
cisco
You make possible

Continuous Integration and Testing for Networks

Steven Carter, Principal Architect
Chris Hocker, Solutions Architect

BRKDEV-3326

cisco Live!
Barcelona | January 27-31, 2020

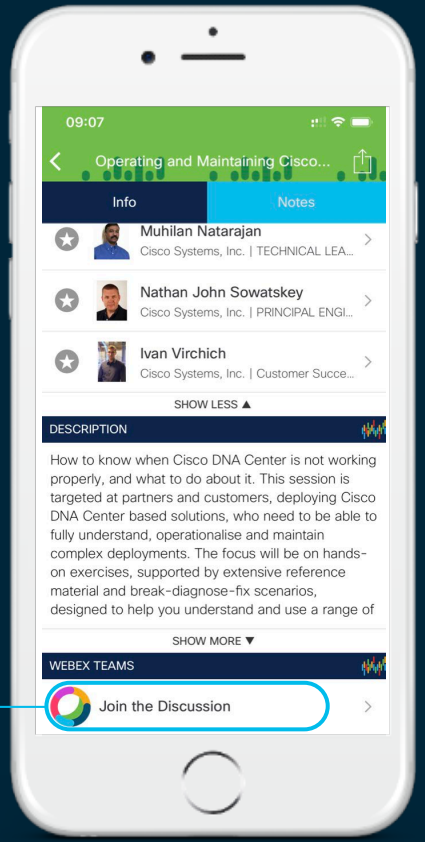
Cisco Webex Teams

Questions?

Use Cisco Webex Teams 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 Webex Teams or go directly to the team space
4. Enter messages/questions in the team space



Agenda

- Introduction
- Automation
- Simulation
- Testing
- DevOps
- Conclusion

Before We Begin

This is an advanced-level session:

- We make some assumptions:
  - Basic understanding of IaC and source control (i.e. git)
  - Basic understanding of Ansible
  - Working knowledge of SD-WAN
- You are open to a new way of operating your infrastructure
  - This is not something that is implemented in a week
  - You'll have/need help
  - Requires an evolution in skill sets

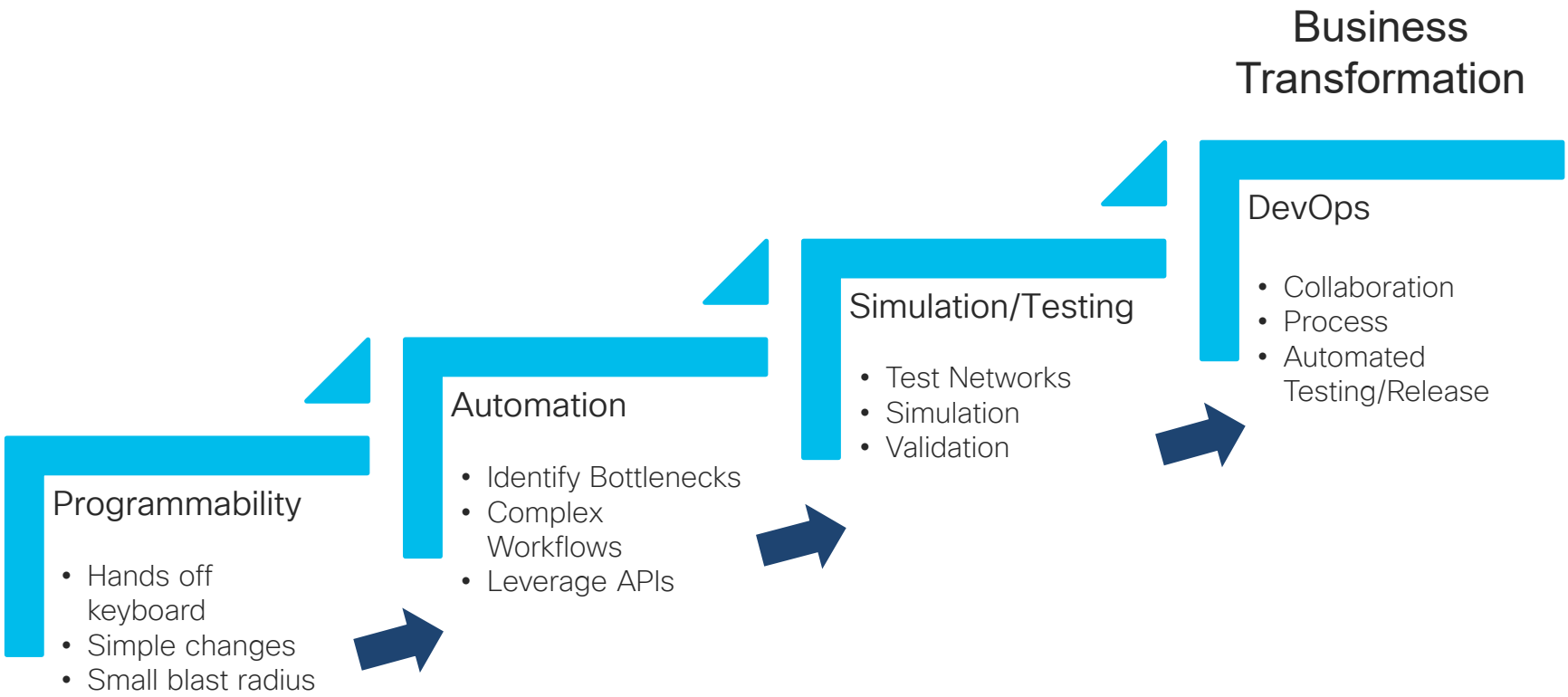
Ask Questions (although we will have to move ahead at times)

Introduction

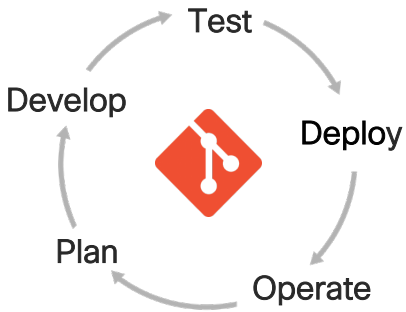
Business Transformation

Time to Value Configuration & Change Automation		Time to Remediation Automated Fault Remediation	
Faster Customer Service On-boarding	Faster Execution of Change Requests	Faster Execution of Maintenance	Faster Troubleshooting and Remediation

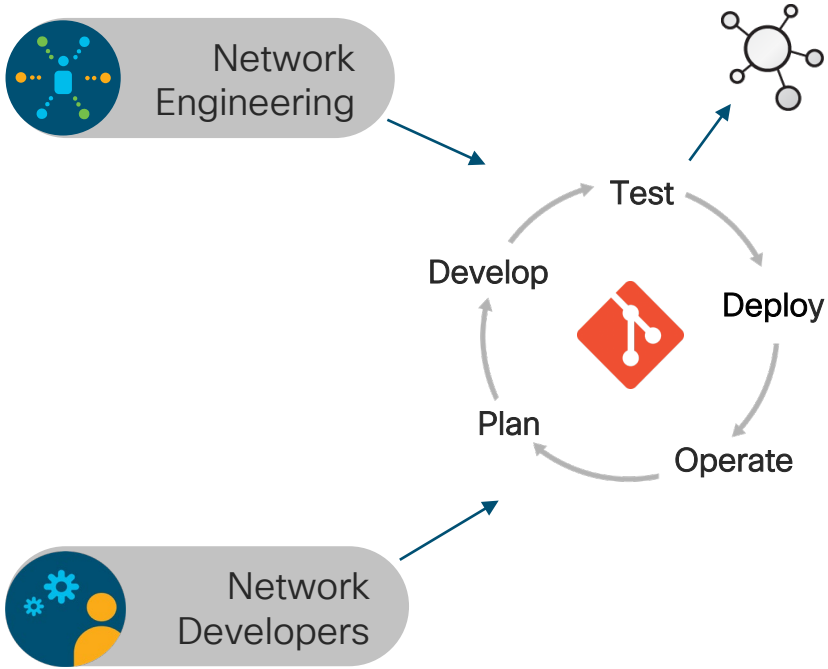
The Goal



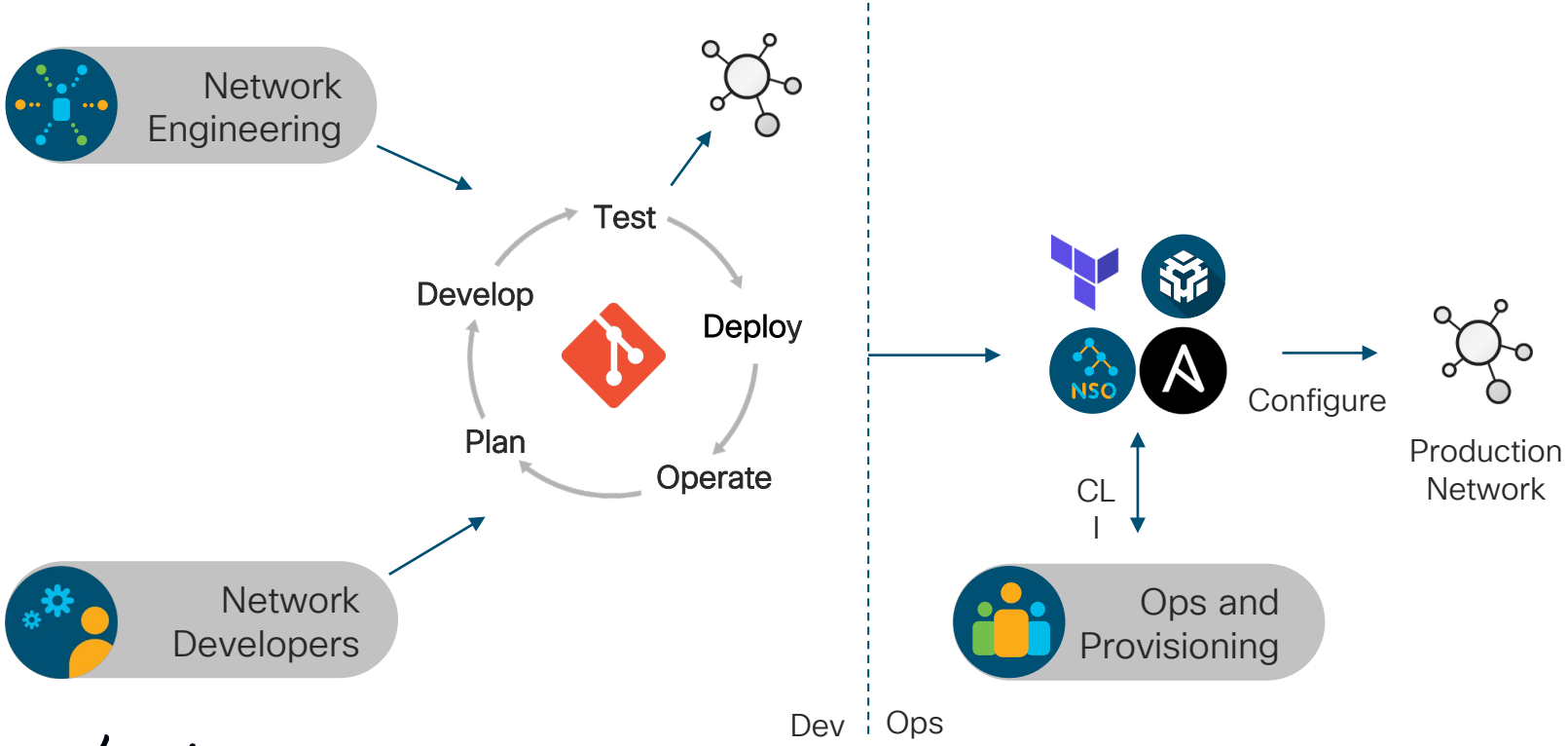
Infrastructure Lifecycle through Code Lifecycle



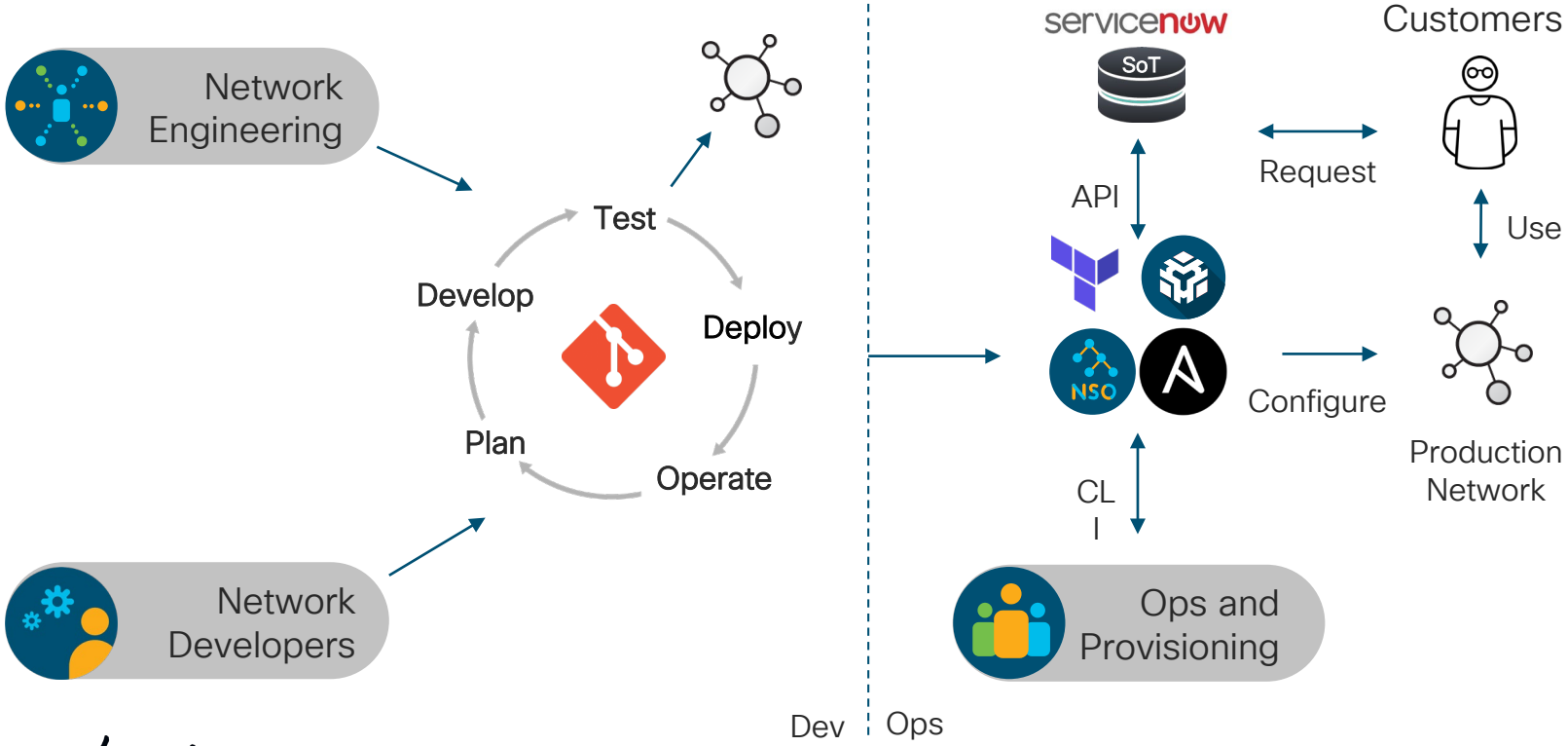
CI: Continuous Integration of Features and Fixes



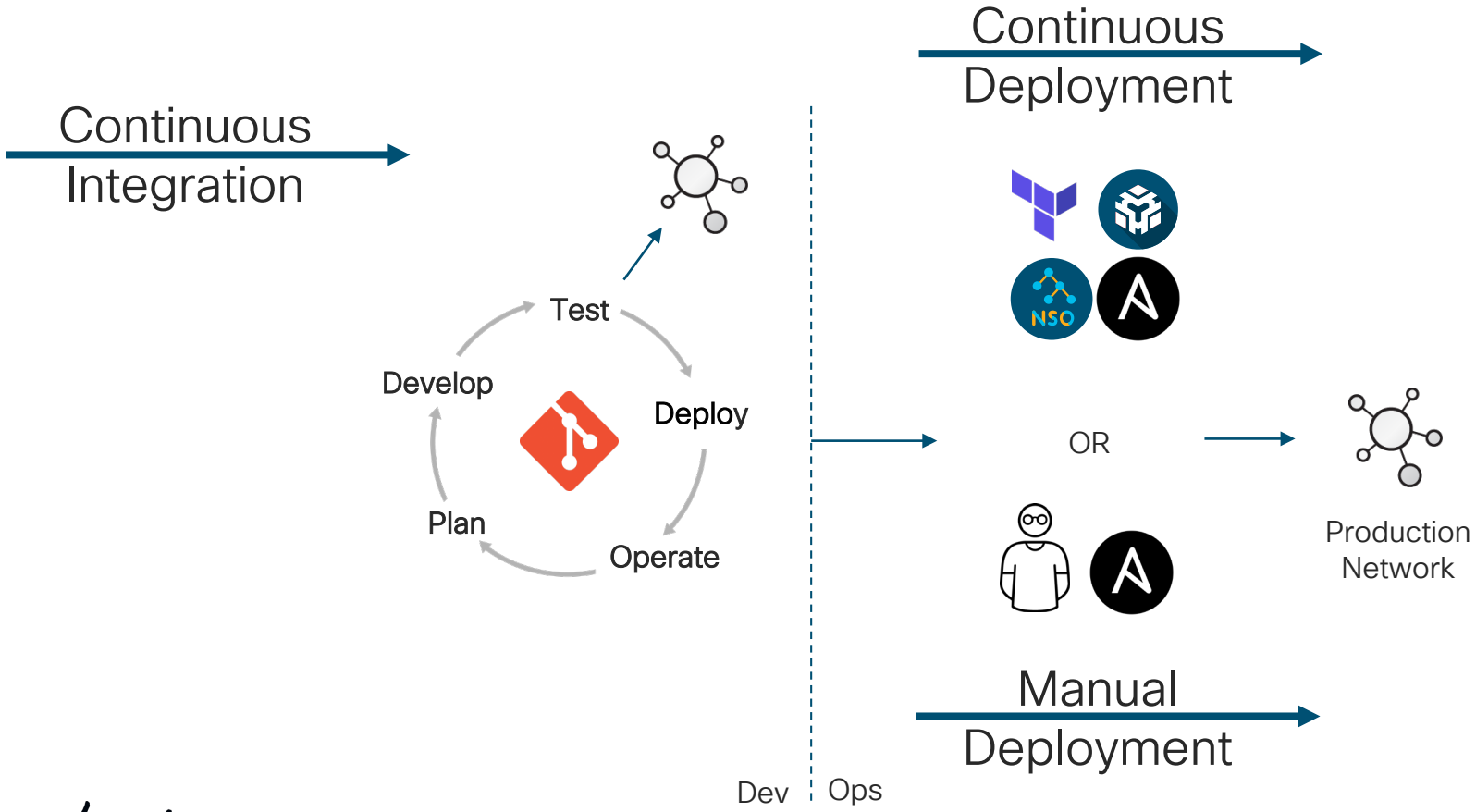
CD: Continuous Deployment of Features and Fixes



Goal: Accelerate Value to Customers



CI vs. CI/CD



Challenges

- It’s hard
- Requires a changes in mindset, organization, and skillset
- Examples never work
- Too many moving parts
- How do you test?
- Where do you test?

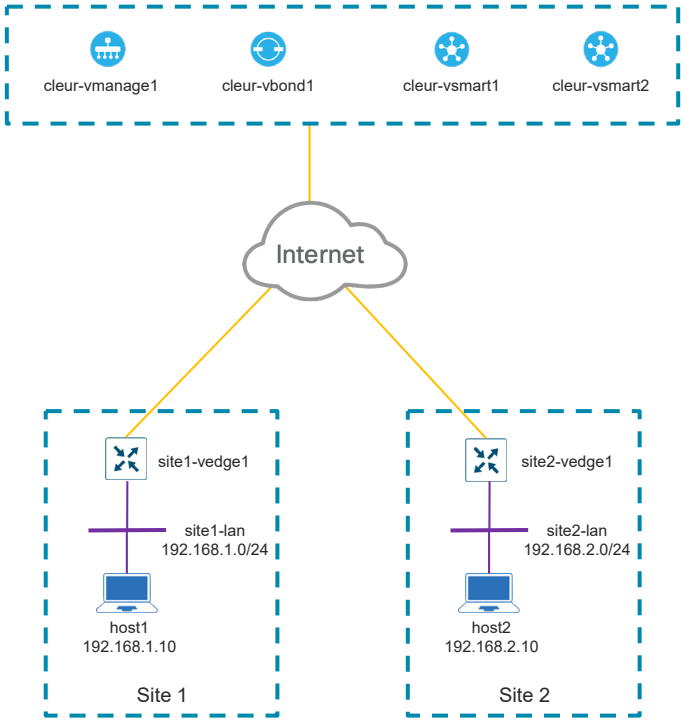
SDWAN-DevOps Repo

https://github.com/CiscoDevNet/sdwan-devops

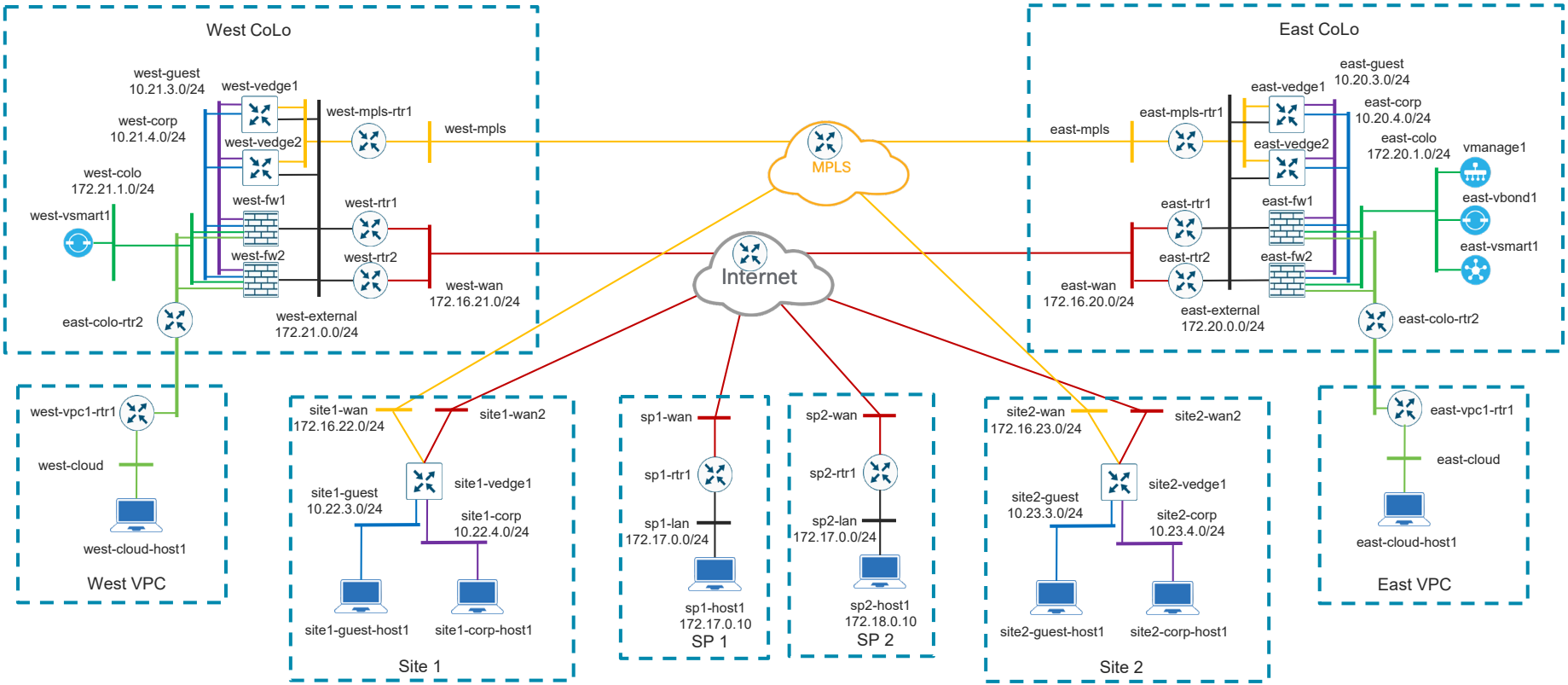
- **Architecture** - Detailed architecture with customer-relevant use cases
- **Simulation** - Dynamic VIRL topology in which to simulate architecture
- **Automation** - Automate deployment in simulated and production environments
- **Testing** - Automated validation or deployments
- **Cisco DevNet** - Learning Labs and Sandboxes to teach and experience the components of the DevOps Bundle

Architecture

Simplified SD-WAN architecture



Real-Life SD-WAN Architectures



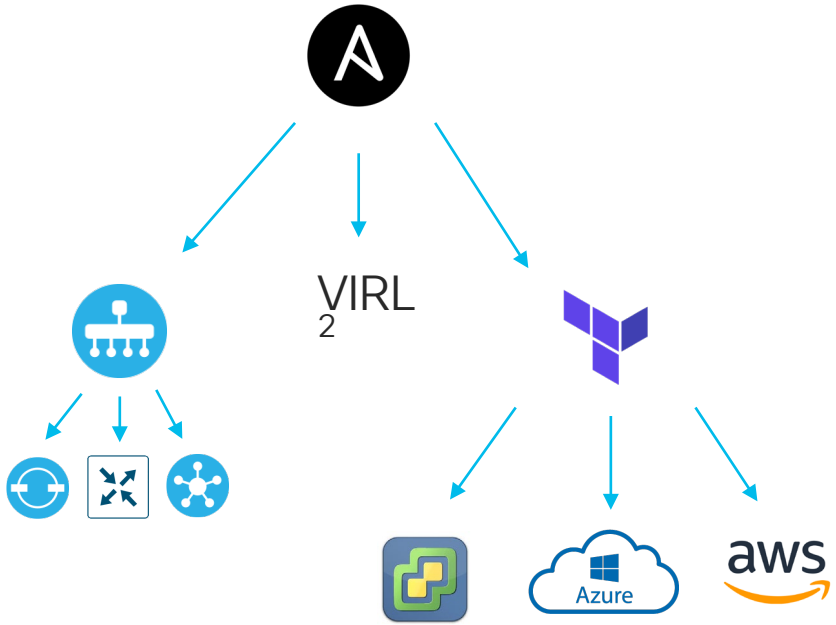
Automation

“If it does not have an API, it does not exist”

Mitchell Hashimoto

Ansible Uses: Overview

- Orchestrate Configuration and Testing
- Generate Day 0 Configuration
- Deliver Data Models:
  - to controllers when able
  - to devices when necessary
- Avoid CLI at all costs
- Test/Prod same where possible



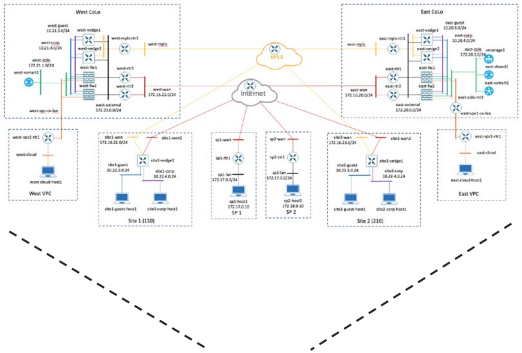
Ansible Uses: Roles

- ansible-viptela
  - Infrastructure agnostic deployment of SD-WAN control plane
  - Provision and configure SD-WAN edge
  - Full template lifecycle (import, export, add, delete, modify, attach, detach)
  - Full policy lifecycle (import, export, add, delete, modify, activate, deactivate)
- ansible-virl
  - Topology lifecycle (Dynamically generate, launch, clean)
  - Dynamic inventory

Inventory-Driven Deployment

```
all:
  vars:
    sdwan_vbond: 192.133.179.21
    pod_name: pod1
  children:
    sdwan:
      children:
        sdwan_control:
          vars:
            vpn0_portgroup: cpn-rtp-colab1
            vpn0_netmask: 24
            vpn0_gateway: 192.133.179.1
            vpn512_portgroup: cpn-rtp-colab1
          children:
            vmanage_hosts:
              vars:
                sdwan_personality: vmanage
                sdwan_device_model: vmanage
              hosts:
                pod1-vmanage1:
                  sdwan_system_ip: 1.1.1.1
                  sdwan_transport_ip: 192.133.179.20
                  ansible_host: "{{ sdwan_transport_ip }}"
                  sdwan_site_id: 1
...
```

- Day 0 - Deploy infrastructure
- Day 1 - Apply configurations
- Day 2 - Update configurations and deploy new edges



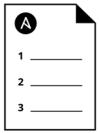
Azure aws

VIRL 2

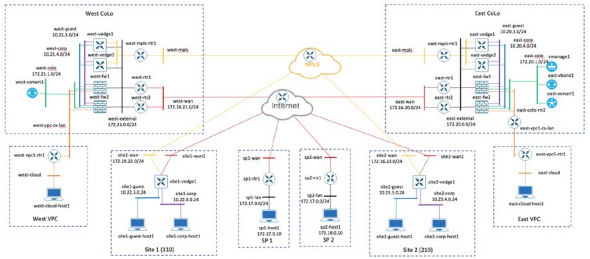
Builds topology

Inventory-Driven Configuration

```
interfaces:
  GigabitEthernet1:
    vrf: Mgmt-intf
    enabled: true
    ip:
      primary: dhcp
  GigabitEthernet2:
    enabled: true
    ip:
      primary: 172.16.21.2/24
  GigabitEthernet3:
    enabled: true
    ip:
      primary: 172.21.0.2/24
      standby:
        address: 172.21.0.1
        group: 0
        priority: 120
        delay: 300

static_routes:
  global:
    - network: 172.21.0.0/16
      fwd_list:
        - fwd: Null0

router:
  bgp:
    id: 65021
    log_neighbor_changes: true
    router_id: 172.16.21.2
    neighbors:
      - id: 172.16.21.1
        remote_as: 65016
      - id: 172.21.0.4
        remote_as: 65020
    address_family:
      global:
        ipv4:
          neighbors:
            - id: 172.16.21.1
              activate: true
            - id: 172.21.0.4
              activate: true
              next_hop_self: yes
          networks:
            - network: 172.21.0.0/16
          aggregate_address:
            - network: 172.21.0.0/16
              summary_only: yes
```



Playbooks define architecture, services and enforce compliance



Key/value pairs in inventory yield specific implementation

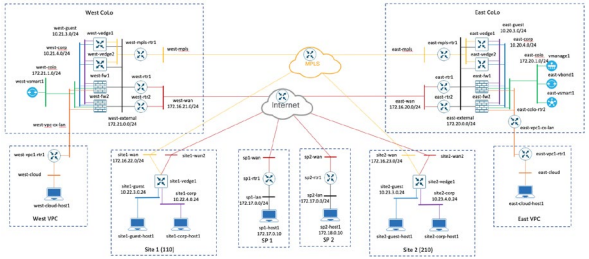
cisco Live!

Flexibility of Data-Driven Automation

```
interfaces:
  GigabitEthernet1:
    vrf: Mgmt-intf
    enabled: true
    ip:
      primary: dhcp
  GigabitEthernet2:
    enabled: true
    ip:
      primary: 172.16.21.2/24
  GigabitEthernet3:
    enabled: true
    ip:
      primary: 172.21.0.2/24
      standby:
        address: 172.21.0.1
        group: 0
        priority: 120
        delay: 300

static_routes:
  global:
    - network: 172.21.0.0/16
      fwd_list:
        - fwd: Null0

router:
  bgp:
    id: 65021
    log_neighbor_changes: true
    router_id: 172.16.21.2
    neighbors:
      - id: 172.16.21.1
        remote_as: 65016
      - id: 172.21.0.4
        remote_as: 65020
    address_family:
      global:
        ipv4:
          neighbors:
            - id: 172.16.21.1
              activate: true
            - id: 172.21.0.4
              activate: true
              next_hop_self: yes
          networks:
            - network: 172.21.0.0/16
          aggregate_address:
            - network: 172.21.0.0/16
              summary_only: yes
```

Jinja

- Encoding
- CLI
- XML
- JSON

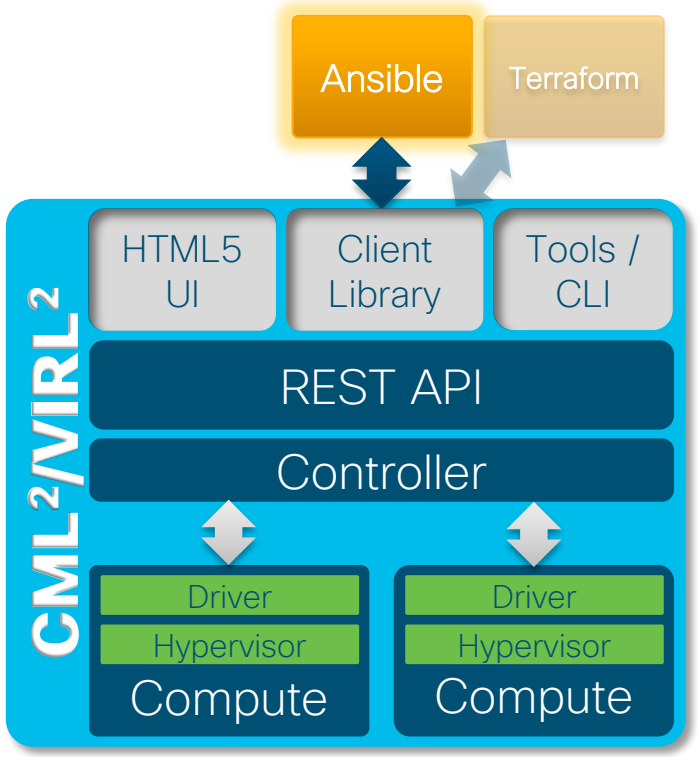


- Transport
- SSH
- NETCONF
- API

Simulation

CML^2 /VIRL^2 – Modern Network Simulation

- Solid REST API Foundation
- Modern UI
- Layered Architecture
- Scalability and Performance
- Persistent labs
- Easy to install and use
- Lightweight
- Rich Ecosystem



Inventory-Driven Topology

network.yml

```
vpn_instances:
  - vpn_id: 0
      interfaces:
        - if_name: eth1
          ip:
            address: 192.133.179.21/24
          tunnel_interface:
          enabled: true
      routes:
        - prefix: 0.0.0.0/0
next_hop:
address: 192.133.179.1
```

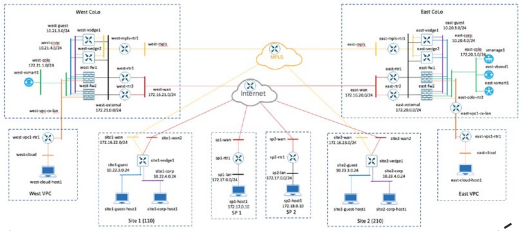
vmware.yml

```
vpn0_portgroup: cpn-rtp-colab1
vpn512_portgroup: cpn-rtp-colab1
servicevpn_portgroup: cpn-rtp-colab1
```

virl.yml

```
virl_config_template: sdwan/vedge.j2
virl_image_definition: "viptela-edge-19.2.1"
```

- Load topology information
- Generate Day0 Config
- Create topology



VIRL 2

Builds topology

Dynamic Inventory

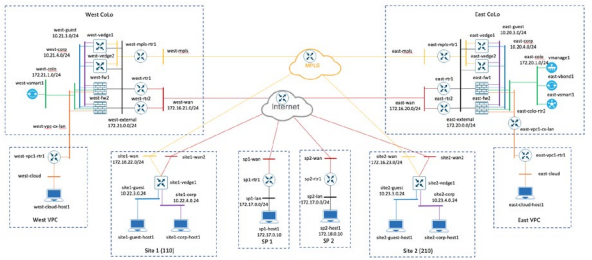
```
interfaces:
  GigabitEthernet1:
    vrf: Mgmt-intf
    enabled: true
    ip:
      primary: dhcp
  GigabitEthernet2:
    enabled: true
    ip:
      primary: 172.16.21.2/24
  GigabitEthernet3:
    enabled: true
    ip:
      primary: 172.21.0.2/24
      standby:
        address: 172.21.0.1
        group: 0
        priority: 120
        delay: 300

static_routes:
  global:
    - network: 172.21.0.0/16
      fwd_list:
        - fwd: Null0

router:
  bgp:
    id: 65021
    log_neighbor_changes: true
    router_id: 172.16.21.2
    neighbors:
      - id: 172.16.21.1
        remote_as: 65016
      - id: 172.21.0.4
        remote_as: 65020
    address_family:
      global:
        ipv4:
          neighbors:
            - id: 172.16.21.1
              activate: true
            - id: 172.21.0.4
              activate: true
              next_hop_self: yes
          networks:
            - network: 172.21.0.0/16
          aggregate_address:
            - network: 172.21.0.0/16
              summary_only: yes
```

VIRL 2

Mgmt IP from VIRL

Key/value pairs from inventory

Overrides production addressing



Test Workflow

Start Simulation

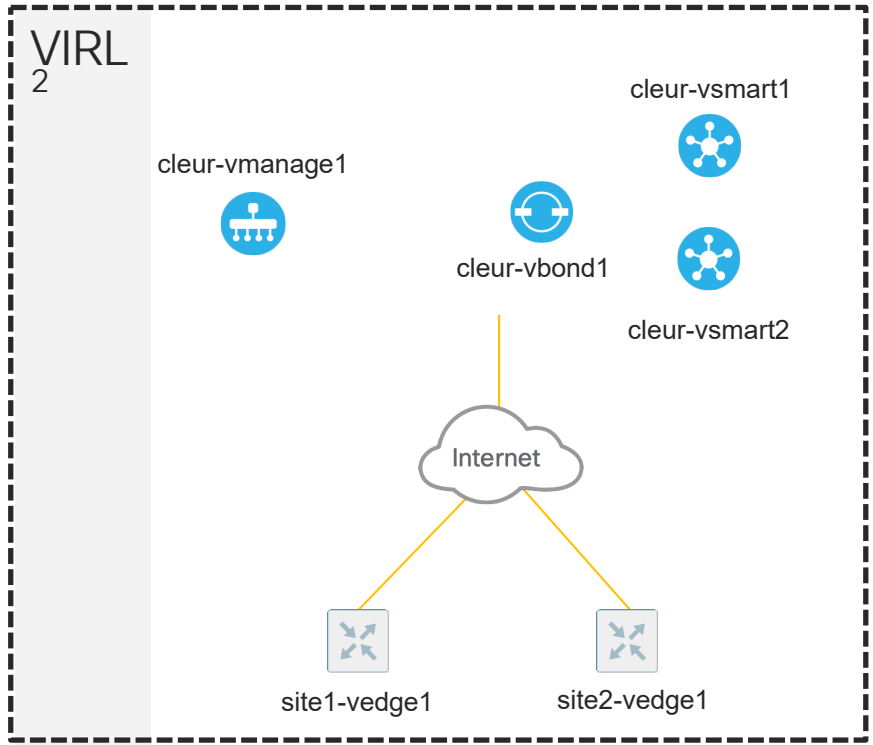
build-virl.yml

`virl_lab`

- Create Lab

`virl_node`

- Start network
- Start control



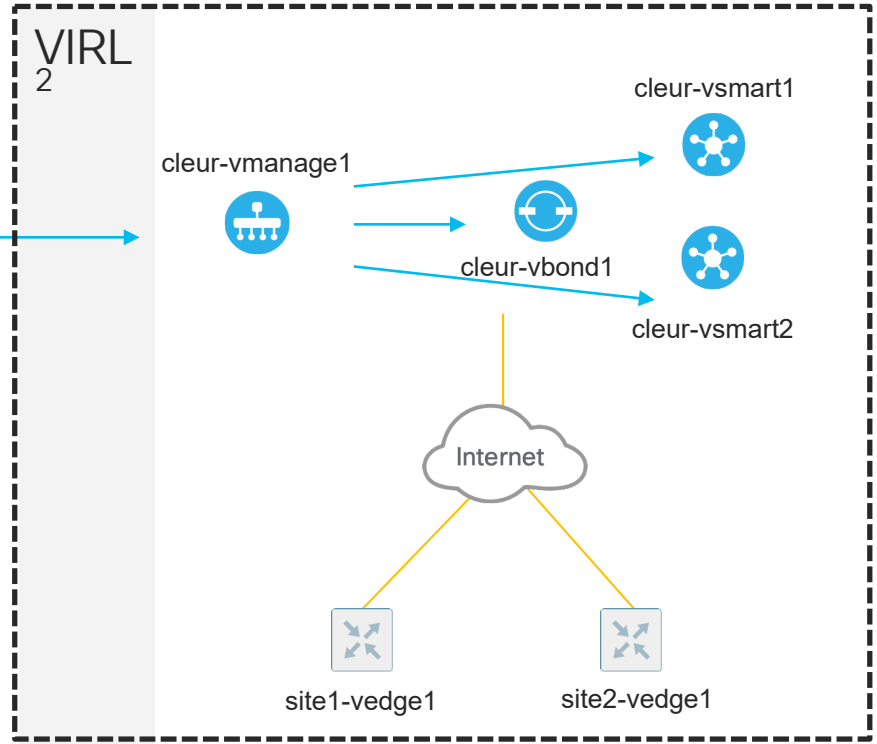
Configure SD-WAN Control Plane

build-virl.yml

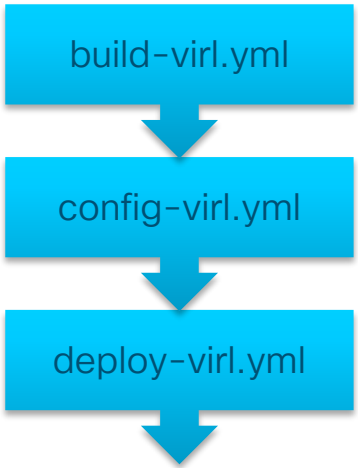
config-virl.yml

`vmanage_*`

- Config vmanage
- Add vbond/vsmart
- Sign Certs
- Load Templates
- Load Policy



Bootstrap and Start Edges

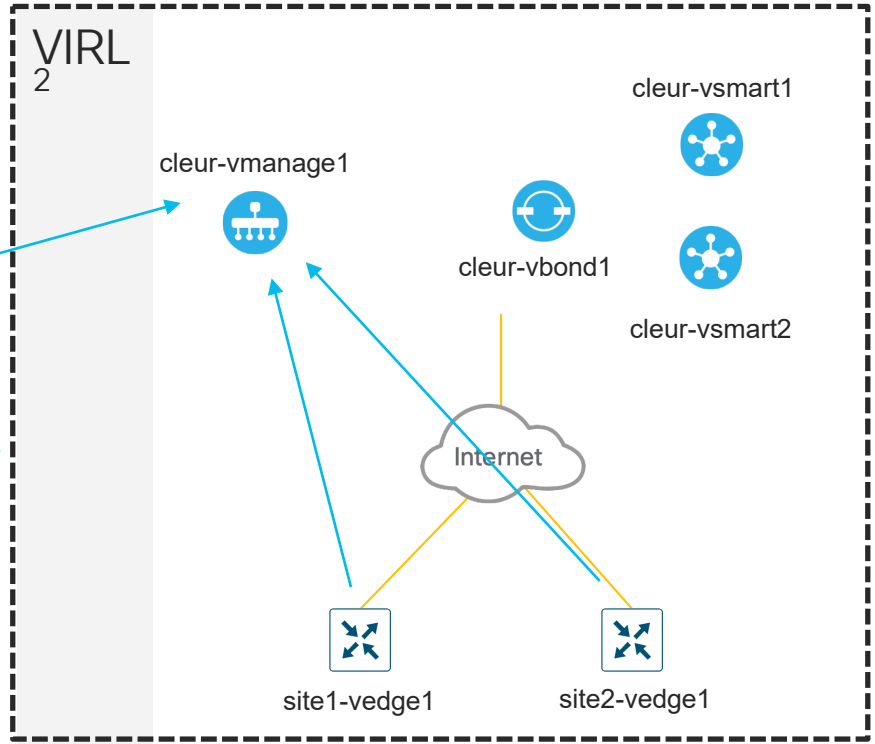


`vmanage_*`

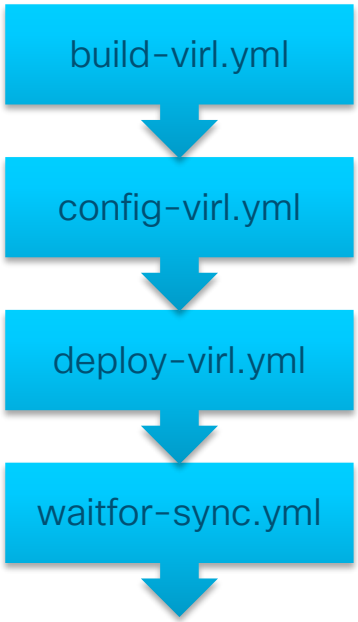
- Attach Templates
- Get Bootstrap

`virl_node`

- Start vEdges

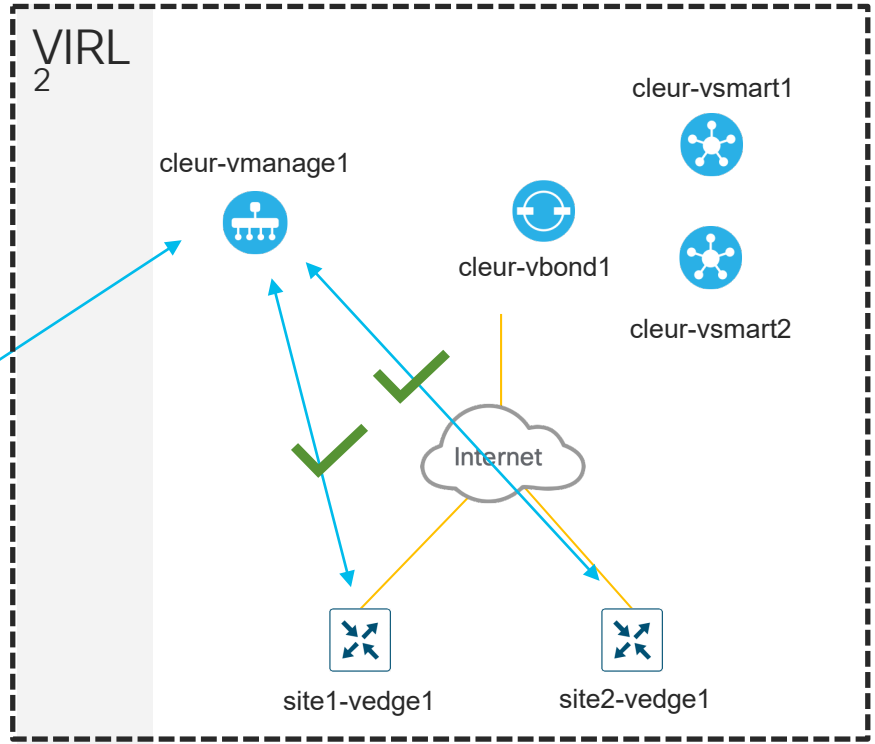


Wait for Devices to Sync

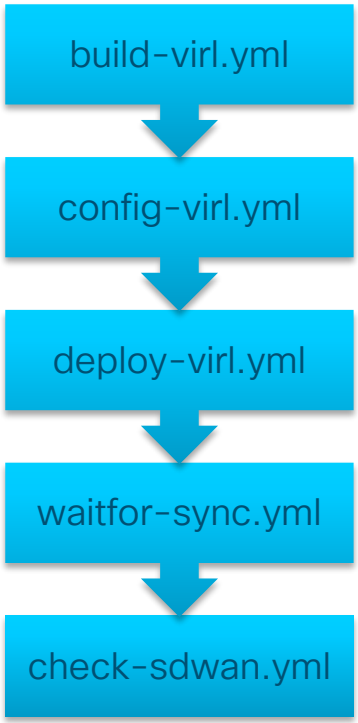


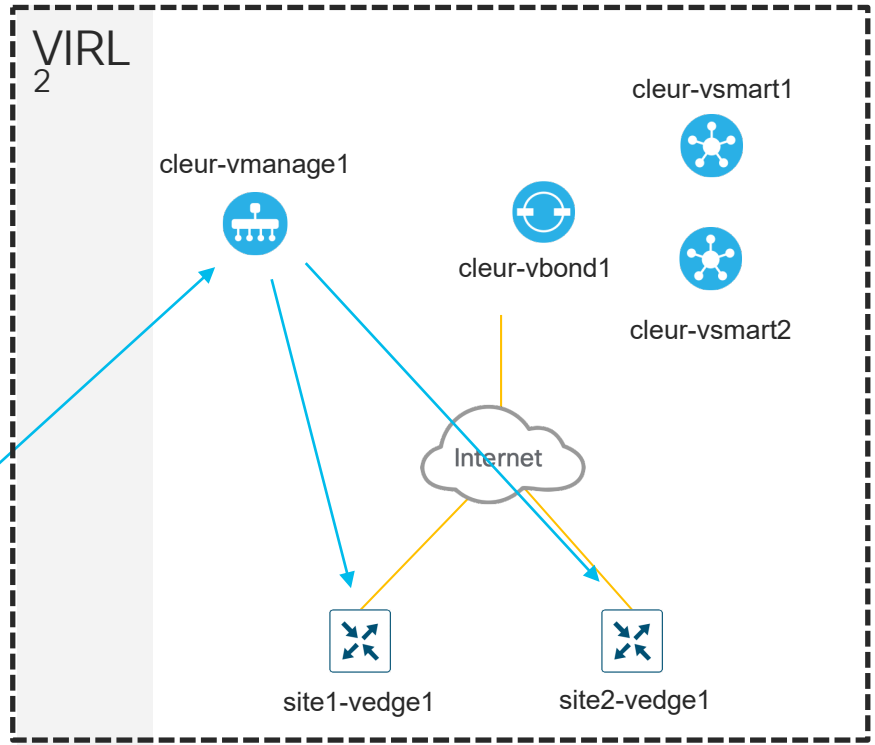
`vmanage_*`

- Poll State



Check for Proper Connectivity





`vmanage_*`

- Run tests

Testing

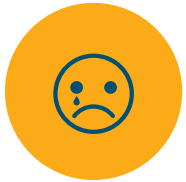
What can we test?

AUTOMATION CODE

CONFIGURATION

SOFTWARE/ FIRMWARE

CONNECTIVITY

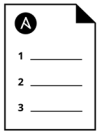


THROUGHPUT

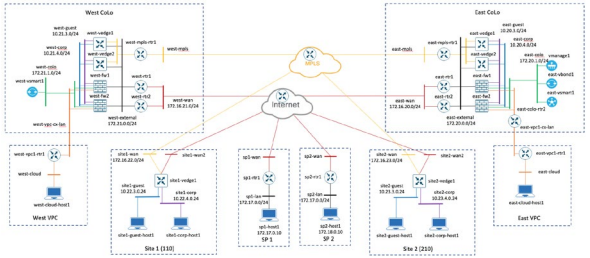
Inventory-Driven Configuration

```
interfaces:
  GigabitEthernet1:
    vrf: Mgmt-intf
    enabled: true
    ip:
      primary: dhcp
  GigabitEthernet2:
    enabled: true
    ip:
      primary: 172.16.21.2/24
  GigabitEthernet3:
    enabled: true
    ip:
      primary: 172.21.0.2/24
      standby:
        address: 172.21.0.1
        group: 0
        priority: 120
        delay: 300

static_routes:
  global:
    - network: 172.21.0.0/16
      fwd_list:
        - fwd: Null0

router:
  bgp:
    id: 65021
    log_neighbor_changes: true
    router_id: 172.16.21.2
    neighbors:
      - id: 172.16.21.1
        remote_as: 65016
      - id: 172.21.0.4
        remote_as: 65020
    address_family:
      global:
        ipv4:
          neighbors:
            - id: 172.16.21.1
              activate: true
            - id: 172.21.0.4
              activate: true
              next_hop_self: yes
          networks:
            - network: 172.21.0.0/16
          aggregate_address:
            - network: 172.21.0.0/16
              summary_only: yes
```

← Change this? →



Playbooks define architecture, services and enforce compliance



Key/value pairs in inventory yield specific implementation

Validate this

Jenkins

- Jenkinsfile defines (most) everything about the tests
- Launches tests in a docker container for consistent environment
  - Loads OS dependencies and Python dependencies (requirements.txt)
- Different Jenkinsfiles for different type of tests (e.g, full, partial, use-case specific)
- VIRL environment with static session ID per environment
- Resource locking
- Speed Kills – Lot's of retries

Validation

- **`check-sdwan.yml`**
  - Performs tests of SD-WAN to verify both connectivity and policy
  - Uses vmanage_nping for testing
- **`check-network.yml`**
  - Performs connectivity tests of the underlay
  - Uses pyATS to perform automated “stare-and-compare” of routing protocols

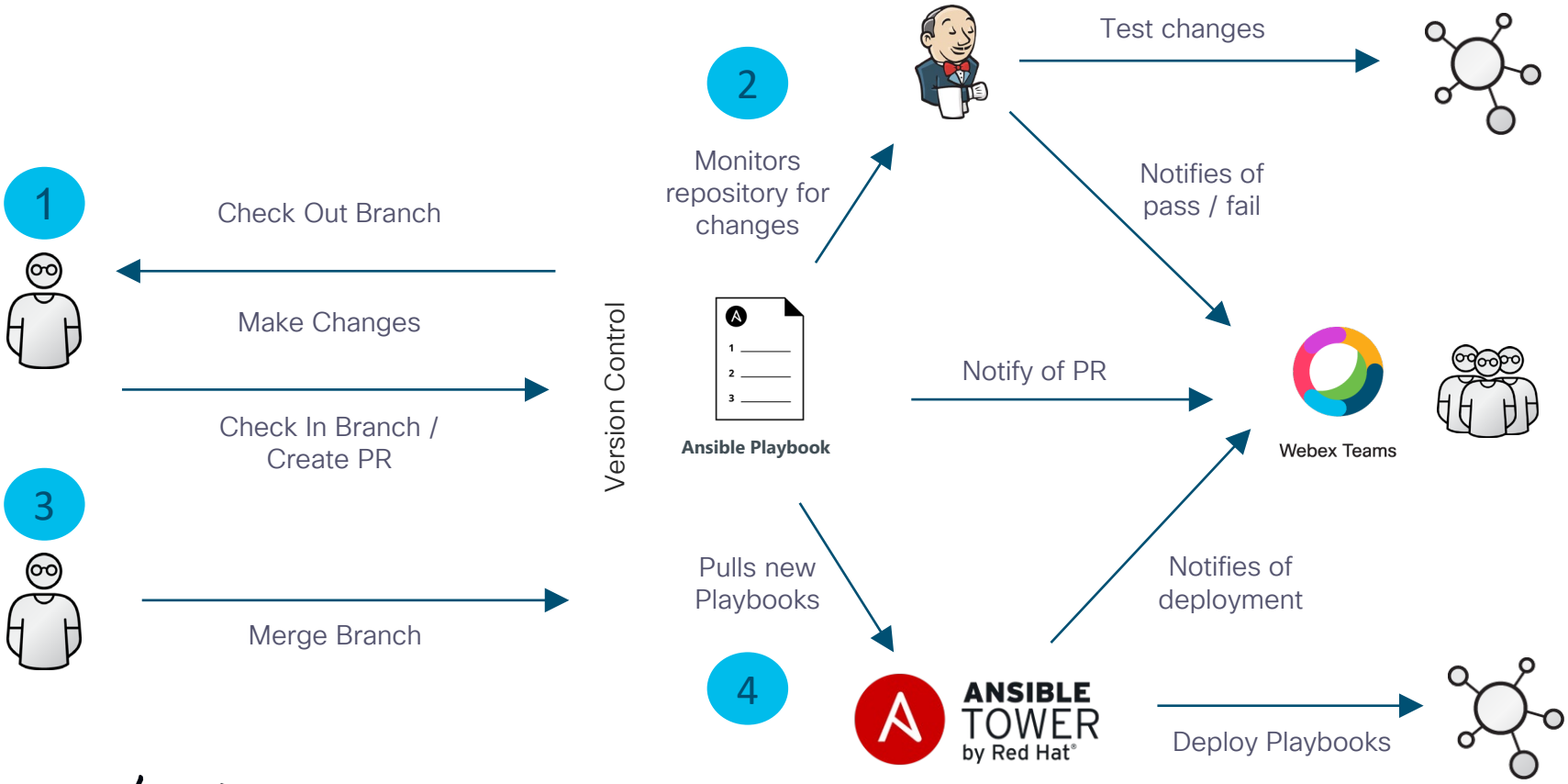
Test Strategies

"Just because you cannot test everything, it does not mean that you should not test anything!" -- Me

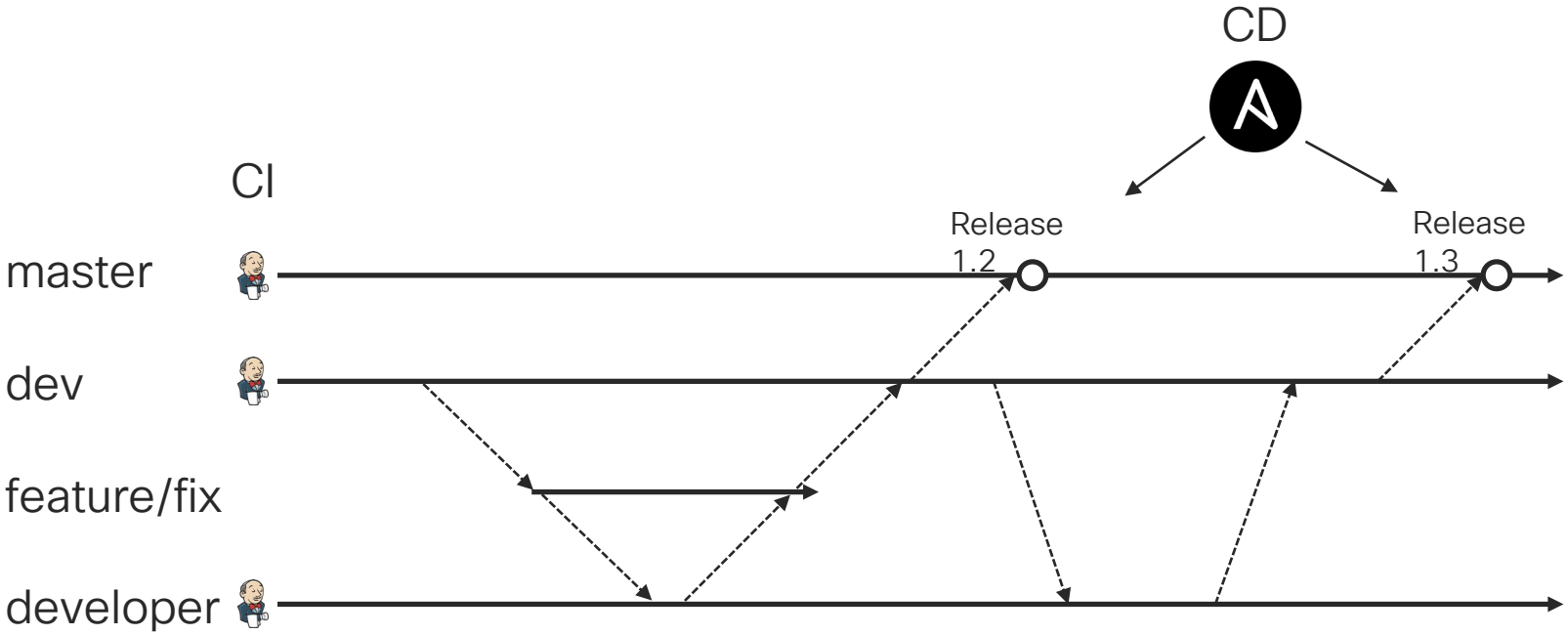
- Test the baseline policy, e.g.:
  - Site to HQ
  - Site to Site
- Test access to critical business functions, e.g.:
  - Site to app server in DC
  - Site to SaaS providers
- Test thing critical to security and/or compliance, e.g.:
  - AAA, Banners
  - Tenancy
  - Guest

DevOps

Network CI Workflow

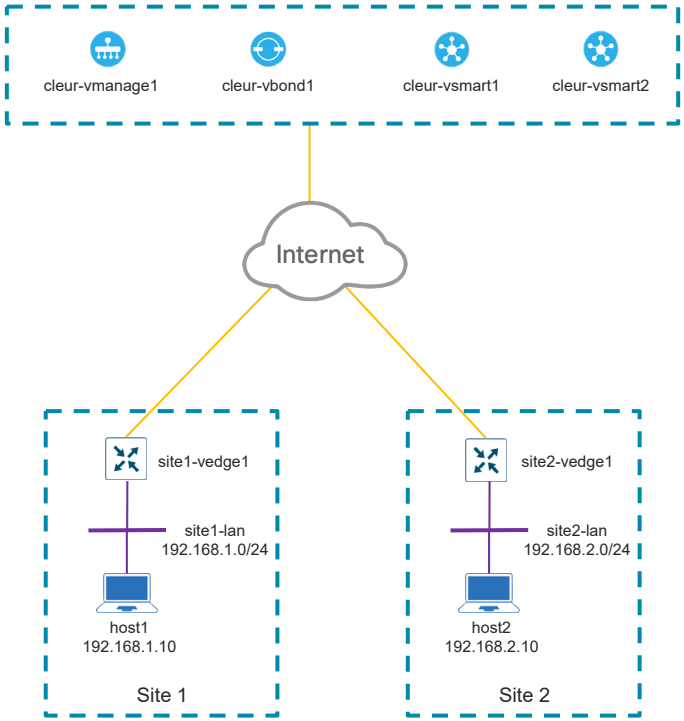


Branching Strategies



Demo

Demo Topology



Common Inventory

network.yml

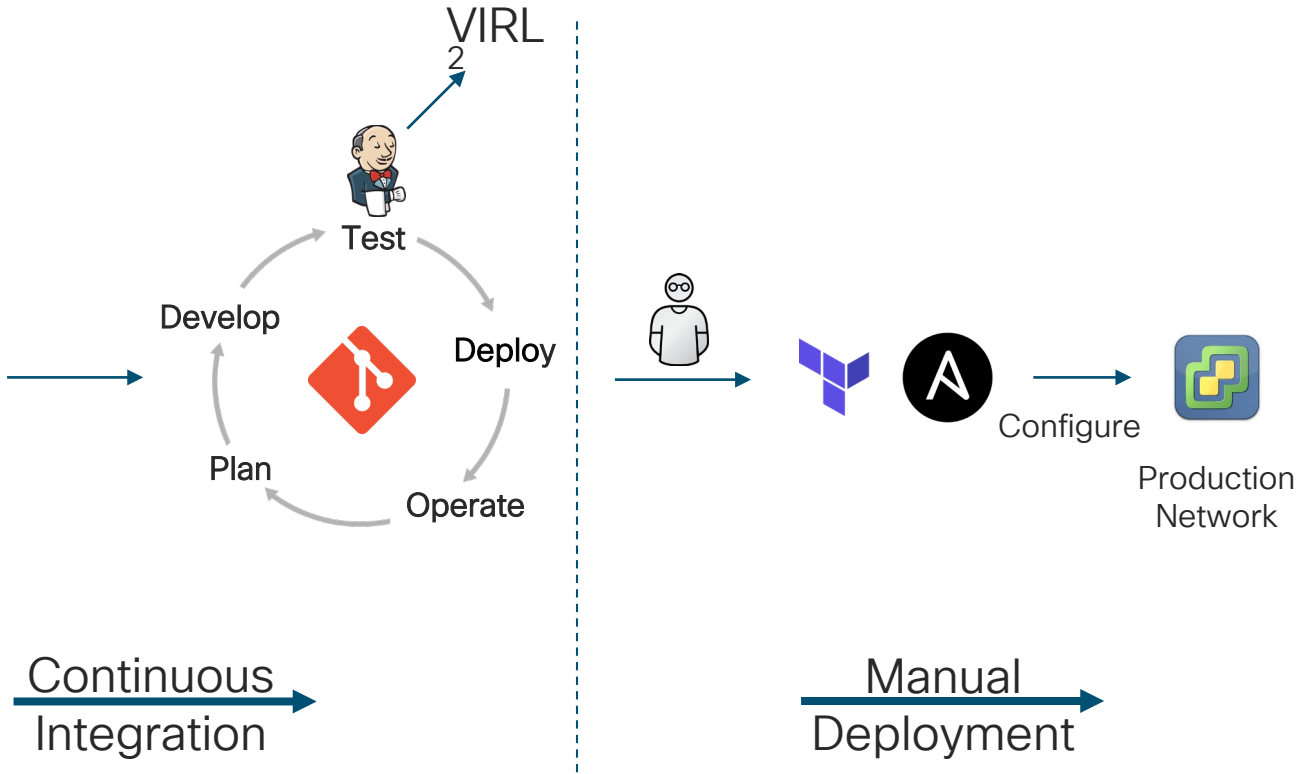
```
vpn_instances:
  - vpn_id: 0
      interfaces:
        - if_name: eth1
          ip:
            address: 192.133.179.21/24
          tunnel_interface:
          enabled: true
      routes:
        - prefix: 0.0.0.0/0
next_hop:
address: 192.133.179.1
```

vmware.yml

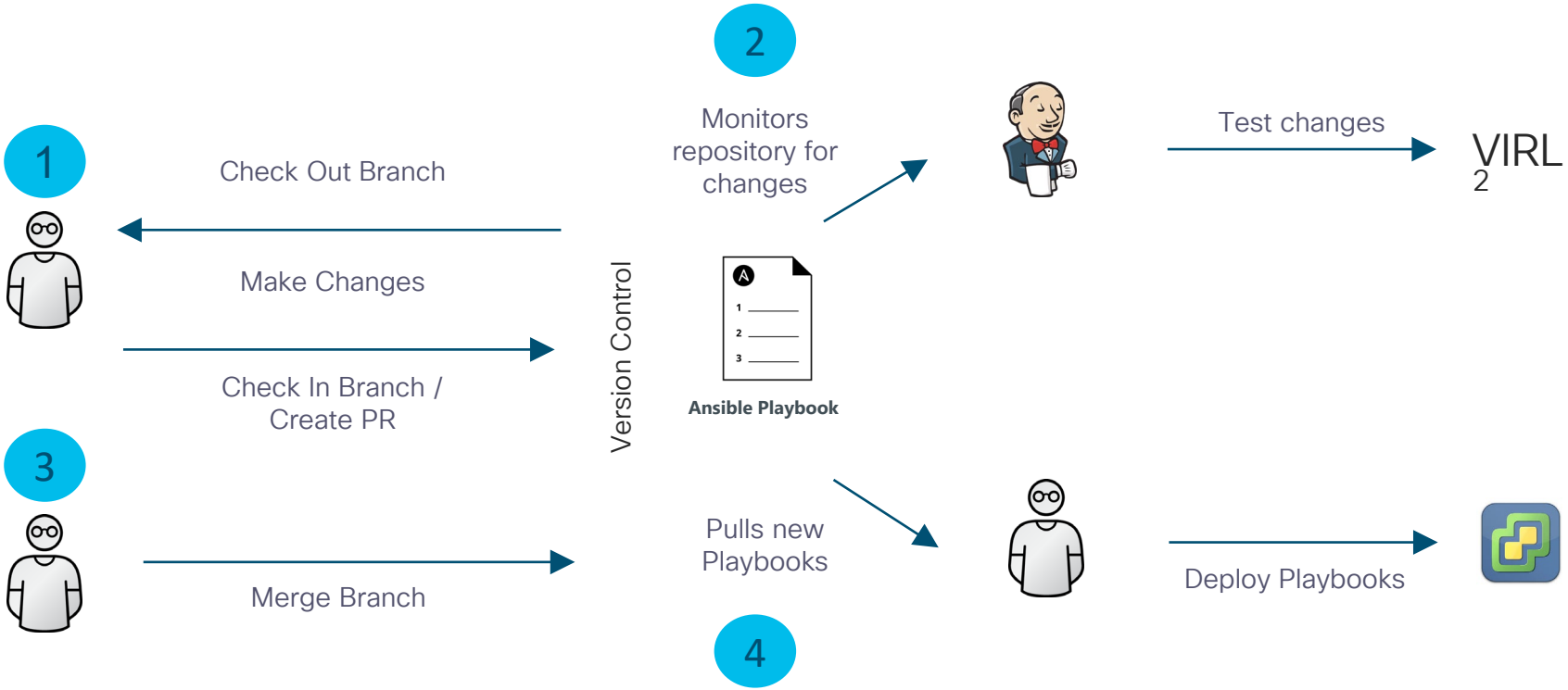
```
vpn0_portgroup: cpn-rtp-colab1
vpn512_portgroup: cpn-rtp-colab1
servicevpn_portgroup: cpn-rtp-colab1
```

virl.yml

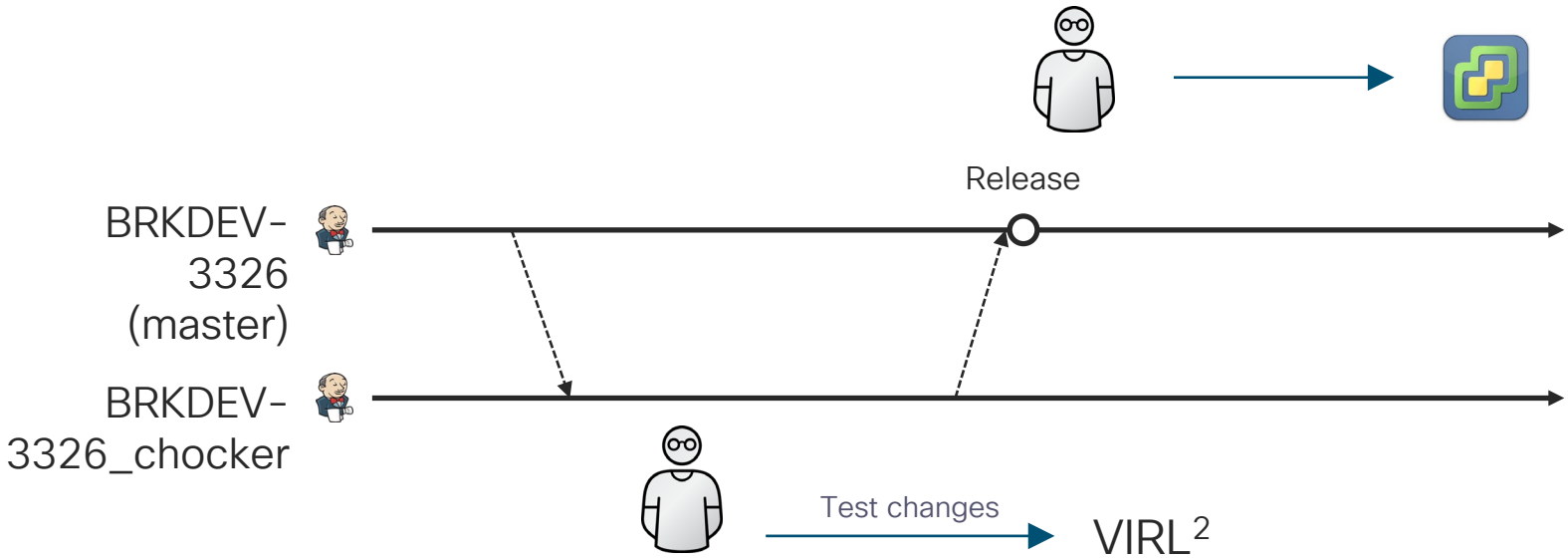
```
virl_config_template: sdwan/vedge.j2
virl_image_definition: "viptela-edge-19.2.1"
```



Demo Workflow



Branching Workflow



Conclusion

Not Ready for full CI/CD?

- Use simulated environment to test changes
- Use automation to provision/configure simulated environment
- Automate testing and validation of manual changes
- Use the Cisco DevNet resource to learn the different components

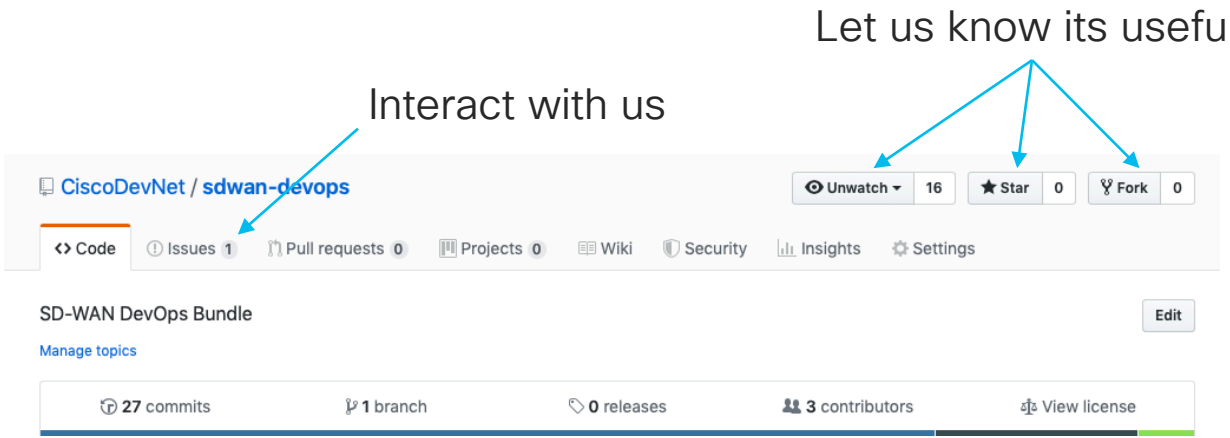
Summary

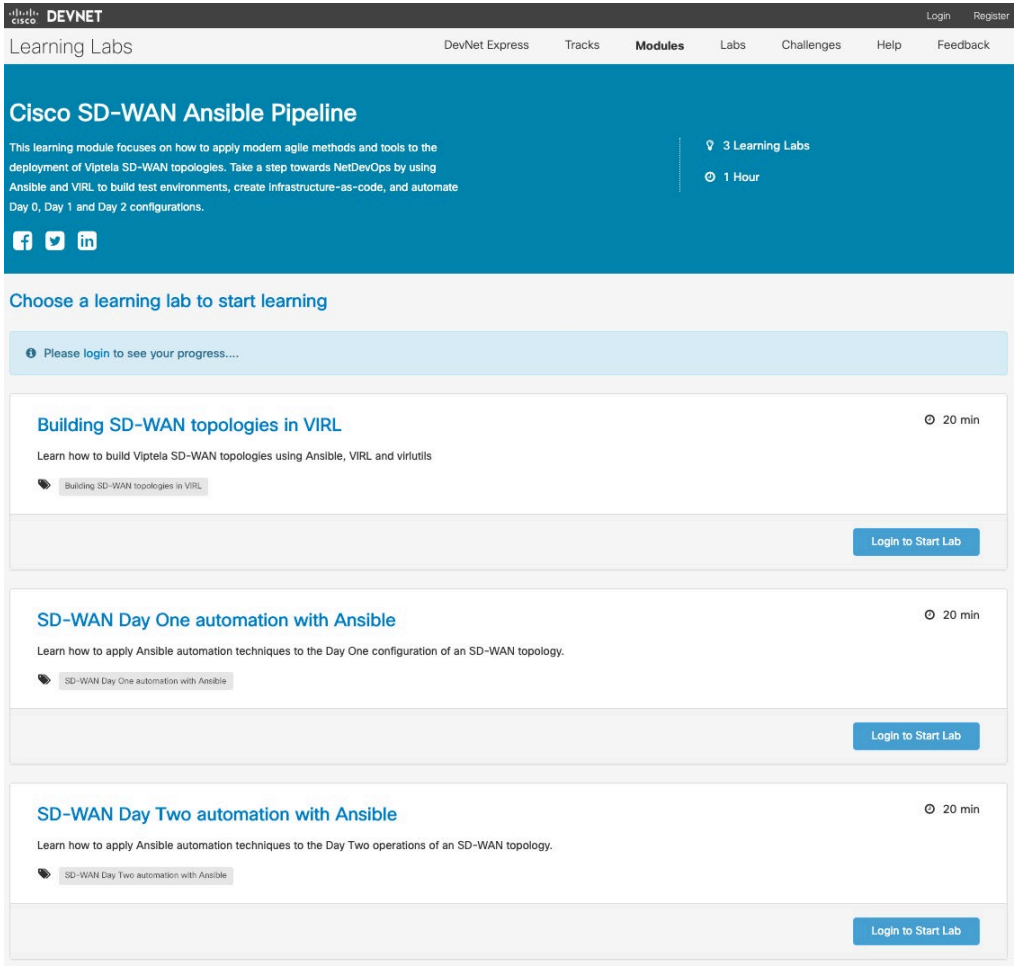
- It’s hard
- There is no one-size fits all
- Complete CI/CD solutions are needed
- No need to do it all at once

Try it out!

Github Repos:

- https://github.com/CiscoDevNet/sdwan-devops
- https://github.com/CiscoDevNet/ansible-(viptela, virl)
- https://github.com/CiscoDevNet/python-viptela





Complete your online session survey

- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (starting on Thursday) to receive your Cisco Live t-shirt.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Content Catalog on ciscolive.com/emea.

Cisco Live sessions will be available for viewing on demand after the event at ciscolive.com.

cisco Live!

Continue your education

Demos in the Cisco Showcase

Walk-In Labs

Meet the Engineer 1:1 meetings

Related sessions

Thank you

You make **possible**