

cisco *Live!*

Let's go

#CiscoLiveAPJC



The bridge to possible

Building a webhook server for monitoring Cisco Catalyst Center operations using Intent APIs and Event Notifications

Hector Morales, Sr. Global TSA, Global Partner Engineering

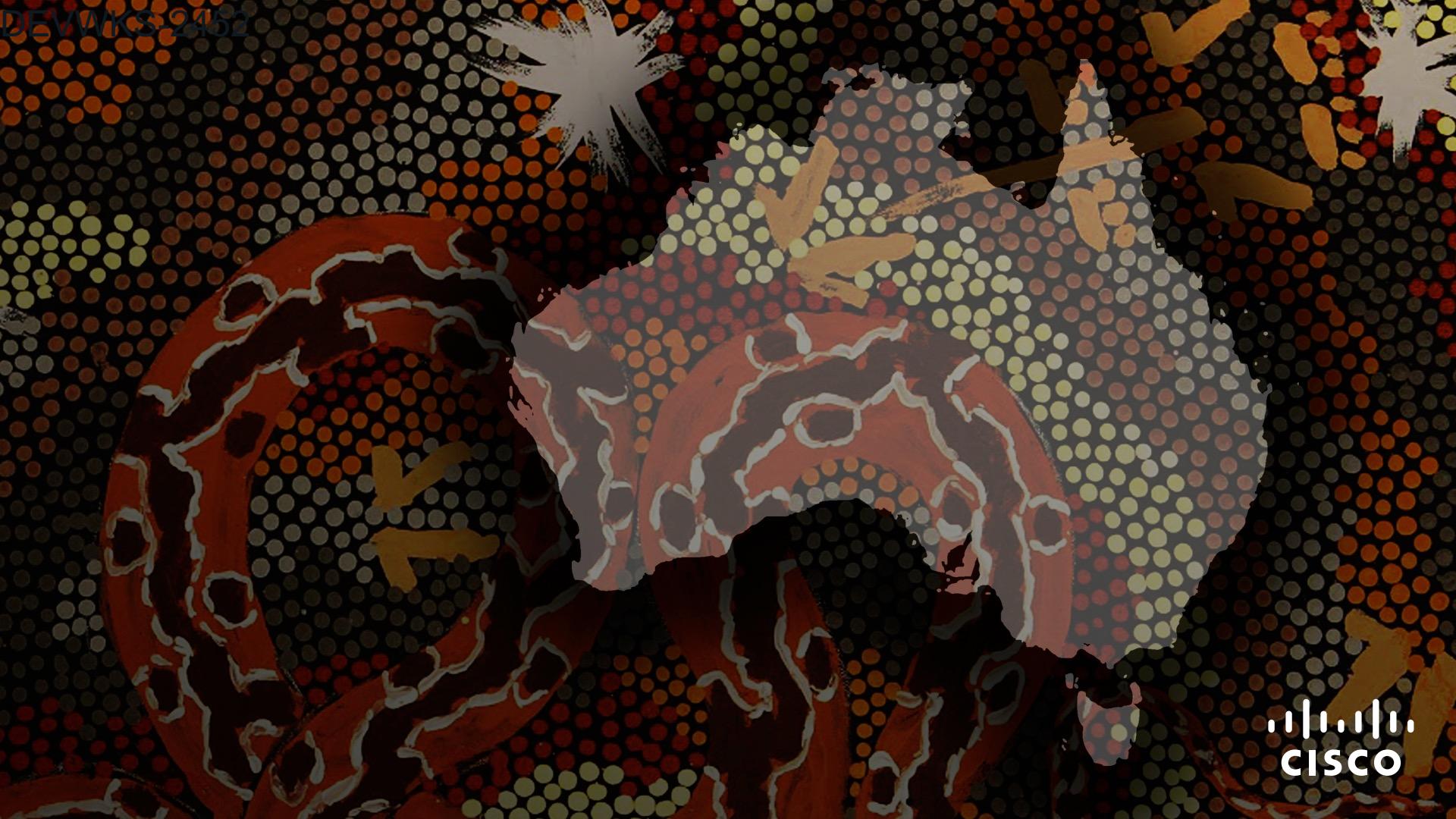
DEVWKS-2452



#CiscoLiveAPJC



DEVWKS-2452




CISCO

Cisco Webex App

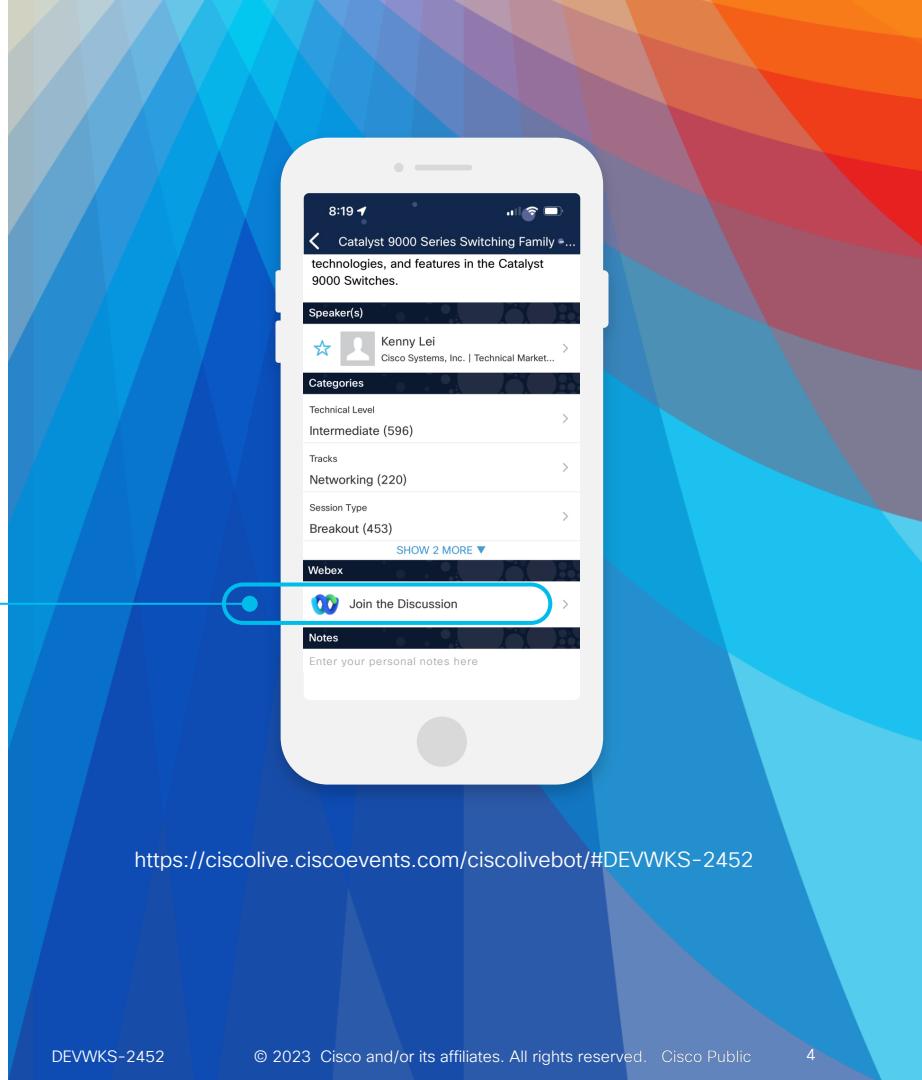
Questions?

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

How

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

Webex spaces will be moderated by the speaker until December 22, 2023.



<https://ciscolive.ciscoevents.com/ciscolivebot/#DEVWKS-2452>



Agenda

- The problem we are trying to solve
- Catalyst Center Event Notification Framework
- Webhook notifications workflow
- Setting up a webhook destination on Catalyst Center

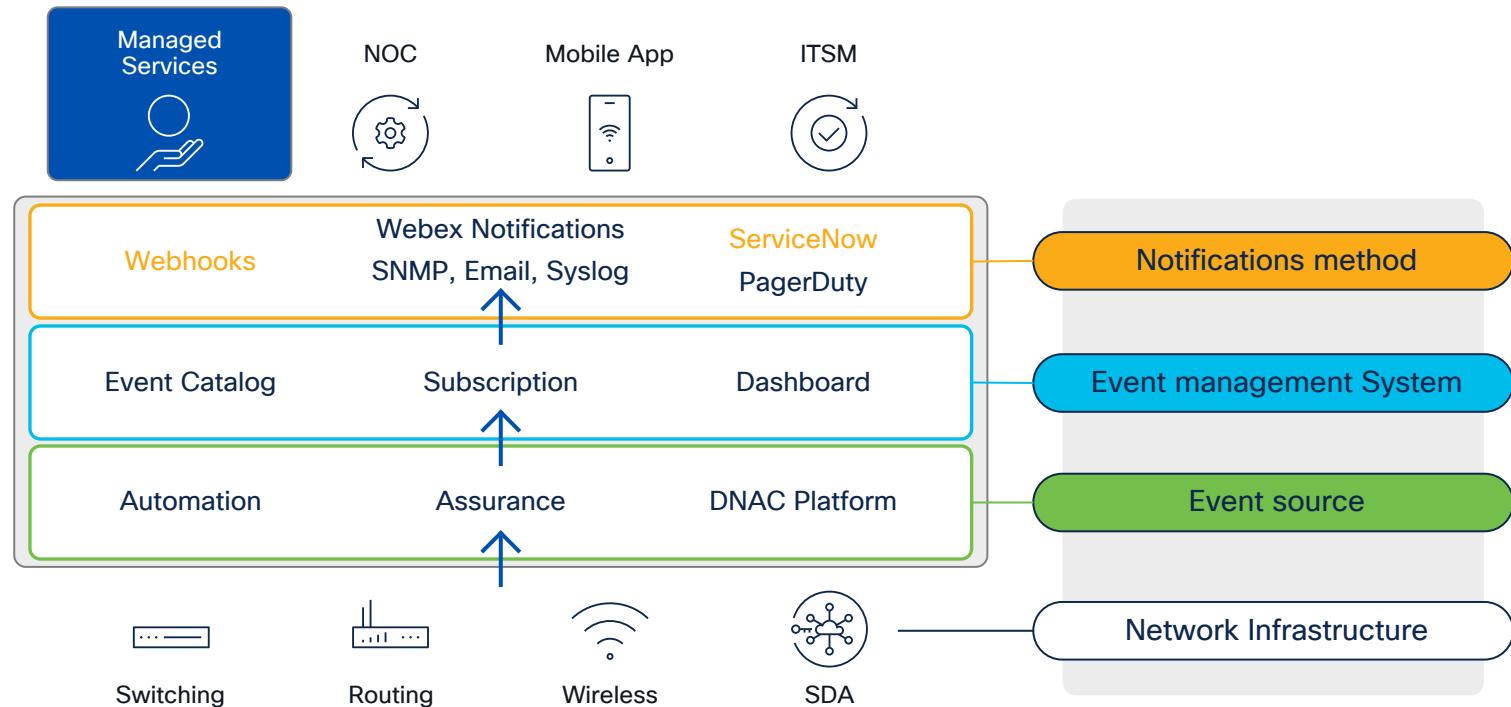
DEVWKS-2452

Problem we are trying to solve

- Most of us, managing Catalyst Center, we have limited capabilities to monitor network events and issues
- Monitoring events and issues is critical to either manage risk or automate events into our support pipelines
- The goal of this session is to create an event and issues emulation framework that can help you to understand:
 - Catalyst Center assurance architecture
 - Events vs Issues
 - Simulate event, enrich issues, insert in your support pipeline

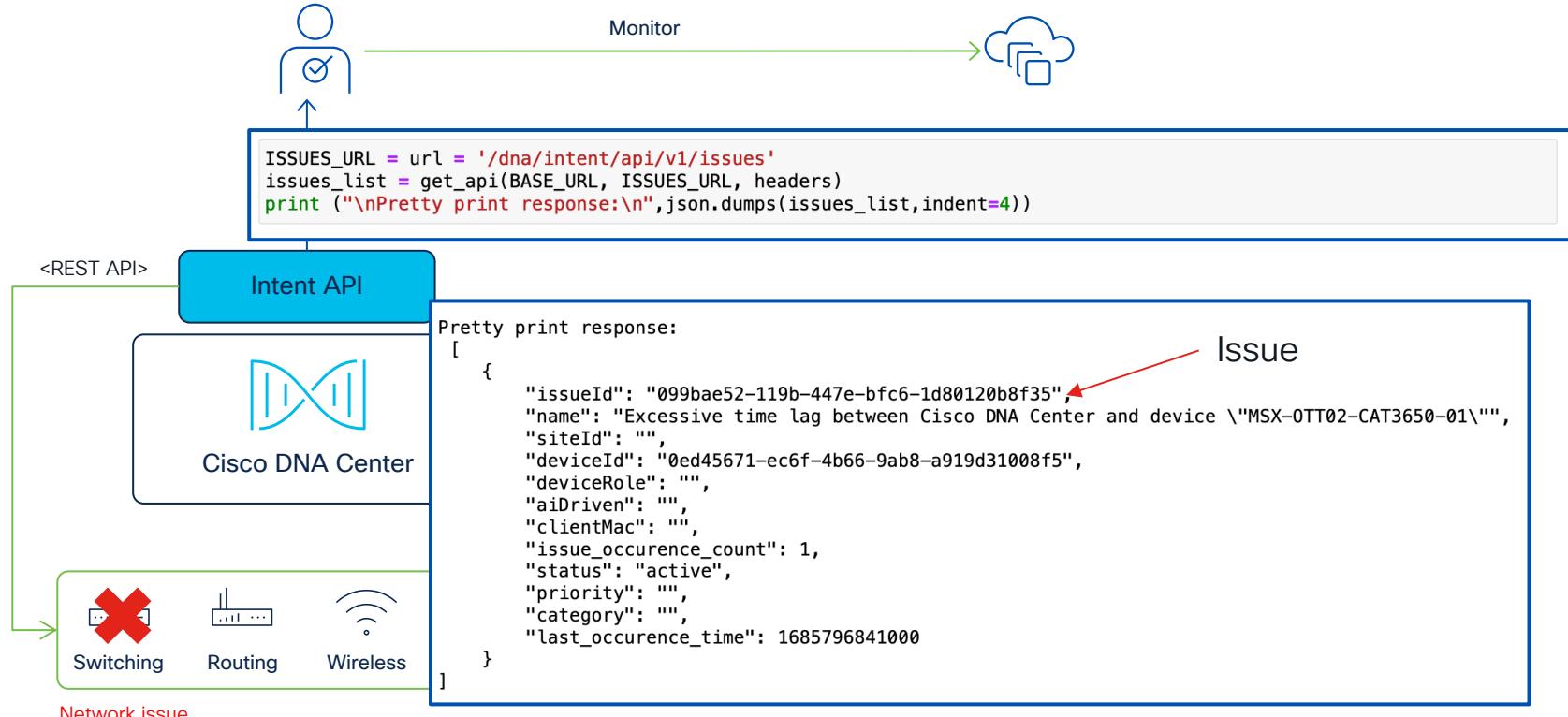
Catalyst Center Event Notification Framework

Catalyst Center Notifications Framework



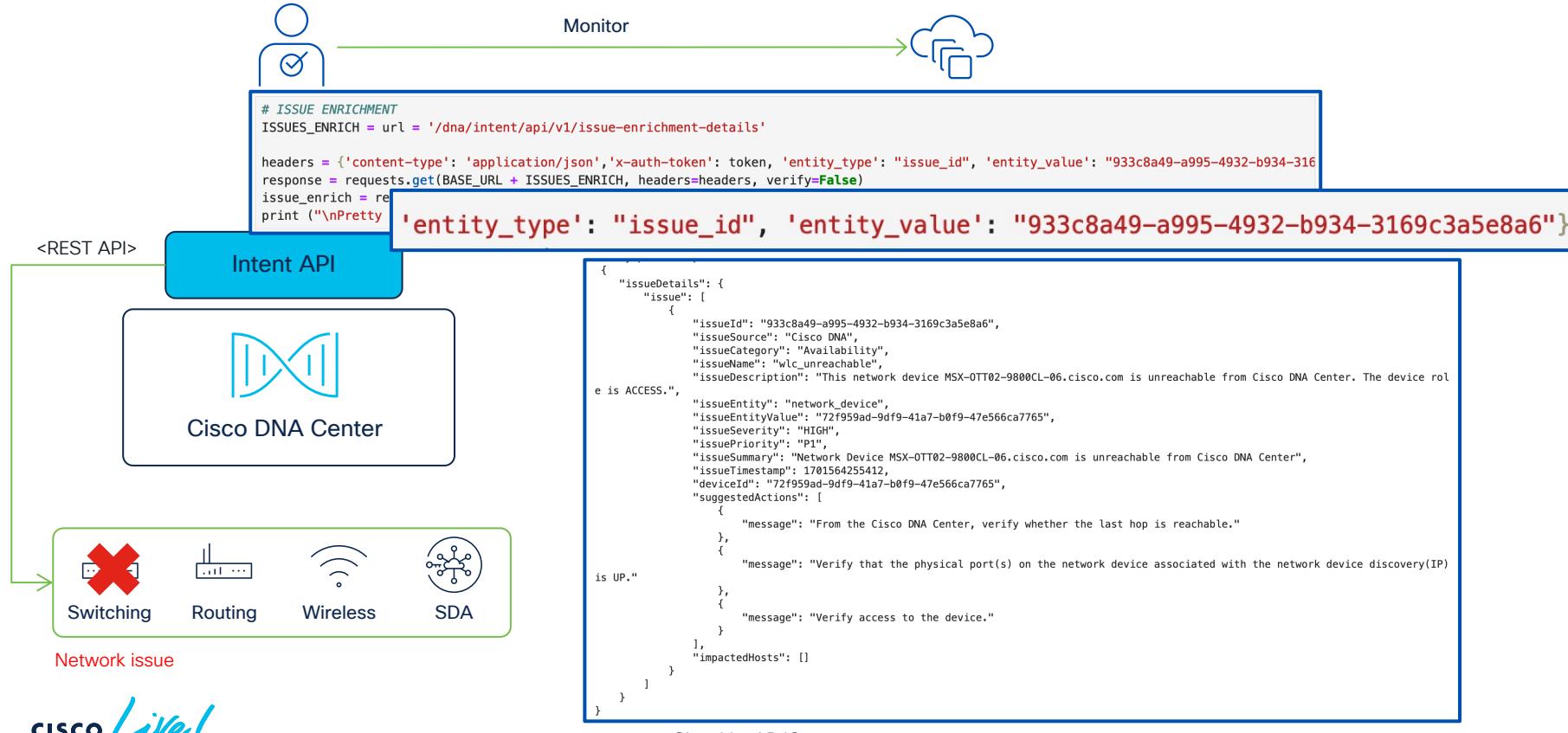
Managing your campus

Get issue using Intent API



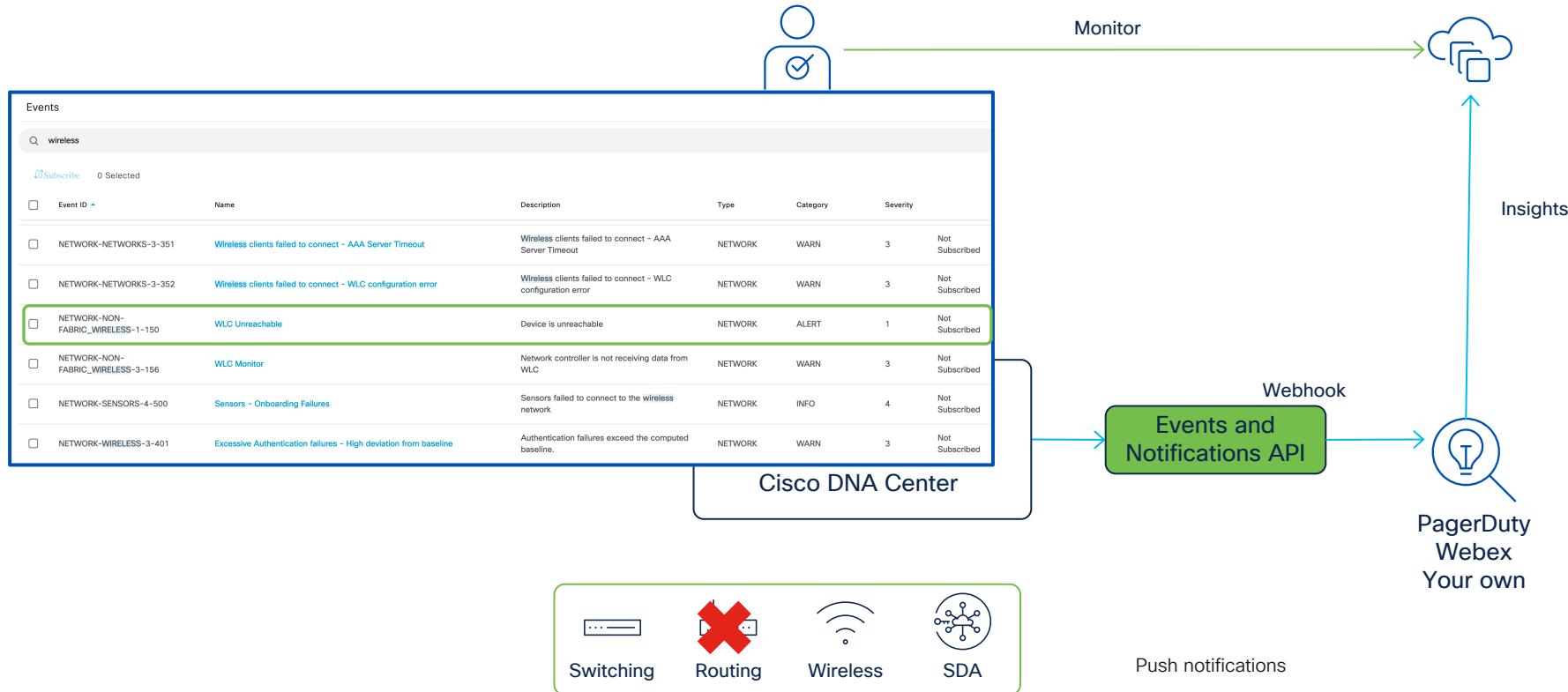
Managing your campus

Issue enrichment using Intent API



Managing your campus

Event notification



Lab prerequisites

1. Git clone the collection
<https://github.com/hemorale/CLAPJ24.git>
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

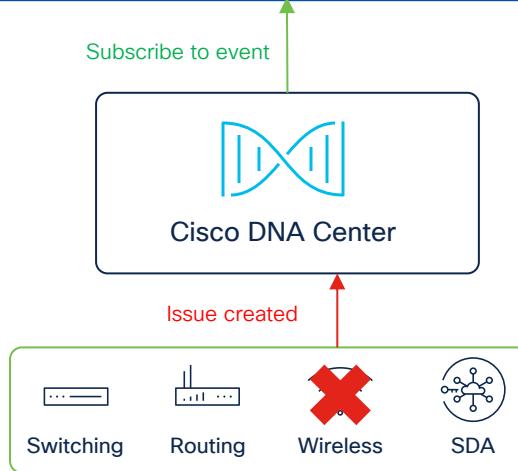
Webhook notifications workflow

Better using event notifications

Event subscription



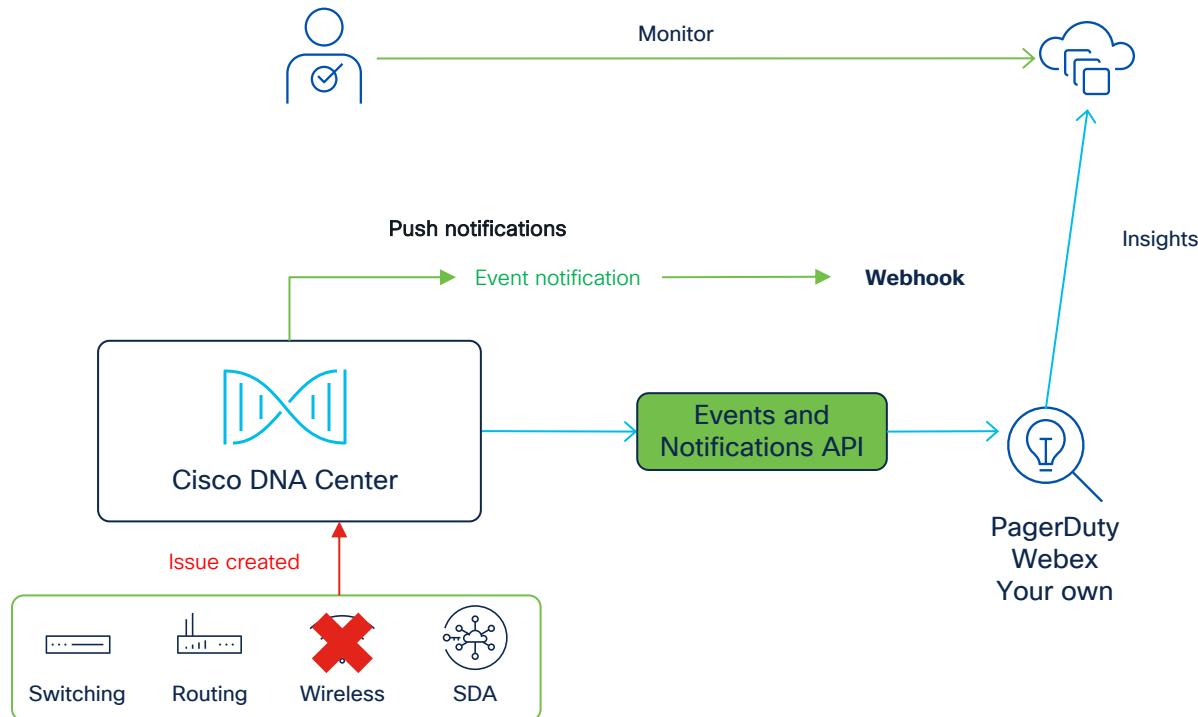
<input type="checkbox"/> NETWORK-NON-FABRIC_WIRELESS-1-150	WLC Unreachable	Device is unreachable	NETWORK	ALERT	1	Not Subscribed
--	-----------------	-----------------------	---------	-------	---	----------------



Better using event notifications

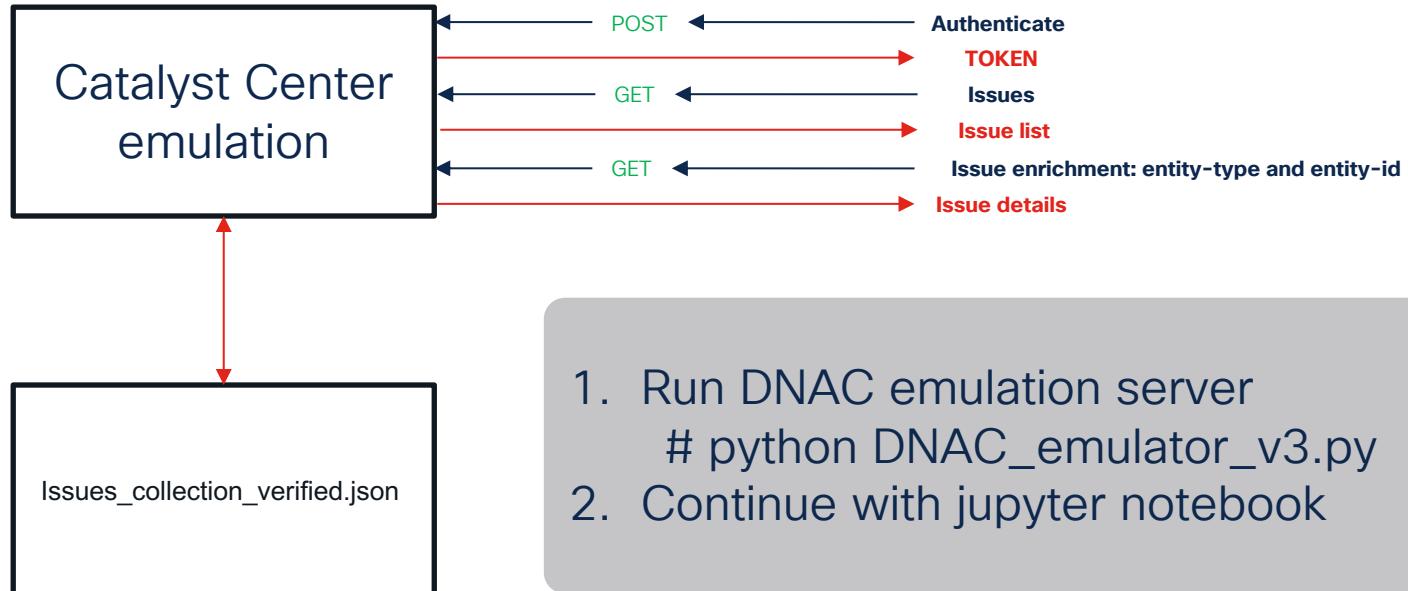
Event subscription

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



1. Run DNAC emulation server
 # python DNAC_emulator_v3.py
2. Continue with jupyter notebook

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)
```

Events_collection_verified.json

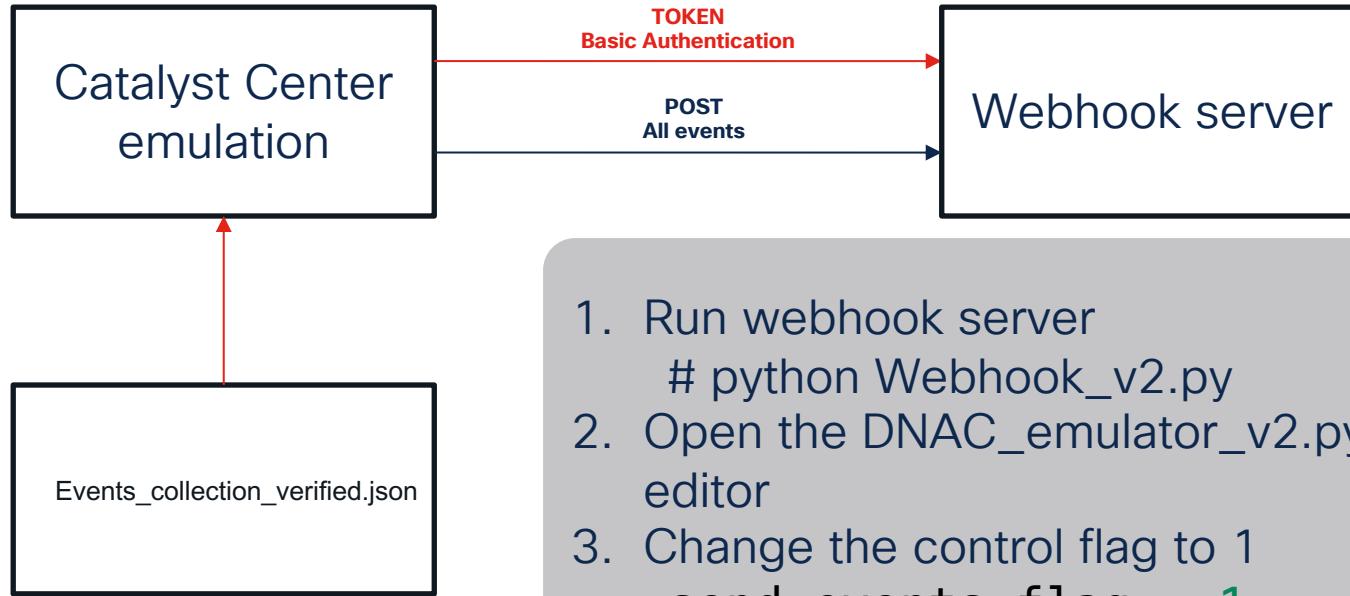
```
# Define valid credentials for Basic Authentication
users = {
    "admin": "CiscoLive100%"
}

@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
```

Lab 3

Creating a webhook server

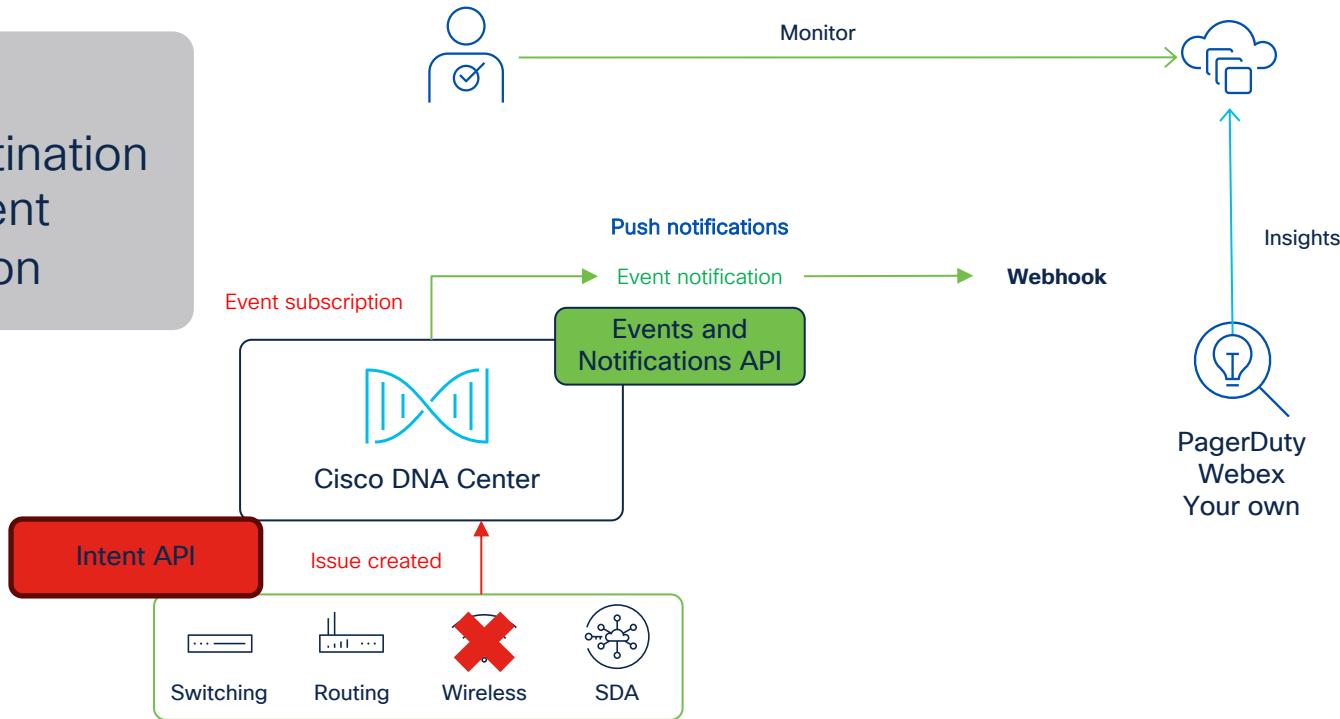


1. Run webhook server
 `# python Webhook_v2.py`
2. Open the DNAC_emulator_v2.py in an editor
3. Change the control flag to 1
 `send_events_flag = 1`

Setting up a webhook destination on Catalyst Center DEMO

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. On the left is a navigation sidebar with icons and labels: Design, Policy, Provision, Assurance, Workflows, Tools, Platform, Activities, Reports, System, and Explore. The 'System' icon is highlighted with a blue box. To its right is a main content area with a dark header bar. Below the header, the title 'External Services' is displayed. Under this title are several sections: 'Umbrella' (with a sub-note 'Register Umbrella with your Cisco DNA Center'), 'Authentication Tokens' (with a sub-note 'Manages connection between Cisco DNA Center and external authentication servers'), 'vManage' (with a sub-note 'Connect Cisco DNA Center to vManage'), 'Cisco AI Analytics' (with a sub-note 'Set up the Cisco artificial intelligence analytics engine'), 'Destinations' (which is highlighted with a blue box), and 'Machine Reasoning Engine' (with a sub-note 'Update the Machine Reasoning Engine to enable automated analysis'). A large blue bracket on the right side of the image encloses the 'External Services' section and the 'Destinations' section.

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 YWRtaW46Q2lzY29M

[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

The screenshot illustrates the Cisco DNA Center interface for subscribing to events. On the left, the navigation menu includes Design, Policy, Provision, Assurance, Workflows, Tools, Platform (selected), Activities, Reports, System, and Explore. The 'Developer Toolkit' section is also visible.

In the center, the 'Events' tab is selected under the 'Cisco DNA Center' heading. A search bar shows 'wlc'. Below it, a list of events includes:

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

A 'Subscribe' button is present next to the search bar. A modal window titled 'WLC Unreachable' provides detailed event information and subscription configuration.

Event Details:

- Description: Device is unreachable
- Event Id: NETWORK-NON-FABRIC_WIRELESS-1-150
- Namespace: ASSURANCE
- Sub Domain: Non-Fabric Wireless
- Category: ALERT
- Cisco DNA Event link: dna/assurance/issue

Tenant Aware: Yes

TAGS: ASSURANCE, wlc_unreachable

MODEL SCHEMA:

REST Schema:

```
1+ {  
2   "Type": "$eventSource$",  
3   "Assurance Issue Priority": "$priority$"  
4   "Assurance Issue Details": "This network device $nwDeviceName$ is unreachable from Cisco DNA Center. The device role is $fabricOrDevice$ and the device ID is $device$."  
5   "Device": "SeventUniqeIds$"  
6   "Assurance Issue Category": "$category$"  
7   "Assurance Issue Name": "Network Device $managementIpAddress$ Is Unreachable From Controller",  
8   "Assurance Issue Status": "$status$"  
9 }
```

Active Subscriptions:

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

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

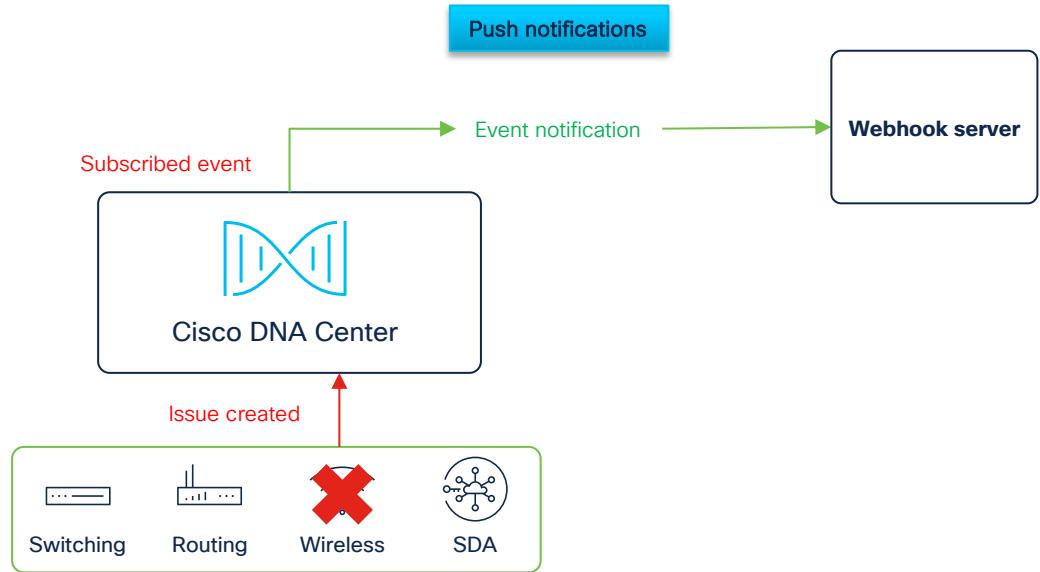
Domain: Connectivity

Sub Domain:

Details:

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

Cancel Publish



Simple flask app running locally

```
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

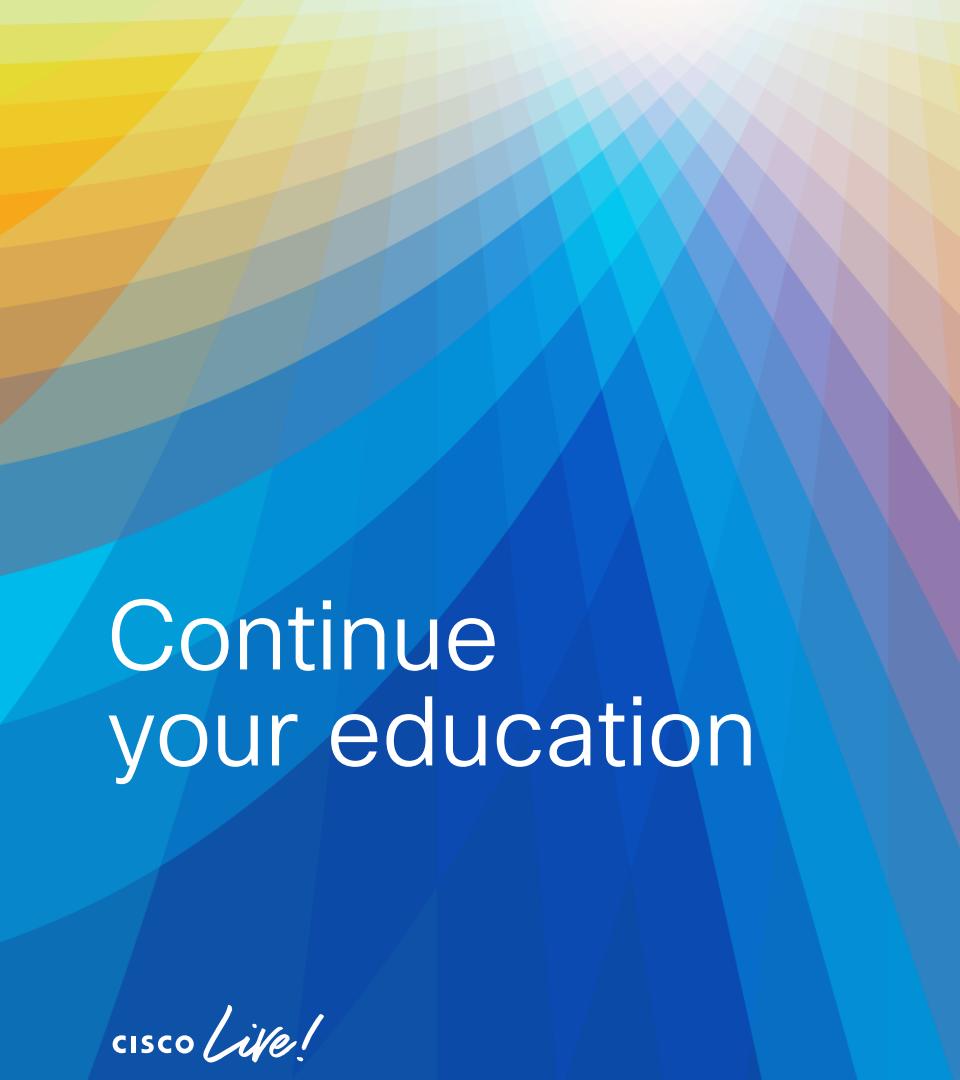
Result

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

Session Surveys

We would love to know your feedback on this session!

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



Continue your education

CISCO *Live!*

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



The bridge to possible

Thank you



#CiscoLiveAPJC



cisco *Live!*

Let's go

#CiscoLiveAPJC

Building a webhook server for monitoring Cisco Catalyst Center operations using Intent APIs and Event Notifications

Hector Morales, Sr. Global TSA, Global Partner Engineering

DEVWKS-2452



#CiscoLiveAPJC



#CiscoLiveAPJC

DEVWKS-2452

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

