



# Automate Network Operation Tasks

with NetDevOps and Source of Truth

Patrick Mosimann - Technical Solutions Architect

Luca Gubler - Systems Engineer, onway (Schweiz) ag

BRKOPS-2357

# Agenda

## Setting the Stage

- 1.1. Introduction
- 1.2. Partner Use Case
- 1.3. Scenario Overview

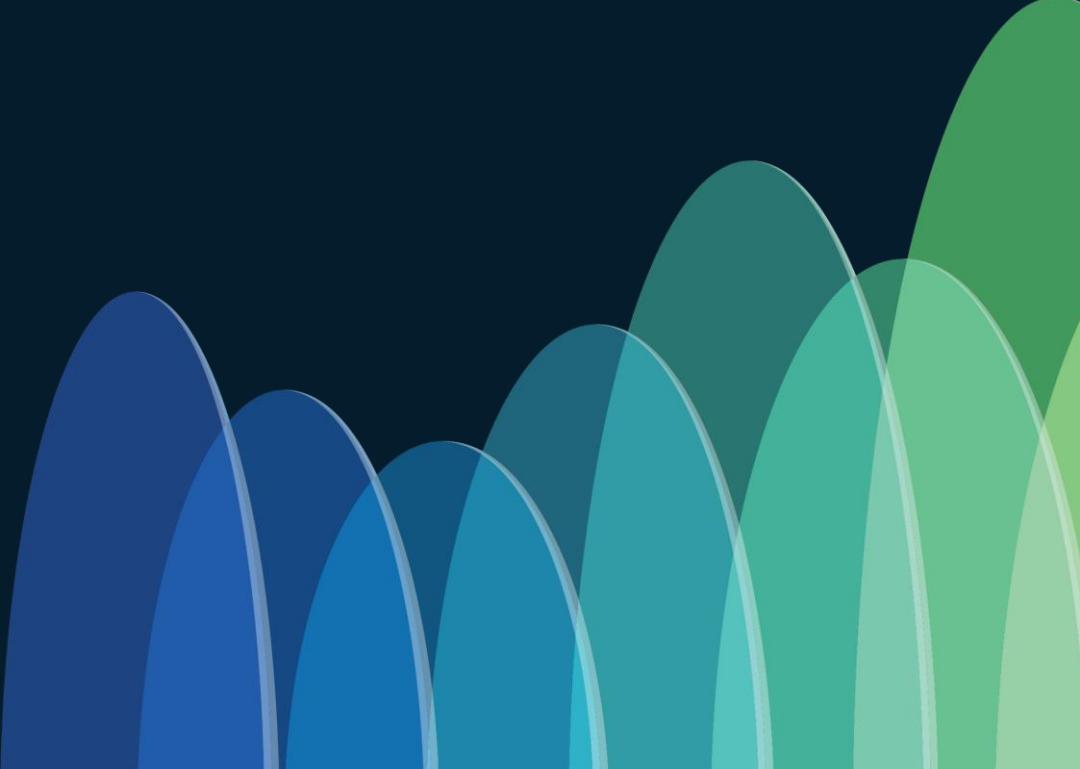
## Technology Deep Dive

- 2.1. Source of Truth
- 2.2. Automation Tools
- 2.3. Testing Framework

## Tying It All Together

- 3.1. Orchestration
- 3.2. Key Takeaways
- 3.3. Call Out

# Setting the Stage



# About Patrick...



18+ years in Networking  
3 years as customer  
3 years as a partner  
3 years as a competitor  
9+ years at Cisco



# Session Introduction

cisco  
The bridge to possible

Introduction to  
Infrastructure as Code  
for Cisco Catalyst Center with Terraform  
Patrick Mosimann, Technical Solutions Architect  
Tobias Spuhler, Technical Solutions Architect

CISCO Live!

CISCO Live!

cisco  
The bridge to possible

Infrastructure as Code  
for Cisco Catalyst Center and Catalyst SD-WAN  
Manager with Terraform

cisco

Automate Network  
Operation Tasks  
with NetDevOps and Source of Truth

Mosimann, Technical Solutions Architect  
Spuhler, Systems Engineer, onway (schweiz) ag

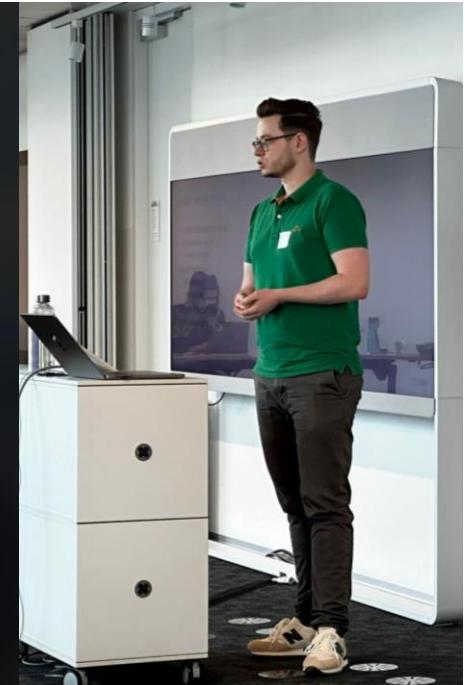


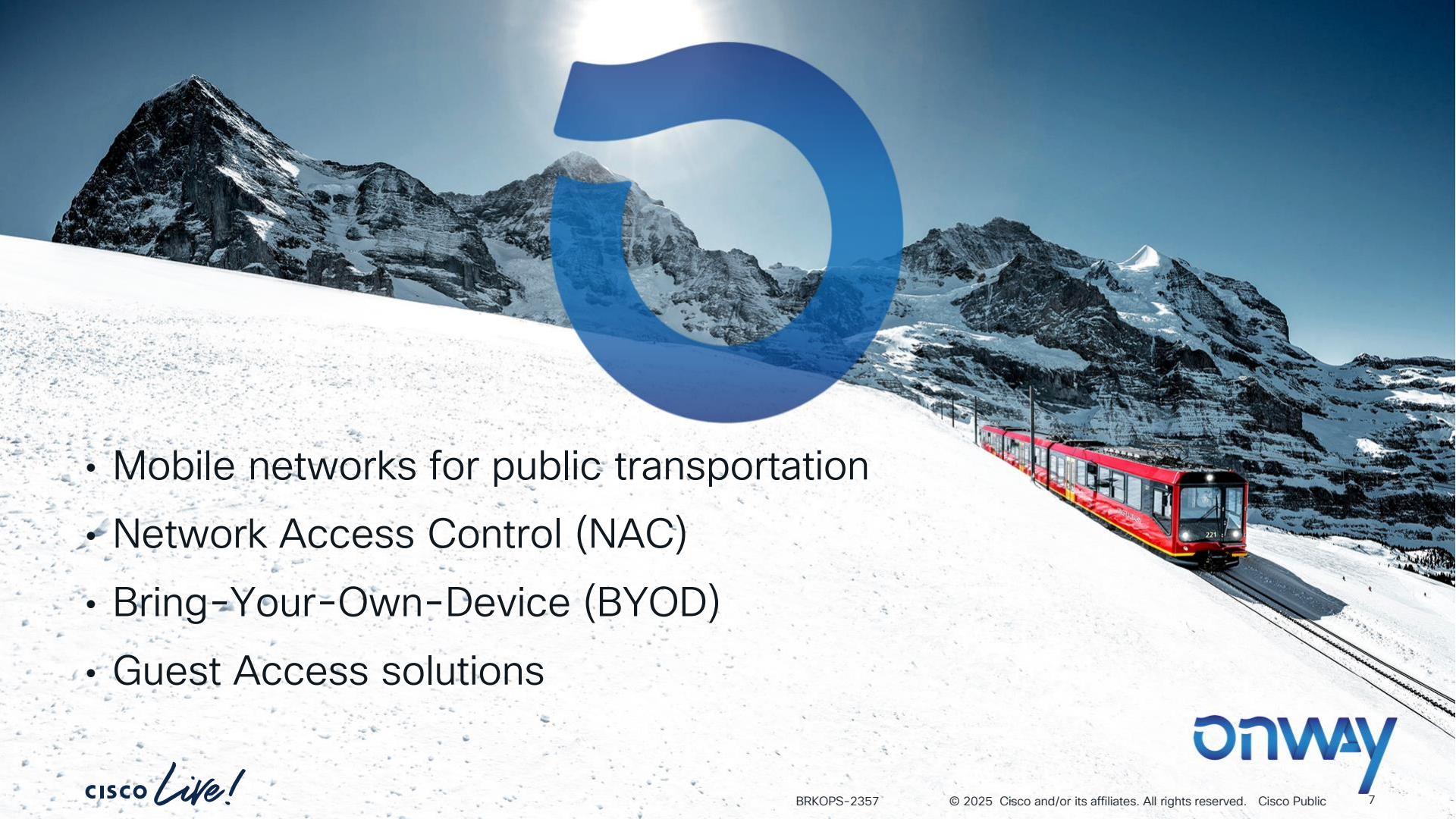
# About Luca...

6+ years in Networking  
6 years as a partner



# 2023::4





- Mobile networks for public transportation
- Network Access Control (NAC)
- Bring-Your-Own-Device (BYOD)
- Guest Access solutions

CISCO Live!

# DevNet Academy



## Theory

Access a wide range of theory materials, including videos and insightful text content, specifically tailored to help you succeed in the Cisco Certified DevNet Expert exam.



## Exercises & Workbook

Engage in practical exercises designed to provide hands-on experience, enhancing your skills and preparing you for real-world challenges.



## Quizzes

Reinforce key concepts and test your understanding with quizzes after each lesson. These quizzes are designed to ensure that you understand the essential elements necessary to pass the exam.



## Coaching and Mentoring

Benefit from personalized coaching and mentoring throughout your certification journey. Luca is here to guide, support, and motivate you every step of the way, helping you navigate through the exam process.



## Community

Join an exclusive community of students to share insights, ask questions, and engage in discussions. This collaborative environment offers a unique opportunity to learn from others' experiences while contributing your own.



## Lifetime Access and Updates

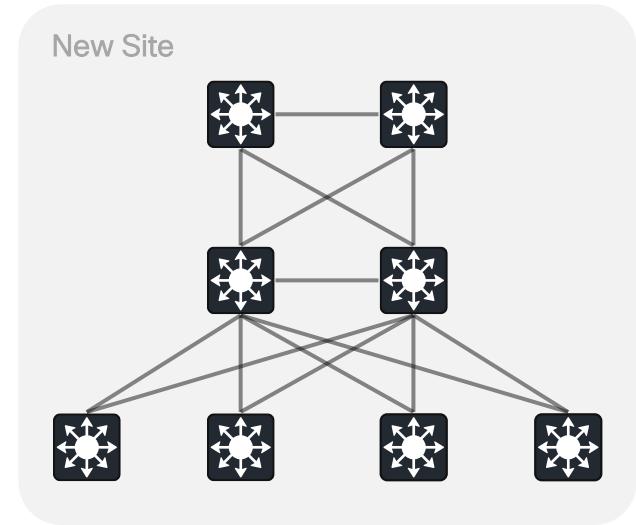
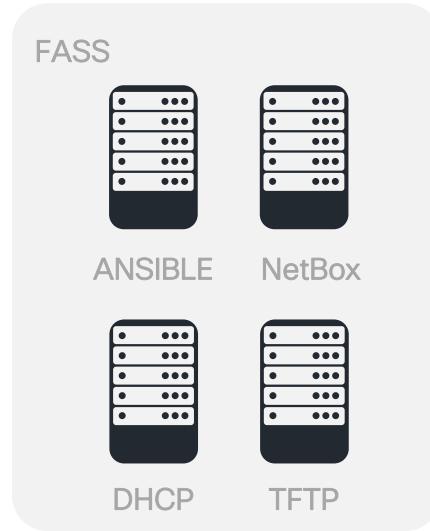
Gain lifetime access to all course materials and receive updates whenever Cisco releases a new blueprint version. Stay current with the latest content and resources to ensure your skills remain sharp and relevant.



- DevNet Expert e-learning
- Theory, videos, exercises, quizzes, and workbook
- Coaching and mentoring
- Sample course with Python Click and Flask-RESTX
- <https://devnet-academy.com>

# Overview

## Fully Automated Staging Setup (FASS)



# Scope

## Fully Automated Staging Setup (FASS)



### Scale

80 routers and switches



### Speed

Staged in less than a day



### Efficiency

Only two engineers on site



### Approach

Data-driven automation with a source of truth

# Components

## Fully Automated Staging Setup (FASS)



NetBox

Single Source of Truth (SSoT)

Centralized inventory



DHCP Server

IP addressing based on MAC address

Pass TFTP server and boot file to device



TFTP Server

Provides Day0 config to each device

Configures mgmt interface and staging user



Ansible

Use NetBox inventory plugin

Push config to each device

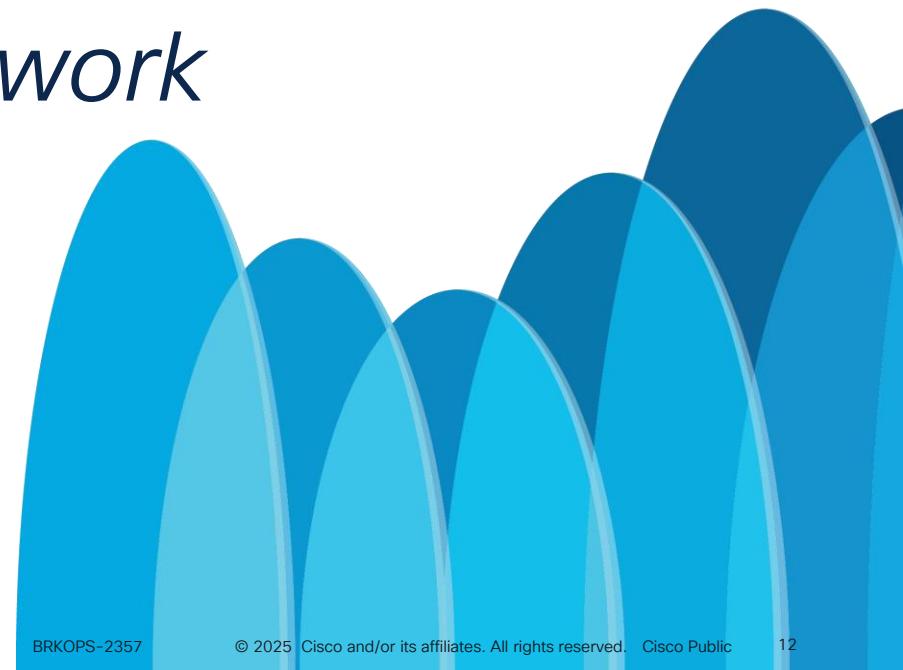


Observium

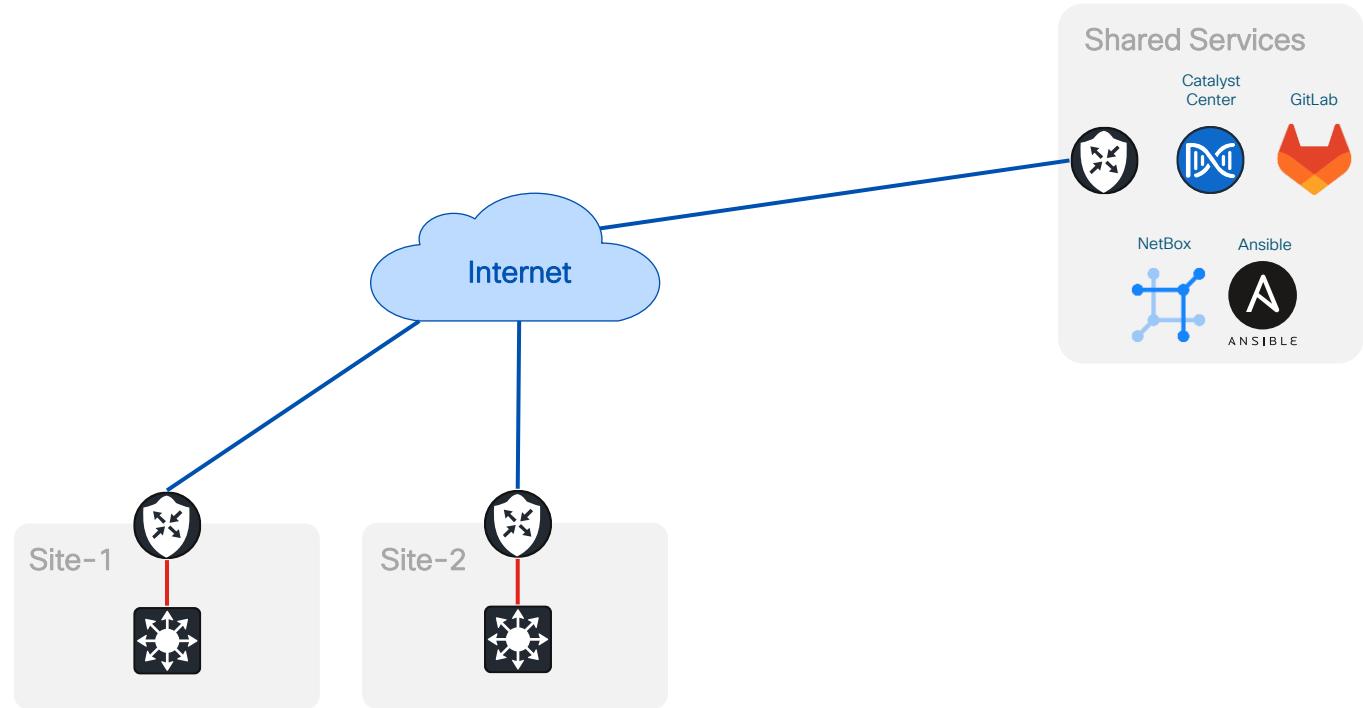
Continuous monitoring

Long-term metrics & Proactive insights

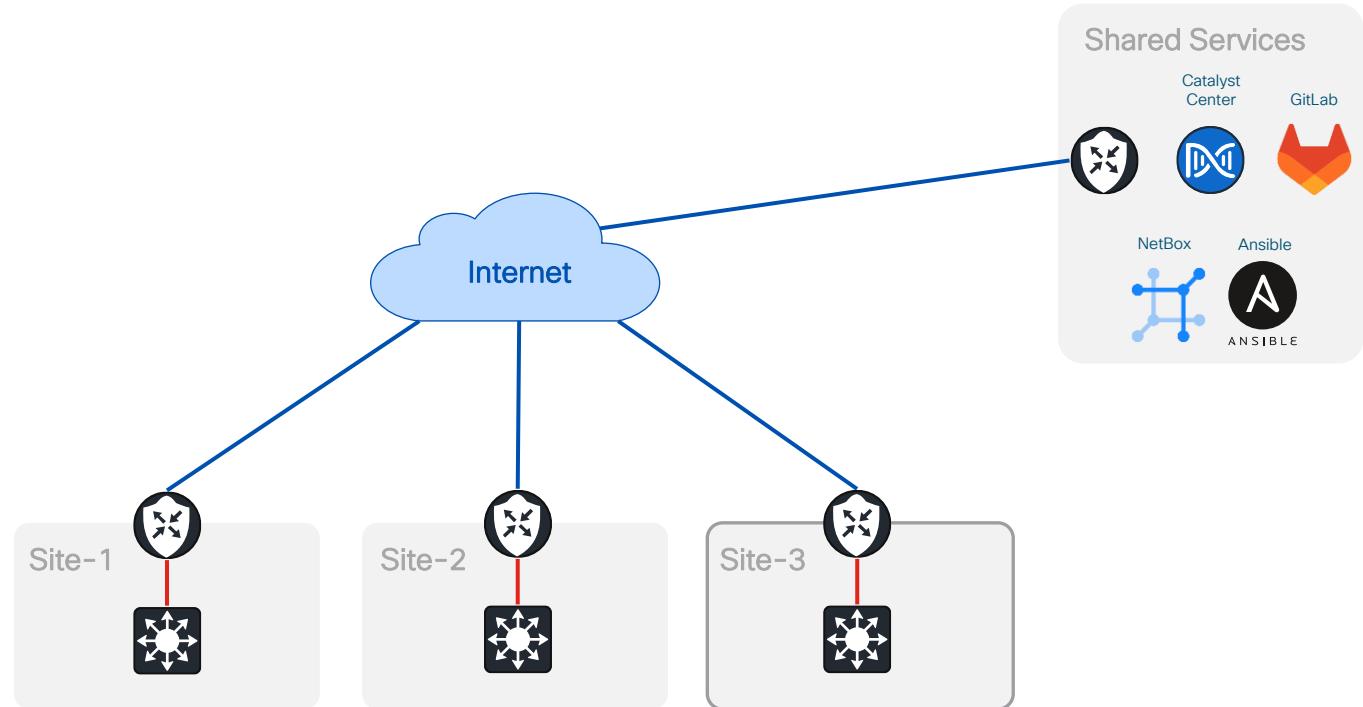
*Automate boring, repetitive tasks  
with data-driven network  
automation!*



# As Is - High Level Layout of the Scenario



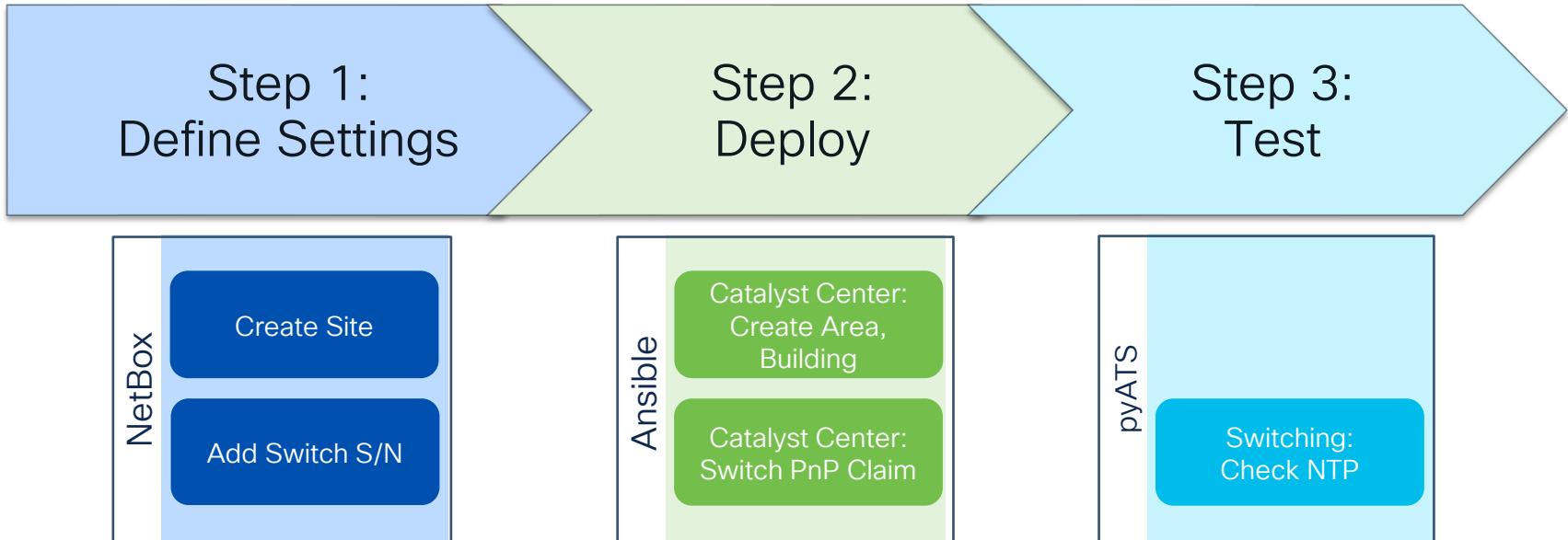
# To Be - High Level Layout of the Scenario



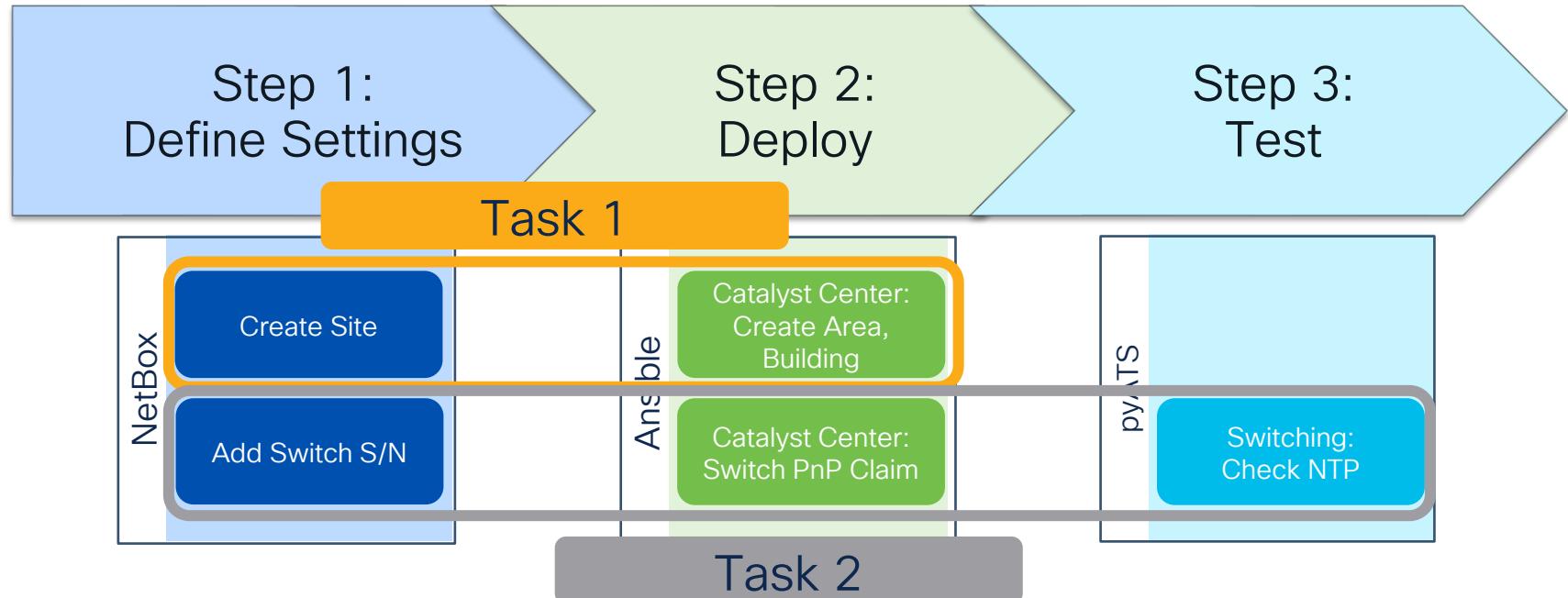
# Business Intent of the Scenario



# Workflow of the Scenario



# Workflow Tasks of the Scenario



The screenshot shows the Netbox web interface on a laptop screen. The left sidebar contains a navigation menu with categories like Organization, Racks, Devices, Connections, Wireless, IPAM, VPN, Virtualization, Circuits, Power, Provisioning, Customization, and a CUSTOMIZATION section with sub-options: Custom Fields, Custom Field Choices, Custom Links, Export Templates, Saved Filters, and Tags. The main content area is titled "Create Site and Locations" with a subtitle "Script to create a new site and associated floors as locations." It features tabs for Script, Source, and Jobs (19). The "Script Data" section includes fields for Tenant\*, Region\*, Site Name\*, Address\*, Number of Floors\*, and Lowest Floor\*. Each field has a descriptive placeholder text below it. A search bar at the top and a notification bell icon are also visible.

netbox

Community

Organization

SITES

Sites + ⚡

Regions

Site Groups

Locations

TENANCY

Tenants

Tenant Groups

CONTACTS

Contacts

Contact Groups

Contact Roles

Contact Assignments

Racks

Devices

Connections

Wireless

IPAM

VPN

Search... dcim.site

Site-3

Created 2025-01-21 08:41 · Updated 2025-01-21 08:44

Bookmark Subscribe Clone Edit

Site Contacts Journal Changelog

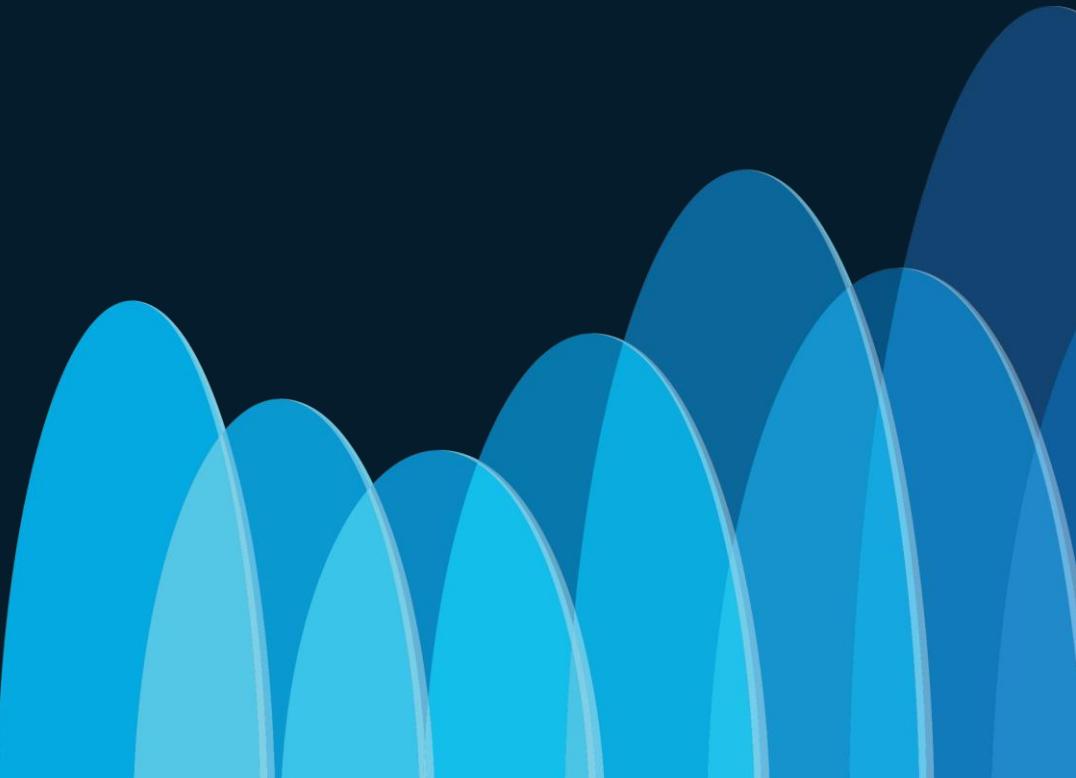
Site	
Region	EMEA
Group	—
Status	Active
Tenant	BRKOPS-2357
Facility	—
Description	—
Time Zone	—
Physical Address	Richtistrasse 7, 8302 Wallisellen, Switzerland
Shipping Address	—
GPS Coordinates	47.409816, 8.590415

Related Objects

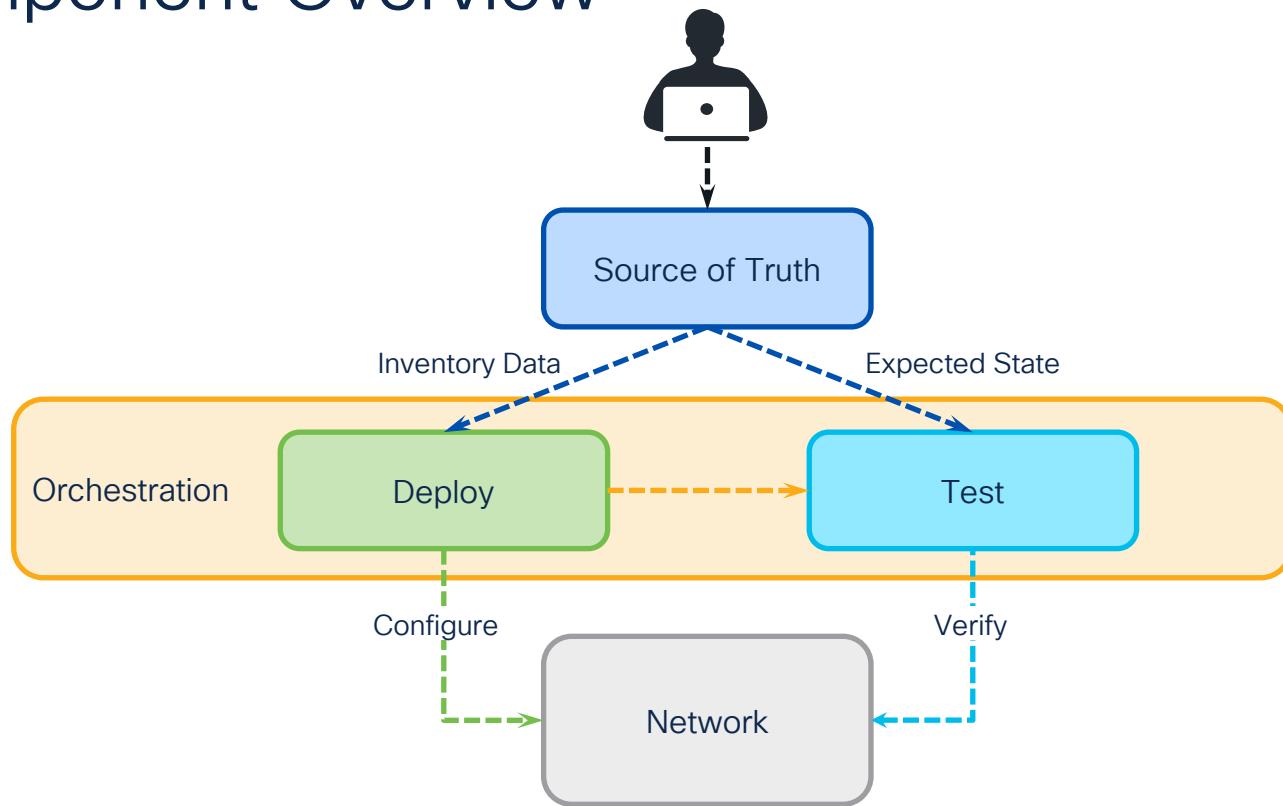
- ASN
- Circuits
- Clusters
- Devices
- Locations
- Power Panels
- Prefixes
- Racks
- Virtual Machines
- VLANs
- VLAN Groups

Images + Add

# Technology Deep Dive

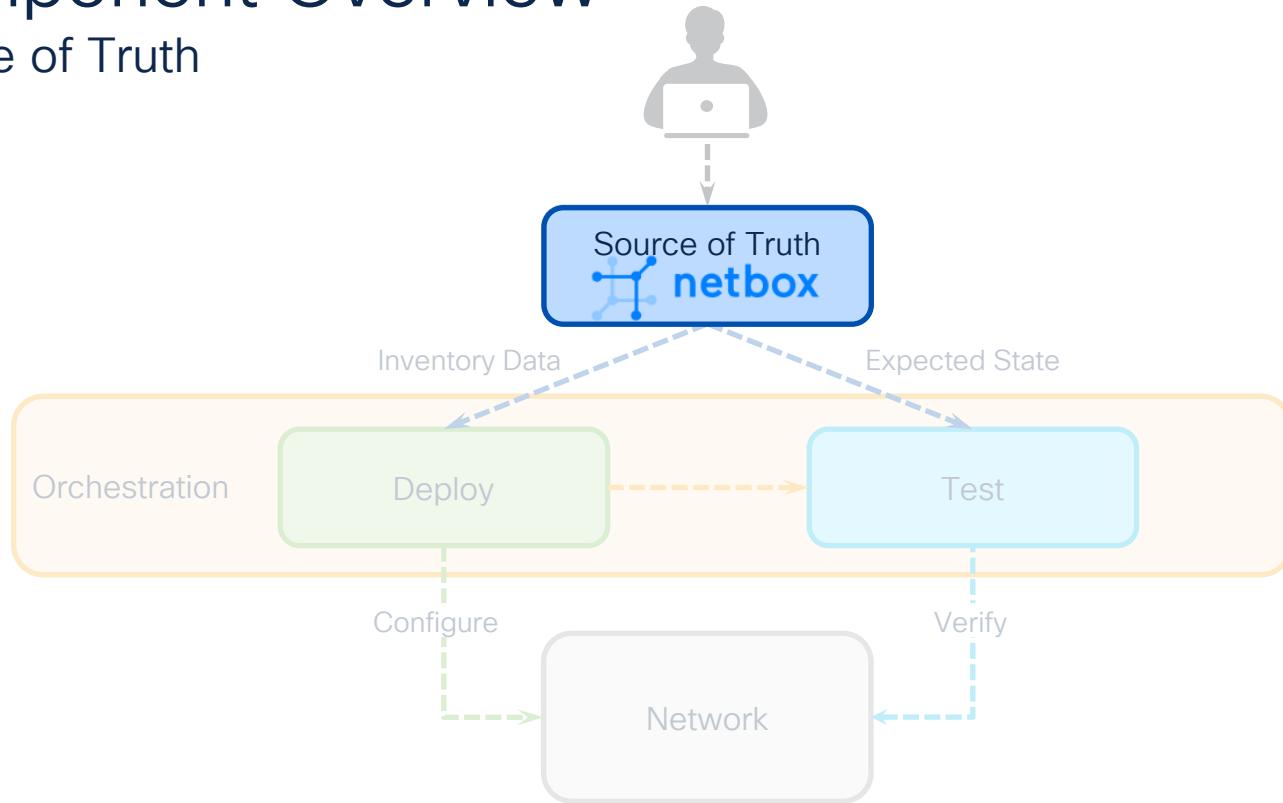


# Component Overview



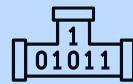
# Component Overview

## Source of Truth



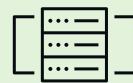
# What Is an IPAM / DCIM?

Source of Truth



IP Address Management  
(IPAM)

IPs, Subnets, VLANs, VRFs,  
etc.



Data Center Infrastructure  
Management (DCIM)

Devices, racks, locations, etc.

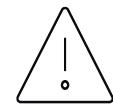


Source of Truth (SoT)

IPAM + DCIM

# Benefits of a Single Source of Truth

## Source of Truth



Reduces human error



Easy administration



Auditing



Eliminates inconsistencies



Ensures data is accurate



No more spreadsheets!

# Config Example

## Source of Truth

Sites / EMEA

### Site-1

Created 2025-01-17 14:16 · Updated 2025-01-17 14:20

Site	Contacts	Journal	Changelog
Site			
Region	EMEA		
Group	—		
Status	Active		
Tenant	BRKOPS-2357		
Facility	—		
Description	—		
Time Zone	—		
Physical Address	Richtistrasse 7, 8302 Wallisellen, Switzerland		
Shipping Address	—		
GPS Coordinates	47.409816, 8.590415		

A playbook.yml

```
1 config:
2   - site:
3     - building:
4       name: "{{ name }}"
5       parent_name: "Global/{{ tenant_name }}/{{ region_name }}"
6       address: "{{ physical_address }}"
7       latitude: "{{ latitude }}"
8       longitude: "{{ longitude }}"
9       country: "{{ physical_address.split(',')[-1].strip() }}"
10      type: building
```

# Why NetBox?

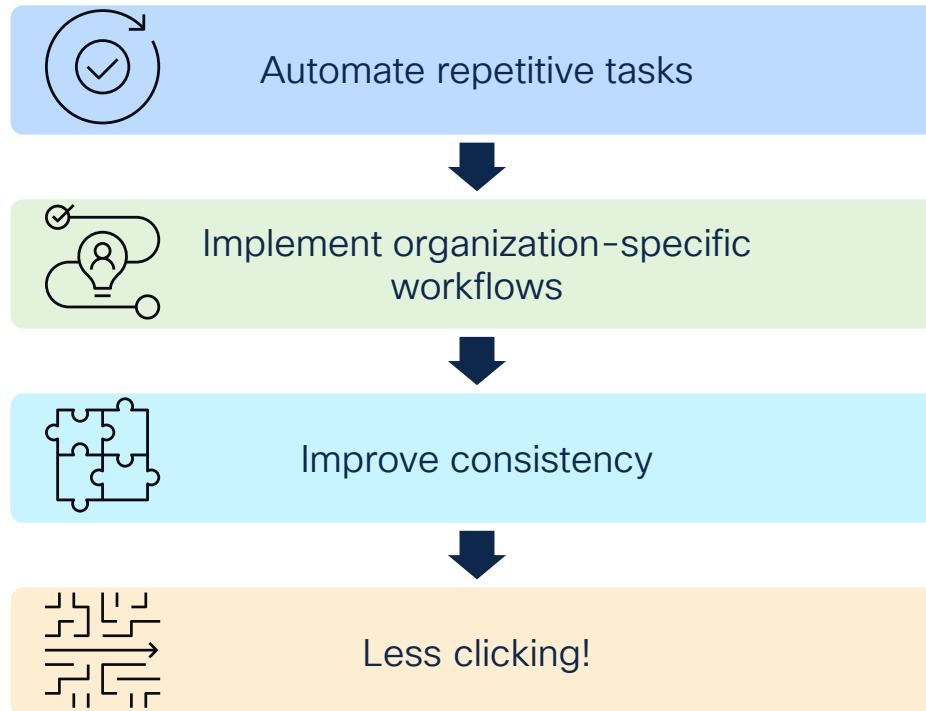
- **Open Source & Popular:**  
Free to use with options for cloud and enterprise plans.
- **API-Driven Automation:**  
Ideal for automating processes with extensive API support.
- **Seamless Ansible Integration:**  
Supports plugins like inventory, or lookup for efficient workflows.





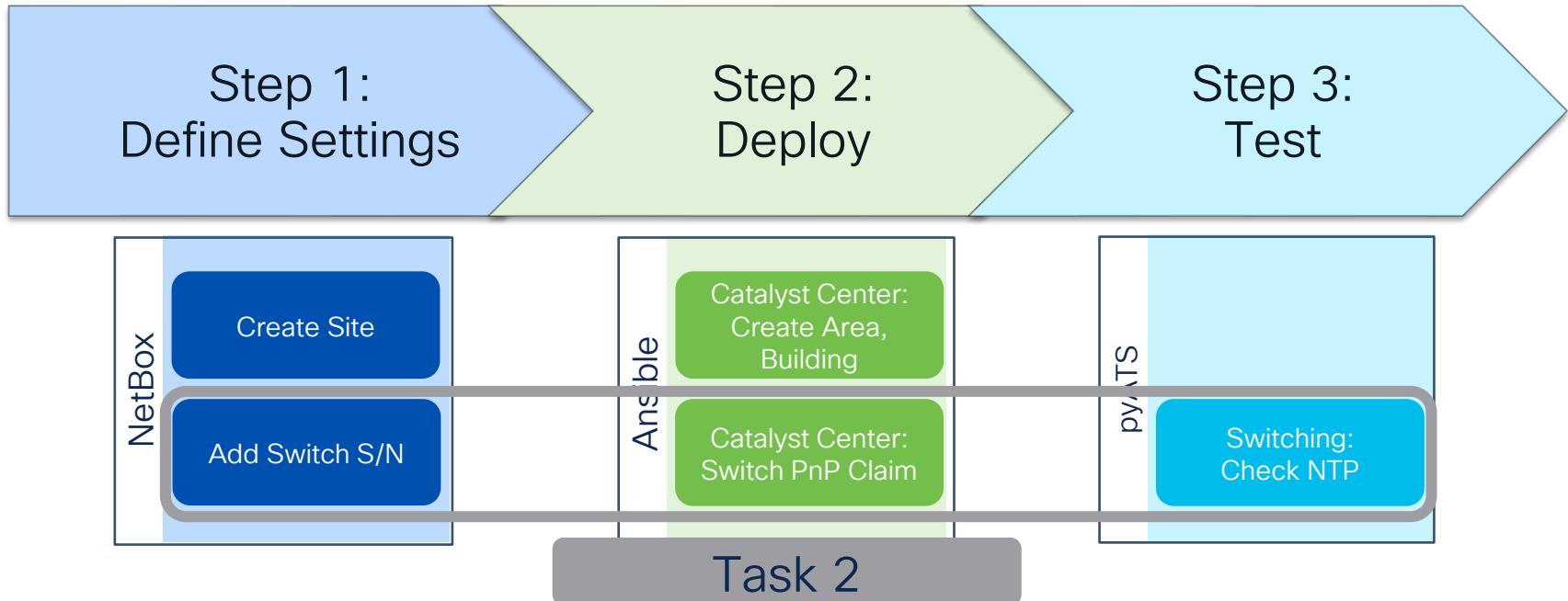
# NetBox Custom Scripts

Source of Truth



# Remember the Workflow Tasks?

Source of Truth



**netbox**  
Community

Organization  
Racks  
Devices  
Connections  
Wireless  
IPAM  
VPN  
Virtualization  
Circuits  
Power  
Provisioning  
Customization  
Operations  
Admin

Search... 💡 🔔

Scripts / 02\_switches

Add Switches to Site

Script to add one or more switches to an existing site.

Script Source Jobs 40

Script Data

Tenant\* -----  
Select the tenant for the new site.

Site\* -----  
Select the site to which the switches will be added.

Device Type\* -----  
Select the type of the switches to add.

Device Role\* -----  
Select the role of the devices to add.

Device Uplink\* -----  
Select uplink interface.

Serial Numbers\* -----  
Enter one or more serial numbers, separated by commas.

netbox

Community

Organization

Racks

Devices

DEVICES

Devices

Modules

Device Roles

Platforms

Virtual Chassis

Virtual Device Contexts

DEVICE TYPES

Device Types

Module Types

Manufacturers

DEVICE COMPONENTS

Interfaces

Front Ports

Rear Ports

Console Ports

Console Server Ports

Power Ports

Search...

9:29 Mosimann 3f3a9e375765

2025-01-21 pamosima Patrick Created Interface GigabitEthernet1/0/7 7ddc42de-bf5a-425d-89b3f3a9e375765

9:29 Mosimann 3f3a9e375765

2025-01-21 pamosima Patrick Created Interface GigabitEthernet1/0/6 7ddc42de-bf5a-425d-89b3f3a9e375765

9:29 Mosimann 3f3a9e375765

2025-01-21 pamosima Patrick Created Interface GigabitEthernet1/0/5 7ddc42de-bf5a-425d-89b3f3a9e375765

9:29 Mosimann 3f3a9e375765

2025-01-21 pamosima Patrick Created Interface GigabitEthernet1/0/4 7ddc42de-bf5a-425d-89b3f3a9e375765

9:29 Mosimann 3f3a9e375765

2025-01-21 pamosima Patrick Created Interface GigabitEthernet1/0/3 7ddc42de-bf5a-425d-89b3f3a9e375765

9:29 Mosimann 3f3a9e375765

2025-01-21 pamosima Patrick Created Interface GigabitEthernet1/0/2 7ddc42de-bf5a-425d-89b3f3a9e375765

9:29 Mosimann 3f3a9e375765

2025-01-21 pamosima Patrick Created Interface GigabitEthernet1/0/1 7ddc42de-bf5a-425d-89b3f3a9e375765

9:29 Mosimann 3f3a9e375765

2025-01-21 pamosima Patrick Created Interface GigabitEthernet0/0 7ddc42de-bf5a-425d-89b3f3a9e375765

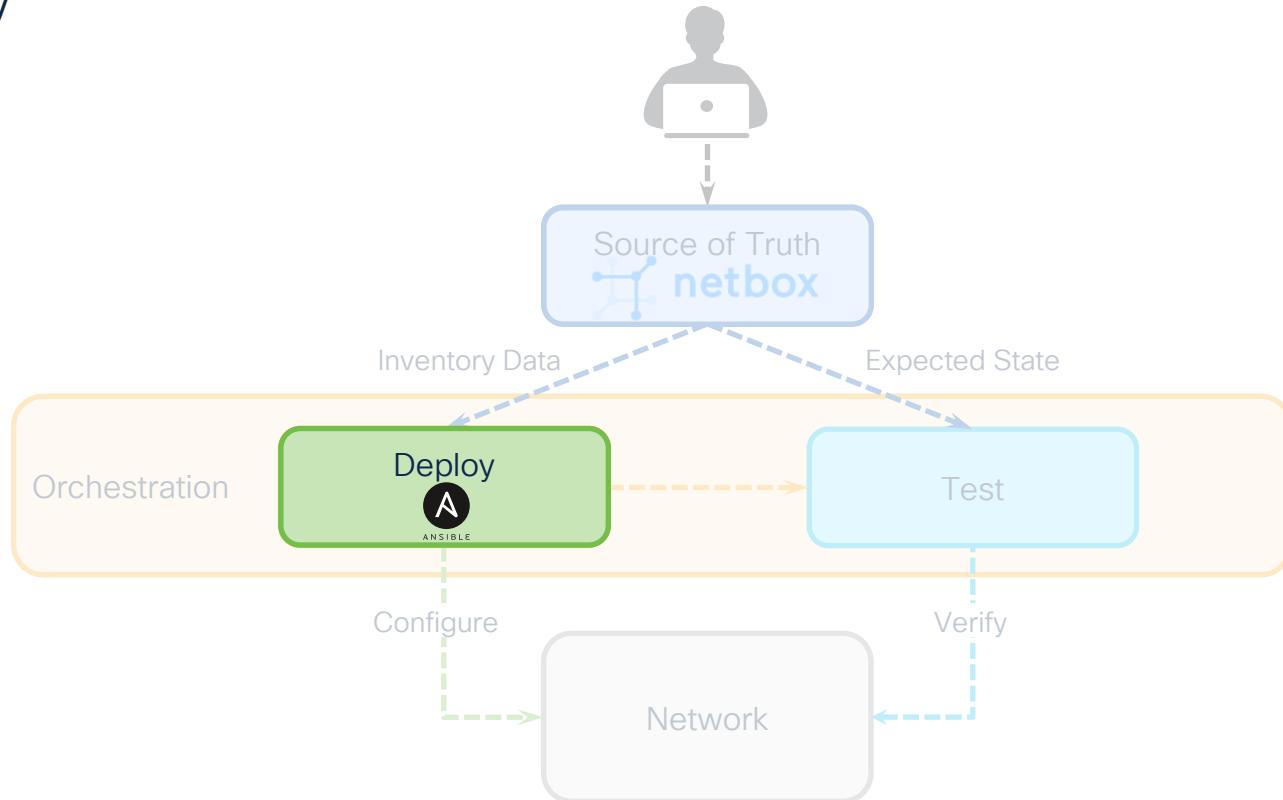
9:29 Mosimann 3f3a9e375765

Showing 1-13 of 13

Change log retention: 90 days

# Component Overview

## Deploy



# Ansible Building Blocks (1/2)

## Deploy

### Playbook

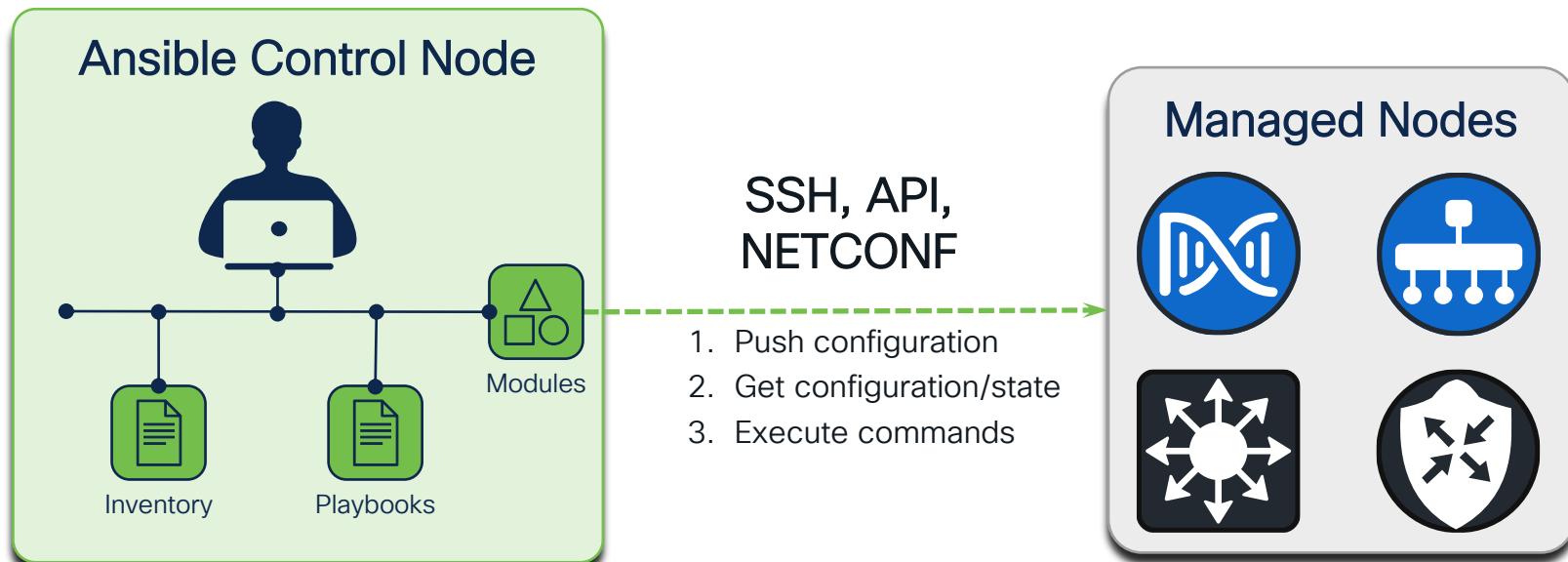
- Repeatable and reusable task
- Simple config management

### Inventory

- Defines hosts and groups
- Stores variables

# Ansible Building Blocks (2/2)

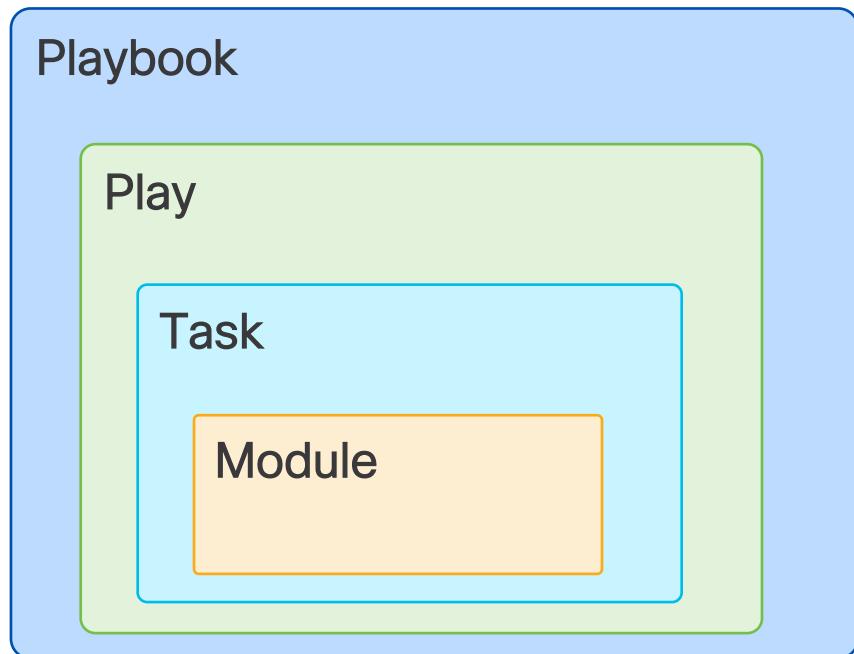
## Deploy



# Ansible Playbooks

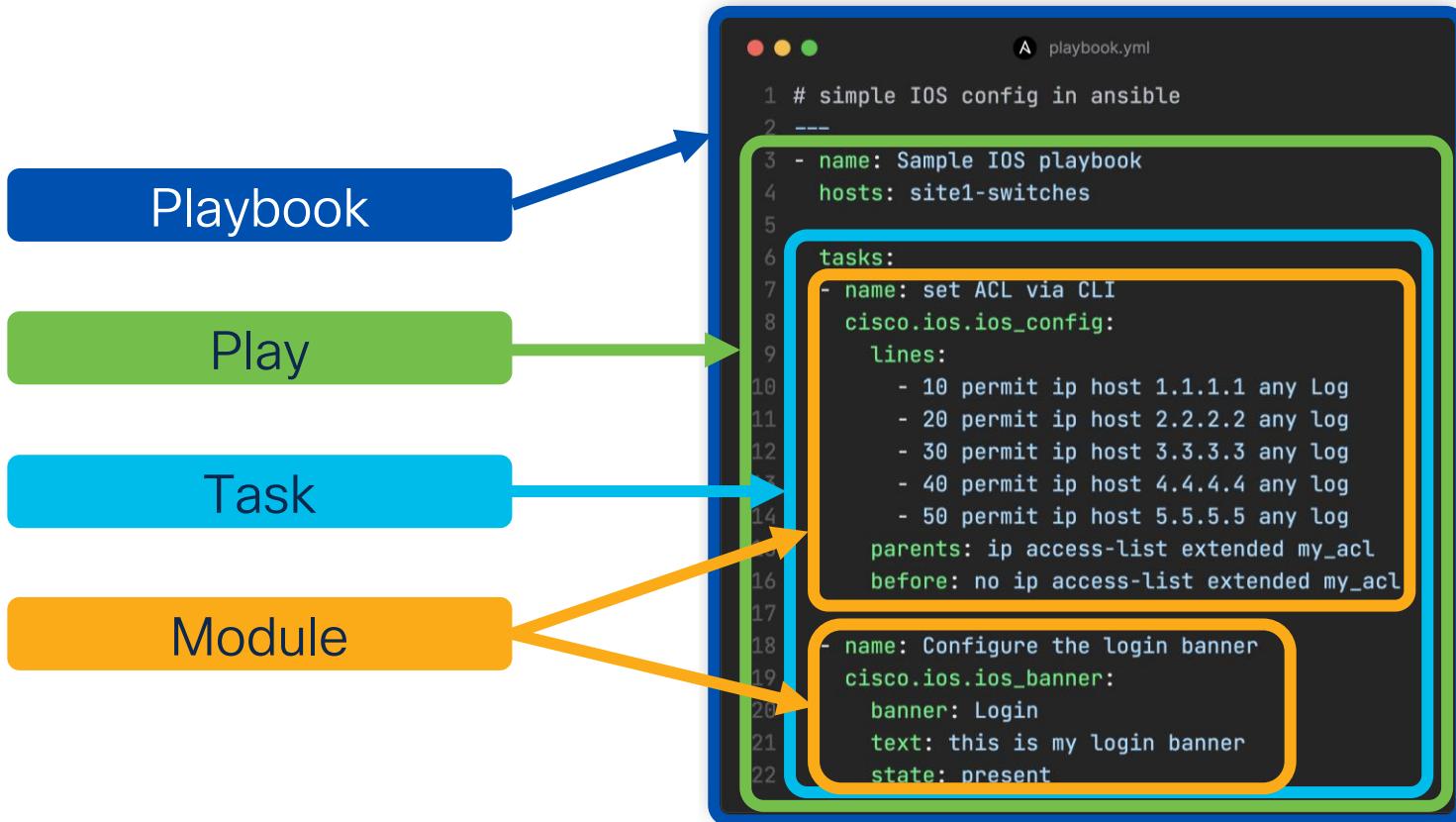
## Deploy

- Playbook
  - Repeatable standard config
- Play
  - A set of tasks
- Task
  - A single action that uses a module
- Module
  - Reusable, standalone scripts



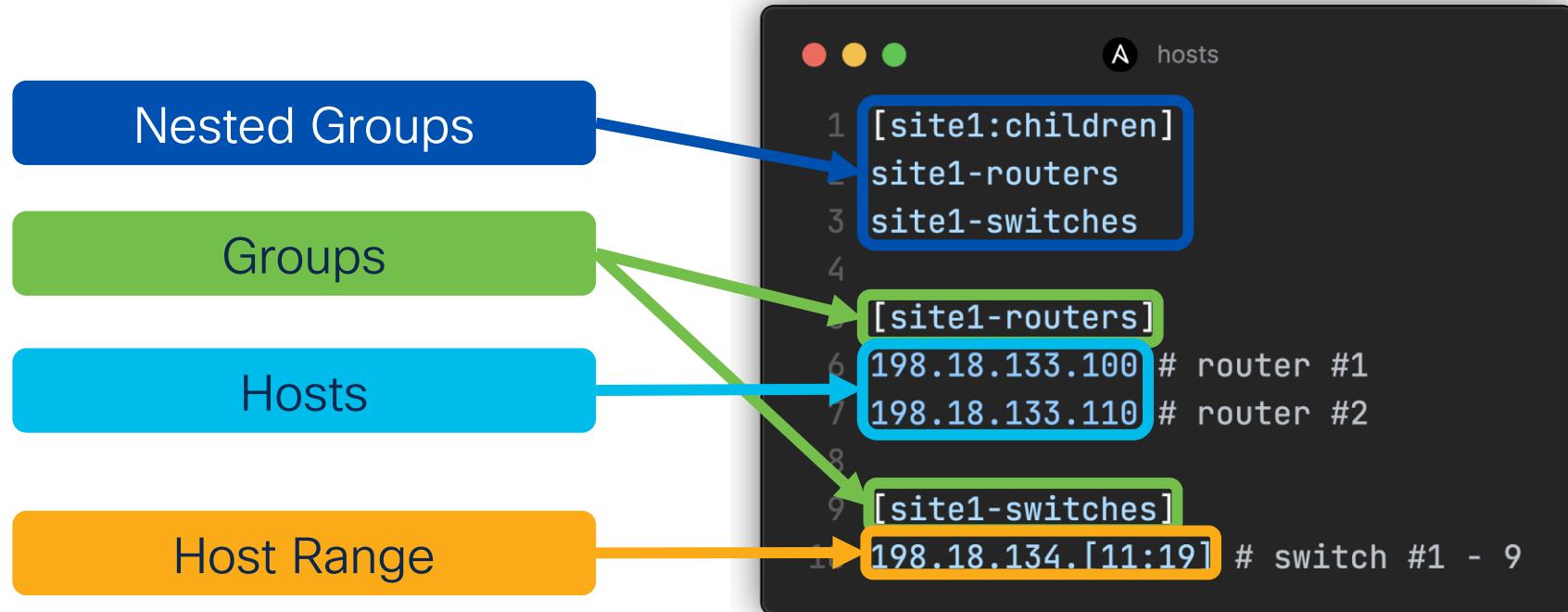
# Ansible Playbook Example

## Deploy



# Ansible Inventory

## Deploy



# Ansible Inventory Plugin

## Deploy

```
 nb_inventory.yml
1 plugin: netbox.netbox.nb_inventory
2 validate_certs: true
3 config_context: false
4 group_by:
5   - device_roles
6 query_filters:
7   - role: switches
8 device_query_filters:
9   - has_primary_ip: true
```

```
 Terminal
1 ansible-inventory --graph -i nb_inventory.yml
2
3 @all:
4   |--@ungrouped:
5   |--@device_roles_switches:
6   |   |--sw1-1
7   |   |--sw2-1
8   |   |--sw3-1
9   |   |--sw4-1
```

Inventory plugins allow users to point at [data sources](#) to compile the [inventory of hosts](#) that Ansible uses to target tasks

# Ansible Idempotency

Deploy

## Definition:

Applying the same playbook multiple times results in the same state without unintended changes.

## Benefits:

- **Efficiency:** Avoids unnecessary actions, saving time.
- **Reliability:** Reduces errors by ensuring systems remain in the desired state.

## Example:

Installs a package only if not already present, maintaining system consistency.

# Why Ansible?

- **Automation & Efficiency:** Automate repetitive tasks to save time and resources.
- **Simplicity & Ease of Use:** YAML syntax makes it accessible and easy to learn.
- **Extensive Modules:** Includes Cisco Catalyst Center and NetBox for flexibility.

# Catalyst Center Ansible Collection

## Deploy

- TAC/Engineering Supported starting with Cisco Catalyst Center version 2.3.7.6
- Identified with the module name including “workflow\_manager”
- Ansible playbooks for common operations for sites, PnP, SWIM, compliance, templates, network settings, ...

The screenshot shows the Ansible Galaxy page for the `cisco.dnac` collection. The page header indicates it is version 6.18.0, updated 22 days ago, with 66,374 downloads. The main navigation tabs are Install, Documentation, Contents, Import log, and Dependencies. Below the tabs, there is a search bar with the query "workflow" and a dropdown menu showing "Modules (21)". The listed modules include: accesspoint\_workflow\_manager, device\_configs\_backup\_workflow, device\_credential\_workflow, discovery\_workflow\_manager, events\_and\_notifications\_workflow, fabric\_sites\_zones\_workflow, inventory\_workflow\_manager, ise\_radius\_integration\_workflow, and network\_compliance\_workflow. To the right of the module list, there is descriptive text about the collection, a note about compatibility, and a compatibility matrix table.

**Ansible Collection - cisco.dnac**

**Ansible Modules for DNA Center**

The dnacenter-ansible project provides an Ansible collection for managing and automating your Cisco DNA Center environment. It consists of a set of modules and roles for performing tasks related to DNA Center.

This collection has been tested and supports Cisco DNA Center 2.3.7.6.

*Note: This collection is not compatible with versions of Ansible before v2.8.*

Other versions of this collection have support for previous Cisco DNA Center versions. The recommended versions are listed below on the [Compatibility matrix](#).

**Compatibility matrix**

The following table shows the supported versions.

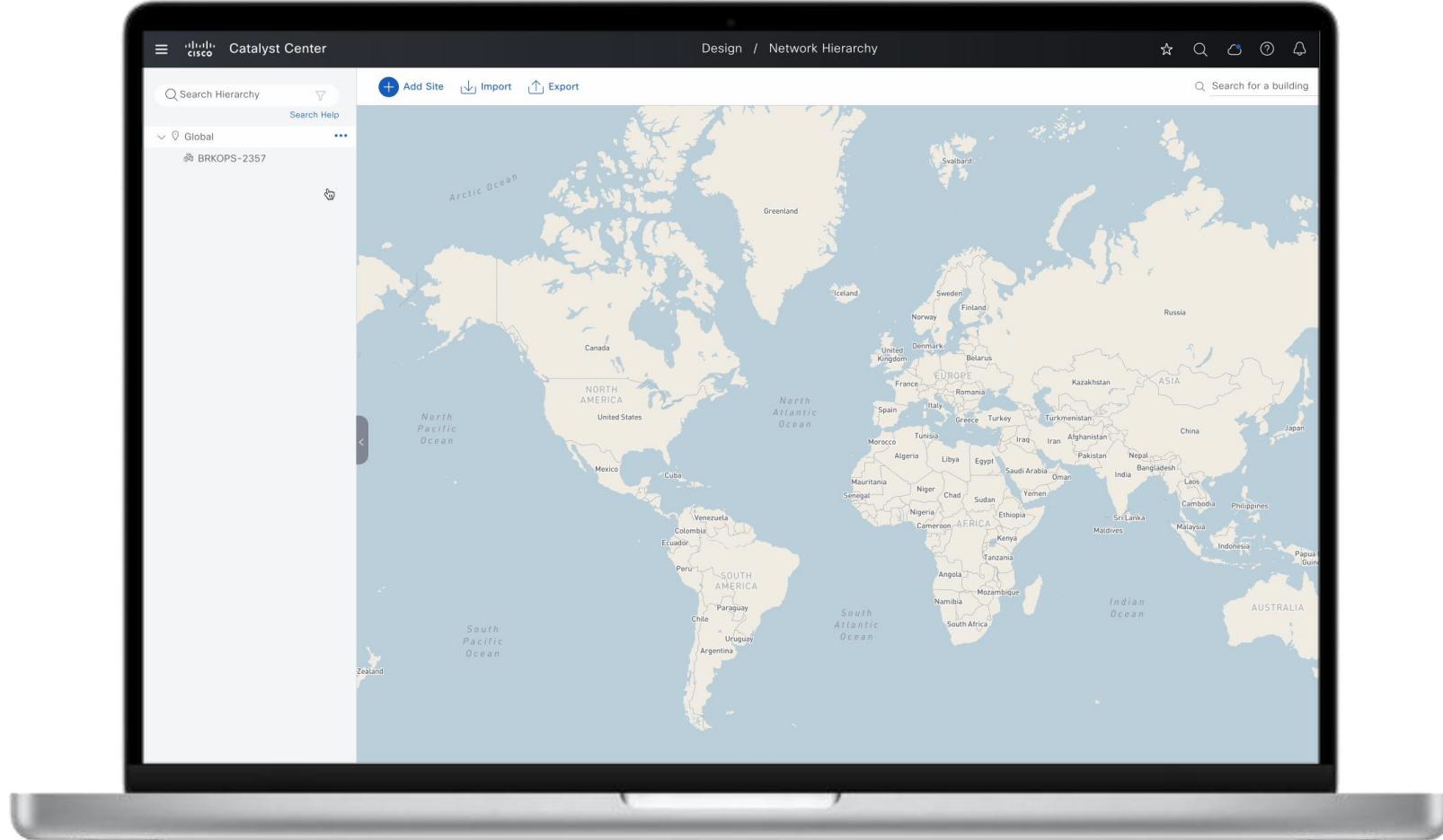
Cisco DNA Center version	Ansible "cisco.dnac" version	Python "dnacentersdk" version
2.2.2.3	3.3.1	2.3.3
2.2.3.3	6.4.0	2.4.11
2.3.3.0	6.6.4	2.5.5

The image shows a laptop screen with a dark-themed terminal window open. The terminal window has two tabs at the top: '01\_create-site-hierarchy.yml' and 'ccc\_site\_hierarchy\_vars.yml'. The 'ccc\_site\_hierarchy\_vars.yml' tab is active, showing its contents. Below the tabs is a horizontal line with three dashes. The main content of the terminal window is an Ansible playbook:

```
---
# Playbook to create areas, building and floors in Catalyst Center
# Usage: ansible-playbook playbooks/01_demo_ccc/01_create-site-hierarchy.yml -i inventory/
ccc_inventory.yml
- name: Create Sites on Cisco Catalyst Center
  hosts: brkops2357
  connection: local
  gather_facts: no
  vars_files:
    - "../../vars/ccc_site_hierarchy_vars.yml"

  # Initialize ccc connection vars
  vars:
    dnac_login: &dnac_login
    dnac_host: "{{ dnac_host }}"
    dnac_username: "{{ dnac_username }}"
    dnac_password: "{{ dnac_password }}"
    dnac_verify: "{{ dnac_verify }}"
```

At the bottom of the terminal window, there is a status bar with the word 'TERMINAL' and a command prompt: '(.venv) Ansible \$'. The overall interface is a dark-themed terminal application.



CISCO Live!

Get the code: <https://github.com/pamosima/BRKOPS-2357>

# Remember the Config Example?

## Deploy

Sites / EMEA

**Site-1**

Created 2025-01-17 14:16 · Updated 2025-01-17 14:20

Site	Contacts	Journal	Changelog
Site			
Region	EMEA		
Group	—		
Status	Active		
Tenant	BRKOPS-2357		
Facility	—		
Description	—		
Time Zone	—		
Physical Address	Richtistrasse 7, 8302 Wallisellen, Switzerland		
Shipping Address	—		
GPS Coordinates	47.409816, 8.590415		

A playbook.yml

```
1 config:
2   - site:
3     - building:
4       name: "{{ name }}"
5       parent_name: "Global/{{ tenant_name }}/{{ region_name }}"
6       address: "{{ physical_address }}"
7       latitude: "{{ latitude }}"
8       longitude: "{{ longitude }}"
9       country: "{{ physical_address.split(',')[-1].strip() }}"
10      type: building
```

```
! nb_inventory.yml ×
plugin: netbox.netbox.nb_inventory
validate_certs: true
config_context: false
flatten_custom_fields: false
site_data: true
virtual_chassis_name: true
group_names_raw: true

query_filters:
- tenant: brkops2357

group_by:
- tenants

compose:
ccc_template_name: custom_fields.ccc_template_name
ccc_pid: custom_fields.ccc_pid
ccc_ip: primary_ip.address
ccc_location: location.name

TERMINAL
> TERMINAL
(.venv) Ansible $
```

zsh - Ansible +

The image shows a laptop screen with a terminal window open. The terminal window is titled "zsh - Ansible". The content of the terminal is a JSON-like structure representing Ansible host variables for a device named "sw3-1". The variables include:

```
_meta": {  
    "hostvars": {  
        "sw3-1": {  
            "ansible_host": "198.18.3.11",  
            "ccc_ip": "198.18.3.11/24",  
            "ccc_pid": "C9KV-UADP-8P",  
            "ccc_template_name": "Ansible_Day0-Template",  
            "custom_fields": {  
                "ccc_pid": "C9KV-UADP-8P",  
                "ccc_template_name": "Ansible_Day0-Template"  
            },  
            "device_roles": [  
                "branch"  
            ],  
            "device_types": [  
                "cisco-c9kv-uadp-8p"  
            ],  
            "dnac_area": "BRKOPS-2357",  
            "dnac_cli_password": "C1sco12345",  
            "dnac_cli_user": "netadmin",  
            "dnac_debug": true,  
            "dnac_dhcp": "198.18.130.11",  
            "dnac_dns_server": "198.18.130.11",  
            "dnac_domain_name": "demo.local",  
            "dnac_host": "{{ lookup('env', 'DNAC_HOST') }}",  
            "dnac_ntp_server": "198.18.133.141",  
            "dnac_password": "{{ lookup('env', 'DNAC_PASSWORD') }}",  
            "dnac_port": 443,  
        }  
    }  
}
```

# Ansible Lookup Plugin

## Deploy

```
 nb_lookup.yml

1 tasks:
2   # query a list of devices
3   - name: Obtain list of devices from NetBox
4     debug:
5       msg: >
6         "Device {{ item.value.display_name }} (ID: {{ item.key }})"
7     loop: "{{ query('netbox.netbox.nb_lookup', 'devices'
8       api_endpoint='http://localhost/',
9       token='<redacted>') }}"

```

Queries NetBox via its API to return virtually any information capable of being held in NetBox.

```
! 01_create-site-hierarchy.yml ×  
---  
# Playbook to create areas, building and floors in Catalyst Center according to Netbox data  
# Usage: ansible-playbook playbooks/02_demo_nb-ccc/01_create-site-hierarchy.yml -i inventory/  
ccc_inventory.yml  
- name: Create Sites on Cisco Catalyst Center, includes Areas, Buildings, and Floors  
  hosts: brkops2357  
  connection: local  
  gather_facts: false  
  
vars:  
  dnac_login: &dnac_login  
    dnac_host: "{{ dnac_host }}"  
    dnac_username: "{{ dnac_username }}"  
    dnac_password: "{{ dnac_password }}"  
    dnac_verify: "{{ dnac_verify }}"  
    dnac_port: "{{ dnac_port }}"  
    dnac_version: "{{ dnac_version }}"  
    dnac_debug: "{{ dnac_debug }}"  
    state: merged  
  
TERMINAL  
> TERMINAL  
(.venv) Ansible $
```

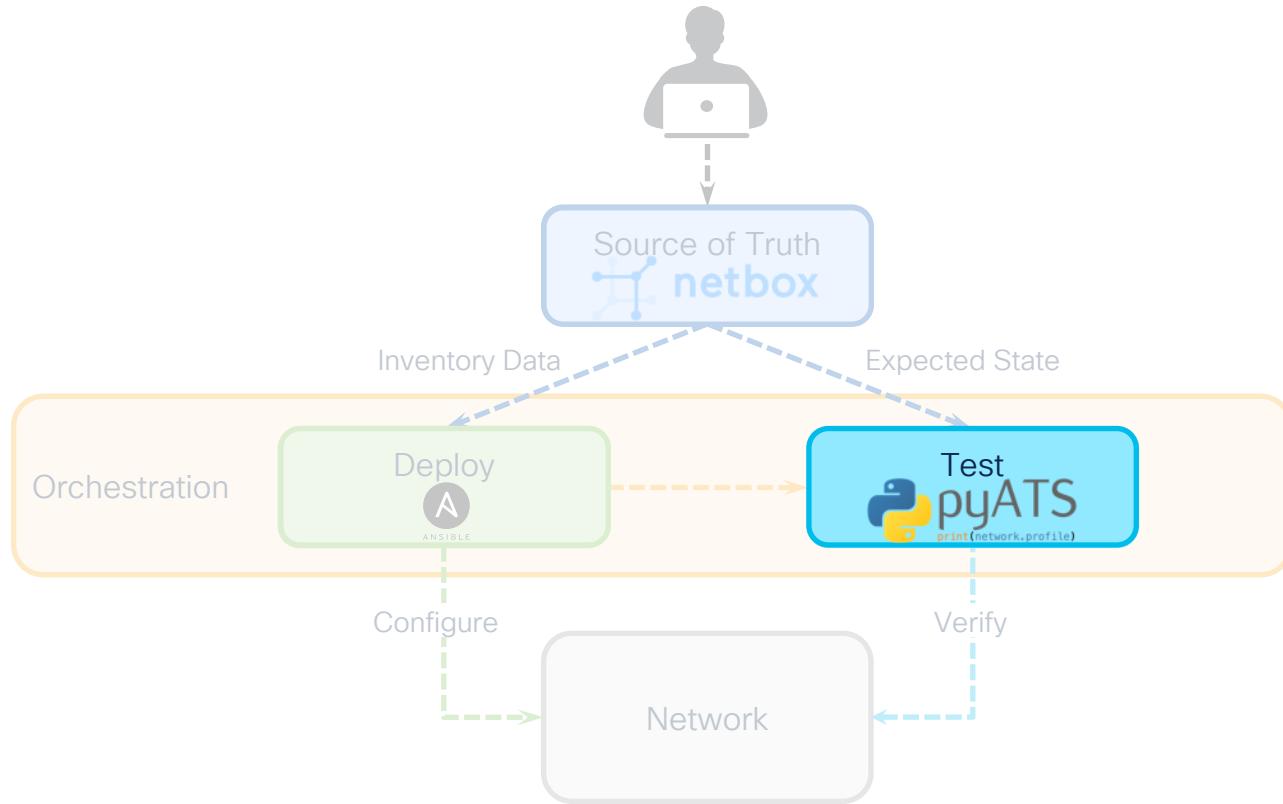
```
TERMINAL
> TERMINAL
zsh - Ansible +
```

```
        "site": {
            "building": {
                "address": "Richtistrasse 7, 8302 Wallisellen, Switzerland",
                "country": "Switzerland",
                "latitude": 47.409816,
                "longitude": 8.590415,
                "name": "Site-3",
                "parent_name": "Global/BRKOPS-2357/EMEA"
            }
        },
        "type": "building"
    },
    {
        "site": {
            "floor": {
                "floor_number": 1,
                "height": 10.0,
                "length": 100.0,
                "name": "Site-3-1",
                "parent_name": "Global/BRKOPS-2357/EMEA/Site-3",
                "rfModel": "Cubes And Walled Offices",
                "units_of_measure": "feet",
                "width": 100.0
            }
        },
        "type": "floor"
    },
    {

```

# Component Overview

Test



# Types of Tests

## Testing



### Unit Testing

Local tests

Check interface, ACL, VLAN



### Integration Testing

Test multiple systems

Routing Table, Port Status



### End-to-End Testing

Complete network test

Ping, Traceroute



### Regression Testing

Check if new feature breaks network

Run validation before and after a change

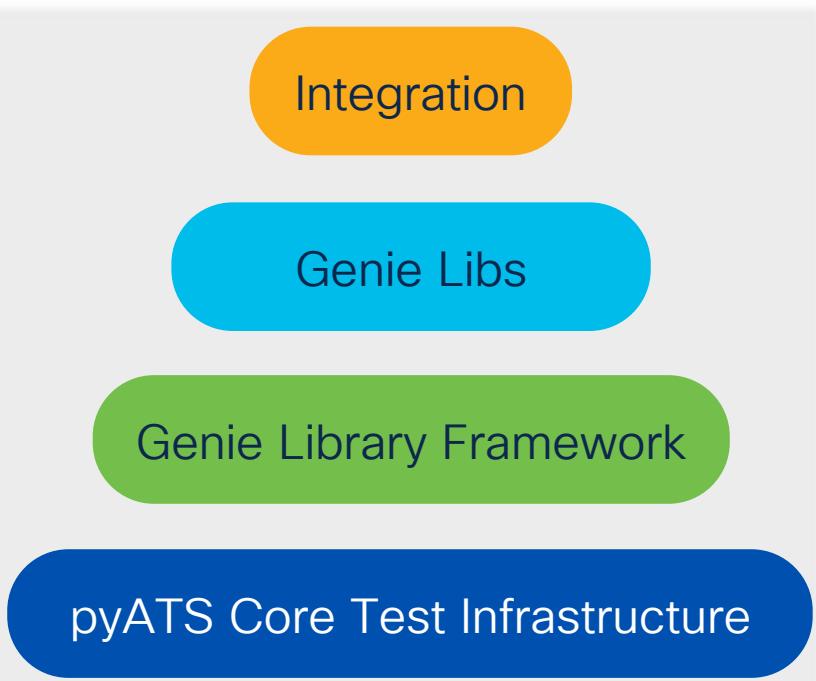
# Why pyATS?

- Automate repetitive network validation tasks
- Ensure operational consistency across devices
- Reduces manual effort and human error
- Fast testing cycles

# Cisco pyATS

## Test

- Testbed
  - Network topology
- ATest Infrastructure
  - Testing framework
- Unicon
  - Device connectivity
- pyATS Library (Genie)
  - Data parsing
- Easypy
  - Runtime environment





# pyATS Testbed

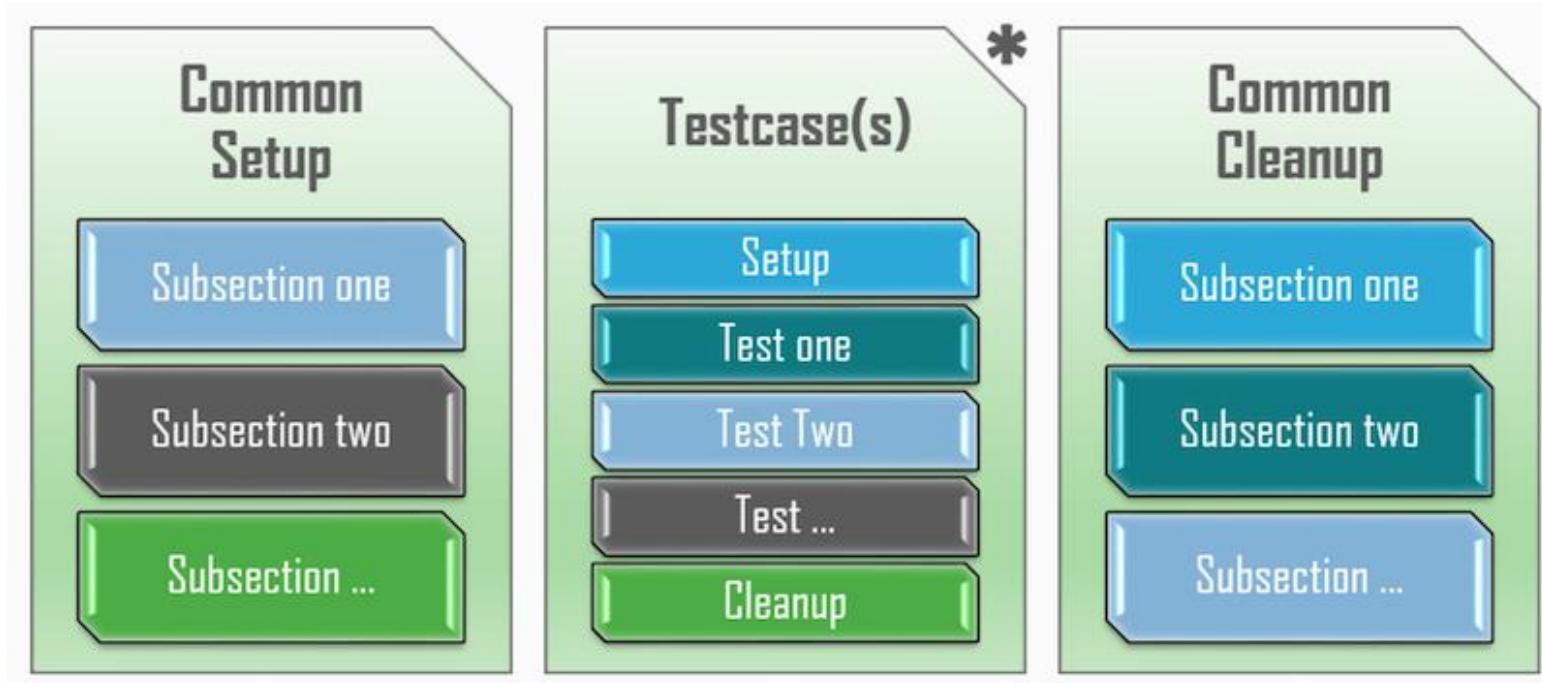
## Test

testbed.yml

```
1 ---  
2 devices:  
3     sw3-1:  
4         type: switch  
5         os: iosxe  
6         credentials:  
7             default:  
8                 username: netadmin  
9                 password: C1sco12345  
10        connections:  
11            mgmt:  
12                protocol: ssh  
13                ip: 198.18.3.11
```

# pyATS ATest Structure

Test



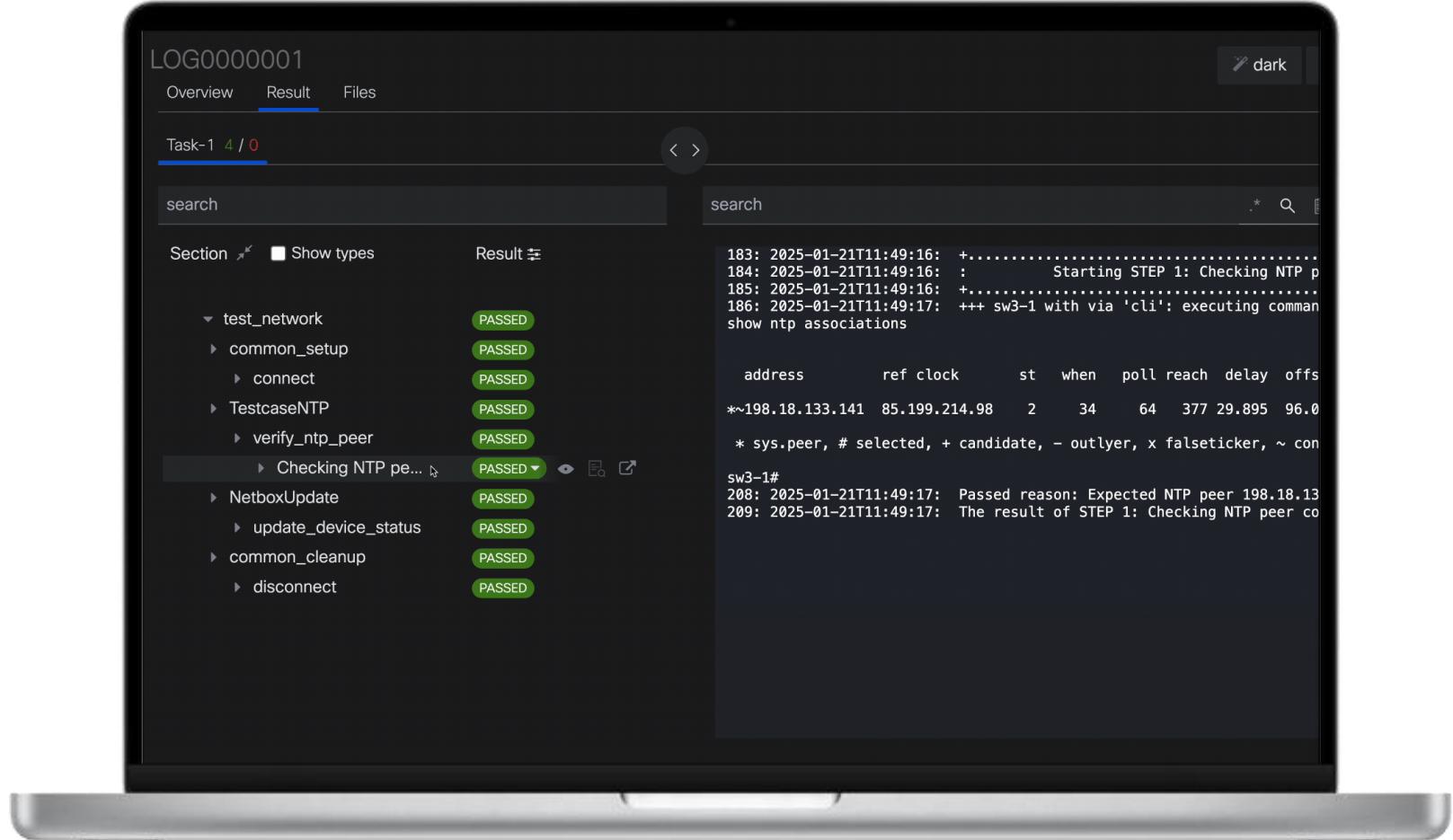
A screenshot of a laptop screen showing a code editor and a terminal window. The code editor window has a dark theme and displays Python code for a test script named 'test\_network.py'. The terminal window below it shows a command-line interface with a prompt '( .venv) pyATS \$'.

```
test_network.py X
from pyats import aetest
from genie.testbed import load
import logging
import pynetbox
import os
from subprocess import call

logging.basicConfig(level=logging.INFO)

import subprocess

class CommonSetup(aetest.CommonSetup):
    @aetest.subsection
    def connect(self, testbed):
        """
        Connect to all devices in the testbed
        """
        logging.info("Connecting to devices...")
        for device_name, device in testbed.devices.items():
            self.log.info(f"Connecting to {device_name}...")
```



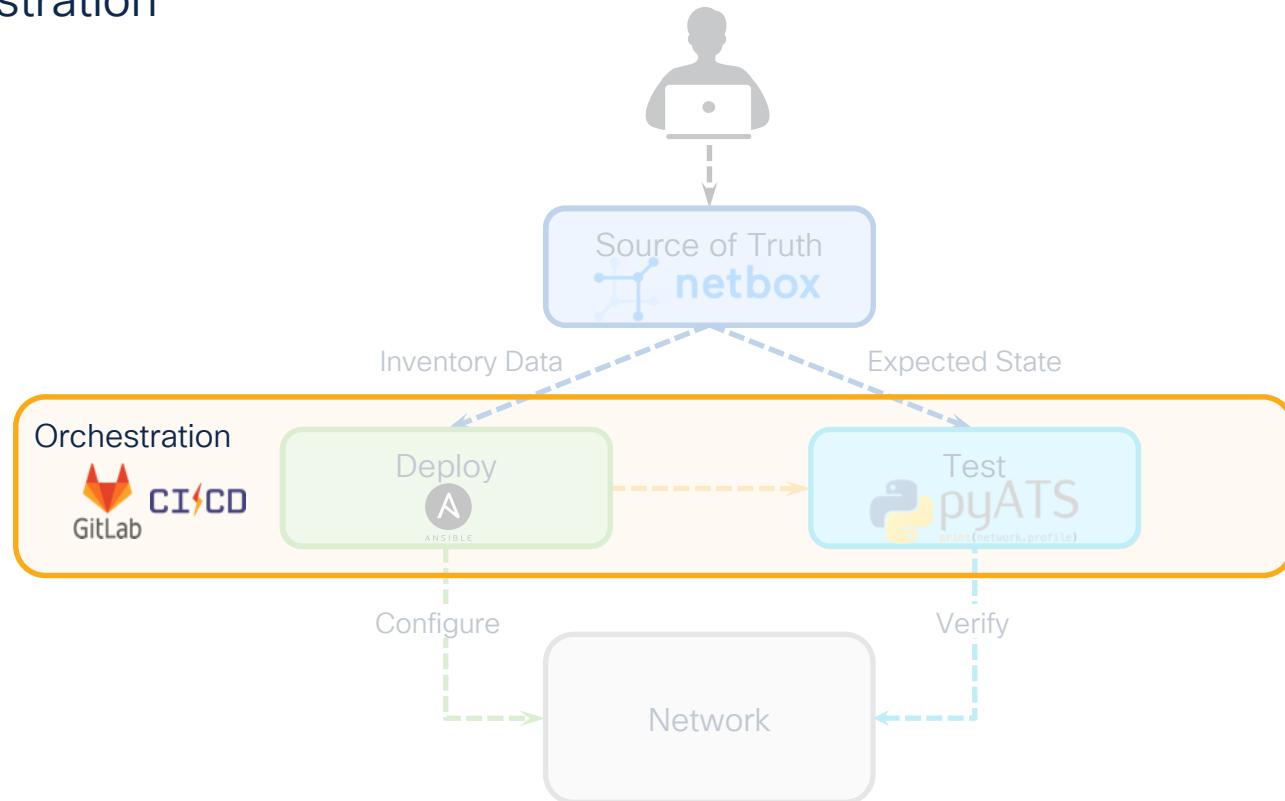
*Automate boring, repetitive tasks  
with data-driven network  
automation!*

# Tying It All Together



# Component Overview

## Orchestration



# GitLab CI/CD

## Orchestration

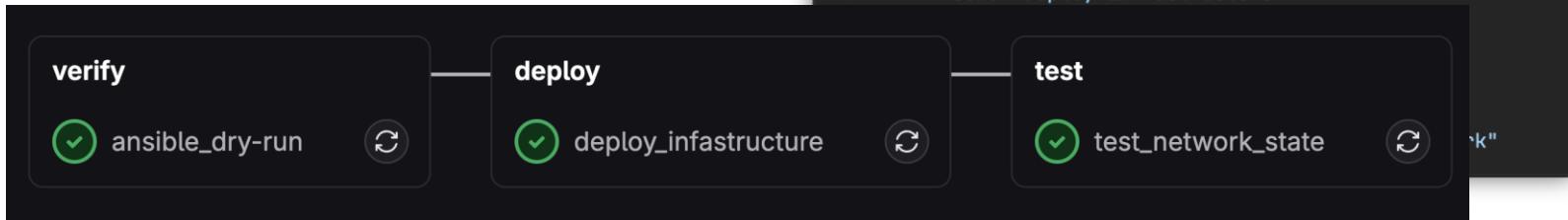
- **.gitlab-ci.yml** File:

- Configuration file defining CI/CD pipeline steps.

- **Pipeline:**

- A series of automated steps (jobs) from code commit to deployment.

```
 .gitlab-ci.yml
1 default:
2   image: git.its-best.ch:5050/ciscolive/brkops-2357:1.2
3
4 stages:
5   - verify
6   - deploy
7   - test
8
9 ansible_dry-run:
10  stage: verify
11  script:
12    - echo "Check Mode & preview changes"
13
14 deploy_infrastructure:
15  stage: deploy
16  script:
17    - echo "Deploy infrastructure"
```



# Why GitLab CI/CD?

- **Workflow Automation:** Run tasks in order with historical tracking for review and auditing.
- **Variable Management:** Store ENV vars centralized and securely.
- **Flexible Runners:** Execute Ansible, and pyATS jobs efficiently.

The screenshot shows a GitHub repository interface for the project "BRKOPS-2357-pipeline".

**Project Sidebar:**

- Pinned
- Issues (0)
- Merge requests (0)
- Manage >
- Plan >
- </> Code >
- Build >
- Secure >
- Deploy >
- Operate >
- Monitor >
- Analyze >
- Settings >
- Help >
- Admin

**Repository Header:**

ciscolive / BRKOPS-2357-pipeline

**Repository Title:**

## B BRKOPS-2357-pipeline

**Code Tab:**

- main
- BRKOPS-2357-pipeline /
- + <button>
- History
- Find file
- Edit
- Code

**Recent Activity:**

Update .gitlab-ci.yml file  
Patrick Mosimann authored 11 minutes ago  
910d50ed

**File List:**

Name	Last commit	Last update
Ansible	triggered by NetBox	3 days ago
pyATS	removed VPN	4 days ago
.gitlab-ci.yml	Update .gitlab-ci.yml file	11 minutes ago
CONTRIBUTING.md	initial commit	19 minutes ago
LICENSE	initial commit	22 minutes ago
README.md	initial commit	22 minutes ago

**README.md Content:**

### BRKOPS-2357 Demo Code (2025)

This repository contains demonstration code from the Cisco Live session BRKOPS-2357.

**Project Information:**

- 71 Commits
- 1 Branch
- 0 Tags
- 55.3 MiB Project Storage
- README
- LICENSE
- CONTRIBUTING
- CI/CD configuration
- + Add CHANGELOG
- Auto DevOps enabled
- + Add Kubernetes cluster
- + Add Wiki
- + Configure Integrations

**Created on:** January 13, 2025

Search... 💡 ⚡

Scripts / 02\_switches / AddSwitchesToSite / 2025-01-21 09:29:03

Results Log threshold Debug ⚙️ Config

Started: 2025-01-21 09:29:03 Duration: 0 minutes, 2.77 seconds Completed

**Log**

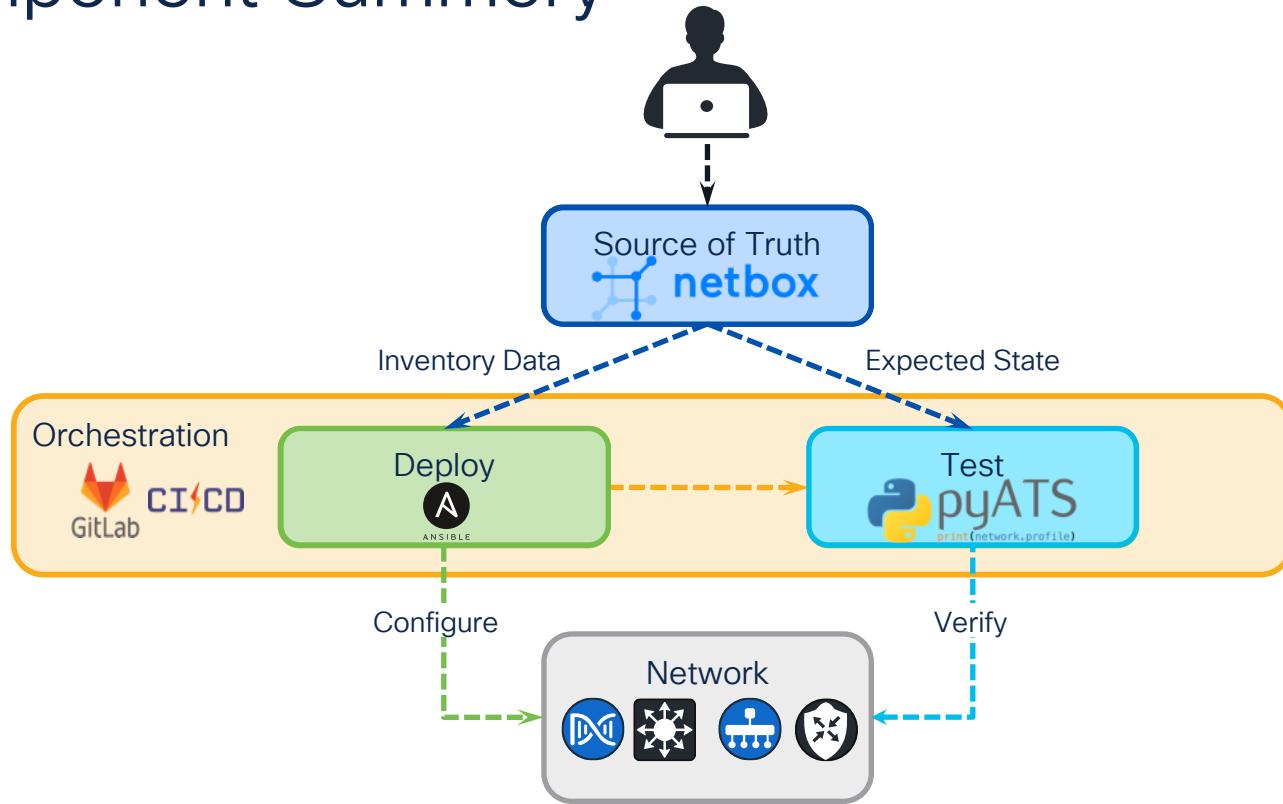
LINE	TIME	LEVEL	OBJECT	MESSAGE
1	2025-01-21T09:29:04.967519+00:00	Success	—	Added switch with serial number CML13SW1 to site Site 1.
2	2025-01-21T09:29:04.967546+00:00	Info	—	Running the CreateIps script...
3	2025-01-21T09:29:05.815651+00:00	Success	—	CreateIps script executed successfully.
4	2025-01-21T09:29:05.815723+00:00	Info	—	Commit is set. Triggering GitLab pipeline...
5	2025-01-21T09:29:06.704454+00:00	Success	—	Pipeline triggered successfully. ↗
6	2025-01-21T09:29:06.704713+00:00	Success	—	Finished adding switches to the site.

Showing 1-6 of 6

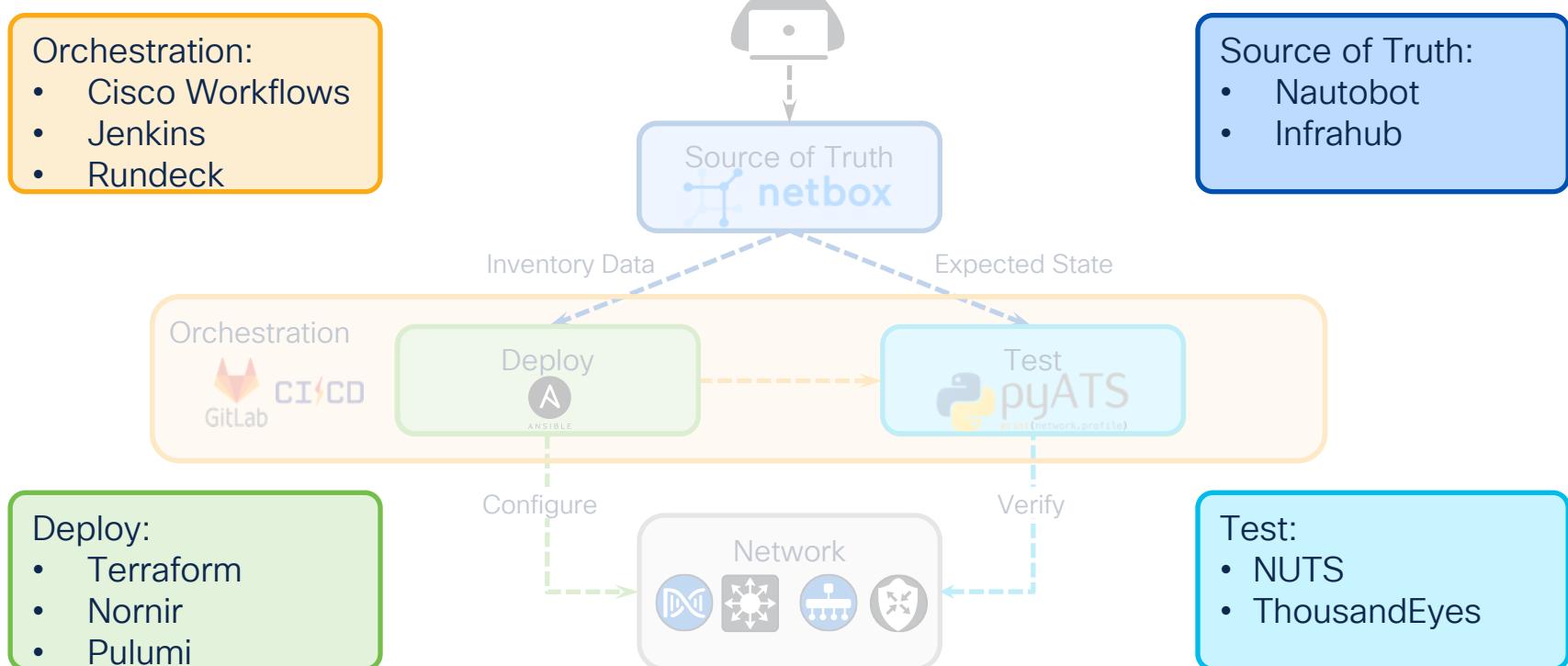
**Output**

None

# Component Summary



# Tool Alternatives



# Procedural vs Declarative

## Tool Alternatives: Deploy

- 1 Prepare ingredients
- 2 Add meat sauce layer to dish
- 3 Add pasta layer to dish
- 4 Repeat step 2+3 several times
- 5 Top with cheese
- 6 Cook in oven at 190C for 25 minutes
- 7 Remove from oven and split up into portions

Procedural



Create a Lasagna

Declarative

# Ansible vs Terraform?

## Tool Alternatives: Deploy



Ansible

Many generic options  
but can result in  
configuration drift

~~VS~~



Terraform

Requires a provider and  
creates immutable  
infrastructure

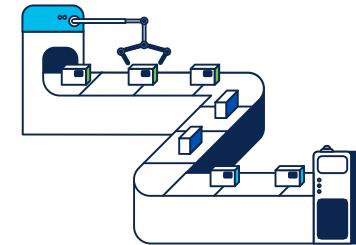
It is not either or, use a combination of both!

Ansible → send commands to provisioned systems  
Terraform → provision immutable infrastructure

# Further Developments (MVP+)

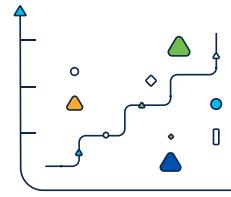
## Future Features:

- Day N Tasks Automation:
  - VLAN Management
  - Switch Port Configuration



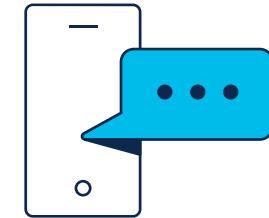
## Integration Plans:

- Cisco Catalyst SD-WAN Manager
- DHCP Server

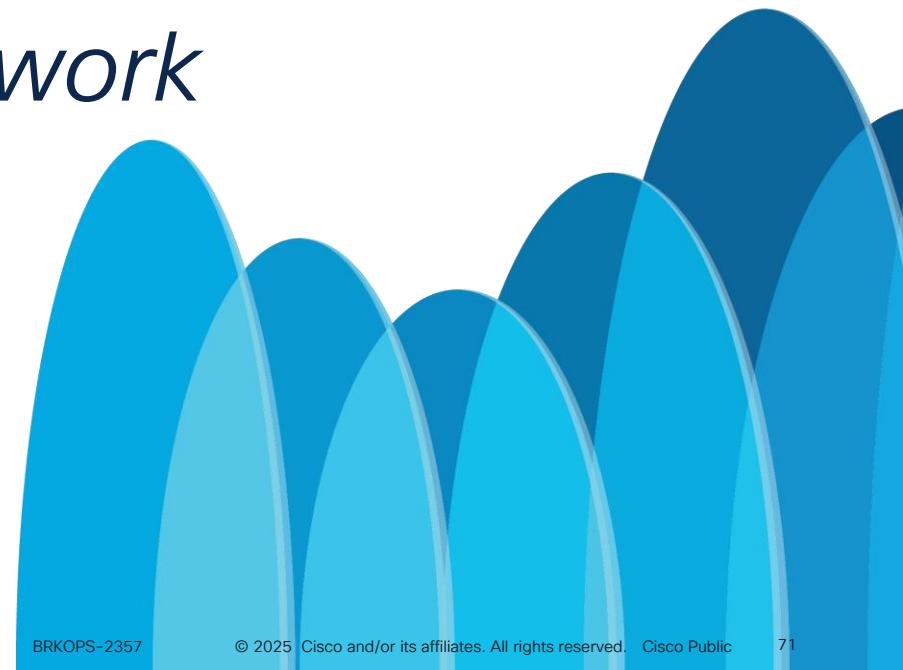


## User Feedback:

- Write Your Idea in the Webex Space
- Contribute to the GitHub Project



*Automate boring, repetitive tasks  
with data-driven network  
automation!*



# Key Takeaways

1

## Reduce Errors:

Ensure data accuracy with a centralized Source of Truth.

2

## Streamline Deployments:

Use Ansible and GitLab CI/CD for fast, reliable setups.

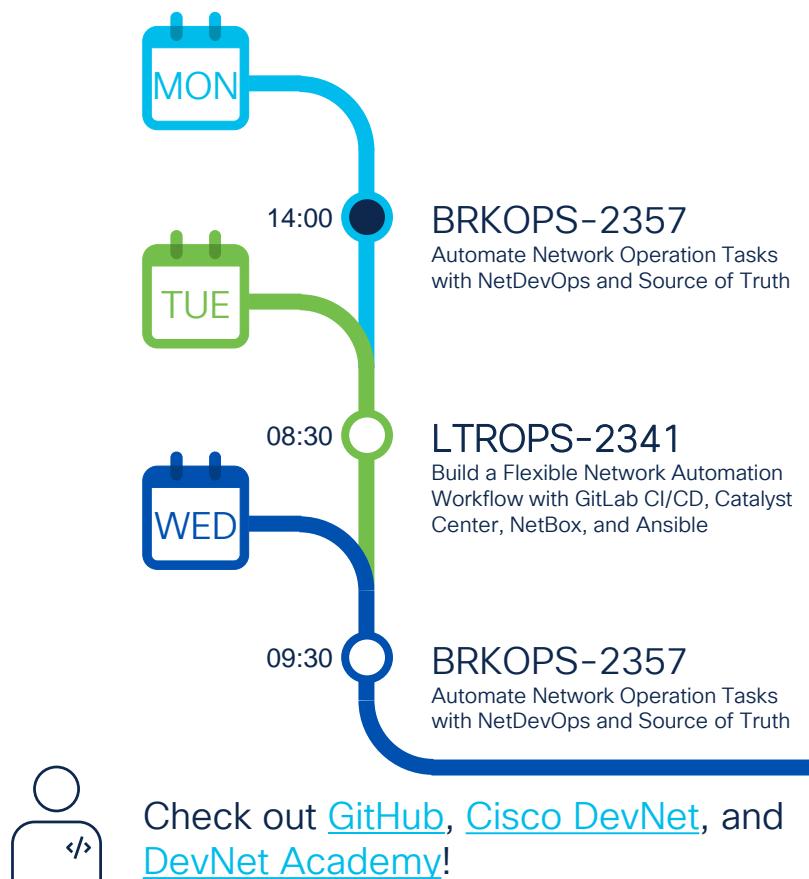
3

## Validate with Confidence:

Automate testing and validation with pyATS.

# Continue your education

CISCO Live!



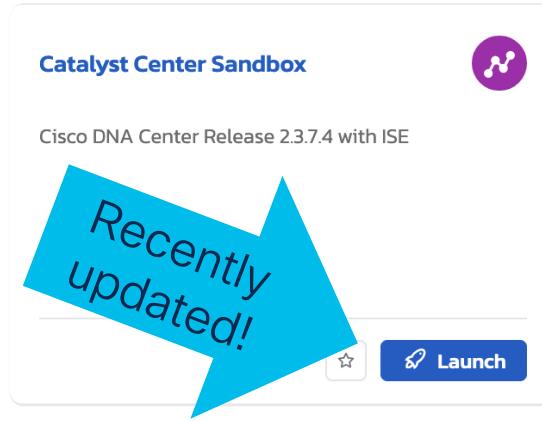
# Call Out

Get the code from GitHub:



[https://github.com/pamosima/  
BRKOPS-2357](https://github.com/pamosima/BRKOPS-2357)

Catalyst Center Sandbox:



- Running version [2.3.7.4](#)
- Include CML, 4 Cat 9kv, ISE
- 10x increased capacity

NetBox Cloud:

sign up for the [Free Plan](#) which includes

- up to 100 devices
- 500 IP addresses
- 10k API requests/month
- 2 operational branches

perfect for a small network/lab/PoC.

# Webex App

## Questions?

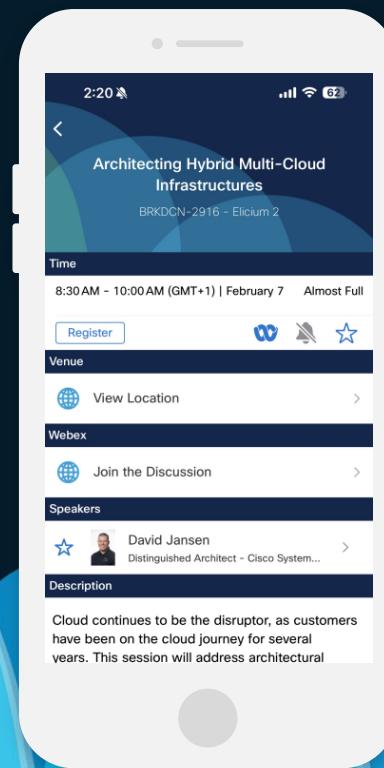
Use the Webex app to chat with the speaker after the session

## How

- 1 Find this session in the Cisco Events 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 February 28, 2025.

CISCO Live!



# Fill Out Your Session Surveys



Participants who fill out a minimum of 4 session surveys and the overall event survey will get a unique Cisco Live t-shirt.

(from 11:30 on Thursday, while supplies last)



All surveys can be taken in the Cisco Events mobile app or by logging in to the Session Catalog and clicking the 'Participant Dashboard'

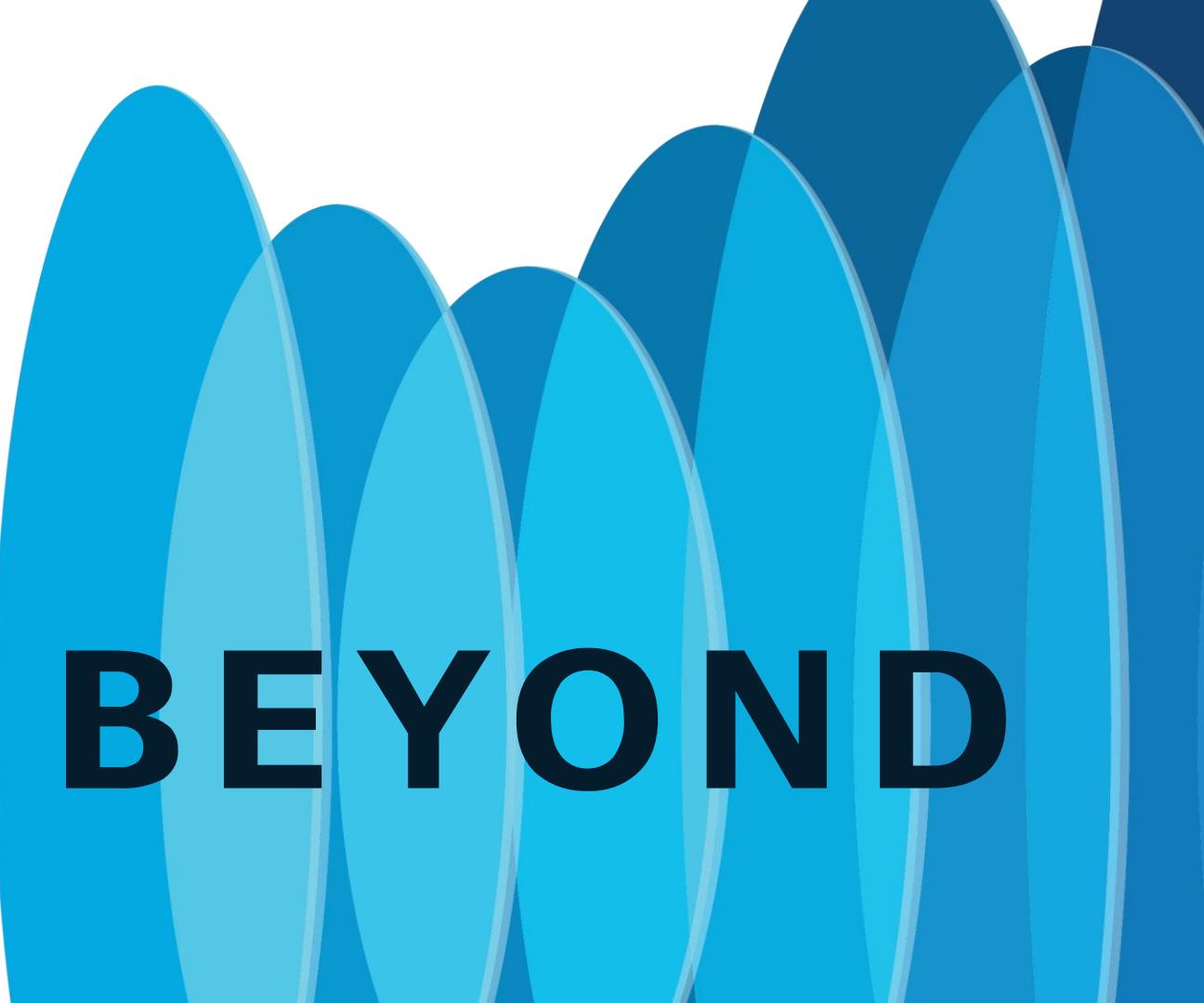


Content Catalog



# Thank you

cisco *Live!*



**GO BEYOND**