. Mature APIs, SDKs, and closed-loop integrations to simplify and streamline ecosystem integration with different services and platforms such as ServiceNow. DevOps module provides faster service delivery using API-based automation workflows, making CatC a headless, efficient operations for large networks and Managed Service Providers. At the same time, by using event-driven management APIs, early issue detection is provided and integration with 3rd party platforms through enhanced notification channels is enabled.
- **AIOps:** Provides improved performances insights. This module provides reduced proactive problem resolution through faster Root Cause Analysis. AI-driven visibility, observability, insights, and troubleshooting ensures the health your customers applications, infrastructure and user experience

IP Address	Name	Username	Password	Preferred Access Method
Entry	ISE 3.0 Undeployed	Entry	Entry	VM Console via dCloud UIChrome or Firefox (once configured)
198.18.133.27	ISE 3.0 (Deployed)	admin	C1sco12345	Chrome or Firefox
198.18.129.100	DNA Center	admin	C1sco12345	Chrome or Firefox
198.18.135.136	Secure Network Analytics (Stealthwatch)	admin	C1sco12345	Chrome or Firefox
198.18.135.137	FlowCollector	admin	C1sco12345	Chrome or Firefox
198.19.11.2	C9800-CL1	admin	C1sco12345	Chrome or FirefoxSSH or Telnet
198.19.11.3	C9800-CL2	admin	C1sco12345	Chrome or FirefoxSSH or Telnet
198.18.134.49	ESXi-Router	admin	C1sco12345	SSH or Telnet
198.18.136.100	Cisco Prime Infrastructure	admin	C1sco12345	Chrome or Firefox
198.18.134.28	Script Server	root	C1sco12345	Chrome or FirefoxSSH
198.18.133.36	Jump Host	admin	C1sco12345	RDP
198.18.133.1	AD	admin	C1sco12345	RDP
198.18.134.1	Client1	admin	C1sco12345	RDP
198.18.134.2	Client2	admin	C1sco12345	RDP
198.18.134.3	Client3	admin	C1sco12345	RDP
198.18.134.4	Client4	admin	C1sco12345	RDP

Table 1: Virtual Machine Addressing and Credentials



VPN	dcloud-rtp-anyconnect.cisco.com
User	v559user1
Password	a115ef

AnyConnect VPN:
Ready to connect.

ISE Posture:
No policy server detected.
Default network access is in effect.

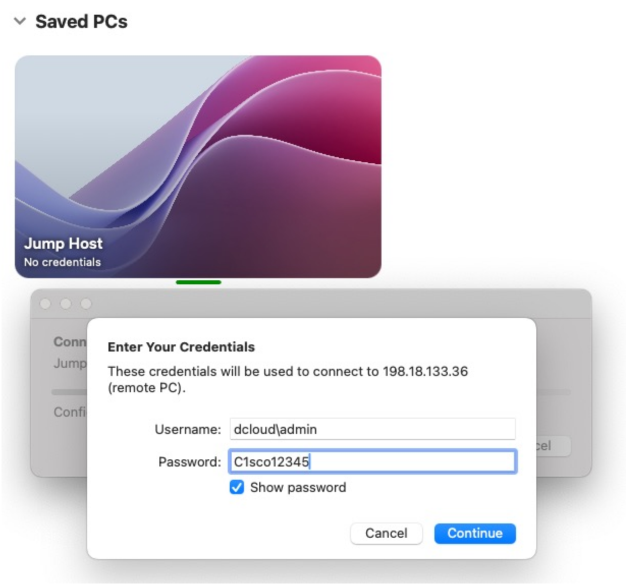
Umbrella:
Umbrella is active.

Figure 5: Step 1 - Connect using anyconnect.

1.3 Lab 1 - Get to know your Catalyst Center

This lab has the goal to provide a basic knowledge of the platform. If you already have worked with Catalyst Center, you can jump to the next section.

1. Access your lab environment by using Cisco Anyconnect®as described previously.
2. Connect to the Windows jump host using the Windows App.
3. Open Google Chrome and click on the DNA Center bookmark on the top left.
4. Use the correct credentials to get access to Catalyst Center.
5. The screen is divided in these sections:
   - Assurance Summary, which provides a summary of healt scores, critical issues, trends and insights. (scroll down).
   - Network Snapshot, which allows to add sites, import images and update devices and also provides a summary of network devices (inventory), QoS policies, network profiles and devices licenses (scroll down).
   - Tools, which are all the tools available for CatC management.
6. Take a few minutes visiting the different tools. At this time, you won't see any report or tool populated as the CatC is just installed. See Figure 9.
7. Click on the 3 vertical line on the top left side (Next to the Cisco DNA Center label). That will open the main menu. The menu has the following options:



Enter Your Credentials

These credentials will be used to connect to 198.18.133.36 (remote PC).

Username: dcloud\admin

Password: C1sco12345

☑ Show password

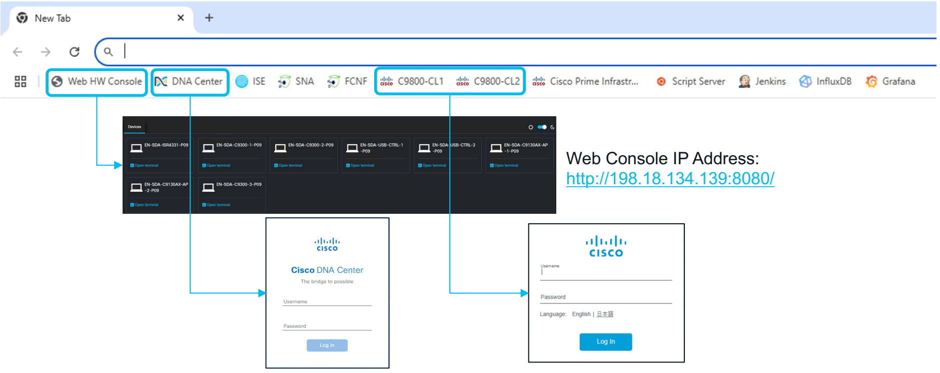
Figure 6: Step 2 – Using Windows App, connect to Jump host.

Figure 7: Step 3 – Using Windows Jump host to access lab applications.

- Design.
- Policy.
- Provision.
- Assurance.
- Workflows.
- Tools.
- Platform.
- Activities.
- Report.
- System.

Figure 10 shows this menu and the options. Take a few minutes to take a look and get familiar with the different options.

8. **NOTE: if you want to return to main page click on the Cisco DNA Center label at the top left. For getting the menu, click again on the 3 vertical lines on the top left.**



Web Console IP Address:
http://198.18.134.139:8080/

Catalyst Center IP Address:
https://198.18.129.100/

WLC IP Address:
https://198.19.11.2/webui/
https://198.19.11.3/webui/

Figure 8: Step 4 - Use Chrome to jump to the web UI as needed.

2 Catalyst Center APIs using Postman Collections

This lab uses the Postman Collections on the `https://github.com/hemorale/CatC-API-Lab`. Due to the limited time we will have, only a couple of collections will be visited but you are free to play with the rest on your own. Download the collections and if you find an error, please email me to hemorale@cisco.com.

2.1 Lab 2 - Load Postman collections into your environment

The goal of this lab is to get you familiar with the CatC Platform APIs.

1. Connect to the dCloud lab using your assigned PoD via Cisco Anyconnect.
2. Using the Windows App, connect into the Jump Host.
3. Open Chrome and download the Postman collections in the jump host.
4. Open Postman. Login into your account or create a new account. You will need it to upload collections and upload the environment variables. **NOTE: if you don't login to an account you can't create what is necessary to use the collections and variables.**
5. Click on Collections. Then click on Import. Select the json files. See Figure 11.

These are the json postman collections you have downloaded. File names corresponds to the description below:

- Catalyst Center API Environment postman environment - provides all environment variables. This should be import in the Environments and this will be used for the rest of the collections.
- Catalyst Center API LAB 100 - Build Hierarchy postman collection – builds the initial CatC network hierarchy.

As described previously, we won't have to test them all but you are more than welcome to download them and in the future, reserve the dCloud to play with them. Again, if you find an error or you have feedback, email me to hemorale@cisco.com.

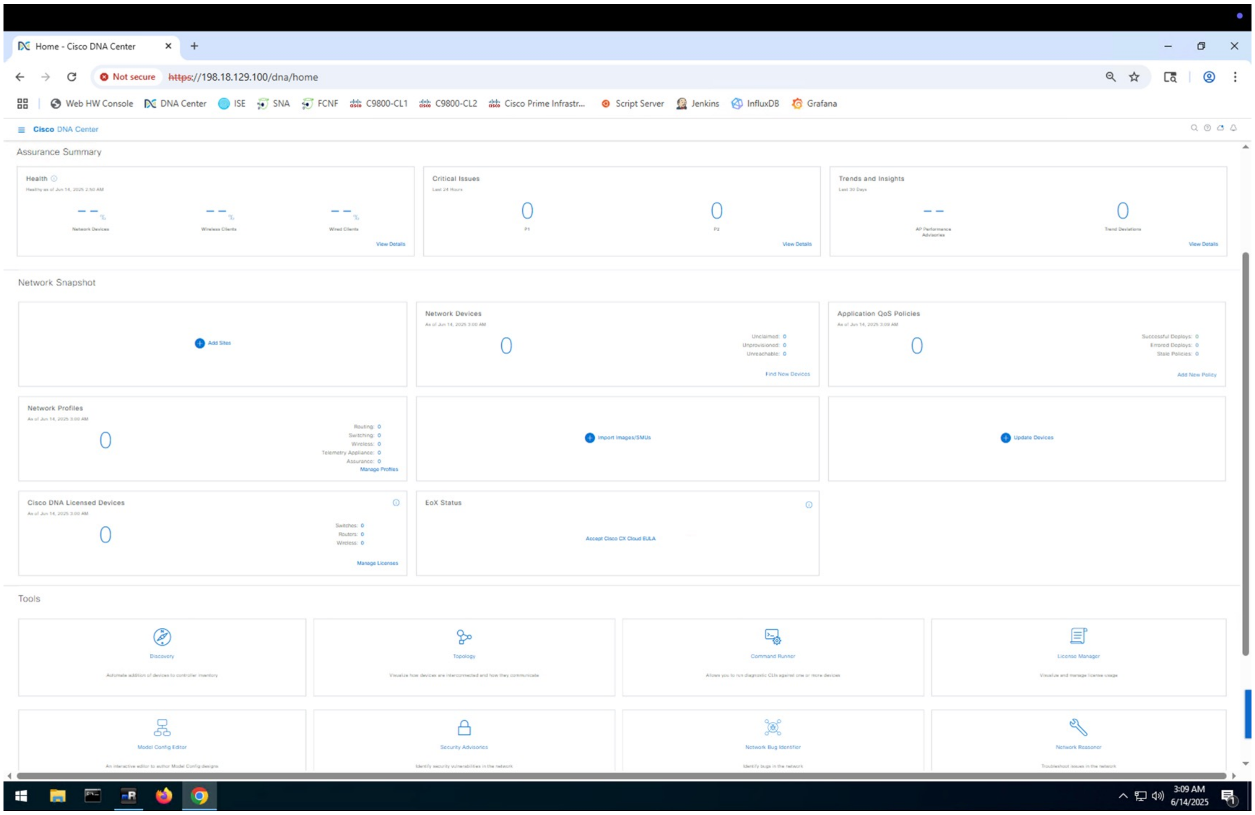


Figure 9: Catalyst Center initial dashboard

2.2 Lab 3 - Catalyst Center Initial Configuration

This lab has the goal to create the basic configuration for the CatC lab environment.

1. In the Windows jump host, open Chrome and click on the DNA Center bookmark (alternatively as you are connected to the dCloud VPN, you can do this in your browser as well using the CatC IP address).

2. Login into CatC using the proper credentials.

3. Navigate to Tools (scroll down).

4. Click on the **Discovery** workflow. It will take you to the discovery dashboard.

5. Click on **Add Discovery**.

6. Click on Next.

7. You are now in the **Discover devices** page. You have to name your discovery job (See the Figure 12). The discovery type is CDP and the IP address is the ISR4331 IP Address 198.18.133.145.

8. You need now to provide the access credentials to the devices. Click on the CLI tab. Add Credentials using the **admin/C1sco12345/C1sco12345** username/password/enable password.

9. Now click on the SNMPv2c Read tab. Add a new V2C credential for the **SNMP Global RO** as the Name/Description and the Read Community as **RO**.

10. Click Next.

11. On Advanced Settings, select **SSH** in Protocol order. For the rest, leave in default. Click next.

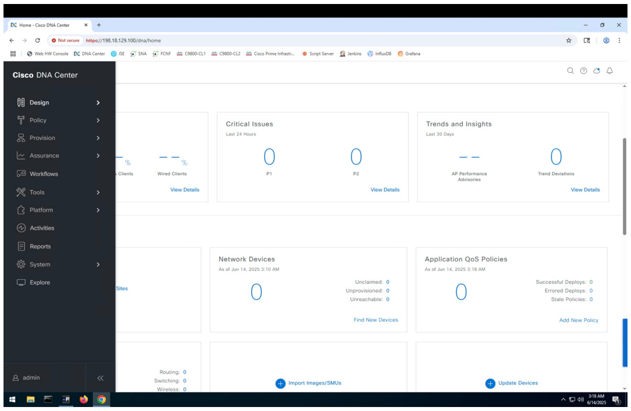


Figure 10: CatC main menu

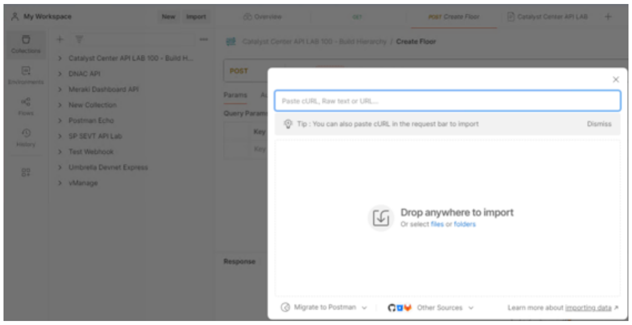


Figure 11: Postman collections for CatC Lab 2

12. In the Assign Device to Site, select **Skip site assignment for now** (This will be another lab).

13. You are ready to schedule the discovery workflow job. Select **Now** and click on **Start Discovery and Telemetry.**

14. You will see that the the Discovery job is in progress. Click on **View Discovery** to see the Status.

15. From the dCloud Portal, open the ISR Console and see the messages. You will see messages like the one show in Figure 13

16. From the CatC dashboard, you can see the status of the discovery process as well as devices discovered. See Figure 14 as a reference. You will see something similar.

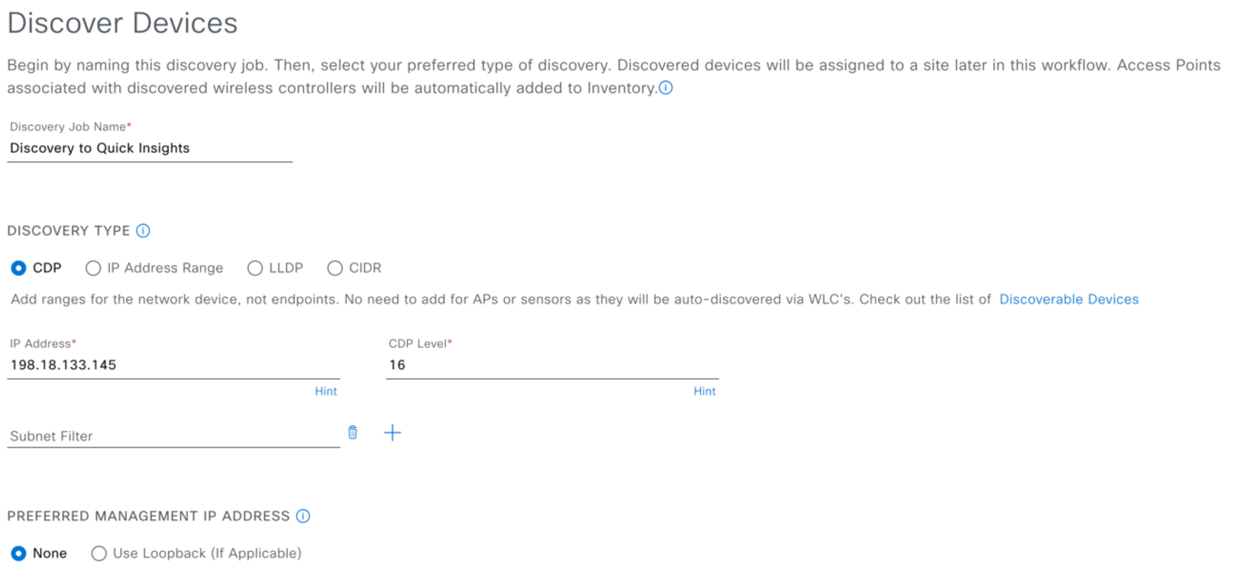
2.3 Lab 4 - Setting up your Postman environment

The goal of these labs is to understand the basics of CatC Intent APIs. To complete this lab, we will use Postman, so we need to set it up. Postman is already installed in the Windows jump host. Alternatively, you can use your own computer and use your own Postman installed locally. Accessing to the dCloud lab requires to connect to the VPN using anyconnect.

1. In Jump host (or your own computer connected to the dCloud VPN), open Postman.

2. Click on the collection Catalyst Center API LAB 100 - Build Hierarchy.

3. Create a new workspace in Postman. **NOTE: you need to login to your account. If you don't have an account, you will to create a new one.** See the Figure 15 for the steps you have to take in Postman.



Discover Devices

Begin by naming this discovery job. Then, select your preferred type of discovery. Discovered devices will be assigned to a site later in this workflow. Access Points associated with discovered wireless controllers will be automatically added to Inventory.ⓘ

Discovery Job Name*
Discovery to Quick Insights

DISCOVERY TYPE ⓘ

● CDP ○ IP Address Range ○ LLDP ○ CIDR

Add ranges for the network device, not endpoints. No need to add for APs or sensors as they will be auto-discovered via WLC's. Check out the list of Discoverable Devices

IP Address*
198.18.133.145

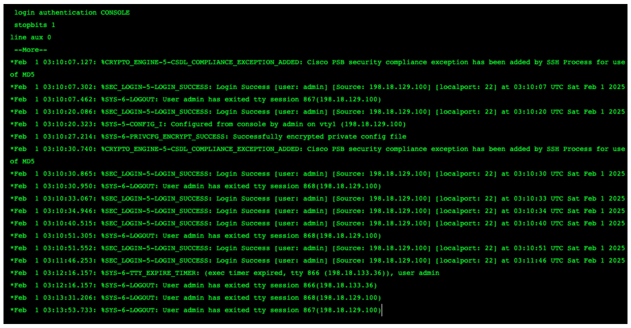
CDP Level*
16

Subnet Filter

PREFERRED MANAGEMENT IP ADDRESS ⓘ

● None ○ Use Loopback (If Applicable)

Figure 12: Discover Devices



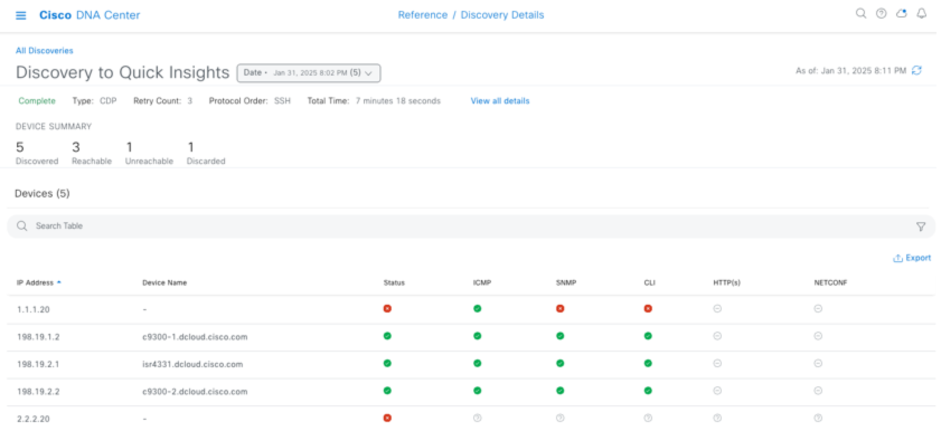
```
 login authentication CONSOLE
 stopbits 1
line aux 0
 --More--
*Feb  1 03:10:07.127: %CRYPTO_ENGINE-5-CSDL_COMPLIANCE_EXCEPTION_ADDED: Cisco PSB security compliance exception has been added by SSH Process for use
of MD5
*Feb  1 03:10:07.302: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: admin] [Source: 198.18.129.100] [localport: 22] at 03:10:07 UTC Sat Feb 1 2025
*Feb  1 03:10:07.462: %SYS-6-LOGOUT: User admin has exited tty session 867(198.18.129.100)
*Feb  1 03:10:20.086: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: admin] [Source: 198.18.129.100] [localport: 22] at 03:10:20 UTC Sat Feb 1 2025
*Feb  1 03:10:20.323: %SYS-5-CONFIG_I: Configured from console by admin on vty1 (198.18.129.100)
*Feb  1 03:10:27.214: %SYS-6-PRIVCFG_ENCRYPT_SUCCESS: Successfully encrypted private config file
*Feb  1 03:10:30.740: %CRYPTO_ENGINE-5-CSDL_COMPLIANCE_EXCEPTION_ADDED: Cisco PSB security compliance exception has been added by SSH Process for use
of MD5
*Feb  1 03:10:30.865: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: admin] [Source: 198.18.129.100] [localport: 22] at 03:10:30 UTC Sat Feb 1 2025
*Feb  1 03:10:30.950: %SYS-6-LOGOUT: User admin has exited tty session 868(198.18.129.100)
*Feb  1 03:10:33.067: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: admin] [Source: 198.18.129.100] [localport: 22] at 03:10:33 UTC Sat Feb 1 2025
*Feb  1 03:10:34.946: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: admin] [Source: 198.18.129.100] [localport: 22] at 03:10:34 UTC Sat Feb 1 2025
*Feb  1 03:10:40.515: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: admin] [Source: 198.18.129.100] [localport: 22] at 03:10:40 UTC Sat Feb 1 2025
*Feb  1 03:10:51.305: %SYS-6-LOGOUT: User admin has exited tty session 868(198.18.129.100)
*Feb  1 03:10:51.552: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: admin] [Source: 198.18.129.100] [localport: 22] at 03:10:51 UTC Sat Feb 1 2025
*Feb  1 03:11:46.253: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: admin] [Source: 198.18.129.100] [localport: 22] at 03:11:46 UTC Sat Feb 1 2025
*Feb  1 03:12:16.157: %SYS-6-TTY_EXPIRE_TIMER: (exec timer expired, tty 866 (198.18.133.36)), user admin
*Feb  1 03:12:16.157: %SYS-6-LOGOUT: User admin has exited tty session 866(198.18.133.36)
*Feb  1 03:13:31.206: %SYS-6-LOGOUT: User admin has exited tty session 868(198.18.129.100)
*Feb  1 03:13:53.733: %SYS-6-LOGOUT: User admin has exited tty session 867(198.18.129.100)
```

Figure 13: ISR Console messages

4. Import postman collection. From the GitHub, you have downloaded the postman collection. You have to import it to your newly created workspace. See Figure 16 to see the actions you need to take to import the collection.

5. Switch environment. In the collection, I have created all the necessary variables you need for the lab, however you have to switch to this environment to use the variable set. Click in any entry of the collection. You will identify a variable because it is in double curly brackets ({{}}). Position the cursor on the variable, for example {{CC}}. There will be a window showing options like ”Enter Value” and below you will see ”Switch Environment”. Click and then select you Catalyst Center API Environment. See the Figure 17 for reference.

6. Verify your environment. Click on the first collection to get the Token. If your environment is correctly switched to use the lab variables, you will see the color of the variable {{CC}} in blue. That indicates your environment is correctly set. **NOTE: call your instructor if you are having issues.** See the Figure as reference.



Discovery to Quick Insights   Date • Jan 31, 2025 8:02 PM (5)   As of: Jan 31, 2025 8:11 PM

Complete   Type: CDP   Retry Count: 3   Protocol Order: SSH   Total Time: 7 minutes 18 seconds   View all details

DEVICE SUMMARY

5 Discovered   3 Reachable   1 Unreachable   1 Discarded

Devices (5)

IP Address	Device Name	Status	ICMP	SNMP	CLI	HTTP(s)	NETCONF
1.1.1.20	-	✗	✓	✗	✗	⊖	⊖
198.19.1.2	c9300-1.dcloud.cisco.com	✓	✓	✓	✓	⊖	⊖
198.19.2.1	isr4331.dcloud.cisco.com	✓	✓	✓	✓	⊖	⊖
198.19.2.2	c9300-2.dcloud.cisco.com	✓	✓	✓	✓	⊖	⊖
2.2.2.20	-	✗	?	?	?	?	?

Figure 14: Discovery process and status

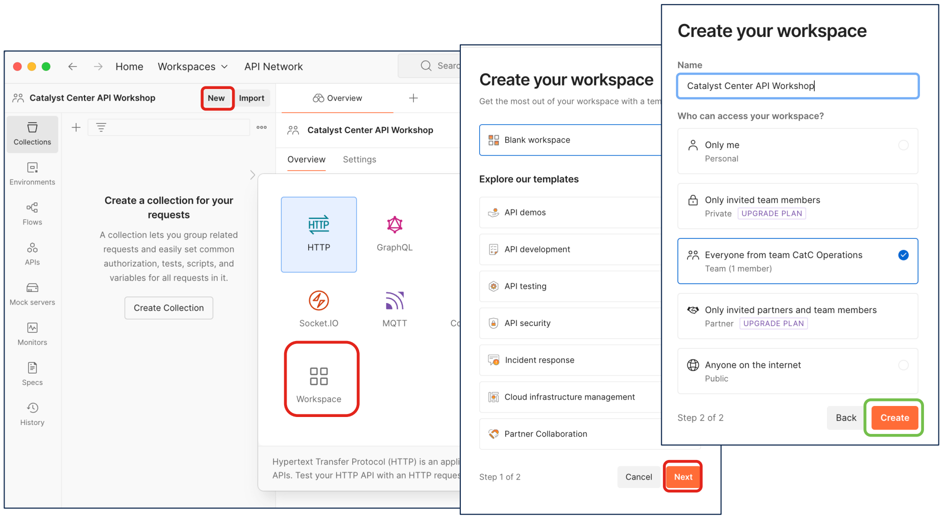

Home
Workspaces
API Network
Catalyst Center API Workshop
New
Import
Overview
Collections
Environments
Flows
APIs
Mock servers
Monitors
Specs
History
Create a collection for your requests
A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.
Create Collection
Catalyst Center API Workshop
Overview
Settings
HTTP
GraphQL
Socket.IO
MQTT
Workspace
Hypertext Transfer Protocol (HTTP) is an appli
APIs. Test your HTTP API with an HTTP reques
Create your workspace
Get the most out of your workspace with a tem
Blank workspace
Explore our templates
API demos
API development
API testing
API security
Incident response
Cloud infrastructure management
Partner Collaboration
Step 1 of 2
Cancel
Next
Create your workspace
Name
Catalyst Center API Workshop
Who can access your workspace?
Only me
Personal
Only invited team members
Private UPGRADE PLAN
Everyone from team CatC Operations
Team (1 member)
Only invited partners and team members
Partner UPGRADE PLAN
Anyone on the internet
Public
Step 2 of 2
Back
Create

Figure 15: Create a new workspace

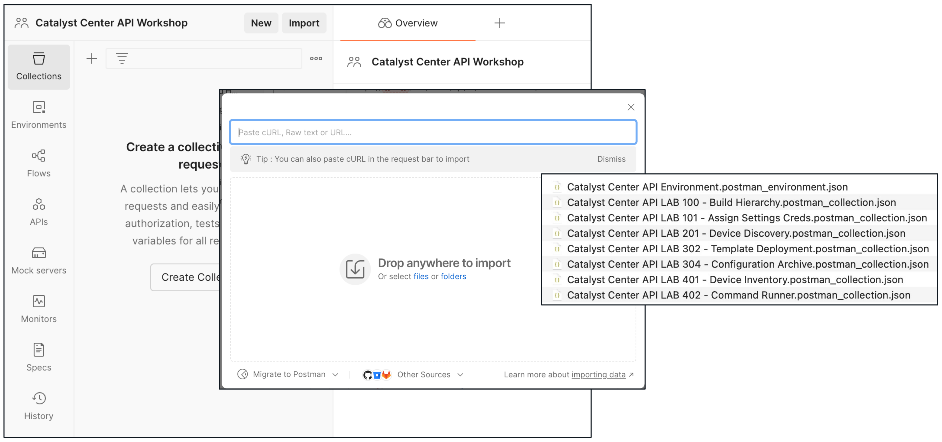
2.4 Lab 5 - Creating a hierarchy using Postman

This lab will show you how to create a hierarchy using CatC APIs. A hierarchy in CatC is needed to have a logical configuration of the network elements placed in logical entities named as Sites, Areas, Building and Floors. In summary, at the end of this lab, the collection will create a hierarchy like the one shown in Figure 19.

1. Creating hierarchy using Postman. Get the Token. Using the first instruction of the collection, you will need to get the Token. The Token is necessary for every API call. See the Figure 20 for reference. Remember to save the Token as you will need to use it. **NOTE**: it is likely that the Token will expiry over the time, so you will need to refresh the Token. As a recommendation, when you create your scripts, get the token in your modules.

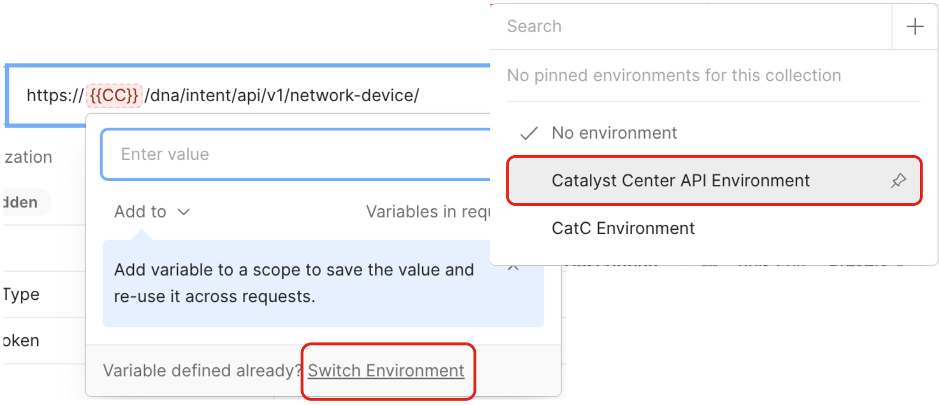
2. Get the Site Ids. The second instruction in the collection is to get the Site Ids from the CatC configuration. See Figure 21 for reference of this step.

3. You will see a response similar to the one shown in Figure 22



- Catalyst Center API Environment.postman_environment.json
- Catalyst Center API LAB 100 - Build Hierarchy.postman_collection.json
- Catalyst Center API LAB 101 - Assign Settings Creds.postman_collection.json
- Catalyst Center API LAB 201 - Device Discovery.postman_collection.json
- Catalyst Center API LAB 302 - Template Deployment.postman_collection.json
- Catalyst Center API LAB 304 - Configuration Archive.postman_collection.json
- Catalyst Center API LAB 401 - Device Inventory.postman_collection.json
- Catalyst Center API LAB 402 - Command Runner.postman_collection.json

Figure 16: Import collection into your workspace



https://{{CC}}/dna/intent/api/v1/network-device/

Add variable to a scope to save the value and re-use it across requests.

Variable defined already? Switch Environment

Figure 17: Switch Environment to choose the lab variables

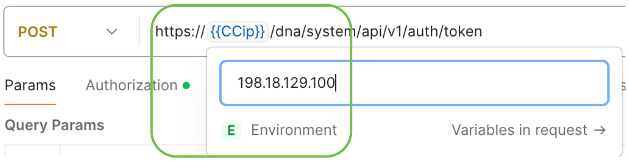
4. Create an area. Now run the third command on the collection to create an area. See the Figure 23 for more reference details.

5. Run again the previous command, what did you notice? Go to Catalyst Center Dashboard and open Network Topology. What do you see now?

6. Try on your own. The next command on the collection is to add a Building. **HINT: use the Catalyst Center Platform API reference for completing this task**

7. Are you observing a 202 Accepted API response? This is called Asynchronous operations. When the Catalyst Center platform returns a 202 (Accepted) HTTP status code, the result body includes a task ID and a URL that you can use to query for more information about the asynchronous task that your original request spawned. For example, you can use this information to determine whether a lengthy task has completed. Go into CatC dashboard and select the menu, then Platform and then Runtime Dashboard. You will see something similar to the Figure 24

8. Use CatC API Platform. Check asynchronous operations. In order to get these operations, you need to use another API called Get Business API Execution Details. Use CatC Dashboard. Go into the Platform and run this API from CatC platform. Use the Task ID. See the Figure 25 for details.


POST

https:// {{CCip}} /dna/system/api/v1/auth/token

198.18.129.100

E Environment

Variables in request →

Figure 18: Verify your variables correctly set in the environment

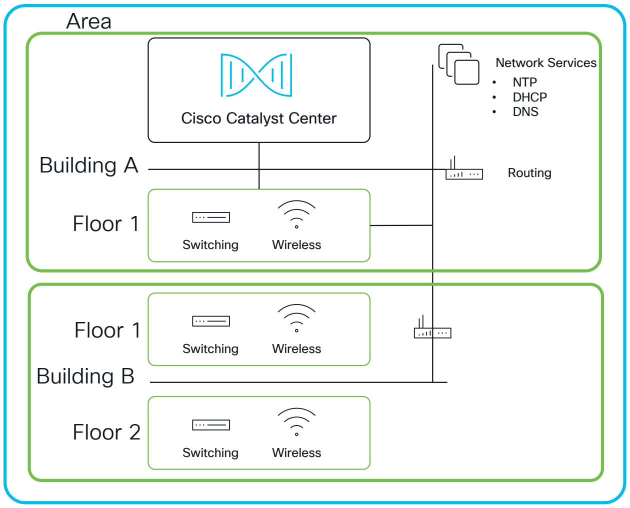
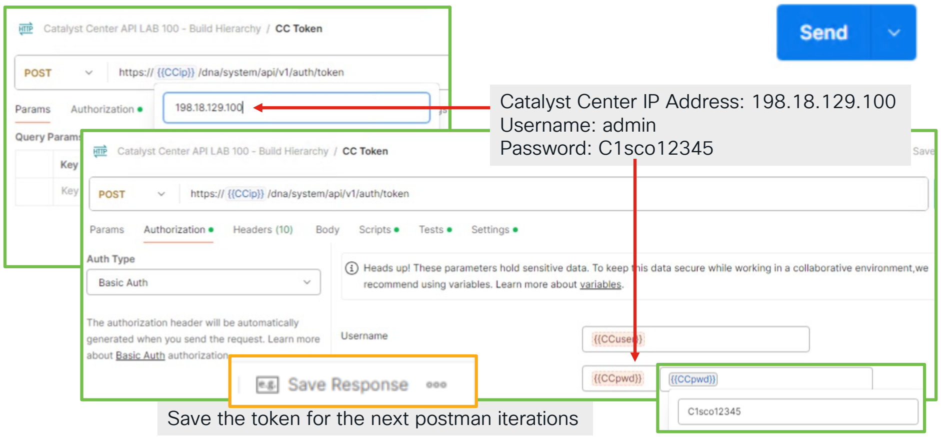


Figure 19: Basic Topology


Catalyst Center API LAB 100 - Build Hierarchy / **CC Token**

POST https:// {{CCip}} /dna/system/api/v1/auth/token

198.18.129.100

Catalyst Center IP Address: 198.18.129.100
Username: admin
Password: C1sco12345

Catalyst Center API LAB 100 - Build Hierarchy / **CC Token**

POST https:// {{CCip}} /dna/system/api/v1/auth/token

Auth Type Basic Auth

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

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 {{CCuser}}

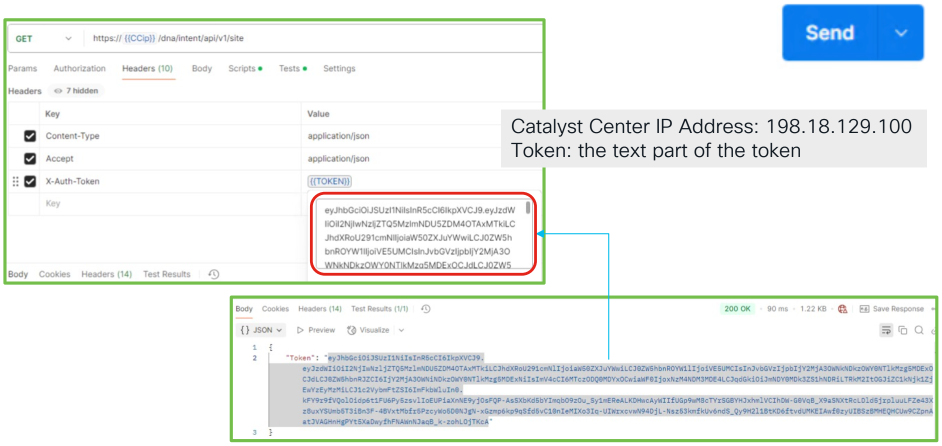
{{CCpwd}}

{{CCpwd}}

C1sco12345

Save the token for the next postman iterations

Figure 20: Get Token



https:// {{CCip}} /dna/intent/api/v1/site

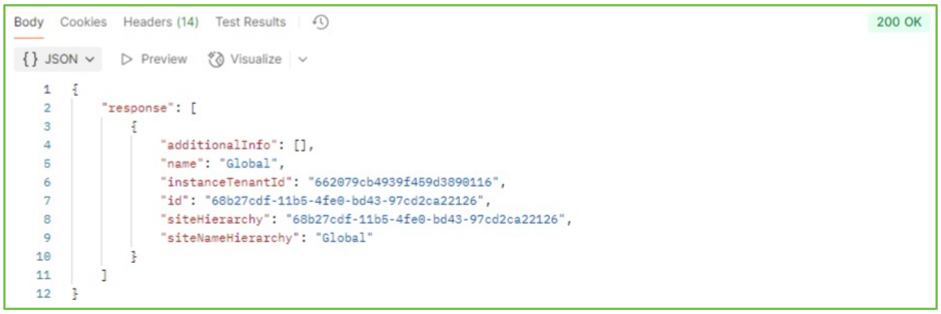
	Key	Value
☑	Content-Type	application/json
☑	Accept	application/json
☑	X-Auth-Token	{{TOKEN}}
	Key	

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdW
IiOiI2NjIwNzljZTQ5MzlmNDU5ZDM4OTAxMTkiLC
JhdXRoU291cmNlIjoiaW50ZXJuYWwiLCJ0ZW5h
bnROYW1lIjoiVE5UMCIsInJvbGVzIjpbIjY2MjA3O
WNkNDkzOWY0NTlkMzg5MDExOCJdLCJ0ZW5
```

Catalyst Center IP Address: 198.18.129.100
Token: the text part of the token

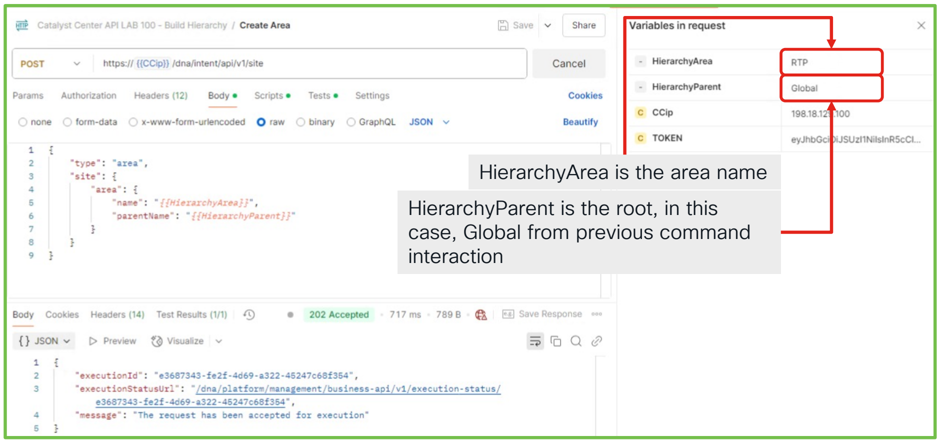
```
{
    "Token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.
        eyJzdWIiOiI2NjIwNzljZTQ5MzlmNDU5ZDM4OTAxMTkiLCJhdXRoU291cmNlIjoiaW50ZXJuYWwiLCJ0ZW5hbnROYW1lIjoiVE5UMCIsInJvbGVzIjpbIjY2MjA3OWNkNDkzOWY0NTlkMzg5MDExO
        CJdLCJ0ZW5hbnRJZCI6IjY2MjA3OWNiNDkzOWY0NTlkMzg5MDExNiIsImV4cCI6MTczODQ0MDYxOCwiaWF0IjoxNzM4NDM3MDE4LCJqdGkiOiJmNDY0MDk3ZS1hNDRiLTRkM2ItOGJiZC1kNjk1Zj
        EwYzEyMzMiLCJ1c2VybmFtZSI6ImFkbWluIn0.
        kFY9r9fVQol0idp6t1FU6Py5zsvlIoEUPiaXnNE9yjOsFQP-AsSXbKd5bYImqb09zOu_Sy1mEReALKDHwcAyWIIfUGp9wM8cTYrSGBYHJxhmlVCIhDW-G0VqB_X9aSNXtRcLDld5jrpluuLFZe43X
        z8uxYSUmb5T3iBn3F-4BVxtMbfr5PzcyWo5D0NJgN-xGzmp6kp9qSfd5vC10nIeMIXo3Iq-UIWrxcvwN94DjL-Nsz53kmfkUv6ndS_Qy9H2l1BtKD6ftvdUMKEIAwf0zyUIBSzBMHEQHCUw9CZpnA
        atJVAGHnHgPYt5XaDwyfhFNAWnNJaqB_k-zohL0jTKcA"
}
```

Figure 21: Get Site IDs



```
{
    "response": [
        {
            "additionalInfo": [],
            "name": "Global",
            "instanceTenantId": "662079cb4939f459d3890116",
            "id": "68b27cdf-11b5-4fe0-bd43-97cd2ca22126",
            "siteHierarchy": "68b27cdf-11b5-4fe0-bd43-97cd2ca22126",
            "siteNameHierarchy": "Global"
        }
    ]
}
```

Figure 22: Site IDs API response



https:// {{CCip}} /dna/intent/api/v1/site

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "type": "area",
    "site": {
        "area": {
            "name": "{{HierarchyArea}}",
            "parentName": "{{HierarchyParent}}"
        }
    }
}
```

Variables in request

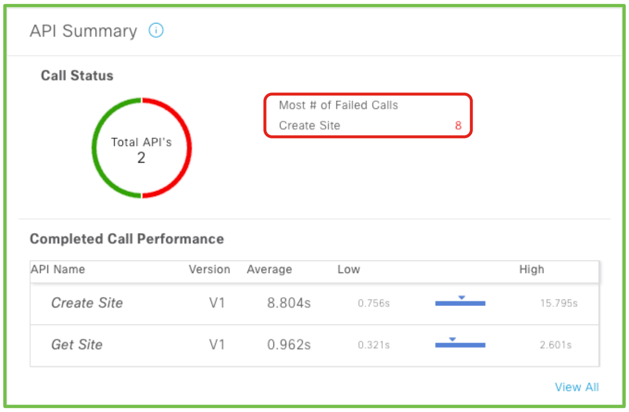
HierarchyArea	RTP
HierarchyParent	Global
CCip	198.18.129.100
TOKEN	eyJhbGciOiJSUzI1NiIsInR5cCI...

HierarchyArea is the area name

HierarchyParent is the root, in this case, Global from previous command interaction

```
{
    "executionId": "e3687343-fe2f-4d69-a322-45247c68f354",
    "executionStatusUrl": "/dna/platform/management/business-api/v1/execution-status/
        e3687343-fe2f-4d69-a322-45247c68f354",
    "message": "The request has been accepted for execution"
}
```

Figure 23: Create an Area



API Summary

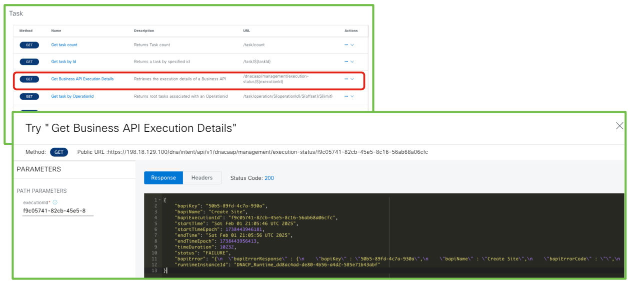
Call Status

Most # of Failed Calls
Create Site 8

Completed Call Performance

API Name	Version	Average	Low	High
Create Site	V1	8.804s	0.756s	15.795s
Get Site	V1	0.962s	0.321s	2.601s

Figure 24: Runtime Dashboard status provides API call status



Task

Method	Name	Description	URL	Actions
GET	Get task count	Returns Task count	/task/count	••• ∨
GET	Get task by Id	Returns a task by specified id	/task/${taskId}	••• ∨
GET	Get Business API Execution Details	Retrieves the execution details of a Business API	/dnacaap/management/execution-status/${executionId}	••• ∨
GET	Get task by OperationId	Returns root tasks associated with an Operationid	/task/operation/${operationId}/${offset}/${limit}	••• ∨

Try " Get Business API Execution Details"

Method: GET Public URL :https://198.18.129.100/dna/intent/api/v1/dnacaap/management/execution-status/f9c05741-82cb-45e5-8c16-56ab68a06cfc

PARAMETERS

PATH PARAMETERS

executionId* f9c05741-82cb-45e5-8

Status Code: 200

```
{
    "bapiKey": "50b5-89fd-4c7a-930a",
    "bapiName": "Create Site",
    "bapiExecutionId": "f9c05741-82cb-45e5-8c16-56ab68a06cfc",
    "startTime": "Sat Feb 01 21:05:46 UTC 2025",
    "startTimeEpoch": 1738443946181,
    "endTime": "Sat Feb 01 21:05:56 UTC 2025",
    "endTimeEpoch": 1738443956413,
    "timeDuration": 10232,
    "status": "FAILURE",
    "bapiError": "{\n  \"bapiErrorResponse\" : {\n    \"bapiKey\" : \"50b5-89fd-4c7a-930a\",\n    \"bapiName\" : \"Create Site\",\n    \"bapiErrorCode\" : \"\",\n
    "runtimeInstanceId": "DNACP_Runtime_dd8ac4ad-de80-4b56-a4d2-585e71b43abf"
}
```

Figure 25: Get Business API Execution Details for asynchronous operations

3 Catalyst Center Intent APIs using Jupyter Notebooks

The Intent API is a REST API that exposes specific capabilities of the Cisco Catalyst Center platform. Intent APIs provide policy-based abstraction of business intent, allowing focus on an outcome rather than struggling with individual mechanisms steps. Intent APIs are RESTFul-based, which makes very easy to use. The RESTful Cisco Catalyst Center Intent API uses HTTPS verbs (GET, POST, PUT, and DELETE) with JSON structures to discover and control the network.

This lab uses Jupyter notebooks. The associated GitHub contains the notebook and instructions to install, in case you need. See `https://github.com/hemorale/CatC-API-Lab/tree/main/Intent%20API`. This lab can run in your own laptop or you can use the script server in dCloud.

3.1 Lab 6 - Install Jupyter notebooks

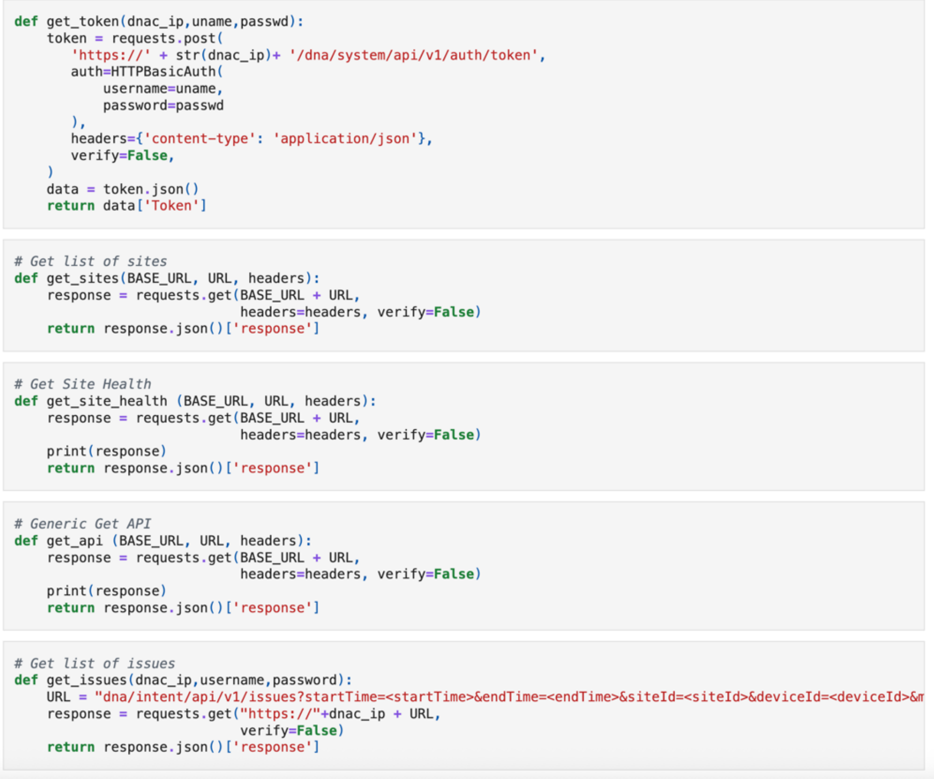
You can skip this section if you already have installed Jupyter notebooks.

1. Install Python if not installed.
2. Create and activate a virtual environment.
3. Run pip install jupyter.
4. Launch Jupyter notebook with jupyter notebook command.
5. (Optional) For a more advanced UI, install and launch JupyterLab using pip.
6. See the file ”Python-Jupyter-Install.md” in github for detailed instructions (`https://github.com/hemorale/CatC-API-Lab/blob/main/Intent%20API/Python-Jupyter-Install.md`

3.2 Lab 7 - Using Intent APIs with Jupyter notebooks

This lab has the goal to take you step by step and explaining how to use Intent APIs.

1. Launch jupyter notebook.
2. Open the ”Intent API/Managed DNAC v3.ipynb” notebook.
3. Let’s analyze the file. The initial cells are functions that creates some API calls. See Figure 26 for reference:
   - Get Token
   - Get Sites
   - Get Site Health
   - A generic Get API function
   - Get Issues
4. Run cell by cell. You won’t see any output.
5. The following cell, will print the Token by using the previously defined Get Token function. Run the cell and see the output. See Figure 27 for detailed reference.
6. Continue running the notebook and get to know the different intent APIs available. If you have any question or doubt, please consult your instructor.



```
def get_token(dnac_ip,uname,passwd):
    token = requests.post(
       'https://' + str(dnac_ip)+ '/dna/system/api/v1/auth/token',
       auth=HTTPBasicAuth(
           username=uname,
           password=passwd
       ),
       headers={'content-type': 'application/json'},
       verify=False,
    )
    data = token.json()
    return data['Token']
```

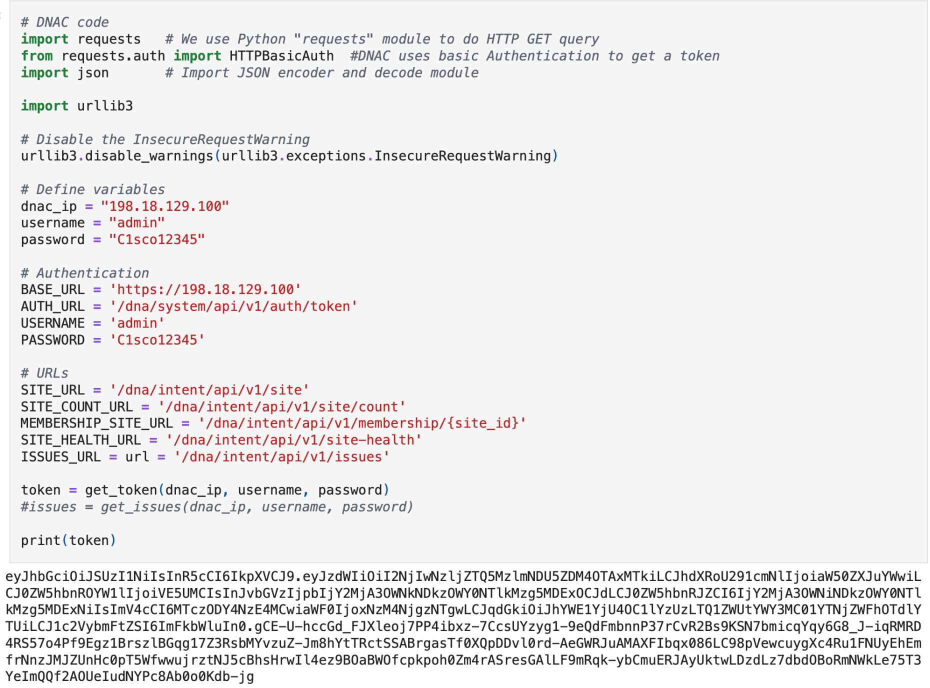
```
# Get list of sites
def get_sites(BASE_URL, URL, headers):
    response = requests.get(BASE_URL + URL,
                            headers=headers, verify=False)
    return response.json()['response']
```

```
# Get Site Health
def get_site_health (BASE_URL, URL, headers):
    response = requests.get(BASE_URL + URL,
                            headers=headers, verify=False)
    print(response)
    return response.json()['response']
```

```
# Generic Get API
def get_api (BASE_URL, URL, headers):
    response = requests.get(BASE_URL + URL,
                            headers=headers, verify=False)
    print(response)
    return response.json()['response']
```

```
# Get list of issues
def get_issues(dnac_ip,username,password):
    URL = "dna/intent/api/v1/issues?startTime=<startTime>&endTime=<endTime>&siteId=<siteId>&deviceId=<deviceId>&m
    response = requests.get("https://"+dnac_ip + URL,
                            verify=False)
    return response.json()['response']
```

Figure 26: Initial cells on Intent APIs



```
# DNAC code
import requests   # We use Python "requests" module to do HTTP GET query
from requests.auth import HTTPBasicAuth  #DNAC uses basic Authentication to get a token
import json       # Import JSON encoder and decode module

import urllib3

# Disable the InsecureRequestWarning
urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)

# Define variables
dnac_ip = "198.18.129.100"
username = "admin"
password = "C1sco12345"

# Authentication
BASE_URL = 'https://198.18.129.100'
AUTH_URL = '/dna/system/api/v1/auth/token'
USERNAME = 'admin'
PASSWORD = 'C1sco12345'

# URLs
SITE_URL = '/dna/intent/api/v1/site'
SITE_COUNT_URL = '/dna/intent/api/v1/site/count'
MEMBERSHIP_SITE_URL = '/dna/intent/api/v1/membership/{site_id}'
SITE_HEALTH_URL = '/dna/intent/api/v1/site-health'
ISSUES_URL = url = '/dna/intent/api/v1/issues'

token = get_token(dnac_ip, username, password)
#issues = get_issues(dnac_ip, username, password)

print(token)
```

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiI2NjIwNzljZTQ5MzlmNDU5ZDM4OTAxMTkiLCJhdXRoU291cmNlIjoiaW50ZXJuYWwiL
CJ0ZW5hbnROYW1lIjoiVE5UMCIsInJvbGVzIjpbIjY2MjA3OWNkNDkzOWY0NTlkMzg5MDExOCJdLCJ0ZW5hbnRJZCI6IjY2MjA3OWNiNDkzOWY0NTl
kMzg5MDExNiIsImV4cCI6MTczODY4NzE4MCwiaWF0IjoxNzM4NjgzNTgwLCJqdGkiOiJhYWE1YjU4OC1lYzUzLTQ1ZWUtYWY3MC01YTNjZWFhOTdlY
TUiLCJ1c2VybmFtZSI6ImFkbWluIn0.gCE-U-hccGd_FJXleoj7PP4ibxz-7CcsUYzyg1-9eQdFmbnnP37rCvR2Bs9KSN7bmicqYqy6G8_J-iqRMRD
4RS57o4Pf9Egz1BrszlBGqg17Z3RsbMYvzuZ-Jm8hYtTRctSSABrgasTf0XQpDDvl0rd-AeGWRJuAMAXFIbqx086LC98pVewcuygXc4Ru1FNUyEhEm
frNnzJMJZUnHc0pT5WfwwujrztNJ5cBhsHrwIl4ez9BOaBWOfcpkpoh0Zm4rASresGAlLF9mRqk-ybCmuERJAyUktwLDzdLz7dbdOBoRmNWkLe75T3
YeImQQf2AOUeIudNYPc8Ab0o0Kdb-jg
```

Figure 27: Get and Print Token

References

Morales, H. (2025*a*), 'Catalyst center api workshop'. Delivered at Cisco Customer Event, London, February 5–6, 2025.

Morales, H. (2025*b*), 'Catalyst center api workshop'. Delivered at Cisco Customer Event, Madrid, April 1–2, 2025.