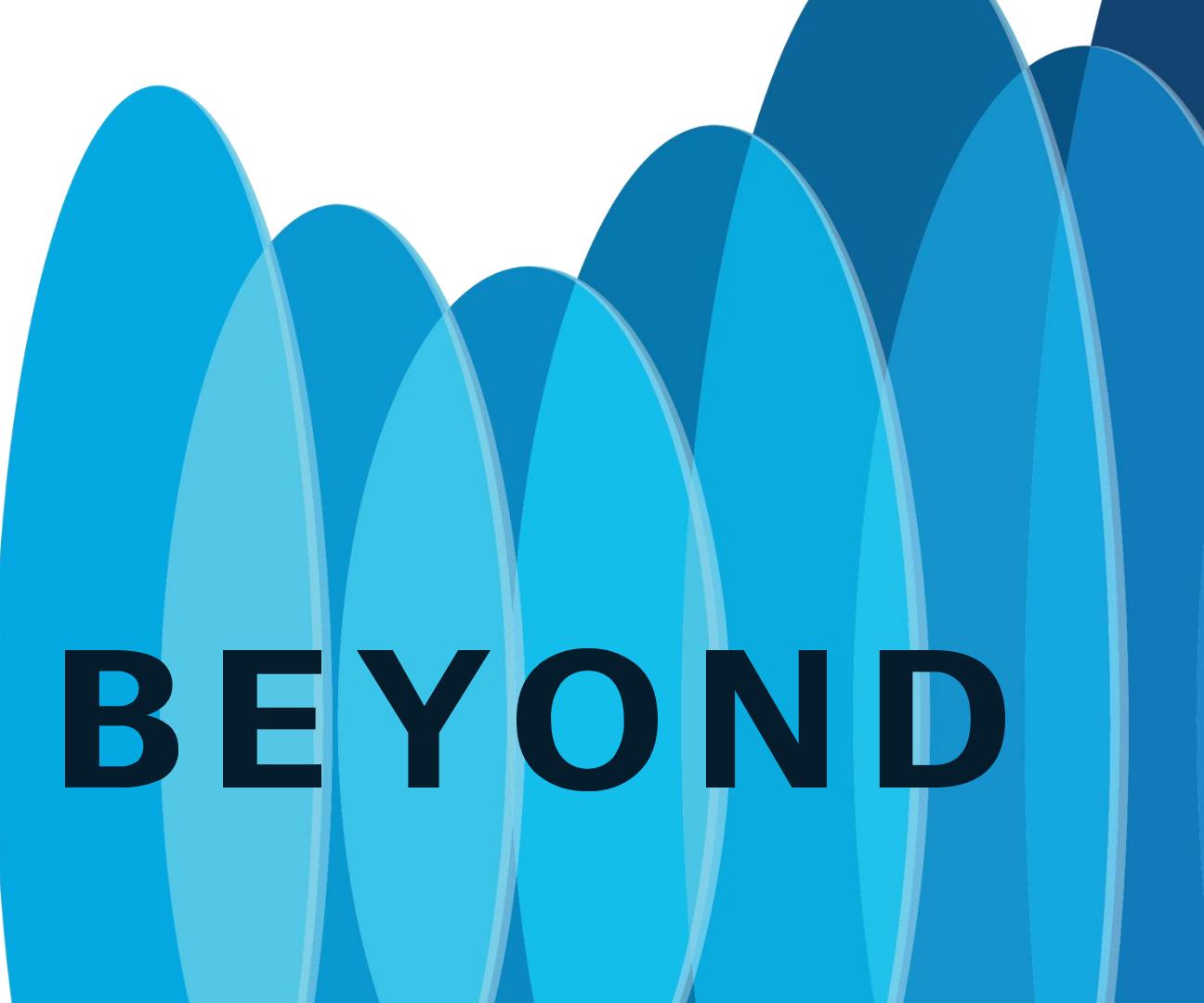


CISCO *Live!*



GO BEYOND



Automation and Management with Catalyst Center APIs

Technical Workshop

Hector Morales Anton Inniss
SE BDM

Silvia Reis
SAM



Agenda

Day 1

9:00-9:30	Introduction and workshop objectives	ALL
9:30-10:30	Catalyst Center deep dive	Cisco
10:30-11:00	Lab introduction	Cisco
11:00-11:30	Break and email	ALL
11:30-13:30	CatC APIs 100-200 level with Postman collections	ALL
13:30-14:30	Lunch	ALL
14:30-16:30	CatC APIs 300-400 level using jupyter notebooks and python	ALL
16:30-17:00	Wrap up day 1	Cisco

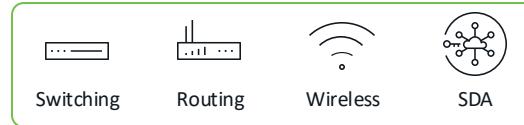
Day 2

9:00-9:30	Review Computacenter use cases	CC
9:30-10:30	Managed Campus Operations Guide	Cisco
10:30-11:00	Challenge: Use Case implementation	CC
11:00-11:30	Break and email	ALL
11:30-13:30	Challenge implementation	CC
13:30-14:00	Challenge presentation	CC
14:00-14:30	Wrap up day 2 and next steps	ALL

Catalyst Center Headless Deployments



The complexity to operate large networks



```
Router# show running-config
Building configuration...
Current configuration : 5980 bytes
!
! Last configuration change at 13:56:48 PST Fri Nov 3 2017 by admin
!
version 16.6
service timestamps debug datetime msec
service timestamps log datetime msec
service call-home
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform shell
!
hostname Router
!
boot-start-marker
boot system tftp /auto/tftp-sjc-users5/c1100-universalk9_ias.16.06.02.SPA.bin 223.255.254.254
boot-end-marker
!
!
vrf definition VRF-example
description VRF-example
!
no logging console
!
aaa new-model
!
!
aaa login success-track-conf-time 1
!
!
aaa session-id common
```

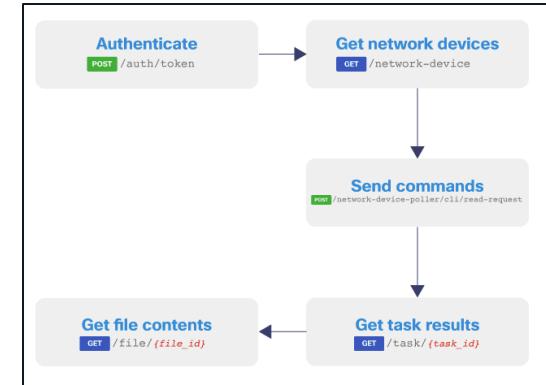
IOS XE CLI

```
- name: Check the running-config against master config
  cisco.ios.ios_config:
    diff_against: intended
    intended_config: "{{ lookup('file', 'master.cfg') }}"

- name: Check the startup-config against the running-config
  cisco.ios.ios_config:
    diff_against: startup
    diff_ignore_lines:
      - htp clock .*
    save_when: modified

- name: Save running to startup when modified
  cisco.ios.ios_config:
    save_when: modified
```

IOS XE Ansible Collection



Cisco Catalyst Center APIs

The complexity to operate large networks

Pros and Cons

	IOS XE CLI	IOS XE Ansible Collection	Cisco Catalyst Center APIs
Day 0	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓
	Fine configuration Prone to human errors	IaC, CI/CD pipelines Requires more expertise	API Platform Single tenant
Day 1	✓ ✓ ✓ ✓	✓ ✓ ✓	✓ ✓ ✓ ✓ ✓
	Data rich Requires additional infra	Leverage community Prone to errors	Extensive integration Requires more coding
Day 2	✓ ✓ ✓ ✓	✓ ✓	✓ ✓ ✓ ✓ ✓
	Information rich Complex to troubleshoot	Leverage community Not ready for Day 2	Event management Requires more coding
Complexity ¹	Very complex	Mid complexity	Low complexity
IDEAL Team size ²	10-15 engineers	5-7 engineers	3-5 engineers

REALITY CHECK: We operate with reduced teams

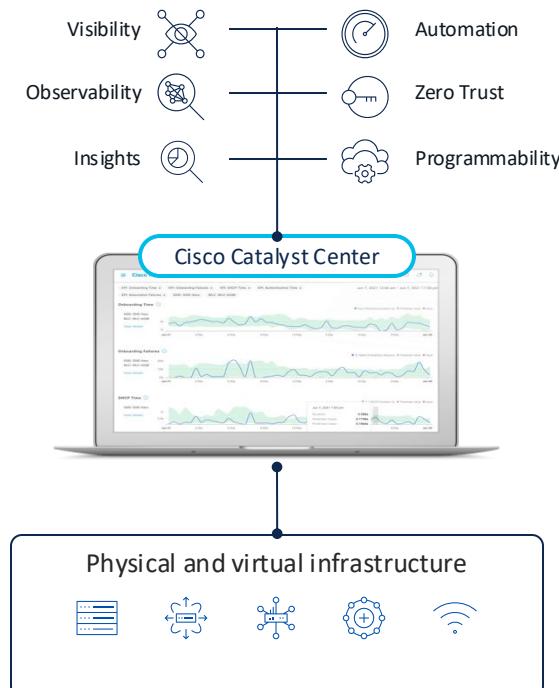
Source: Own research with public data from top 100 customers and service providers, 2024

1 - Network size: 20,000 devices in a single campus

2 – Necessary team to support single campus

Catalyst Center simplifies network operations

Enabling IT admin to deliver business outcomes, smarter and faster at scale



Small teams' productivity increase through automation

NetOps

Increase Scale

- Providing customers with business resiliency, continuity, and quick time to value
- Enabling customers to achieve Compliance of network with config policies
- Automation to simplify the creation and maintenance of customers networks

SecOps

Improved Security

- AI-driven security to classify endpoints and enforce security policies for a complete zero trust workplace solution
- Automate end point visibility, classification, and grouping

DevOps

Improved Service Delivery

- Mature APIs, SDKs, and closed-loop integrations to simplify and streamline ecosystem integration
- Faster service delivery using API-based automation workflows
- Early issue detection and integration with 3rd party platforms through enhanced notification channels

AIOps

Improved Performance Insights

- Reduced proactive problem resolution through faster Root Cause Analysis
- AI-driven visibility, observability, insights, and troubleshooting to ensure the health your customers applications, infrastructure and user experience

Catalyst Center as a platform

Powered for headless integration

Event Notifications

- Assurance Issues
- AI/ML Insights
- System Health
- Integration Connectivity
- License Management
- Webhooks
- PagerDuty
- Email
- Syslog
- SNMP

Northbound REST APIs

- Network Inventory
- Network Topology
- Network Design
- Provisioning
- SWIM, PnP
- Path Trace
- Assurance
- SDA
- Templates
- RMA
- Config Archive
- Sensors

IT Ecosystem Integrations

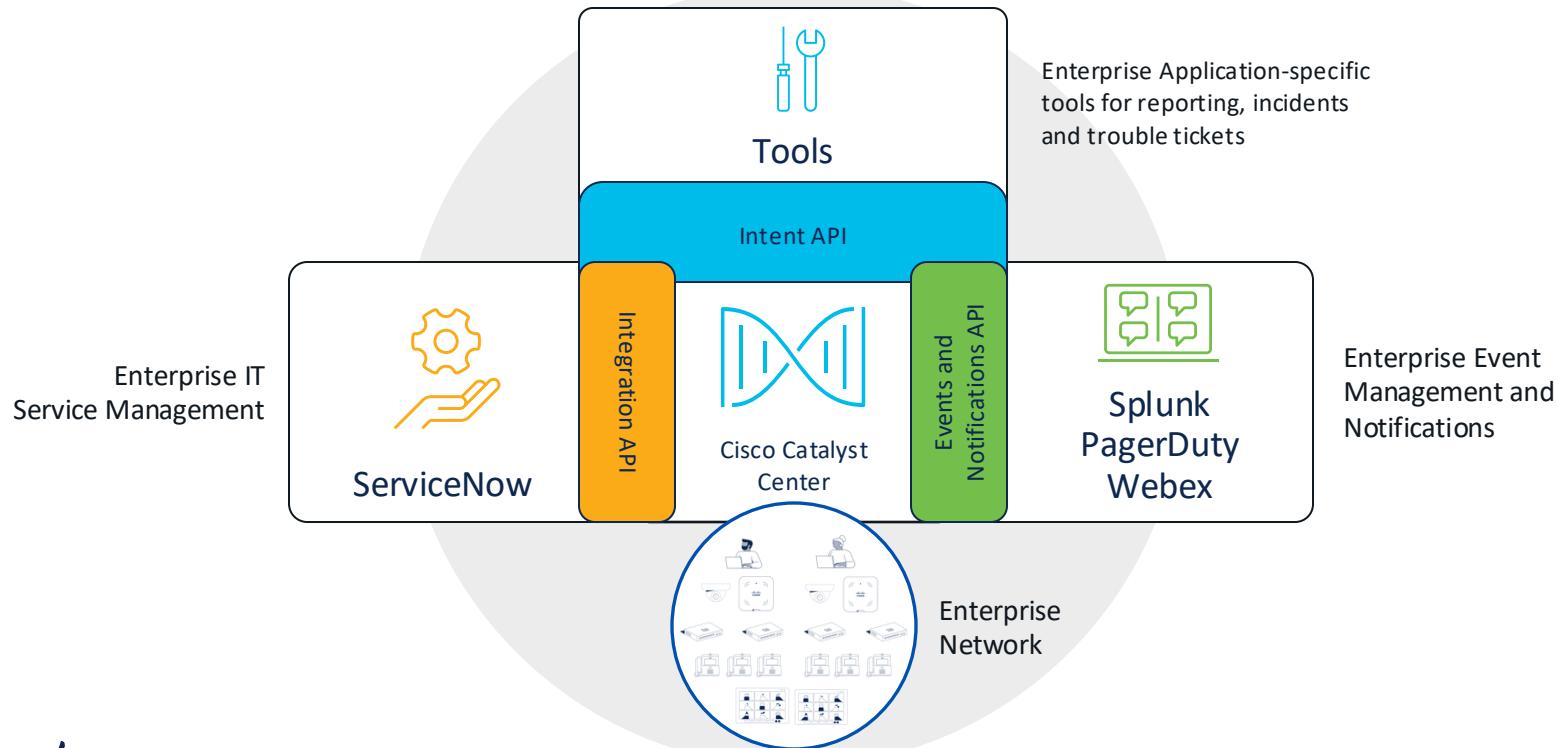
- IT Service Management
- IP Address Management
- Reporting
- Wireless Planning
- Alerting
- SIEM

Developer Resources

- Sample Code, Videos
- Python SDK, Ansible, Terraform
- Cisco DevNet
 - Sandboxes, Learning Labs
 - Developer Guides
 - Sample Code

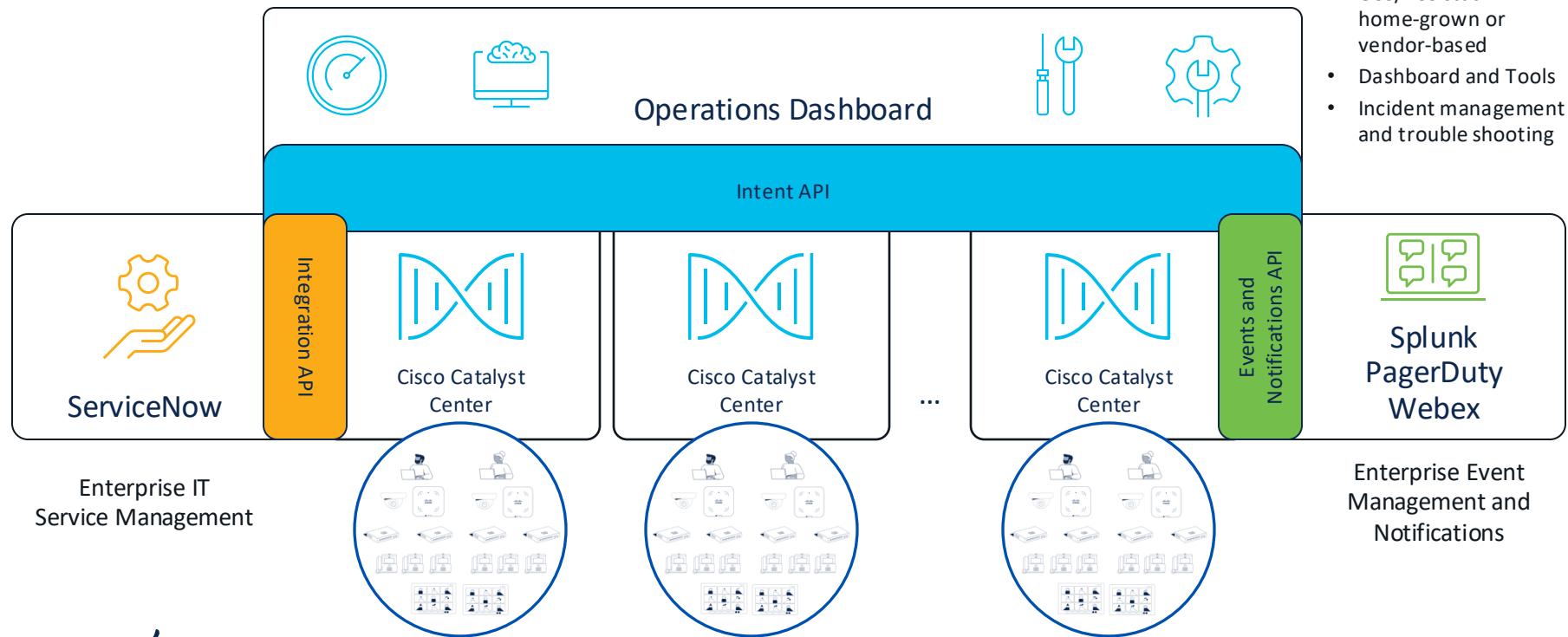
Cisco Catalyst Center enterprise architecture

Align to ITIL framework



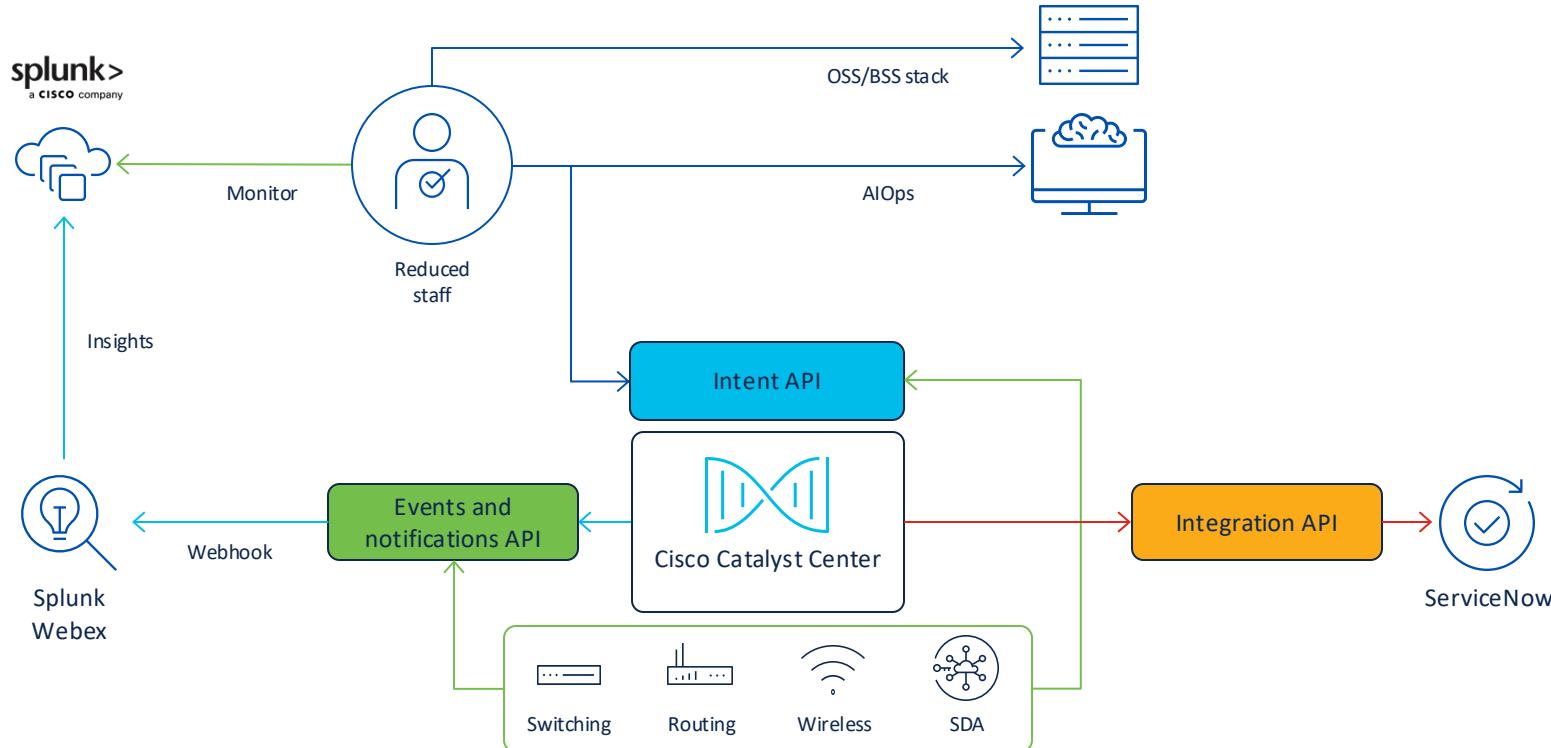
Cisco Catalyst Center architecture

Headless operations: operate Catalyst Center at scale



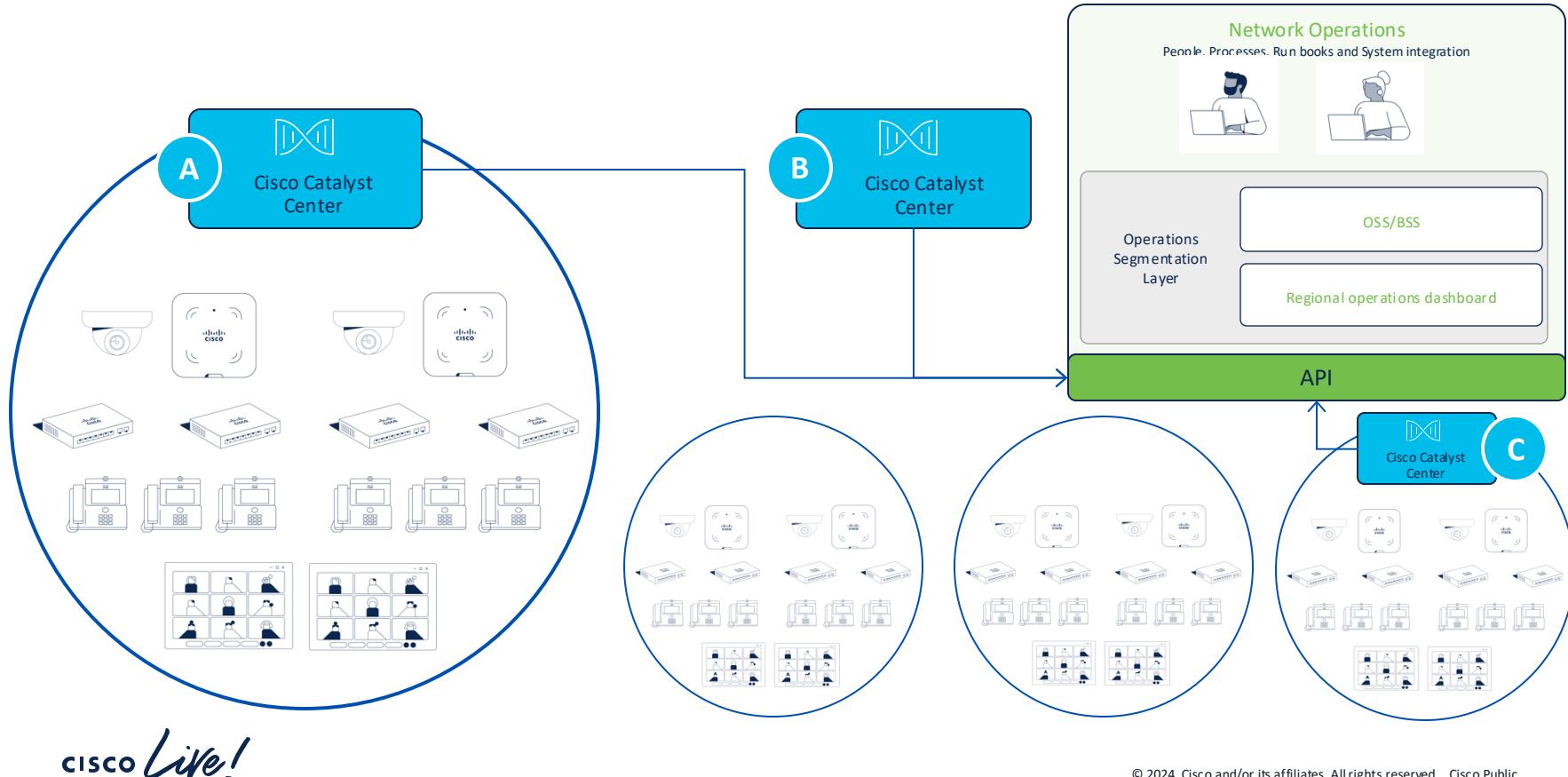
A high-level view

Catalyst Center headless deployment example



Catalyst Center APIs

What a headless scenario looks like



Cisco Catalyst Center API platform

Event and Notifications API

Provides a notification handler when specific events are triggered, such as Network Assurance and Automation (SWIM) events.

Enables external systems to take actions in response to an event: execute a software upgrade action in response to an out of compliance notification of network devices

Notifications may also be triggered by internal Catalyst Center events. For example, Assurance events can be customised for IT Service Management incidents.

Intent API

The Intent API is a REST API that exposes specific capabilities of the Cisco Catalyst Center platform.

Provides policy-based abstraction of business intent, allowing focus on an outcome rather than struggling with individual mechanisms steps.

The RESTful Cisco Catalyst Center Intent API uses HTTPS verbs (GET, POST, PUT, and DELETE) with JSON structures to discover and control the network.

Integration API

Provides mechanisms for integrating Network Assurance end-to-end workflows and data with third-party IT Service Management (ITSM) solutions.

ITSM integration minimises handoffs, reduces duplication of issues, and optimizes processes such as approval- and pre-approval chains,

Cisco Catalyst Center also Integrates with Reporting and Analytics capabilities for capacity planning, asset management, compliance control, and auditing.

Cisco Catalyst Center

Platform overview

Cisco DNA Center

Platform - Overview

Welcome to the Cisco DNA Center Platform. Programmatically access your network through Intent APIs, integrate with your preferred IT systems to create end-to-end solutions and add support for multi-vendor devices.

Bundles
Bundles are easy to use feature sets for consuming Intent APIs, integrations, events and notifications. View all the available bundles, enable relevant bundles and customize the configuration preferences to consume events as per your application(s) or IT system(s) needs.

Developer Toolkit
Discover APIs to manage your network, configure integration flows and access network data to analyze, export and visualize complex reports.

Runtime Dashboard
Get insights into API usage, view events published to IT systems such as number of API calls, response time(s), events published, bundles activated etc.

Configurations
View and set global or bundle specific settings to manage your integration configurations and modify event specific settings.

Notifications

BUNDLE UPDATE
The Cisco DNA Center REST API bundle is enabled.
Sep 20 2022, 08:47 am
[View Details](#) | [Dismiss](#)

BUNDLE UPDATE
Disabled the Cisco DNA Center REST API bundle.
Sep 20 2022, 08:47 am
[View Details](#) | [Dismiss](#)

BUNDLE UPDATE
The Cisco DNA Center REST API bundle is enabled.
Aug 19 2022, 07:37 pm
[View Details](#) | [Dismiss](#)

Options for CatC Labs

- DevNet Catalyst Center Sandbox
- dCloud

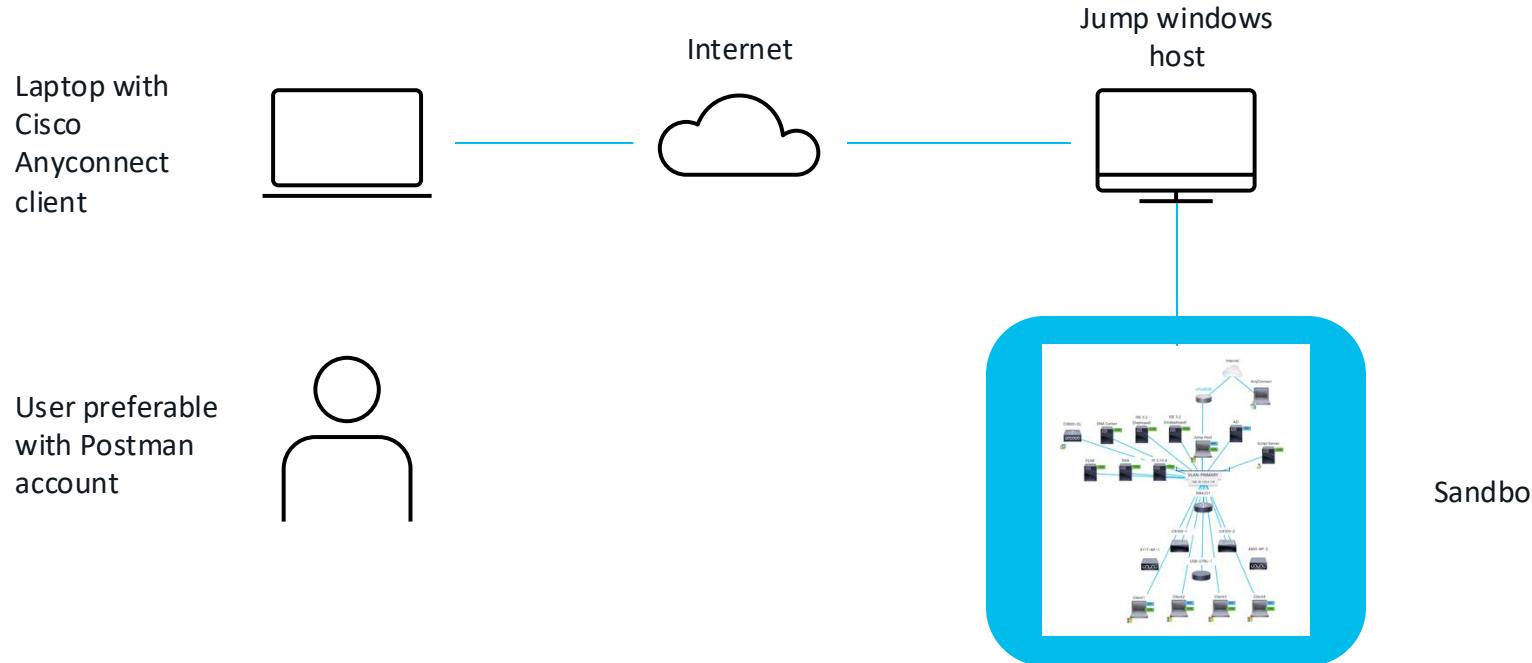
CatC API documentation

- In CatC in the Platform overview
- In DevNet
 - <https://developer.cisco.com/docs/dna-center/cisco-dna-center-2-3-7-api-overview/>

Lab introduction

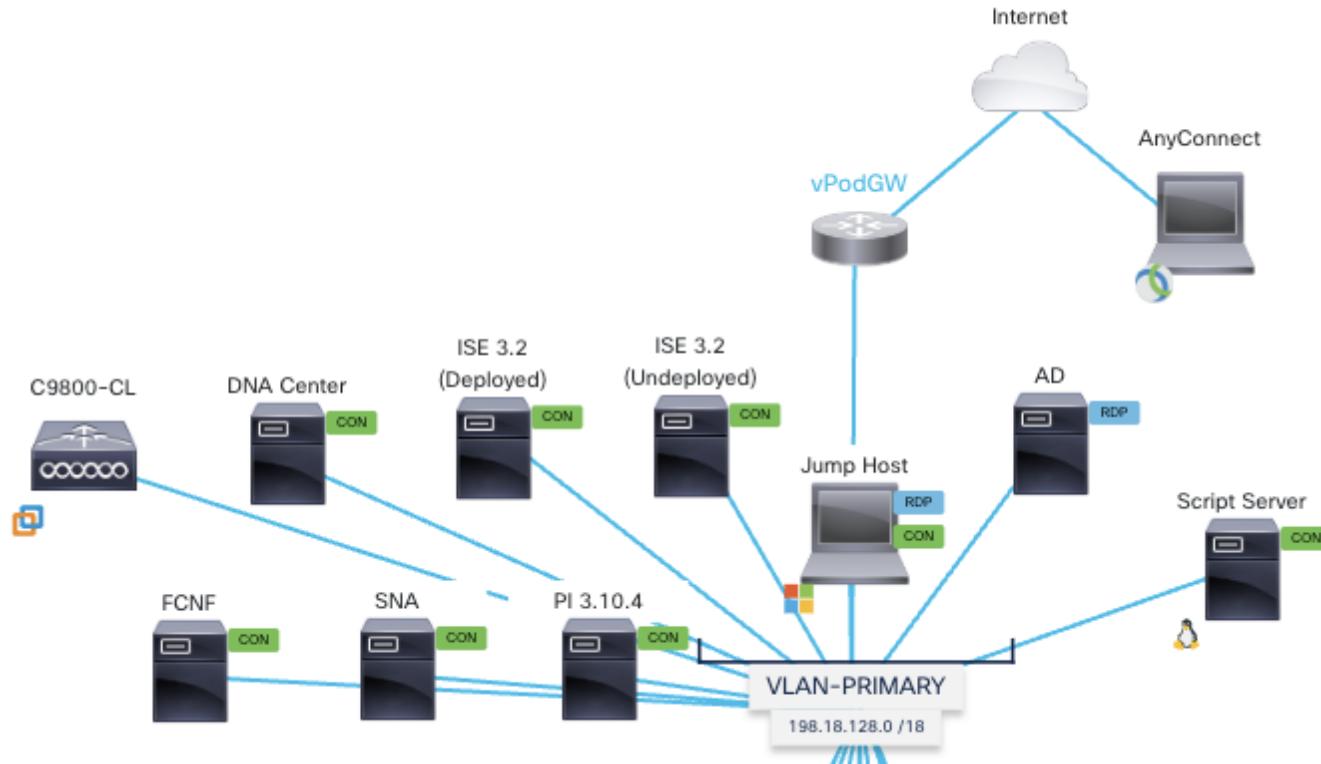
Lab description

Prerequisites



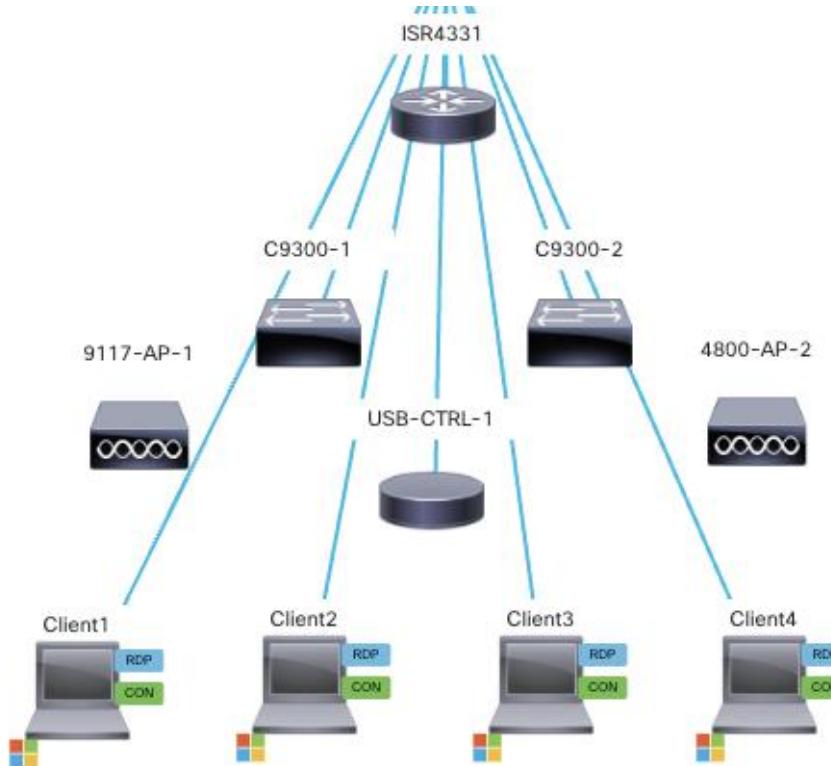
Cisco Enterprise Networks Sandbox v4.1

Jump host and servers including CatC



Cisco Enterprise Networks Sandbox v4.1

Network devices and clients



What is in this environment?

Virtual Machines

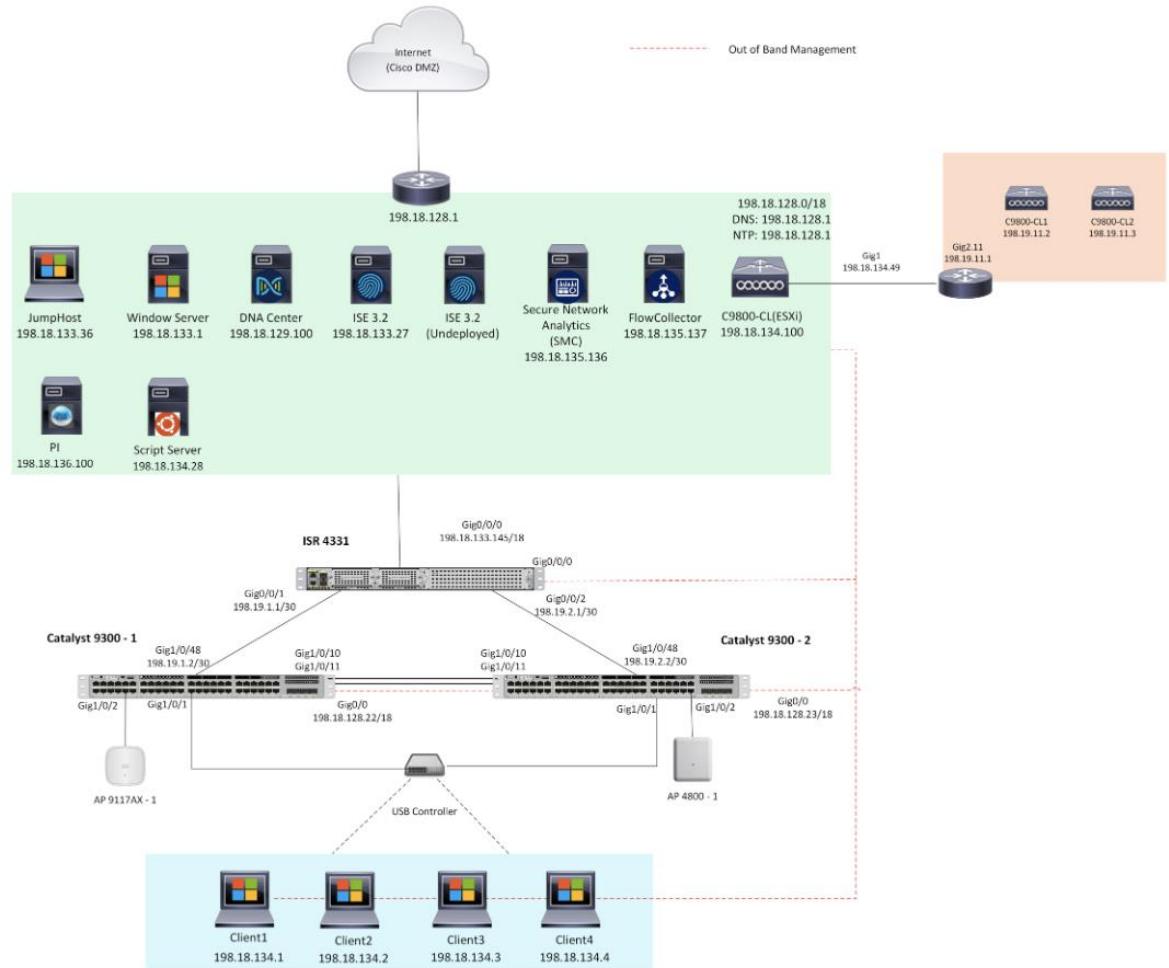
- DNA Center 2.3.5.5
- Identity Services Engine (ISE) 3.2 Patch 5 – (Deployed)
- Identity Services Engine (ISE) 3.2 – (Undeployed)
- Secure Network Analytics (Stealthwatch) 7.4.2
- FlowCollector 7.4.2
- Cisco Prime Infrastructure 3.10.4
- Wireless LAN Controllers – C9800 running IOS-XE 17.12.3 code
- 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 programming 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.

What is in this environment?

Hardware devices

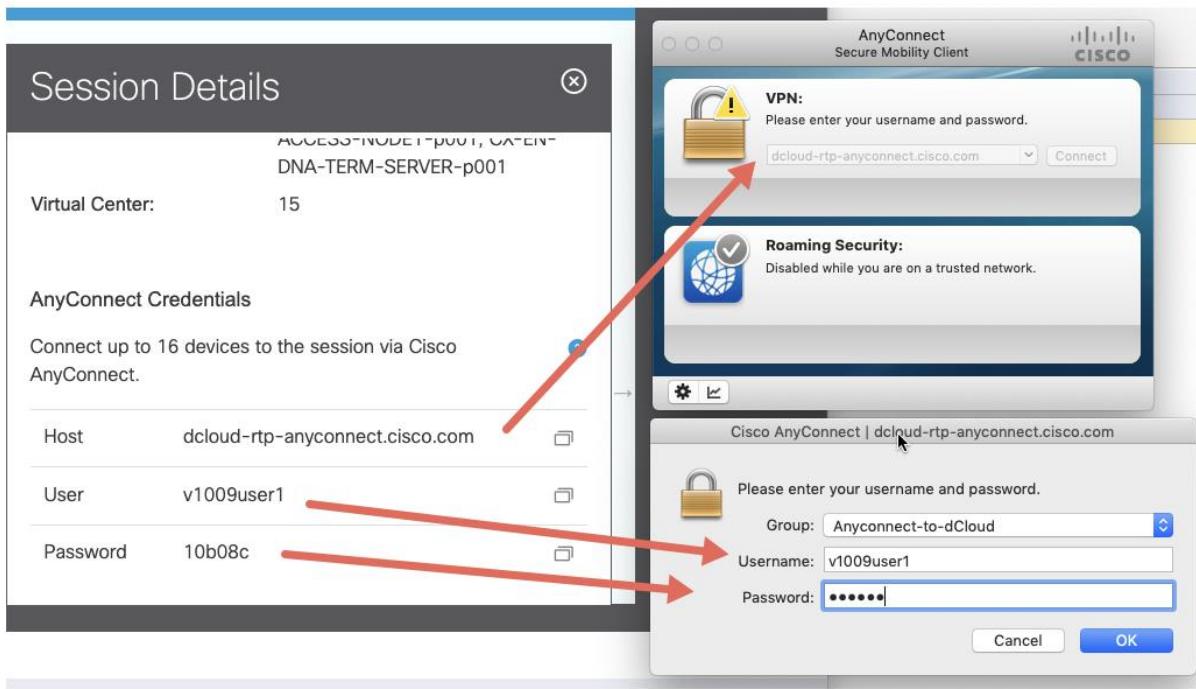
- ISR 4331 Router - 17.09.02a IOS-XE Code
- Catalyst 9300 Switches - 17.09.02 IOS-XE Code with Embedded Wireless Controller (EWC) and ThousandEyes Enterprise Agent
- 9117 Access Point
- 4800 Access Points
- Silex Controllers (2 Wired NICs)

Topology



How to access your environment?

Print session details
the day before



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 in the Virtual Machines section that follows.
3. Use your local RDP client to connect to RDP-capable Virtual Machines as outlined in Virtual Machines section that follows.

Virtual Machine Addressing and Credentials

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

Cisco Secure Client

The image shows the Cisco Secure Client application window. On the left, a sidebar lists configuration sections: Endpoint Kits, Public NAT IP / Internal NAT IP / Proxy, AnyConnect Credentials, Phone Numbers, and DNS. The AnyConnect Credentials section is expanded, displaying the following information:

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

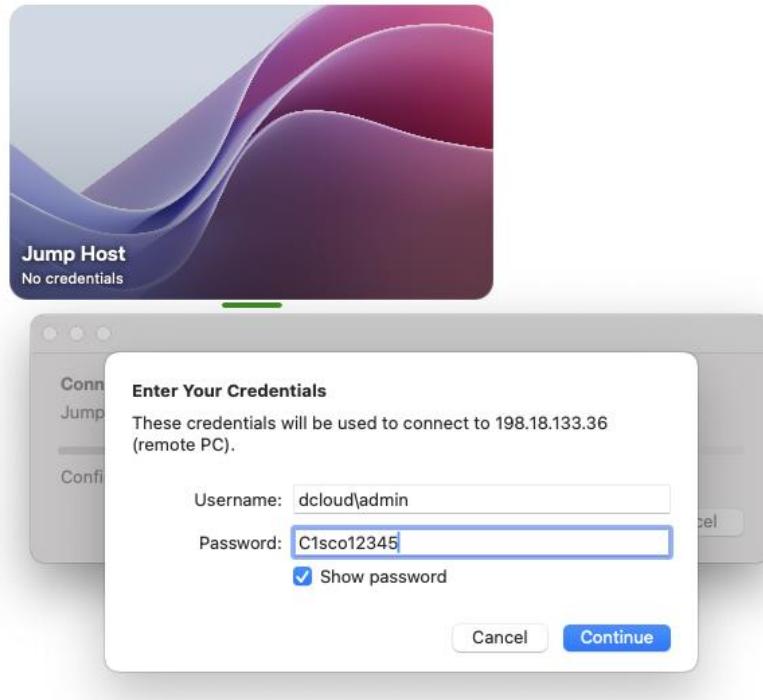
The main pane displays three status cards:

- AnyConnect VPN:** Ready to connect. A "Connect" button is present.
- ISE Posture:** No policy server detected. Default network access is in effect. A "Scan Again" button is present.
- Umbrella:** Umbrella is active.

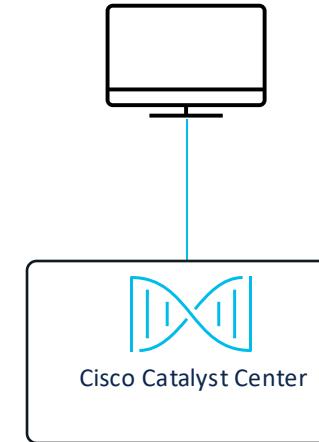
At the bottom of the main pane are settings and refresh icons.

RDC to jump host

▼ Saved PCs



Jump windows host



Windows Jump Host



CatC APIs
100-200 level
Postman
collections

GitHub

 CatC-API-Workshop Public

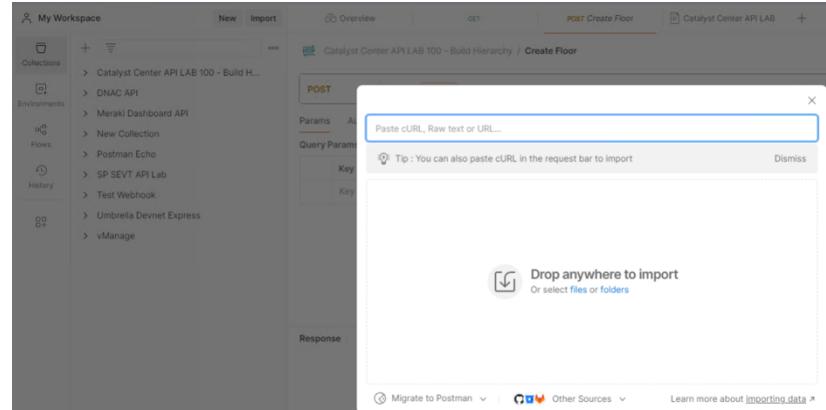
main 1 Branch 0 Tags Go to file Add file Code

 hemorale Add files via upload	7251092 · 6 minutes ago	 4 Commits
 Managed_Campus_scripts Add files via upload	53 minutes ago	
 Postman Collections Add files via upload	53 minutes ago	
 Managed_Campus_Ops_Guide.pdf Add files via upload	6 minutes ago	
 README.md Update README.md	8 minutes ago	

<https://github.com/hemorale/CatC-API-Workshop>

Postman collections

- Connect to the dCloud lab using your assigned PoD via Cisco Anyconnect
- Using the Windows App, connect into the Jump Host
- Open Chrome and download the Postman collections in the jump host.
- Open Postman. If necessary login into your account or create a new account. You will need it to upload collections
- Click on Collections. Then click on Import. Select the json files.

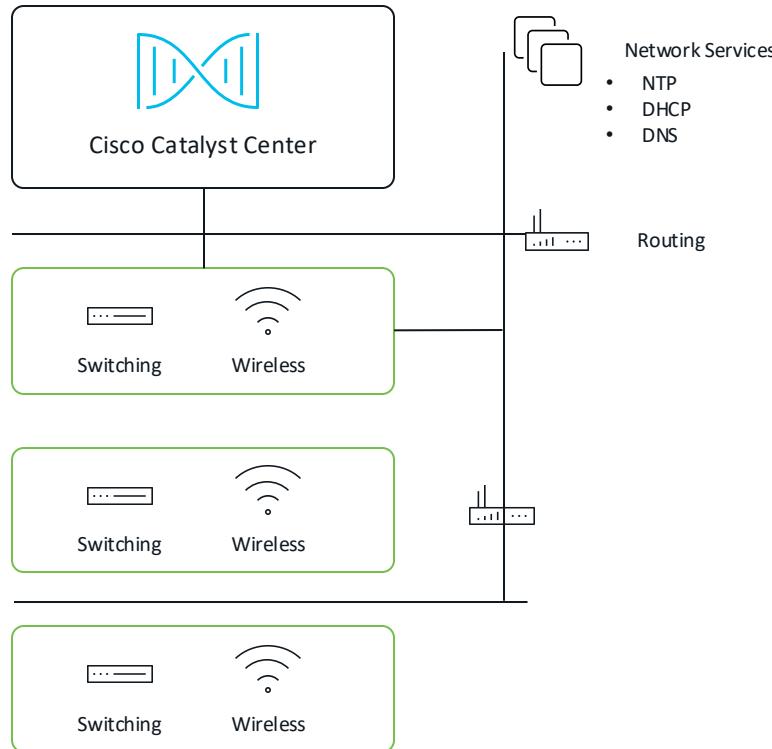


Postman collections

- **Catalyst Center API Environment.postman_environment.json** - 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.json** – builds the initial CatC network hierarchy
- **Catalyst Center API LAB 101 - Assign Settings Creds.postman_collection.json** - assign all CatC lab credentials
- **Catalyst Center API LAB 201 - Device Discovery.postman_collection.json** - device discovery
- **Catalyst Center API LAB 302 - Template Deployment.postman_collection.json** – deploys a switch template
- **Catalyst Center API LAB 304 - Configuration Archive.postman_collection.json** - configuration archive
- **Catalyst Center API LAB 401 - Device Inventory.postman_collection.json** – device inventory
- **Catalyst Center API LAB 402 - Command Runner.postman_collection.json** – command runner

Catalyst Center basic configuration

Topology discovery



Connect to Catalyst Center

1. In Jump host, open Chrome
2. Click on DNA Center
3. Login
4. Click on Add sites
5. Click on Discovery
6. Add Discovery
7. Click Next
8. Discovery type CDP
9. IP Address: 198.18.133.145
10. Use Loopback
11. Click Next

Catalyst Center basic configuration

 Add CLI Credentials

GLOBAL CREDENTIALS

Name/Description*	CLI Global Credentials	
Password*	C1sco12345	
View Password Criteria		
 Add V2C Read Credentials		

Username*
admin

Enable Password
C1sco12345


[View Password Criteria](#)

- Skip site assignment for now. We will add later from inventory.
- Start discovery
- Open Hardware console on dcloud portal
- Open ISR console
- See messages
- Click on view discovery

GLOBAL CREDENTIALS

Name/Description*	SNMP Global RO
Read Community*	RO
 View Password Criteria	



Discovery process

```
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)
```

Discovery process

Tools / Discovery / Dashboard

Discoveries (1) [+ Add Discovery](#)

Search Table ✖

Scheduled discoveries [Export](#) As of: Jan 31, 2025 8:08 PM [⟳ Refresh](#)

Discovery Name ▾	Type	Status	IP Address	Reachable Devices	Actions
Discovery to Quick Insights	CDP	In Progress	198.18.133.145	3	...

Discovery process

Cisco DNA Center Reference / Discovery Details

All Discoveries 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

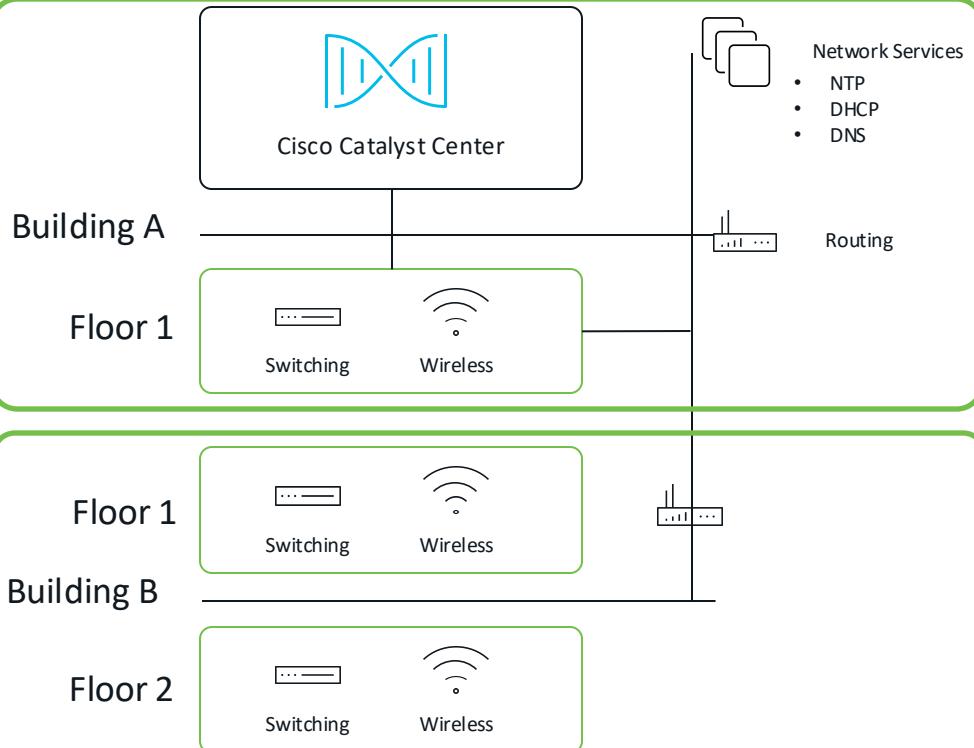
5	3	1	1
Discovered	Reachable	Unreachable	Discarded

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	-	✗	?	?	?	?	?

Catalyst Center basic configuration

Area



Using APIs to create Topology map

1. In Jump host, open Postman
2. Click on the collection Catalyst Center API LAB 100 - Build Hierarchy
3. This collection will create a hierarchy like this one

Creating hierarchy using Postman

Get the Token

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

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

Params Authorization • 198.18.129.100

Query Param

Key

Key

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

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

Params Authorization • Headers (10) Body Scripts • Tests • Settings •

Auth Type Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about [Basic Auth](#) authorization.

Username {{CCuser}}

Save Response

Save

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

Save the token for the next postman iterations

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Creating hierarchy using Postman

Get SiteIDs

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

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9eyJzdWJlOiI2NjIwNzljZTQ5MzlmNDU5ZDM4OTAxMTkILCJhdXRoU291cmNljoiaW50ZXJuYWwiLCJ0ZW5hbnROYW1lijoive5UMCiIslnJvbGVzIjpbljY2MjA30WNkNDkzOWy0NT1kMzg5MDEx0Cj0ZWhbnR3ZCIEiJy2MjA30WNlNDkzOWy0NT1kMzg5MDExNiIsImV4cI6MTcz0DQ0MDYxOCwiawF0IjoxNz4NDM3MDE4LCjqdGkiO1JmNDy0MDk3Zs1hNDRIlTRkM2It0GjiZC1kNjk1ZjEwYzEyMzMilC1c2VybmtZS16ImFkbWluIn0.kFY9z9fVQo1dp6t1Fu6Py5sv1loEUPiaXnNE9yjo5FQP-AsSXbKd5bYImqb09z0u_Sy1mEReALKDhwcaYIIIfUGp9wMBcTYrSGBYHJxhmlVCIhDW-G0VqB_X9aNxtRcLDld5jrp1uuLFZe43Xz8uxYSumb6T3i8n3P-4BVxtMbfz5PzcyWo500Njgn-xGzmp6kp9qSfd5vC1o3IeMIXo3Iq-UIWrxcvwN94Djl-Nsz53kmfkUv6ndS_Qy9H2l1BtK06ftvdUMKEIAwf0zyUIBSzBMHEQHCuW9CzpnaatJVAGhNhgPYt5xaDwyfhFNawanJaqcB_k-zohLojTKcA"

Creating hierarchy using Postman

Get SiteIDs



The screenshot shows a Postman interface with a green border. At the top, there are tabs: Body, Cookies, Headers (14), Test Results, and a refresh icon. To the right of the tabs, it says "200 OK". Below the tabs, there are three buttons: "JSON" (selected), "Preview", and "Visualize". The main area contains a JSON response with line numbers from 1 to 12 on the left. The JSON object has a single key "response" which contains an array with one element. This element is an object with properties: "additionalInfo": [], "name": "Global", "instanceTenantId": "662079cb4939f459d3890116", "id": "68b27cdf-11b5-4fe0-bd43-97cd2ca22126", "siteHierarchy": "68b27cdf-11b5-4fe0-bd43-97cd2ca22126", and "siteNameHierarchy": "Global".

```
1 {  
2     "response": [  
3         {  
4             "additionalInfo": [],  
5             "name": "Global",  
6             "instanceTenantId": "662079cb4939f459d3890116",  
7             "id": "68b27cdf-11b5-4fe0-bd43-97cd2ca22126",  
8             "siteHierarchy": "68b27cdf-11b5-4fe0-bd43-97cd2ca22126",  
9             "siteNameHierarchy": "Global"  
10        }  
11    ]  
12 }
```

Creating hierarchy using Postman

Create Area

CCCCpwd
C1sco12345
CCuser

Variables in request:

- HierarchyArea: RTP
- HierarchyParent: 198.18.129.100
- CCip: 198.18.129.100
- TOKEN: eyJhbGciOiJSUzI1NiIsInR5cCI...

HierarchyArea is the area name

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

Run again the previous command, what did you notice?
Go to Catalyst Center Dashboard and open Network Topology

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

```
1 {  
2   "type": "area",  
3   "site": {  
4     "area": {  
5       "name": "{{HierarchyArea}}",  
6       "parentName": "{{HierarchyParent}}"  
7     }  
8   }  
9 }
```

Body Cookies Headers (14) Test Results (1/1) 202 Accepted 717 ms 789 B Save

```
1 {  
2   "executionId": "e3687343-fe2f-4d69-a322-45247c68f354",  
3   "executionStatusUrl": "/dna/platform/management/business-api/v1/execution-status/e3687343-fe2f-4d69-a322-45247c68f354",  
4   "message": "The request has been accepted for execution"  
5 }
```

Creating hierarchy using Postman

Add Building

Using Postman collection, try on
your own

Creating hierarchy using Postman

Add Building

Add Building X

Area contains other areas and/or buildings. Buildings contain floors and floor plans.

Building Name*

Parent
Global ▼

Address (i)
eg : 150 W Tasman Dr, San Jose ...

Latitude*
eg : 37.338 Longitude*
eg : -121.832

Cancel Add

What is wrong?

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.

Use CatC API Platform

Check asynchronous operations

Platform/Runtime Dashboard

API Summary ⓘ

Call Status

Total API's 2

Most # of Failed Calls Create Site 8

Completed Call Performance

Try "Get Business API Execution Details"

Method: GET Public URL :https://198.18.129.100/dna/intent/api/v1/dnacap...

PARAMETERS

executionId* ⓘ f9c05741-82cb-45e5-8

PATH PARAMETERS

Response Headers Status Code: 200

```
1 * {  
2     "bapiKey": "50b5-89fd-4c7a-930a",  
3     "bapiName": "Create Site",  
4     "bapiExecutionId": "f9c05741-82cb-45e5-8c16-56ab68a06cf",  
5     "startTime": "Sat Feb 01 21:05:46 UTC 2025",  
6     "startTimeEpoch": 1738443946181,  
7     "endTime": "Sat Feb 01 21:05:56 UTC 2025",  
8     "endTimeEpoch": 1738443956413,  
9     "timeDuration": 10232,  
10    "status": "FAILURE",  
11    "bapiError": "\n        \"bapiErrorResponse\" : {\n            \"bapiKey\" : \"50b5-89fd-4c7a-930a\",  
12            \"bapiName\" : \"Create Site\",  
13            \"bapiErrorCode\" : \"\",\n            \"runtimeInstanceId\" : \"DNACP_Runtime_dd8ac4ad-de80-4b56-a4d2-585e71b43abf\"\n        }\n    }  
14 }
```

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	/dnacap/management/execution-status/\${executionId}	... ▾
GET	Get task by OperationId	Returns root tasks associated with an Operationid	/task/operation/\${operationId}/\${offset}/\${limit}	... ▾
GET	Get tasks	Returns task(s) based on filter criteria	/task	... ▾
GET	Get task tree	Returns a task with its children tasks by based on their id	/task/\${taskId}/tree	... ▾

Creating hierarchy using Postman

Add floor

The screenshot shows the Postman interface with the following details:

- Request URL:** https://{{CCip}}/dna/intent/api/v1/site
- Method:** POST
- Body (JSON):**

```
1 {
2     "type": "floor",
3     "site": {
4         "floor": {
5             "name": "{{HierarchyFloor}}",
6             "parentName": "{{HierarchyParent}}/{{HierarchyArea}}/{{HierarchyBldg}}",
7             "rfModel": "Free Space",
8             "width": "100",
9             "length": "100",
10            "height": "10"
11        }
12    }
}
```
- Variables in request:**

HierarchyFloor	2nd Floor
HierarchyParent	Global
HierarchyArea	RTP
HierarchyBldg	HQ
CCip	198.18.129.100
TOKEN	eyJhbGciOiJSUzI1NiIsInR5cCI...
- Response:** 202 Accepted
- Body (JSON Response):**

```
1 {
2     "executionId": "ff2ca6ed-41e7-40b3-be0c-f32014e91fc7",
3     "executionStatusUrl": "/dna/platform/management/business-api/v1/execution-status/
4         ff2ca6ed-41e7-40b3-be0c-f32014e91fc7",
5     "message": "The request has been accepted for execution"
}
```

CatC APIs 300-400 level using jupyter notebooks



CatC APIs 300 level – Intent APIs



This lab will use jupyter notebooks

- You can do this in your laptop
- Install 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.
 5. (Optional) For a more advanced UI, install and launch JupyterLab.
- See the file in github for detailed instructions
- Directory for this lab:
 - <https://github.com/hemorale/CatC-API-Workshop/blob/main/Intent%20and%20Event%20Management/Managed%20DNAC%20v3.ipynb>

Authentication



POST

Authentication API

API to obtain an access token. The token obtained using this API is required to be set as value to the X-Auth-Token HTTP Header for all API calls to Cisco DNA...

/auth/token

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

```
# DNAC code
# Define variables
dnac_ip = <Your DNAC DNS/IP>
username = <Your DNAC username>
password = <Your DNAC password>

token = get_token(dnac_ip, username, password)

print(token)
```



General instructions to call a DNAC API

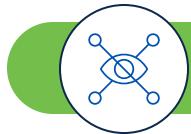
```
# Headers
headers = {'content-type': 'application/json', 'x-auth-token': token}

sites = get_sites(BASE_URL, SITE_URL, headers)
```

```
# Authentication
BASE_URL = 'https://dnac-79.infra.ciscomsx.com'
AUTH_URL = '/dna/system/api/v1/auth/token'

# 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 = '/dna/intent/api/v1/issues'
```

```
# 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']
```

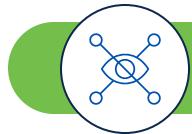


Monitoring your network

Site

GET	Get Site Health ^{Intent}	Returns Overall Health information for all sites	/site-health
GET	Get Site ^{Intent}	Get site using siteNameHierarchy/siteld/type ,return all sites if these parameters are not given as input.	/site
GET	Get Membership ^{Intent}	Getting the site children details and device details.	/membership/\${siteld}
GET	Issues ^{Intent}	Intent API to get a list of global issues, issues for a specific device, or issue for a specific client device's MAC address.	/issues

Monitoring your network



```
# Headers  
headers = {'content-type': 'application/json', 'x-auth-token': token}  
  
sites = get_sites(BASE_URL, SITE_URL, headers)
```

```
# Authentication  
BASE_URL = 'https://dnac-79.infra.ciscomsx.com'
```

```
# 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'
```

```
# Print nicely only Name, Site Name Hierarchy and Namespace
```

```
import json  
  
json_tree = json.dumps(sites, indent=4)  
  
data = json.loads(json_tree)  
  
for item in data:  
    name = item.get('name')  
    site_name_hierarchy = item.get('siteNameHierarchy')  
    namespace = None  
    item_id = item.get('id')  
  
    additional_info = item.get('additionalInfo', [])  
    for info in additional_info:  
        namespace = info.get('nameSpace')  
        print("Id:", item_id)  
        print("Name:", name)  
        print("Site Name Hierarchy:", site_name_hierarchy)  
        print("Namespace:", namespace)  
        print()
```

```
Id: 816225fd-9d87-406c-aadc-b8973519f02c  
Name: Building-Regression  
Site Name Hierarchy: Global/Area-Regression/Building-Regression  
Namespace: UMBRELLA
```

```
Id: 816225fd-9d87-406c-aadc-b8973519f02c  
Name: Building-Regression  
Site Name Hierarchy: Global/Area-Regression/Building-Regression  
Namespace: Location
```

```
Id: 816225fd-9d87-406c-aadc-b8973519f02c  
Name: Building-Regression  
Site Name Hierarchy: Global/Area-Regression/Building-Regression  
Namespace: ETA
```

Monitoring your network

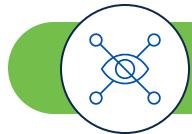


```
ISSUES_URL = url = '/dna/intent/api/v1/issues'
issues_list = get_api(BASE_URL, ISSUES_URL, headers)
print ("\nPretty print response:\n",json.dumps(issues_list,indent=4))
```

```
Pretty print response:
[
    {
        "issueId": "099bae52-119b-447e-bfc6-1d80120b8f35", ↑
        "name": "Excessive time lag between Cisco DNA Center and device \"MSX-OTT02-CAT3650-01\"", 
        "siteId": "", 
        "deviceId": "0ed45671-ec6f-4b66-9ab8-a919d31008f5", 
        "deviceRole": "", 
        "aiDriven": "", 
        "clientMac": "", 
        "issue_occurrence_count": 1, 
        "status": "active", 
        "priority": "", 
        "category": "", 
        "last_occurrence_time": 1685796841000
    }
]
```

Issue

Monitoring your network



```
# 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'
```

```
site_health = get_site_health (BASE_URL, SITE_HEALTH_URL, headers)

print ("\nPretty print response:\n",json.dumps(site_health,indent=4))
```

```
json_tree = json.dumps(site_health,indent=4)
data = json.loads(json_tree)

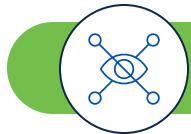
for site in data:
    site_name = site.get("siteName")
    site_id = site.get("siteId")
    healthy_network_device_percentage = site.get("healthyNetworkDevicePercentage")
    healthy_clients_percentage = site.get("healthyClientsPercentage")
    number_of_clients = site.get("numberOfClients")
    number_of_network_device = site.get("numberOfNetworkDevice")

    # Print the extracted information
    print("Site Name:", site_name)
    print("Site ID:", site_id)
    print("Healthy Network Device Percentage:", healthy_network_device_percentage)
    print("Healthy Clients Percentage:", healthy_clients_percentage)
    print("Number of Clients:", number_of_clients)
    print("Number of Network Devices:", number_of_network_device)
    print()
```

Site Name: All Sites
Site ID: All Sites
Healthy Network Device Percentage: 50
Healthy Clients Percentage: 100
Number of Clients: 1
Number of Network Devices: 2

Site Name: Amsterdam
Site ID: 9b592581-9388-4665-80d1-c2c30f1c4013
Healthy Network Device Percentage: None
Healthy Clients Percentage: None
Number of Clients: None
Number of Network Devices: None

Site Name: Area-Regression
Site ID: 50f4f3e5-0379-4ef2-9c59-27bd87366f47
Healthy Network Device Percentage: 50
Healthy Clients Percentage: 100
Number of Clients: 1
Number of Network Devices: 2



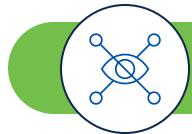
Monitoring your network

Topology

GET	Get VLAN details	Returns the list of VLAN names	/topology/vlan/vlan-names
GET	Get Physical Topology	Returns the raw physical topology by specified criteria of nodeType	/topology/physical-topology
GET	Get L3 Topology Details	Returns the Layer 3 network topology by routing protocol	/topology/l3/\${topologyType}
GET	Get Site Topology	Returns site topology	/topology/site-topology
GET	Get topology details	Returns Layer 2 network topology by specified VLAN ID	/topology/l2/\${vlanID}
GET	Get Overall Network Health ^{Intent}	Returns Overall Network Health information by Device category (Access, Distribution, Core, Router, Wireless) for any given point of time	/network-health

Monitoring your network

Topology



```
PHYSICAL_TOPOLOGY_URL = '/dna/intent/api/v1/topology/physical-topology'  
physical_topology_list = get_api(BASE_URL, PHYSICAL_TOPOLOGY_URL, headers)  
print ("\nPretty print response:\n",json.dumps(physical_topology_list,indent=4))
```

```
# Physical topology with device type, name, IP and Node Type  
json_tree = json.dumps(physical_topology_list,indent=4)  
data = json.loads(json_tree)
```

```
# Display deviceType, label, ip, and nodeType  
for node in data['nodes']:  
    print("Device Type:", node['deviceType'])  
    print("Label:", node['label'])  
    print("IP:", node['ip'])  
    print("Node Type:", node['nodeType'])  
    print()
```

Cisco Catalyst

Device Type: Cisco Catalyst 3650 Switch Stack
Label: MSX-OTT02-CAT3650-01
IP: 10.85.189.167
Node Type: device

Device Type: Third Party Device
Label: Street Left Camera
IP: 192.168.128.6
Node Type: device

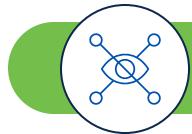
Device Type: Third Party Device
Label: MR28 - Upper floor
IP: 192.168.1.208
Node Type: device

Device Type: Cisco Catalyst 9800-CL Wireless Controller for Cloud
Label: MSX-OTT02-9800CL-06.cisco.com
IP: 10.85.189.78
Node Type: device

Meraki

Monitoring your network

Topology beauty example



```
from anytree import Node, RenderTree

# Parse JSON data
parsed_data = physical_topology_list

# Create a dictionary to store nodes by their IDs
nodes_dict = {}

# Create the root node
root = Node("Root")

# Create nodes and store them in the dictionary
for node_data in parsed_data['nodes']:
    node_id = node_data['id']
    label = node_data['label']
    parent_id = node_data['customParam']['parentNodeId']

    node = Node(label, parent=nodes_dict.get(parent_id, root))
    nodes_dict[node_id] = node

# Print the tree
for pre, fill, node in RenderTree(root):
    print(f"{pre}{node.name}")
```

Cisco Catalyst

Meraki

Root

```
MSX-OTT02-CAT3650-01
Street Left Camera
MR28 - Upper floor
MSX-OTT02-9800CL-06.cisco.com
MR30H-Floor-2
MR28 - First floor
MR20-HomeOffice-Floor2
MS120-HomeOffice
MX64W-HomeOffice
MR30H-Floor-1
MR20-HomeOffice-Floor1
```



Monitoring your network

Issues

GET

Issues^{Intent}

Intent API to get a list of global issues, issues for a specific device, or issue for a specific client device's MAC address.

/issues

```
{ □
  "issueId": "04f6c371-5b98-433b-98b2-375d67b6e95c",
  "name": "Excessive time lag between Cisco DNA Center and device \"MSX-OTT02-CAT3650-01\"",
  "siteId": "",
  "deviceId": "0ed45671-ec6f-4b66-9ab8-a919d31008f5",
  "deviceRole": "",
  "aiDriven": "",
  "clientMac": "",
  "issue_occurrence_count": 1,
  "status": "active",
  "priority": "",
  "category": "",
  "last_occurrence_time": 1682364608000
```



Monitoring your network

Issues

GET	Get Issue Enrichment Details ^{Intent}	Enriches a given network issue context (an issue id or end user's Mac Address) with details about the issue(s), impacted hosts and suggested actions for...	/issue-enrichment-details
	<pre>"issueDetails":{ □ "issue": [□ { □ "issueId": "04f6c371-5b98-433b-98b2-375d67b6e95c", "issueSource": "Cisco DNA", "issueCategory": "Device", "issueName": "device_time_drift", "issueDescription": "The time on Cisco DNA Center and Device \\\"MSX-OTT02-CAT3650-01\\\"", "issueEntity": "network_device", "issueEntityValue": "0ed45671-ec6f-4b66-9ab8-a919d31008f5", "issueSeverity": "HIGH", "issuePriority": "P3", "issueSummary": "Excessive time lag between Cisco DNA Center and device \\\"MSX-OTT02-CA", "issueTimestamp": 1682364608000, "deviceId": "0ed45671-ec6f-4b66-9ab8-a919d31008f5", }] }</pre>	<pre>"suggestedActions": [□ { □ "message": "Check time on the Device", "steps": [□ { □ "entityId": "0ed45671-ec6f-4b66-9ab8-a919d31008f5", "description": "Check system time", "command": "show clock", "stepType": "command-Runner", "runButton": null }] },]</pre>	



Monitoring your network

Device

GET

[Get Device list](#)

Returns list of network devices based on filter criteria such as management IP address, mac address, hostname, etc. You can use the .* in any value to conduct...

/network-device

GET

[Get Device Count](#)

Returns the count of network devices based on the filter criteria by management IP address, mac address, hostname and location name

/network-device/count

GET

[Get Device Summary](#)

Returns brief summary of device info such as hostname, management IP address for the given device Id

/network-device/\${id}/brief

GET

[Get Functional Capability by Id](#)

Returns functional capability with given Id

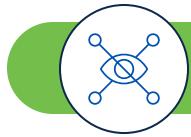
/network-device/functional-capability/\${id}

GET

[Get Device Config for all devices](#)

Returns the config for all devices

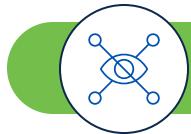
/network-device/config



Monitoring your network

Client

Client API Endpoints			
GET	Get Overall Client Health ^{<i>Intent</i>}	Returns Overall Client Health information by Client type (Wired and Wireless) for any given point of time	/client-health
GET	Client Proximity ^{<i>Intent</i>}	This intent API will provide client proximity information for a specific wireless user. Proximity is defined as presence on the same floor at the same time as...	/client-proximity
GET	Get Client Enrichment Details ^{<i>Intent</i>}	Enriches a given network End User context (a network user-id or end user's device Mac Address) with details about the user, the devices that the user is...	/client-enrichment-details
GET	Get Client Detail ^{<i>Intent</i>}	Returns detailed Client information retrieved by Mac Address for any given point of time.	/client-detail



Monitoring your network

Health and Performance

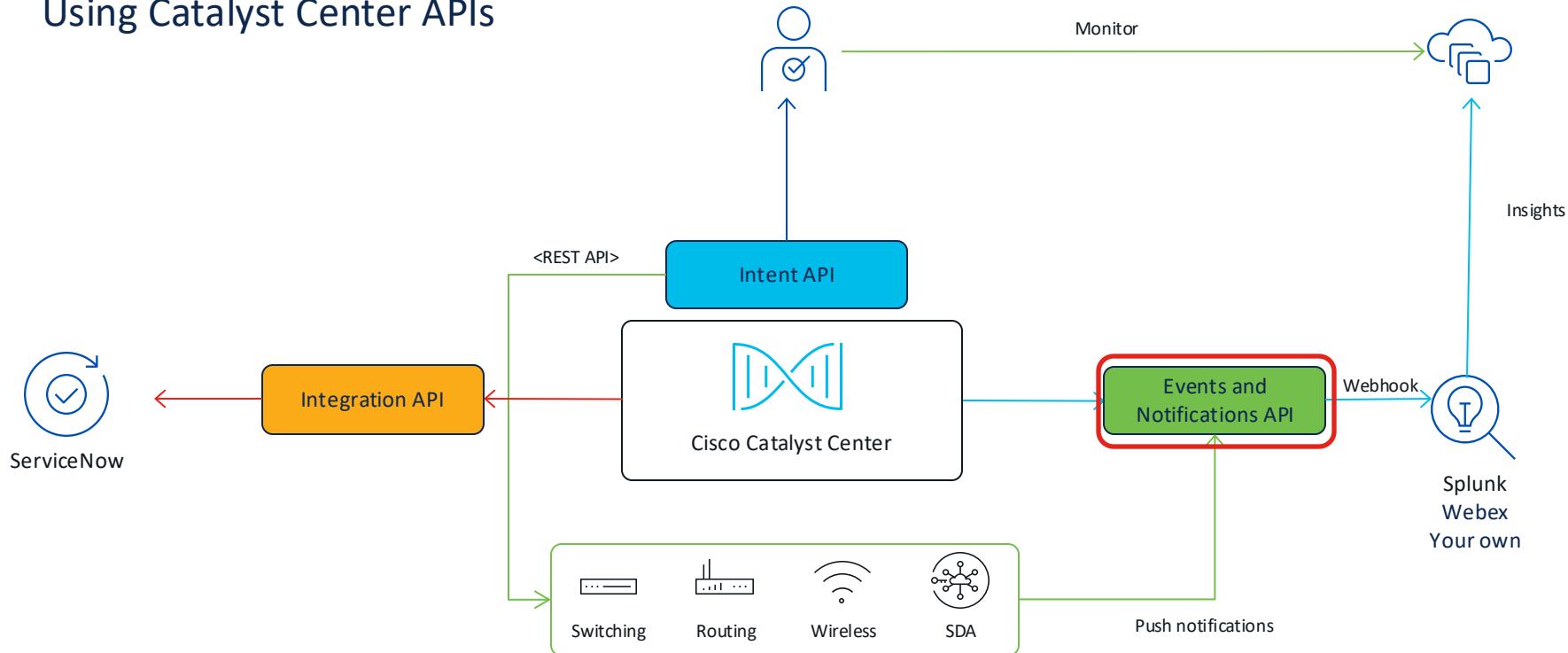
```
# Get System Health
def get_system_health (BASE_URL, URL, headers):
    response = requests.get(BASE_URL + URL,
                           headers=headers, verify=False)
    print(response)
    return response.json()|
```

Note: this response does not have the 'response' key in the API response

GET	System Health Count API ^{Intent}	This API gives the count of the latest system events	/diagnostics/system/health/count
GET	System Health API ^{Intent}	This API retrieves the latest system events	/diagnostics/system/health
GET	System Performance Historical API ^{Intent}	This API retrieves the historical performance indicators . The data can be retrieved for the last 3 months.	/diagnostics/system/performance/history
GET	System Performance API ^{Intent}	This API gives the aggregated performance indicators. The data can be retrieved for the last 3 months.	/diagnostics/system/performance

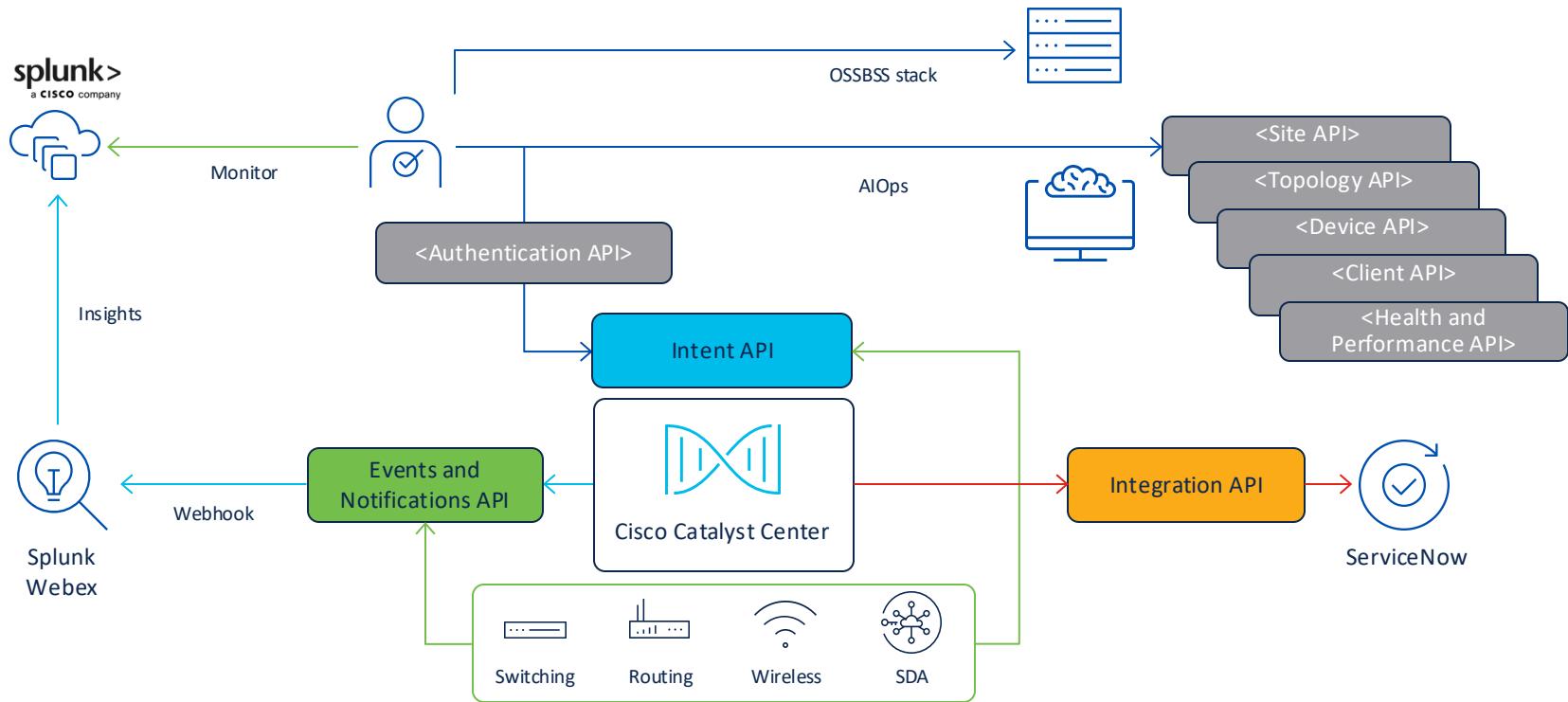
Headless operations

Using Catalyst Center APIs



Catalyst Center headless operations

Recap



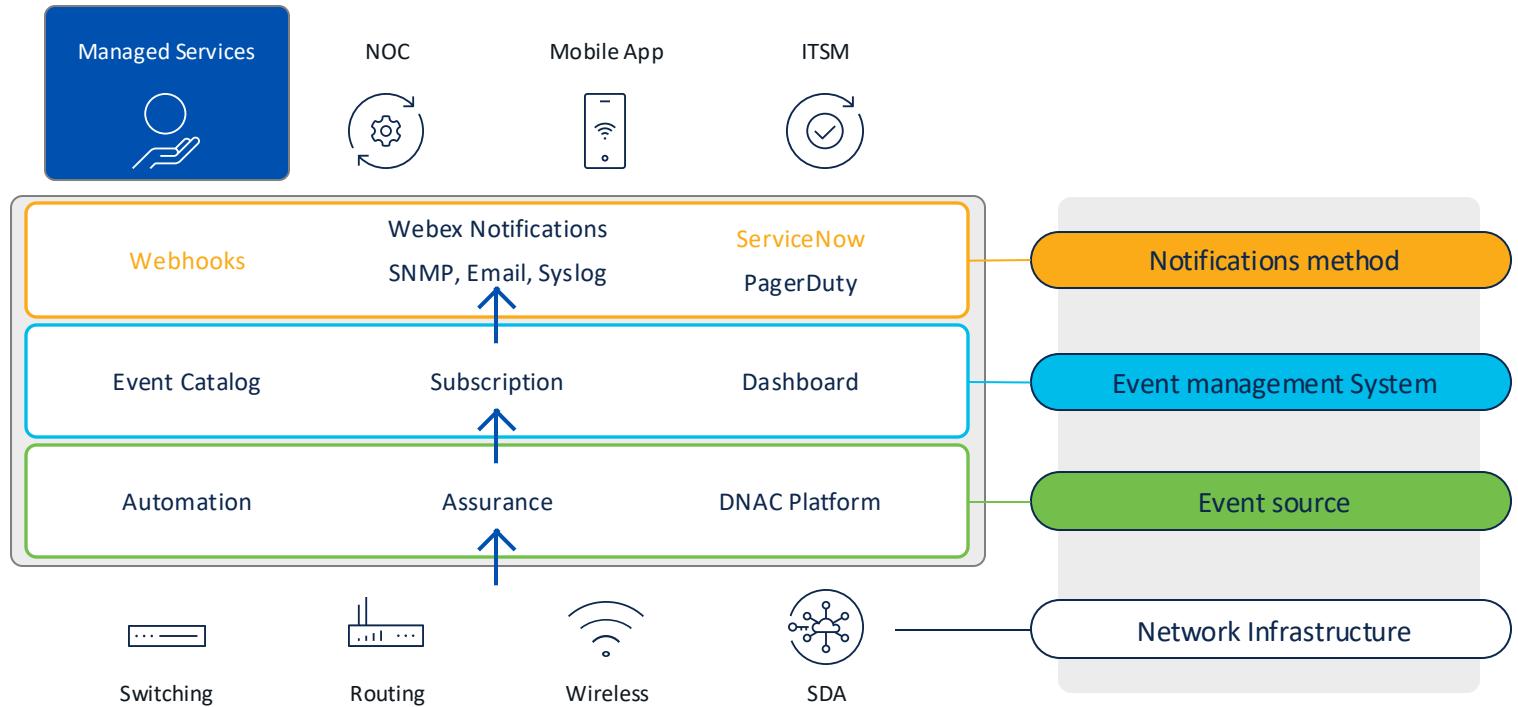
CatC APIs 400 level – Event Management API



This lab will use python

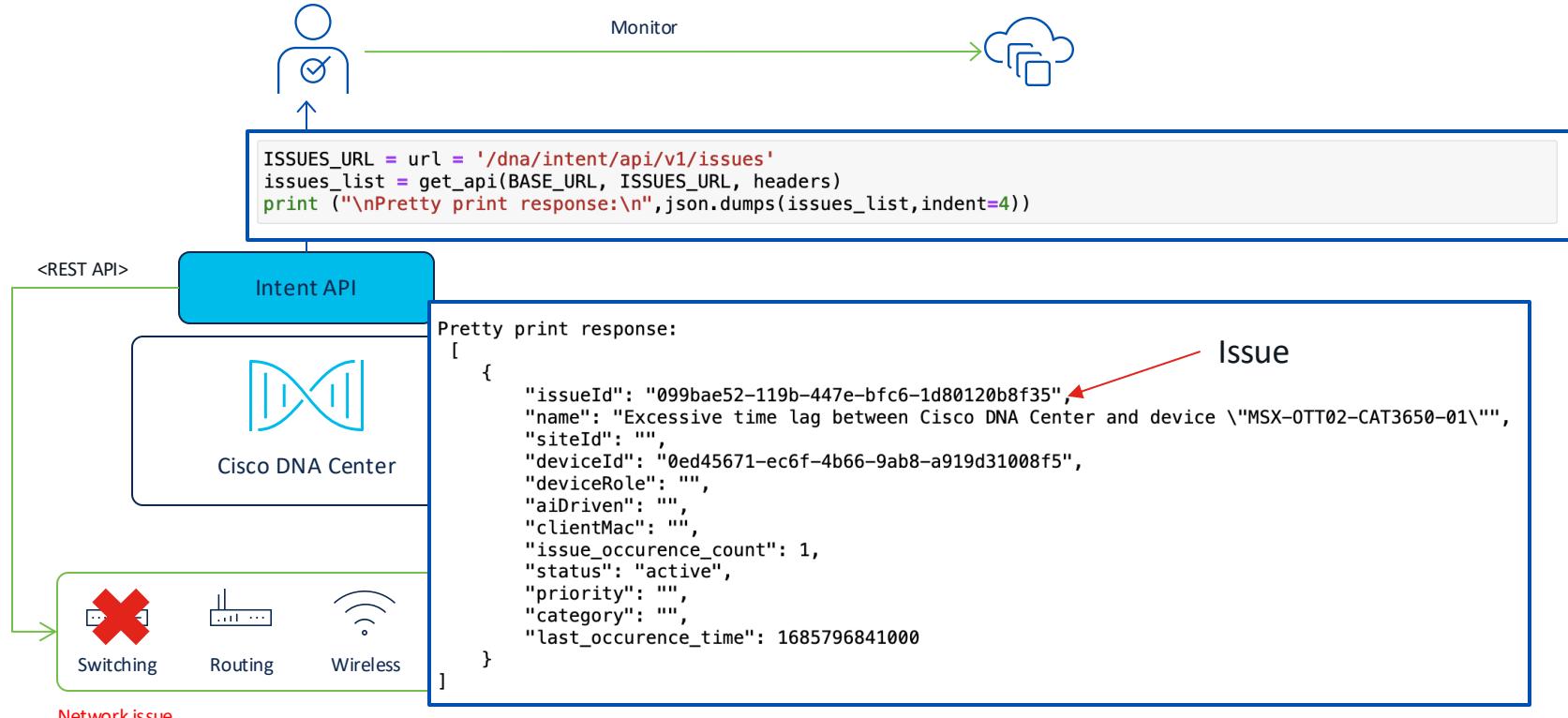
- You can do this in your laptop
- You already have installed Python, you might need additional packages.
- Directory for this lab:
 - <https://github.com/hemorale/CatC-API-Workshop/blob/main/Intent%20and%20Event%20Management/>

Catalyst Center Notifications Framework



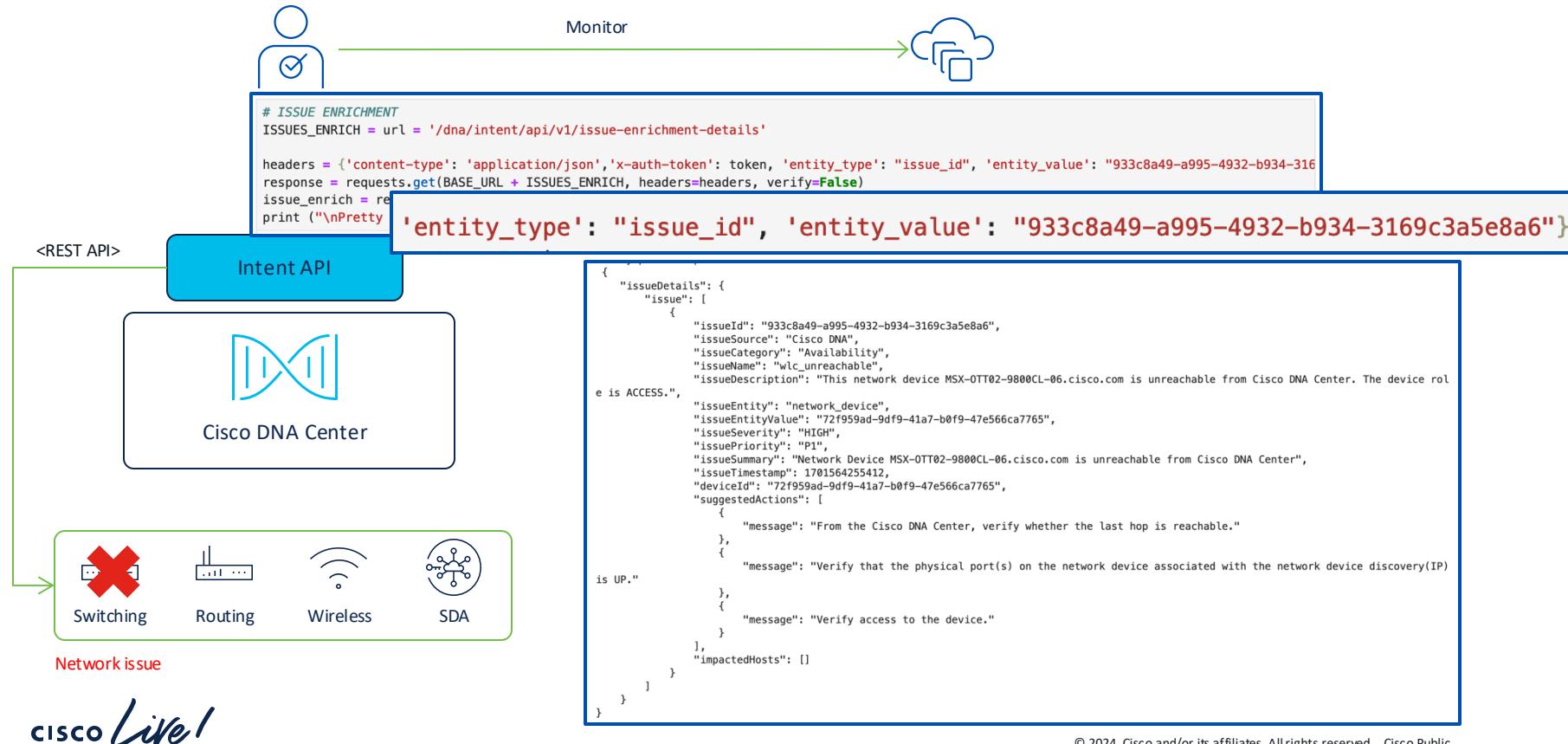
Managing your campus

Get issue using Intent API



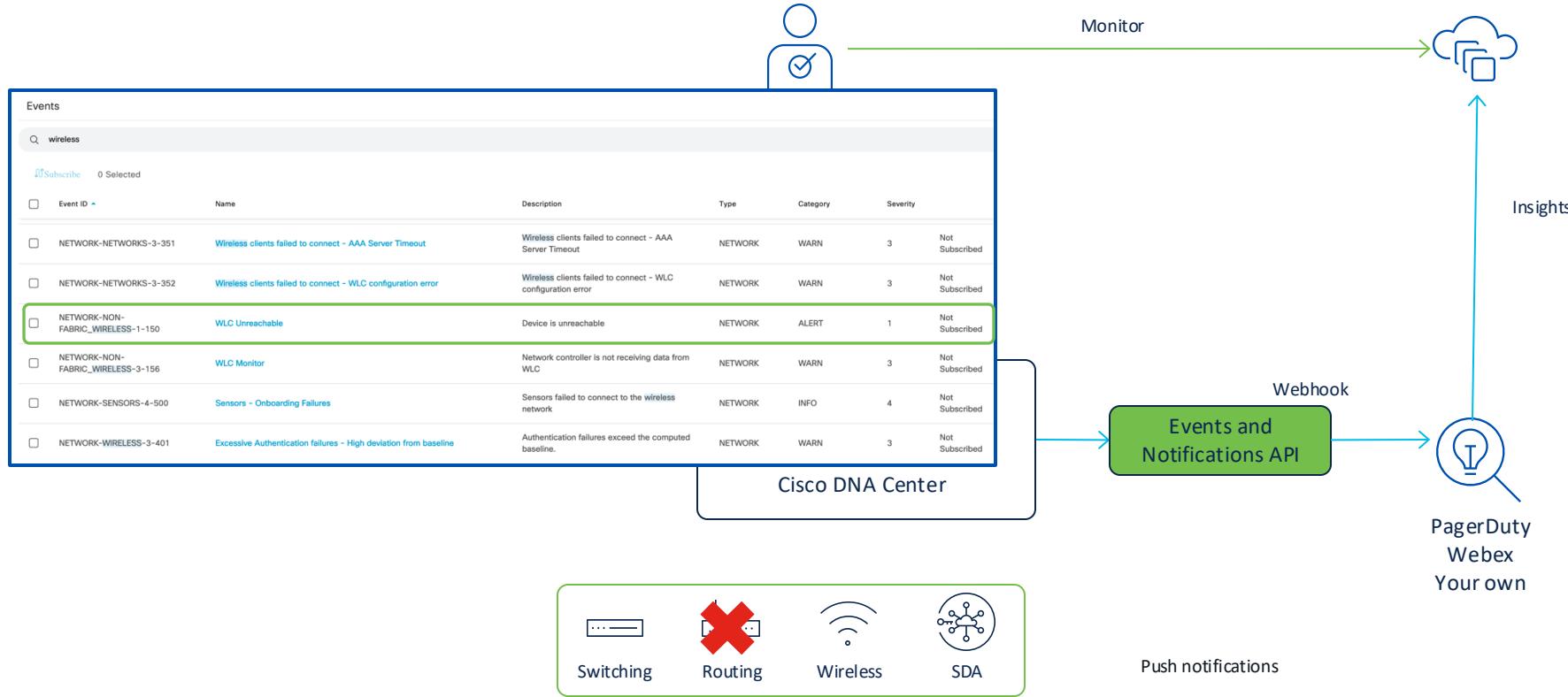
Managing your campus

Issue enrichment using Intent API



Managing your campus

Event notification



Lab prerequisites

1. Git clone the collection
<https://github.com/hemorale/CatC-API-Workshop/tree/main/Intent%20and%20Event%20Management>
2. Pip install all requirements
3. Open the jupyter notebook

Lab 1

Find the common issue between these collections

Issues_collection_verified.json

Issues_enrichment_verified.json

Events_collection_verified.json

'issueld'

'issueld' in issueDetails{}, issue[]

'instanceld'

1. Open jupyter notebook
2. Reusable python script: findissues.py

Webhook notifications workflow

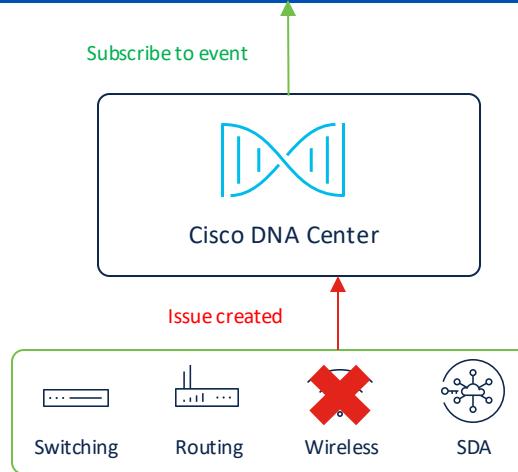


Better using event notifications

Event subscription



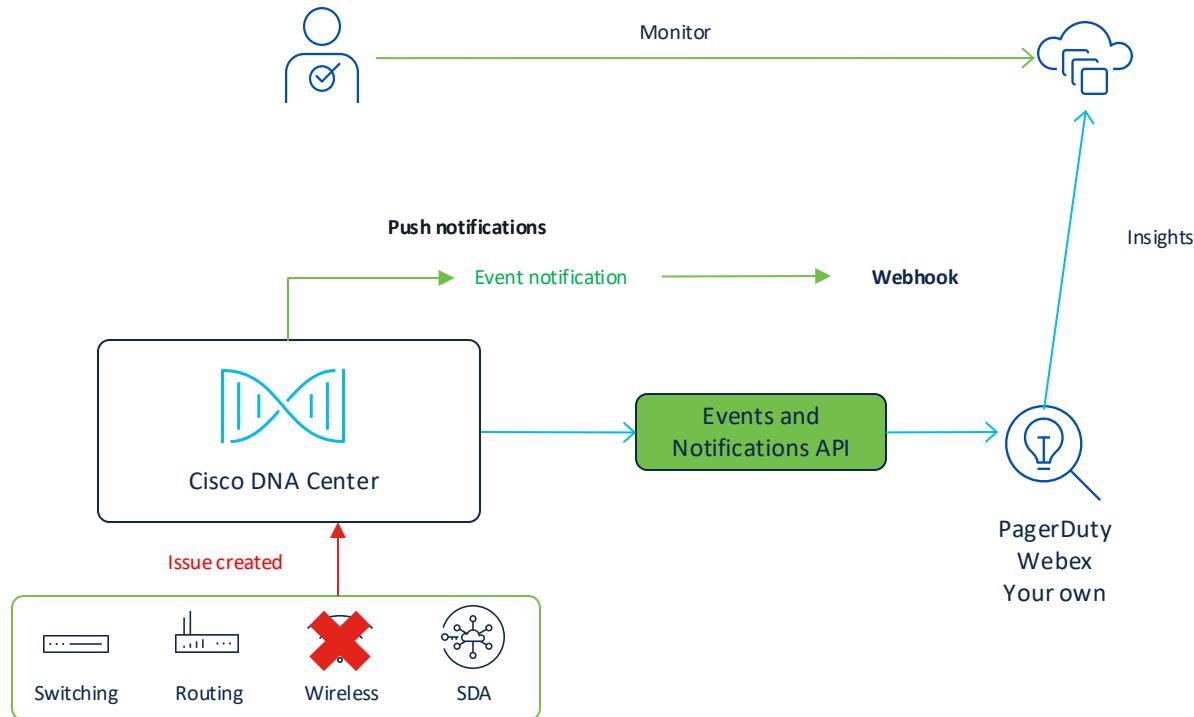
<input type="checkbox"/> NETWORK-NON-FABRIC_WIRELESS-1-150	WLC Unreachable	Device is unreachable	NETWORK	ALERT	1	Not Subscribed
--	-----------------	-----------------------	---------	-------	---	----------------



Better using event notifications

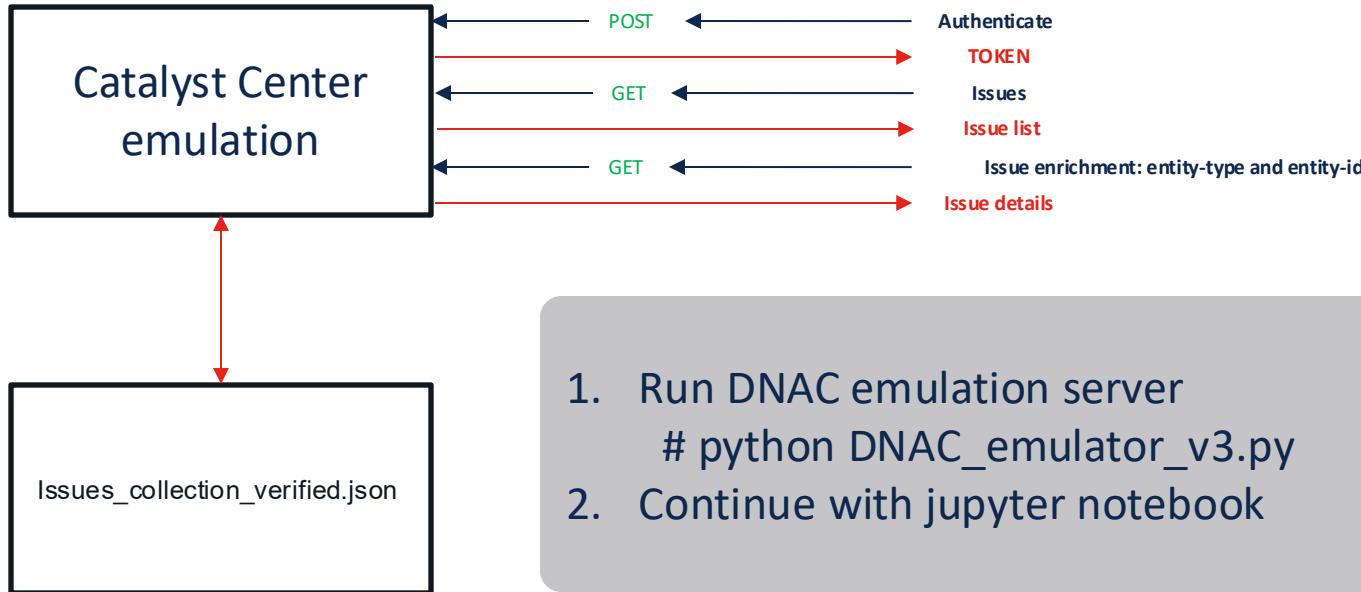
Event subscription

Once an issue is triggered by the assurance engine and if the event is subscribed, an event is sent out to the remote webhook location



Lab 2

Emulating Catalyst Center to get issues and issue enrichment



The webhook server

Emulating events and post them

```
def send_periodic_events():
    global send_events_flag, post_interval
    while True:
        if send_events_flag:
            try:
                with open('events_collection_verified.json', 'r') as file:
                    events_data = json.load(file)

                # Encode the credentials
                credentials = f'{VALID_USERNAME}:{VALID_PASSWORD}'
                encoded_credentials = base64.b64encode(credentials.encode()).decode()

                # Set up headers with Basic Authentication
                headers = {
                    'Authorization': f'Basic {encoded_credentials}',
                    'content-type': 'application/json'
                }

                # Sending POST request
                post_url = 'http://127.0.0.1:5001/dna/events/' # Update to your target server's URL
                response = requests.post(post_url, json=events_data, headers=headers, verify=True)
                print("POST request sent. Status Code:", response.status_code)
            except Exception as e:
                print("Error sending POST request:", str(e))
            time.sleep(post_interval)
```

```
# Define valid credentials for Basic Authentication
users = {
    "admin": "CiscoLive100%"
}

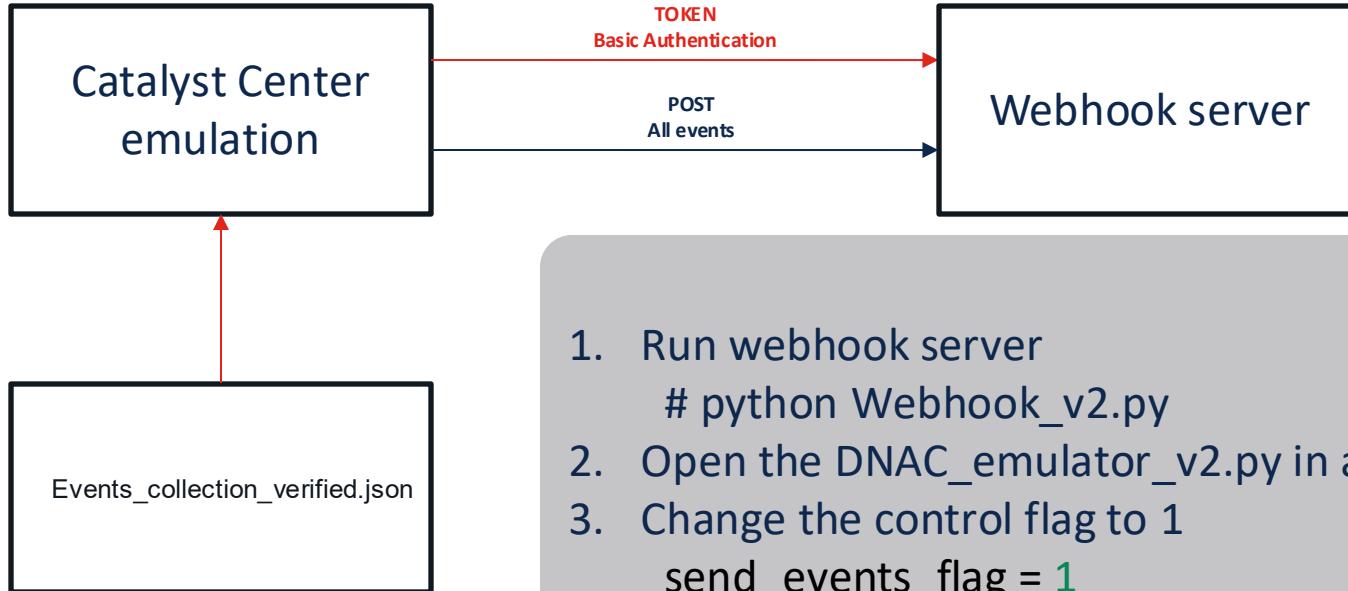
@auth.verify_password
def verify_password(username, password):
    if username in users and users[username] == password:
        return username

@app.route('/dna/events/', methods=['POST'])
@auth.login_required
def receive_events():
    data = request.json
    print("Received POST request with data:", json.dumps(data, indent=4))
    return jsonify({"message": "Data received successfully"}), 200
```

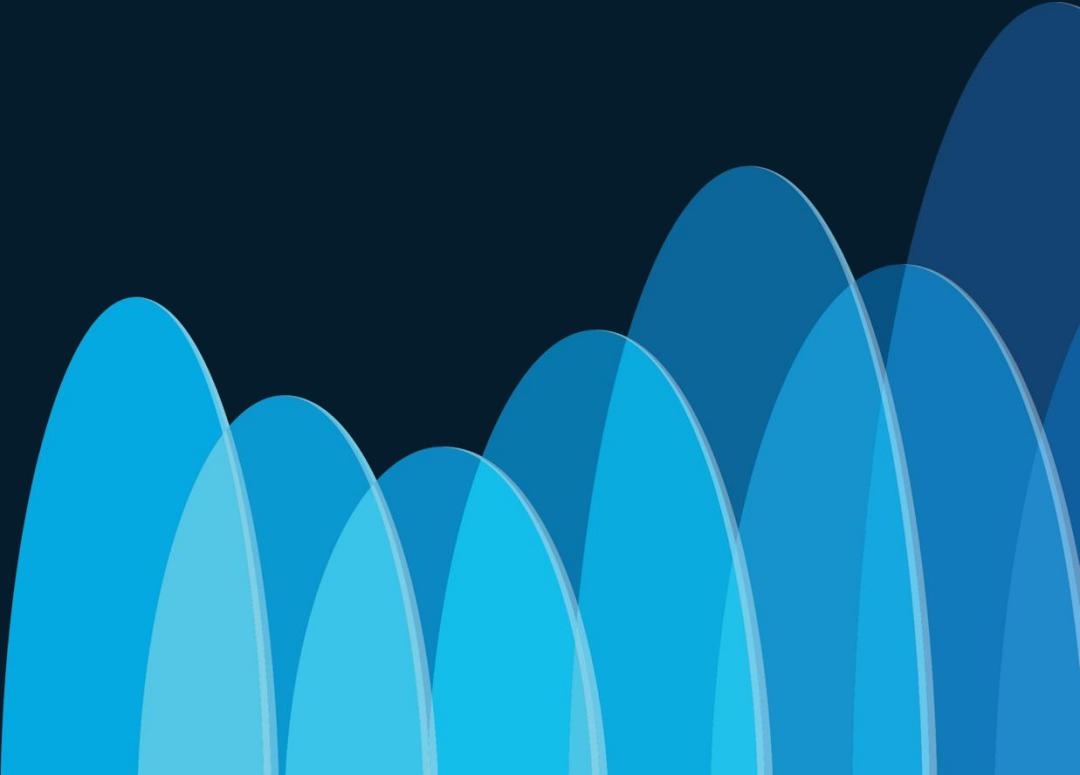
Events_collection_verified.json

Lab 3

Creating a webhook server



Setting up a webhook destination on Catalyst Center

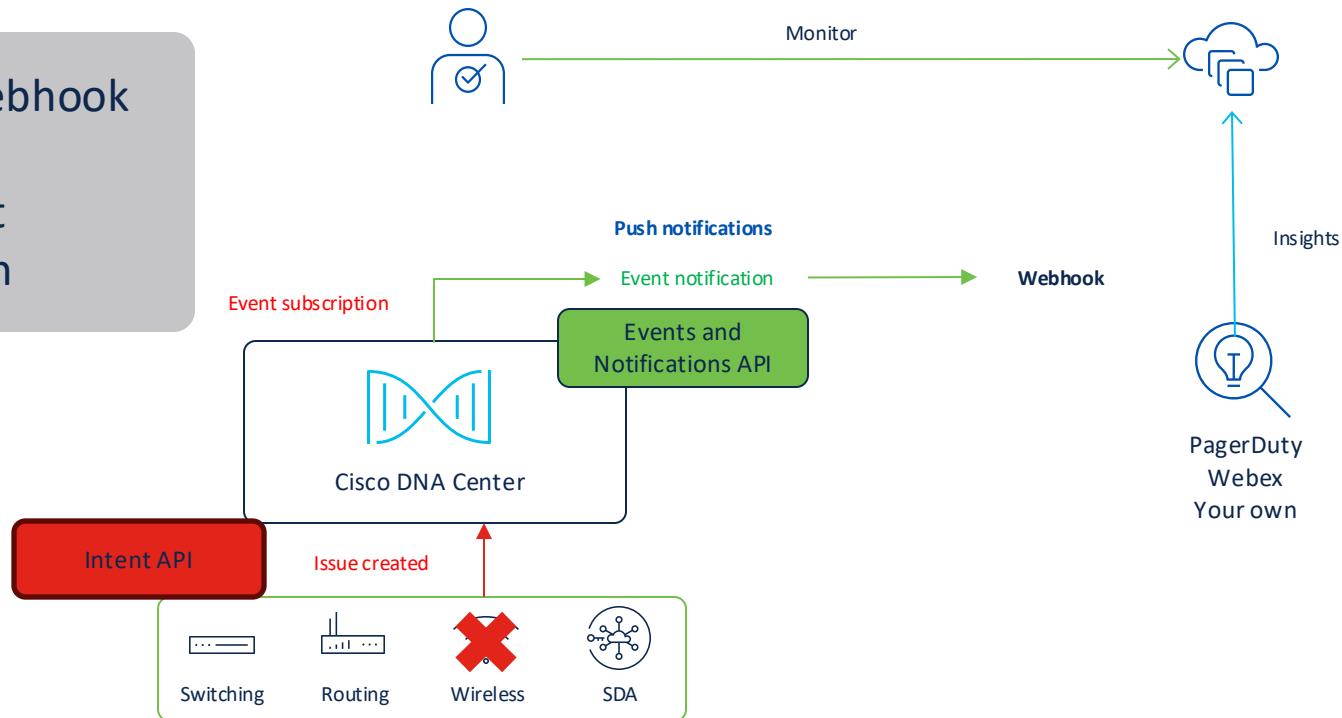


This lab will use python

- You can do this in your laptop
- You will need Ngrok and the Webhook_server.py script
- Detailed Ngrok installation instructions are in the collection
- Directory for this lab:
 - <https://github.com/hemorale/CatC-API-Workshop/blob/main/Intent%20and%20Event%20Management/>

Event notifications flow

1. Setup a new webhook destination
2. Subscribe event
3. Test notification



Setting up a destination

Webhook

The screenshot shows the Cisco DNA Center interface. The left sidebar lists various navigation items like Design, Policy, Provision, Assurance, Workflows, Tools, Platform, Activities, Reports, System, and Explore. The 'System' item is expanded, showing sub-options like System 360, Software Up, Settings, Data Platform, System Health, Users & Roles, Backup & Recovery, and Disaster Recovery. The 'Settings' option is highlighted. A modal window titled 'External Services' is open over the main content area. It contains sections for 'Umbrella' (Register Umbrella with your Cisco DNA Center), 'Authentication and Policy Servers' (Specify the servers that authenticate Cisco DNA Center users. Cisco Identity Services Engine (ISE) servers can also supply policy and user information), 'Authentication Tokens' (Manages connection between Cisco DNA Center and external authentication servers), 'vManage' (Connect Cisco DNA Center to vManage), 'Cisco AI Analytics' (Set up the Cisco artificial intelligence analytics service), and 'Destinations' (Configure various types of destinations). The 'Destinations' section is highlighted with a blue border. Below the modal, the main content area has a title 'Destinations' and a sub-instruction 'Configure various types of destinations to deliver event notifications from Cisco DNA Center Platform'. It lists five destination types: Webhook, Email, Syslog, SNMP, and ITSM. The 'Webhook' option is underlined, indicating it is selected. A detailed description for the Webhook destination says 'Configure the REST Endpoint to receive Audit logs and Events from Cisco DNA Center Platform'.

Destinations

Configure various types of destinations to deliver event notifications from Cisco DNA Center Platform

Webhook Email Syslog SNMP ITSM

Configure the REST Endpoint to receive Audit logs and Events from Cisco DNA Center Platform

Webhook provisioning

Add Webhook

Name*
Webhook Test

Description
Webhook Test

URL*
webhook.cisco.com

Trust Certificate
 Yes No

Method*
POST

Authentication
 Basic Token No Auth

Headers
Header Name Header Value
Authorization Basic YWRtaW46Q2lzY29MaXZlMTAwJQ==

[Add](#)

[Save](#)

Trust Certificate: **YES**

Method: **POST**

Authentication: **Basic**

Headers:

'Authorization': 'Basic Encoded64(Username+Password)'

```
import base64

# Define valid credentials
VALID_USERNAME = 'admin'
VALID_PASSWORD = 'CiscoLive100%'

# Encode the credentials
credentials = f'{VALID_USERNAME}:{VALID_PASSWORD}'
encoded_credentials = base64.b64encode(credentials.encode()).decode()

print(encoded_credentials)
print(f'Basic {encoded_credentials}')
```

YWRtaW46Q2lzY29MaXZlMTAwJQ==
Basic YWRtaW46Q2lzY29MaXZlMTAwJQ==

Name	Description	URL	Method
Webhook Test	Webhook Test	https://webhook.cisco.com	POST

Subscribe to events

Cisco DNA Center

- Design
- Policy
- Provision
- Assurance
- Workflows
- Tools
- Platform
- Activities
- Reports
- System
- Explore

Overview

Manage

Developer Toolkit

Cisco DNA Center

APIs Integration Flows Multivendor Support Events

Events

wlc

Subscribe 0 Selected

Event ID	Name
NETWORK-DEVICES-2-106	AP disconnected from WLC
NETWORK-DEVICES-2-152	WLC Reboot Crash
NETWORK-DEVICES-2-153	WLC Power Supply Failure
NETWORK-DEVICES-3-154	WLC Memory High Utilization
NETWORK-DEVICES-3-155	AP License Exhausted on WLC
NETWORK-FABRIC_WIRELESS-1-307	Fabric WLC to MapServer Connection Lost
NETWORK-NETWORKS-3-352	Wireless clients failed to connect
NETWORK-NON-FABRIC_WIRELESS-1-150	WLC Unreachable
NETWORK-NON-FABRIC_WIRELESS-3-156	WLC Monitor

Subscribe

1 Event selected

Name*
WLC unreachable|

Subscription Type
REST

Select an existing instance. Or Click [here](#) to create a new instance.

Subscription Endpoint
Webhook Test

Model Schema

REST Schema

```
1+ { "Type": "$eventSource$", "Assurance Issue Priority": "$priority$", "Assurance Issue Details": "This network device $nwDeviceName$ is unreachable from Cisco DNA Center. The device role is $fabricOrDeviceRole$. Device: $deviceUniqueIds$.", "Assurance Issue Category": "$category$", "Assurance Issue Name": "Network Device $managementIpAddress$ Is Unreachable From Controller", "Assurance Issue Status": "$status$"}
```

Cancel Subscribe

CC Webhooks notifications workflow

Notification

Event Name: WLC Unreachable

Event Id: NETWORK-NON-FABRIC_WIRELESS-1-150

Namespace: ASSURANCE

Context: EXTERNAL

Source: EXTERNAL

Type: NETWORK

Category: ALERT

Severity: 1

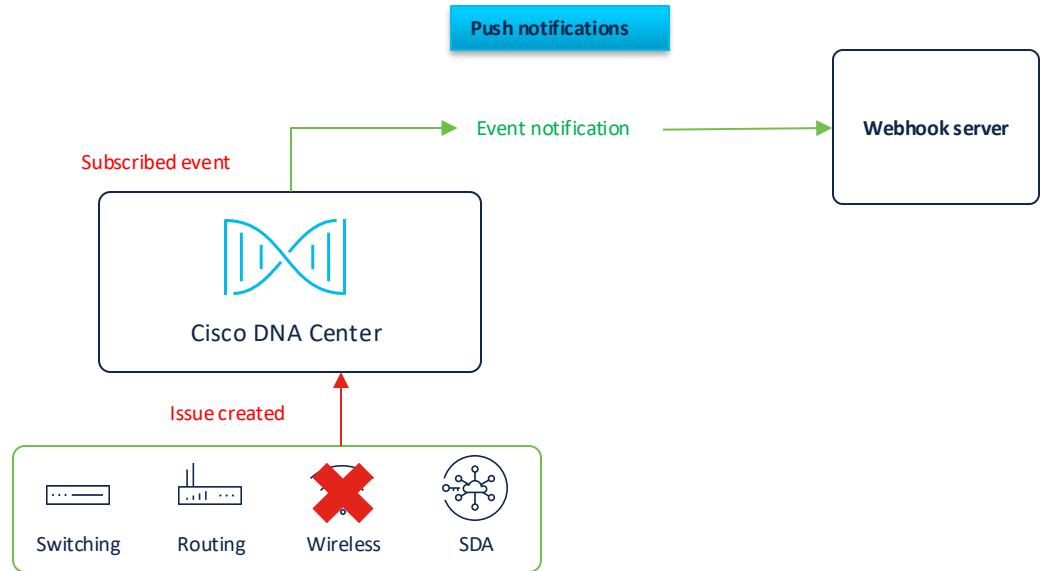
Domain: Connectivity

Sub Domain:

Details:

```
{  
  "Type": "$eventSource$"  
  "Assurance Issue Priority": "$priority$"  
  "Assurance Issue Details": "This network device $mDeviceName$ is unreachable from Cisco DNA Center. The Device": "$deviceUniqueId$"  
  "Assurance Issue Category": "$category$"  
  "Assurance Issue Name": "Network Device $managementIpAddress$ Is Unreachable From Controller",  
  "Assurance Issue Status": "$status$"  
}
```

Cancel Publish



Try on your own

```
Received POST request with data: {  
    "version": "1.0.0",  
    "instanceId": "8b3d20dc-f161-4f4d-97fc-56cb6d1dec8a",  
    "eventId": "NETWORK-NON-FABRIC_WIRELESS-1-150",  
    "namespace": "ASSURANCE",  
    "name": "WLC Unreachable",  
    "description": "Device is unreachable",  
    "type": "NETWORK",  
    "category": "ALERT",  
    "domain": "Connectivity",  
    "subDomain": "Non-Fabric Wireless",  
    "severity": 1,  
    "source": "EXTERNAL",  
    "timestamp": 1701690738552,  
    "details": {  
        "Type": "",  
        "Assurance Issue Priority": "",  
        "Assurance Issue Details": "This network device is unreachable from Cisco DNA Center. The device role is .",  
        "Device": "",  
        "Assurance Issue Category": "",  
        "Assurance Issue Name": "Network Device Is Unreachable From Controller",  
        "Assurance Issue Status": ""  
    },  
    "ciscoDnaEventLink": "https://<IP_ADDRESS>/dna/assurance/issueDetails?issueId=",  
    "note": "To programmatically get more info see here - https://<ip-address>/dna/platform/app/consumer-portal/developer-toolkit/apis?apiId=8684-39bb-4e89-a6e4",  
    "context": "EXTERNAL",  
    "userId": null,  
    "i18n": null,  
    "eventHierarchy": null,  
    "message": null,  
    "messageParams": null,  
    "parentInstanceId": null,  
    "network": null,  
    "dnacIP": null  
}  
127.0.0.1 - - [04/Dec/2023 22:52:22] "POST /dna/events/ HTTP/1.1" 200 -
```

Notification

Event Name

WLC Unreachable

Event Id

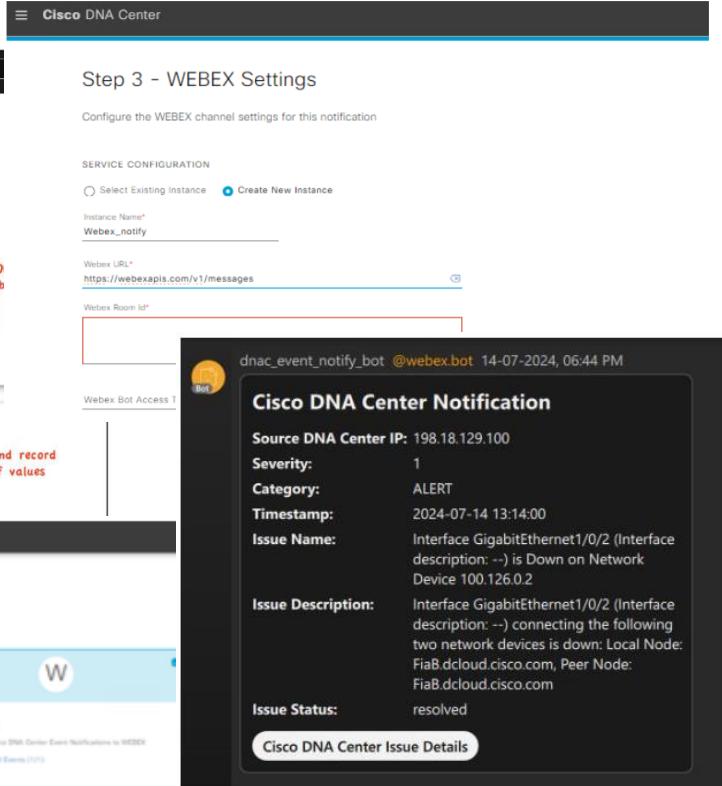
NETWORK-NON-FABRIC_WIRELESS-1-150

Result

Subscription Name	Connector Type	Status	Message
WLC unreachable	REST	✓ SUCCESS	OK

Best Practices

Webex Bot as webhook destination



The screenshot shows the Cisco DNA Center interface. A prominent message says: "Cisco DNA Center Event Notification Bot1 is one step closer to becoming a reality." Below it, there are fields for "Bot access token" and "Bot ID". To the right, a callout box says: "** Click CO in your Tab" and "** Click COPY and record in your Table of values". At the bottom, there's a "Cisco DNA Center Notification" card with details like Source DNA Center IP: 198.18.129.100, Severity: 1, Category: ALERT, and Issue Description: Interface GigabitEthernet1/0/2 (Interface description: --) is Down on Network Device 100.126.0.2.

Step 3 - WEBEX Settings
Configure the WEBEX channel settings for this notification

SERVICE CONFIGURATION

Select Existing Instance Create New Instance

Instance Name*: Webex_notify

Webex URL*: https://webexapis.com/v1/messages

Webex Room ID*: [redacted]

Webex Bot Access T [redacted]

Congratulations! Cisco DNA Center Event Notification Bot1 is one step closer to becoming a reality.

Cisco DNA Center Event Notification Bot1

Bot name*: Cisco DNA Center Event Notification Bot1

Bot username*: [redacted] dhweber.bot

Icon*: [redacted] Bot is available

Description*: Provide some details about what your bot does. Hint: benefits users, and most importantly, how a user can gain value from it. The description should be under 1900 characters. You can use bullet lists, and Markdown syntax. If your app is listed on the Webex App Hub, this field will be filled in as the bot's description.

Supported markdown: 1384 characters remaining

By creating this app, you accept the Terms of Service and Privacy Statement

Create a New Notification

Step 2 - Select Channels
Choose the notification channels

EMAIL Send an Email notification
Supported Events (1/1)

PAGERDUTY POST Cisco DNA Center Event Notifications to PagerDuty
Supported Events (1/1)

REST Send the data via HTTP post API
Supported Events (1/1)

SYSLOG Send data to a Syslog server
Supported Events (1/1)

WEBEX POST Cisco DNA Center Event Notifications to WEBEX
Supported Events (1/1)

More information on [salesconnect](#) (access for partners)

CISCO Live!

Automated bulk upgrade of network devices

Vulnerability assessment for devices managed by multiple Catalyst-Center

The objective of this use case is to streamline and provision the process of upgrading network devices managed by multiple Catalyst Centers driven by vulnerability assessments to ensure that all network devices are up-to-date with the recommended security patches and firmware versions as suggested by Catalyst Center.

1

Perform Security Advisory Retrieval

```
post_url = "https://"+ip+"/api/system/v1/auth/token"
url_sites = f"https://{ip_address}/dna/intent/api/v1/site"
url_membership = f"https://{ip_address}/dna/intent/api/v1/membership/{site_id}"
url_advisory = f"https://{ip_address}/dna/intent/api/v1/security-advisory/device/{key}/advisory"
```

2

Perform Software Image Management on the
devices on network

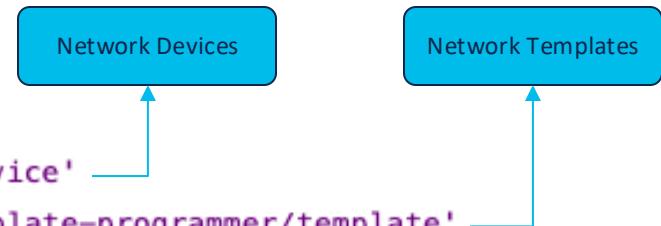
```
url_images = f"https://{ip}/dna/intent/api/v1/image/importation"
url_distribute = f"https://{ip}/dna/intent/api/v1/image/distribution"
url_device = f"https://{ip}/dna/intent/api/v1/network-device"
url_activate = f"https://{ip}/dna/intent/api/v1/image/activation/device"
url_task = f"https://{host}/api/v1/task/{taskid}"
```

More information on [salesconnect](#) (access for partners)

Pushing customized template

Devices spread across multiple sites (topology, inventory, template)

To automate the deployment of customised configuration templates to network devices distributed across multiple sites by leveraging APIs to manage topology, inventory, and template application, thereby ensuring consistent and efficient configuration management, and reducing manual intervention.



```
post_url = "https://"+ip+"/api/system/v1/auth/token"
url_device = f'https://{{ip}}/dna/intent/api/v1/network-device'
url_show_template = f'https://{{ip}}/dna/intent/api/v1/template-programmer/template'
url_search_template = f'https://{{ip}}/dna/intent/api/v1/template-programmer/template'
url_template = f'https://{{ip}}/dna/intent/api/v1/template-programmer/template/{{templateId}}'
url_template_apply = f'https://{{ip}}/dna/intent/api/v2/template-programmer/template/deploy'
url = f'https://{{host}}/api/v1/task/{{taskid}}'
url = f'https://{{host}}/dna/intent/api/v1/template-programmer/template/deploy/status/{{deploymentId}}'
```

More information on [salesconnect](#) (access for partners)



Thank you

CISCO *Live!*



GO BEYOND