cisco Live!

#CiscoLive

The bridge to possible

Ansible and DCNM!

Parity in Function and Value to Operate Your Fabric

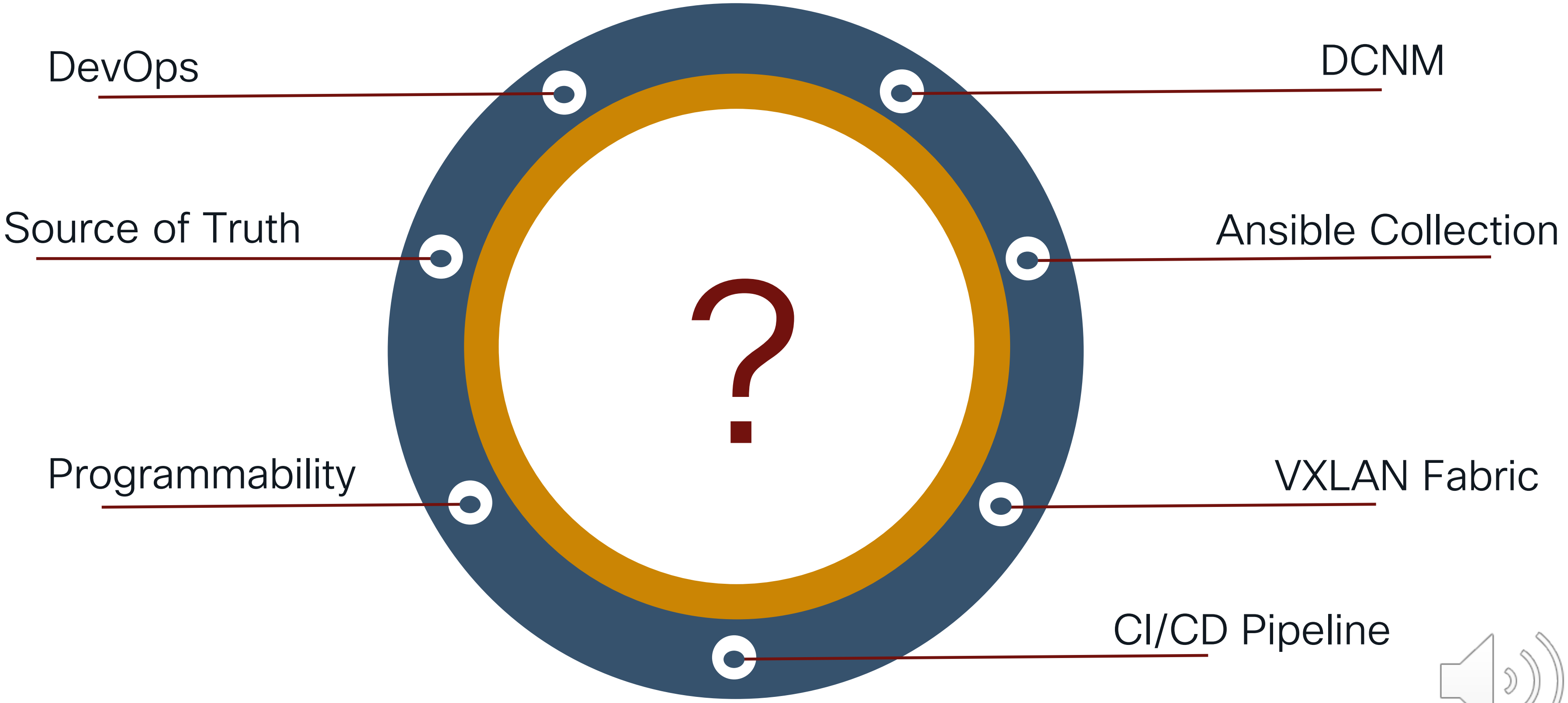
Mike Wiebe, Technical Leader Engineering

BRKDCN-2512

Let's Meet Cliff

- Managing VXLAN EVPN fabric
- Using DCNM Version 11.5(1)
- Wants to improve automation

What If …


DevOps
Source of Truth
Programmability
DCNM
Ansible Collection
VXLAN Fabric
CI/CD Pipeline
?

Cisco DCNM Ansible Collection Solves This!

Agenda

- Introduction
- Install the DCNM Ansible Collection
- Automate using the DCNM Ansible Collection
- DCNM Ansible and DevOps Workflows
- DCNM Ansible Collection Roadmap
- Conclusion

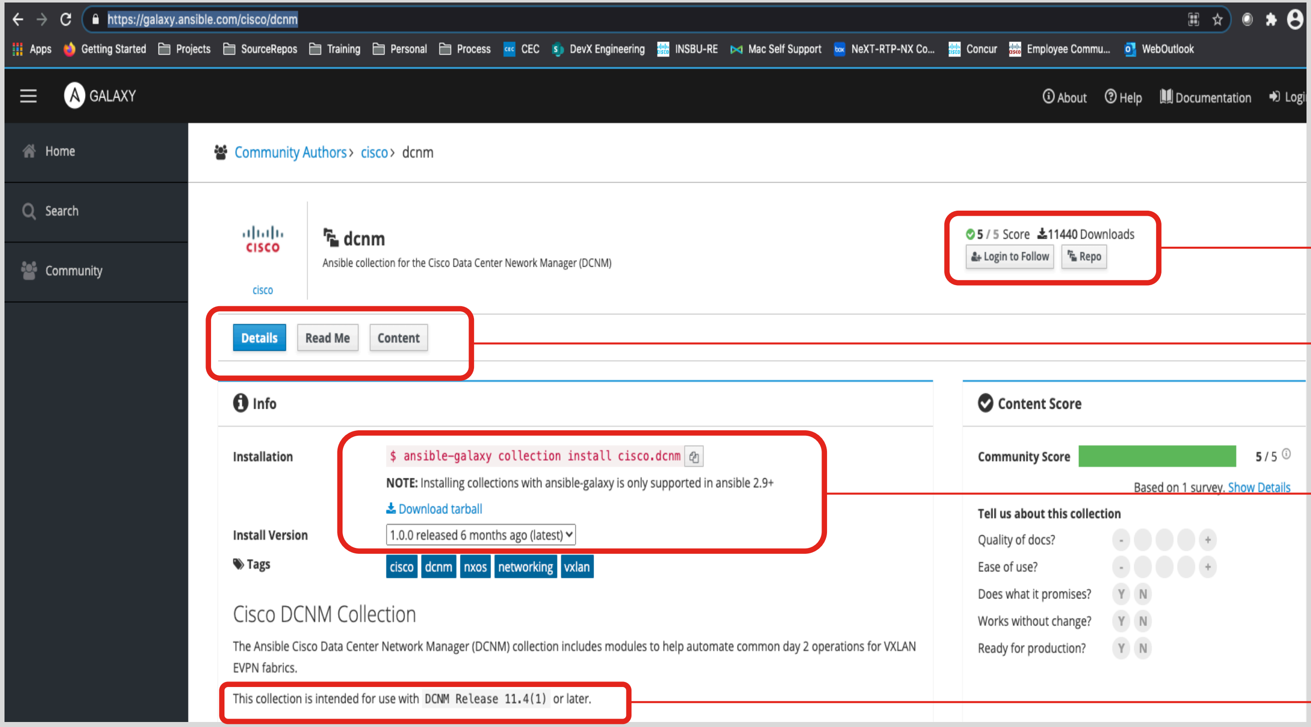
Brief Ansible Collection Intro

- Distribution format for Ansible content
- First introduced in Ansible 2.8 (tech preview)
- Fully supported in Ansible 2.9
- Non-base plugins/modules moved out of ansible/ansible in Ansible 2.10

Installing the DCNM Ansible Collection

- Collection available on https://galaxy.ansible.com/cisco/dcnm
- Install using ansible using ansible-galaxy
  - `ansible-galaxy collection install cisco.dcnm`
- Ansible uses the Fully Qualified Collection Name (FQCN)
  - Namespace: cisco
  - Collection Name: dcnm
- DCNM module references in the playbook must use the FQCN

DCNM Ansible Galaxy Collection Site



dcnm

Ansible collection for the Cisco Data Center Nework Manager (DCNM)

5 / 5 Score 11440 Downloads

Installation

```
$ ansible-galaxy collection install cisco.dcnm
```

NOTE: Installing collections with ansible-galaxy is only supported in ansible 2.9+

Download tarball

Install Version: 1.0.0 released 6 months ago (latest)

Tags: cisco dcnm nxos networking vxlan

Cisco DCNM Collection

The Ansible Cisco Data Center Network Manager (DCNM) collection includes modules to help automate common day 2 operations for VXLAN EVPN fabrics.

This collection is intended for use with `DCNM Release 11.4(1)` or later.

Community Score 5 / 5

Based on 1 survey. Show Details

Tell us about this collection

Quality of docs?	- +
Ease of use?	- +
Does what it promises?	Y N
Works without change?	Y N
Ready for production?	Y N

Link to repo

Read Me

Collection Version & Install Cmd

DCNM Version

Demo – Install DCNM Collection

Agenda

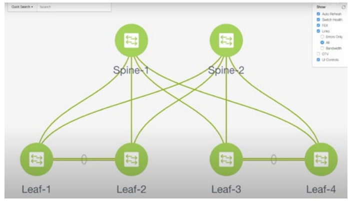
- Introduction
- Install the DCNM Ansible Collection
- Automate using the DCNM Ansible Collection
- DCNM Ansible and DevOps Workflows
- DCNM Ansible Collection Roadmap
- Conclusion

DCNM Ansible Collection Details

- Latest Collection Version: 1.0.0 – Lan Fabric
- Ansible DCNM Modules

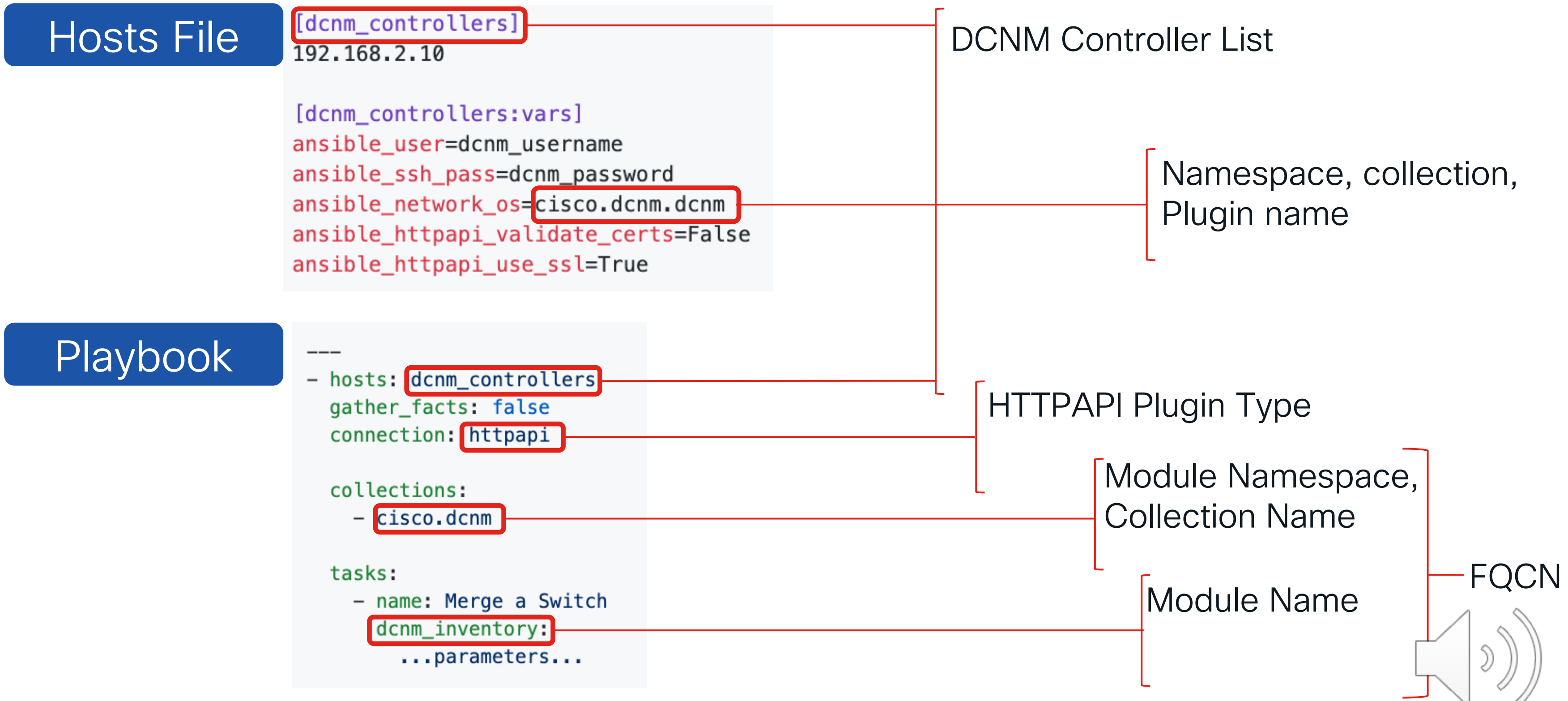
  5 Modules in the current collection

  - cisco.dcnm.dcnm_inventory
  - cisco.dcnm.dcnm_vrf
  - cisco.dcnm.dcnm_network
  - cisco.dcnm.dcnm_interface
  - cisco.dcnm.dcnm_rest



HTTPAPI Connection Plugin

- Configuration required to use the connection plugin



```
[dcnm_controllers]
192.168.2.10

[dcnm_controllers:vars]
ansible_user=dcnm_username
ansible_ssh_pass=dcnm_password
ansible_network_os=cisco.dcnm.dcnm
ansible_httpapi_validate_certs=False
ansible_httpapi_use_ssl=True
```

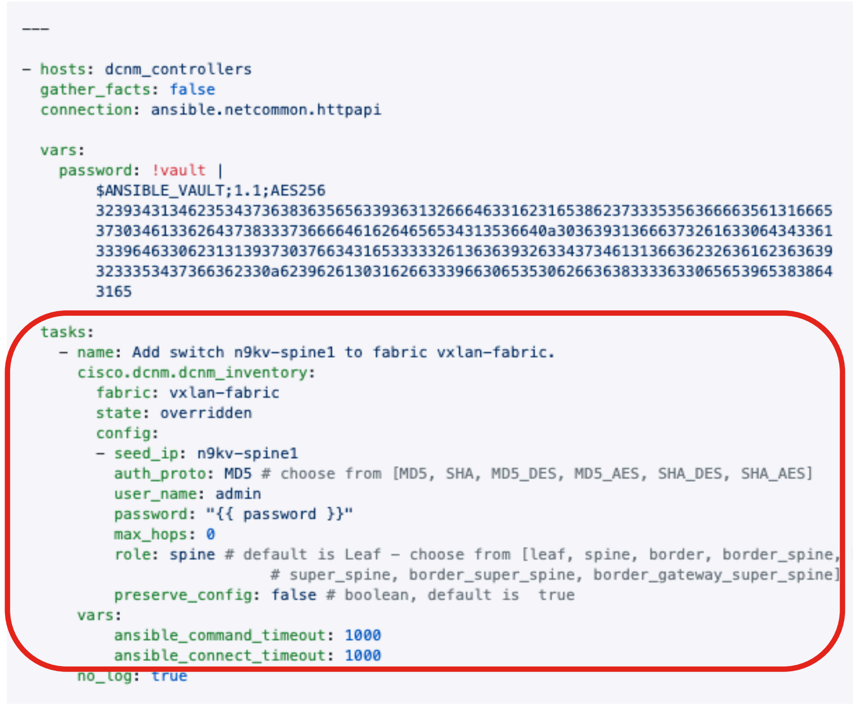
```
---
- hosts: dcnm_controllers
  gather_facts: false
  connection: httpapi

  collections:
    - cisco.dcnm

  tasks:
    - name: Merge a Switch
      dcnm_inventory:
        ...parameters...
```

dcnm_inventory module

- Manage switches for an existing fabric
  - Add and remove switches and assign roles (spine, leaf, boarder etc…)



```
---

- hosts: dcnm_controllers
  gather_facts: false
  connection: ansible.netcommon.httpapi

  vars:
    password: !vault |
        $ANSIBLE_VAULT;1.1;AES256
        32393431346235343736383635656339363132666463316231653862373335356366663561316665
        37303461336264373833373666646162646565343135363640a303639313666373261633064343361
        33396463306231313937303766343165333332613636393263343734613136636232636162363639
        3233353437366362330a623962613031626633396630653530626636383333633065653965383864
        3165

  tasks:
    - name: Add switch n9kv-spine1 to fabric vxlan-fabric.
      cisco.dcnm.dcnm_inventory:
        fabric: vxlan-fabric
        state: overridden
        config:
        - seed_ip: n9kv-spine1
          auth_proto: MD5 # choose from [MD5, SHA, MD5_DES, MD5_AES, SHA_DES, SHA_AES]
          user_name: admin
          password: "{{ password }}"
          max_hops: 0
          role: spine # default is Leaf - choose from [leaf, spine, border, border_spine,
                                  # super_spine, border_super_spine, border_gateway_super_spine]
          preserve_config: false # boolean, default is  true
      vars:
        ansible_command_timeout: 1000
        ansible_connect_timeout: 1000
      no_log: true
```

- Brownfield or Greenfield
- Must specify each device separately.
- Timeout
  - ansible_command >= 1000s
  - ansible_connect >= 1000s
- Password encrypted using vault
- Supported States
  - merged, overridden, deleted, query

dcnm_vrf module

- Create and attach vrf object to switches

Ansible Playbook

```
- name: MERGED - Create, Attach and Deploy new VRF
  cisco.dcnm.dcnm_vrf:
    fabric: "{{ ansible_it_fabric }}"
    state: merged
    config:
    - vrf_name: ansible-vrf-int1
      vrf_id: 9008011
      vlan_id: 500
      attach:
       - ip_address: "{{ ansible_switch1 }}"
       - ip_address: "{{ ansible_switch2 }}"
      deploy: true
```

- vrf_id and vlan_id are optional
  - Auto-populated by DCNM
- deploy flag controls device configuration
- Supported States:
  - merged, replaced, overridden, deleted, query

dcnm_network module

- Create and attach network object to switches

Ansible Playbook

```
- name: MERGED - Create, Attach and Deploy new NET
  cisco.dcnm.dcnm_network:
    fabric: "{{ ansible_it_fabric }}"
    state: merged
    config:
    - net_name: ansible-net13
      vrf_name: Tenant-1
      net_id: 7005
      vlan_id: 1500
      gw_ip_subnet: '192.168.30.1/24'
      attach:
      - ip_address: "{{ ansible_switch1 }}"
        ports:
            - "{{ ansible_sw1_int1 }}"
            - "{{ ansible_sw1_int2 }}"
      deploy: True
```

- net_id and vlan_id are optional
  - Auto-populated by DCNM
- deploy flag controls device configuration
- Supported States:
  - merged, replaced, overridden, deleted, query

dcnm_interface module

- Create, modify and remove interfaces

Ansible Playbook

```
- name: Create loopback interfaces
  cisco.dcnm.dcnm_interface:
    fabric: "{{ ansible_it_fabric }}"
    state: merged
    config:
      - name: lo100
        type: lo
        switch:
          - "{{ ansible_switch1 }}"
        deploy: true
        profile:
          admin_state: true
          mode: lo
          ipv4_addr: 192.169.10.1
          ipv6_addr: fd01::0201
          cmds:
            - no shutdown
          description: "loopback interface 100"
```

- Supported types:
  - Ethernet (access, trunk)
  - Routed Interface, sub-interface
  - Port-channel, Virtual PC
  - Loopback
- Profile has unique interface specific parameters
- Supported States:
  - merged, replaced, overridden, deleted, query

dcnm_rest module

- Query or Configure DCNM using any available DCNM APIs

Ansible Playbook

```
- name: create template
  cisco.dcnm.dcnm_rest:
    method: POST
    path: /fm/fmrest/config/templates/template
    json_data: |
      {
        "content": "##template properties
\nname=demo_template;\ndescription = ;\ntags =
;\nuserDefined = true;\nsupportedPlatforms =
All;\ntemplateType = POLICY;\ntemplateSubType =
DEVICE;\ncontentType = TEMPLATE_CLI;\nimplements =
;\ndependencies = ;\npublished = false;\n##\n\n##template
variables\n##\n##template content\n##"
      }
":null}'
```

- Module can be used until feature specific module can be developed
- This example creates a new template in DCNM.
- User must know the rest path and provide raw json_data.

Module States Explained – (dcnm_vrf)

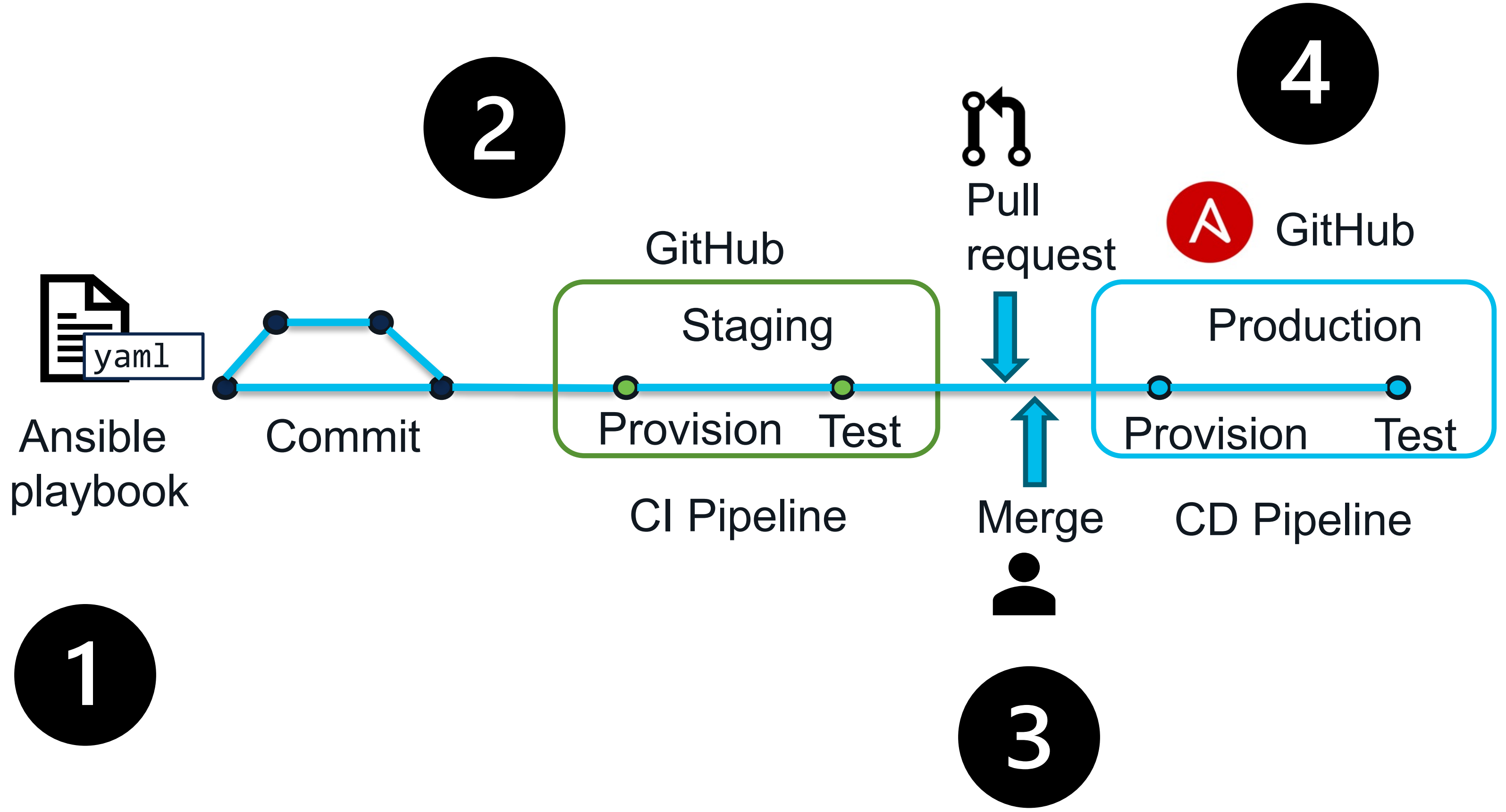
Merged	VRF created or update supported properties
Replaced	VRF created or completely replaced – Source of Truth
Overridden	VRF created or completely replaced and VRFs not specified in the playbook are deleted - Source of Truth
Deleted	VRF deleted
Query	Current state of VRFs returned by the module – parameters act as query filter

Demo – Automate with Ansible and DCNM

Agenda

- Introduction
- Install the DCNM Ansible Collection
- Automate using the DCNM Ansible Collection
- DCNM Ansible and DevOps Workflows
- DCNM Ansible Collection Roadmap
- Conclusion

CI/CD Pipeline – Network as Code



Demo – CI/CD Pipeline with Ansible DCNM

Agenda

- Introduction
- Install the DCNM Ansible Collection
- Automate using the DCNM Ansible Collection
- DCNM Ansible and DevOps Workflows
- DCNM Ansible Collection Roadmap
- Conclusion

DCNM Ansible Roadmap

Module Extensions:

- VRF LITE support extensionsn for dcnm_vrf module
- Multisite support for dcnm_vrf and dcnm_network modules

New Modules:

- Image Management
  - Upload images to DCNM
  - Upgrade fabric switches
- Install and apply RPM’s and SMU’s
- Customized Template and Policy Creation/Association with fabric switches
- L4-L7 Service Insertion

Agenda

- Introduction
- Install the DCNM Ansible Collection
- Automate using the DCNM Ansible Collection
- DCNM Ansible and DevOps Workflows
- DCNM Ansible Collection Roadmap
- Conclusion

Continue your education

- Demos in the Cisco campus
- Meet the engineer 1:1 meetings
- Walk-in labs
- Related sessions

The bridge to possible

Thank you

#CiscoLive