

CISCO *Live!*

GO BEYOND

#CiscoLiveAPJC



DevOps Tools Fight for Relevance!

Adrian ILIESIU, Developer Advocate
@aidevnet
DEVNET-1280



#CiscoLiveAPJC

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 November 15, 2024.



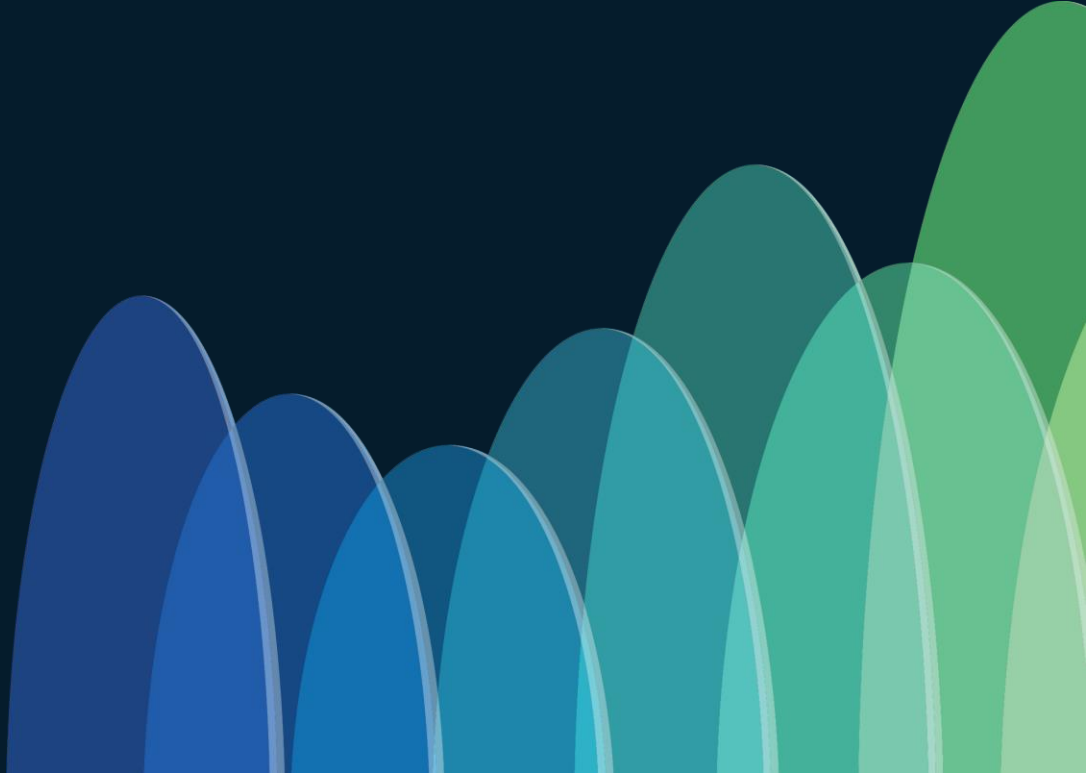
<https://ciscolive.ciscoevents.com/ciscolivebot/#DEVNET-1280>



Agenda

- What is Infrastructure as Code (IaC)?
- Ansible
 - Ansible Collections
 - Ansible Concepts
- Terraform
 - Terraform Concepts
- Ansible and Terraform Comparison
- Next Steps
- Resources

What is Infrastructure as Code (IaC)?

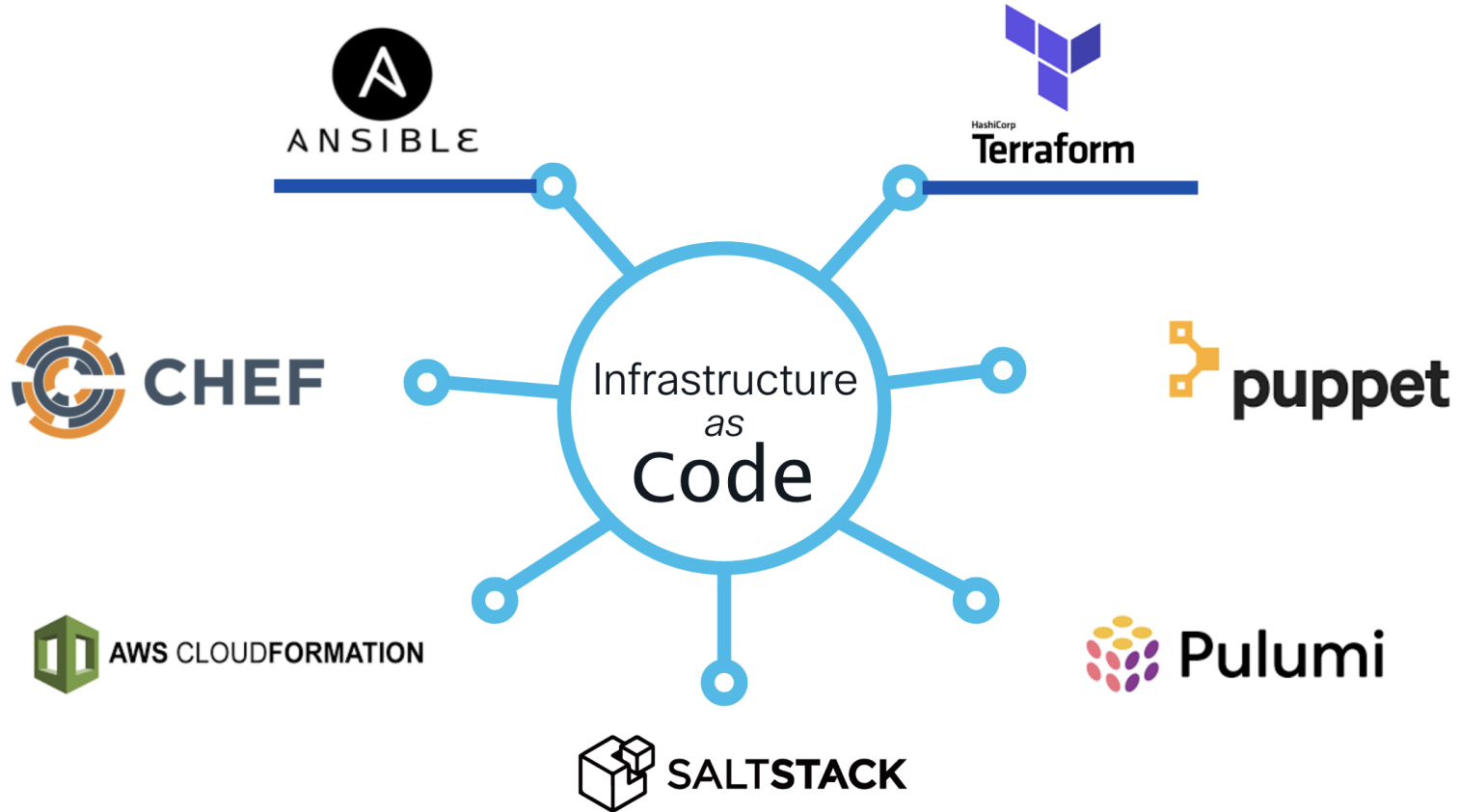


What is Infrastructure as Code (IaC)?

- Managing infrastructure can be tedious
- Network operators connect to devices and make changes to the configuration
 - CLI – “finger net”
 - Web browser – “point and click” aka “ClickOps”
- Most think of building/managing Cloud Infrastructure
- Define what the intended state of infrastructure should be
- Automation tools read & apply changes to devices to match the intended state

The management & provisioning of computer infrastructure through code and data structures instead of direct device management.

Infrastructure as Code Tools

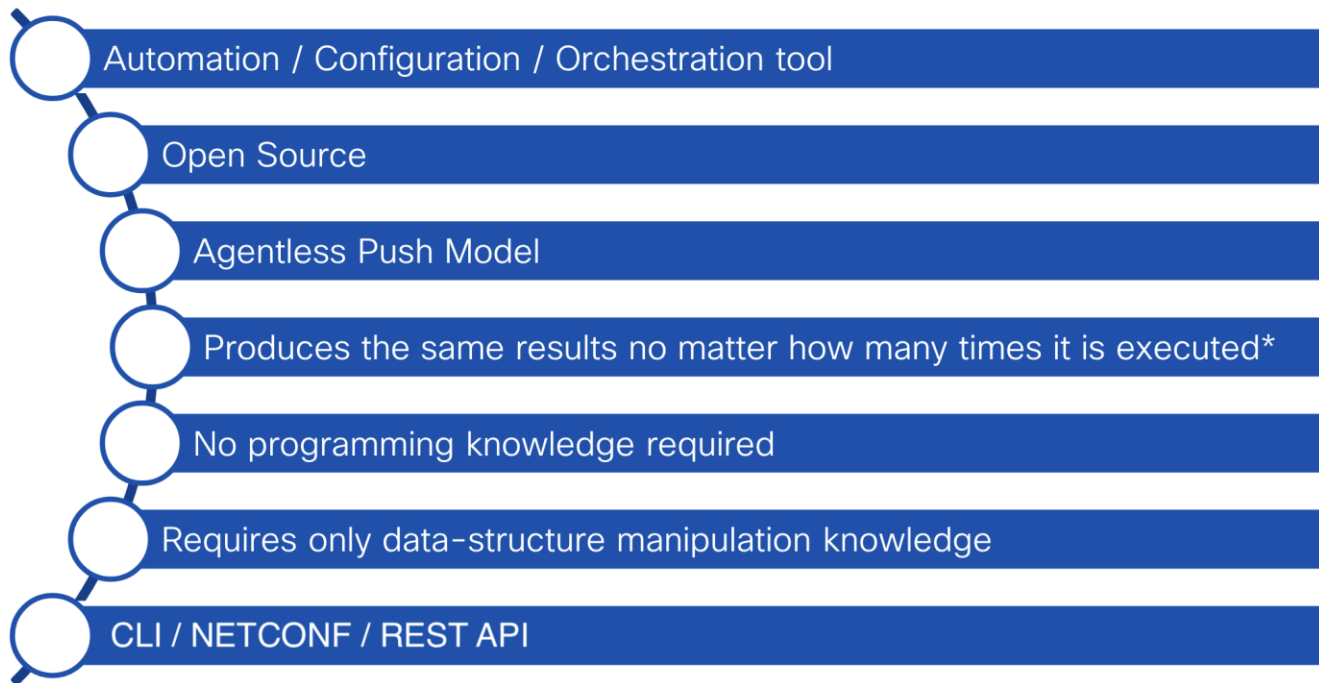


Ansible

CISCO *Live!*

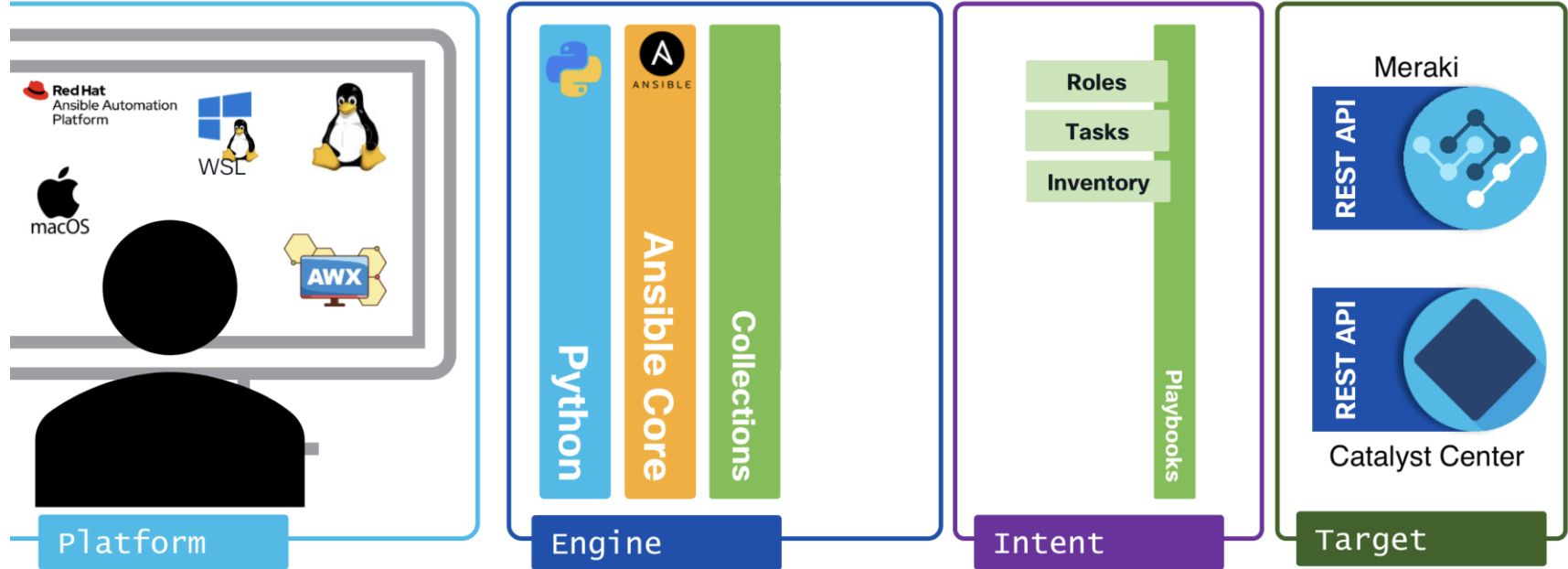


What is Ansible?



*idempotent

What makes up Ansible?



Installing Ansible

Python Virtual Environments



- You should use a **virtual environment**
- Virtual environments allow for installing Ansible inside a contained area with a specific version of Python
- Makes it possible to run different Python scripts that require different versions of Python and libraries

Ansible Install

Core or Everything

```
$ pip install ansible-core
```

- Ansible installs only the **core components**
- Collections must be installed separately
- Smaller footprint and more control
- Assures install of latest collections versions

```
$ pip install ansible
```

- Ansible installs **all collections**
- Complete package consumes much more disk space
- Might not install the latest version of the collection

https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html

Ansible Collections

What are Ansible collections?

- Introduced in Ansible 2.9
- Collections allow vendors to de-couple their Ansible capabilities from the core Ansible release schedule
- Uses Ansible Galaxy as delivery method
- Collections can be installed in any location with the `-p` flag

```
$ ansible-galaxy collection install cisco.meraki cisco.dnac cisco.ios
```

Installing Ansible Collections

Command

```
(ansible) parallels@ubuntu-linux-22-04-02-desktop:~/ansible_test$ ansible-galaxy collection install cisco.meraki cisco.dnac cisco.ios
Starting galaxy collection install process
Process install dependency map
Starting collection install process
Downloading https://galaxy.ansible.com/api/v3/plugin/ansible/content/published/collections/artifacts/cisco-ios-8.0.0.tar.gz to /home/parallels/.ansible/tmp/ansible-local-1779456tv5_m84/tmp09ihp5ue/cisco-ios-8.0.0-ebqocon1
Installing 'cisco.ios:8.0.0' to '/home/parallels/.ansible/collections/ansible_collections/cisco/ios'
Downloading https://galaxy.ansible.com/api/v3/plugin/ansible/content/published/collections/artifacts/ansible-netcommon-6.1.2.tar.gz to /home/parallels/.ansible/tmp/ansible-local-1779456tv5_m84/tmp09ihp5ue/ansible-netcommon-6.1.2-r63cft1t
cisco.ios:8.0.0 was installed successfully
Installing 'ansible.netcommon:6.1.2' to '/home/parallels/.ansible/collections/ansible_collections/ansible/netcommon'
Downloading https://galaxy.ansible.com/api/v3/plugin/ansible/content/published/collections/artifacts/ansible-utils-4.1.0.tar.gz to /home/parallels/.ansible/tmp/ansible-local-1779456tv5_m84/tmp09ihp5ue/ansible-utils-4.1.0-fokfejil
ansible.netcommon:6.1.2 was installed successfully
Installing 'ansible.utils:4.1.0' to '/home/parallels/.ansible/collections/ansible_collections/ansible/utils'
Downloading https://galaxy.ansible.com/api/v3/plugin/ansible/content/published/collections/artifacts/cisco-meraki-2.18.1.tar.gz to /home/parallels/.ansible/tmp/ansible-local-1779456tv5_m84/tmp09ihp5ue/cisco-meraki-2.18.1-ttazvknm
ansible.utils:4.1.0 was installed successfully
Installing 'cisco.meraki:2.18.1' to '/home/parallels/.ansible/collections/ansible_collections/cisco/meraki'
Downloading https://galaxy.ansible.com/api/v3/plugin/ansible/content/published/collections/artifacts/cisco-dnac-6.13.3.tar.gz to /home/parallels/.ansible/tmp/ansible-local-1779456tv5_m84/tmp09ihp5ue/cisco-dnac-6.13.3-vwe5zyjc
cisco.meraki:2.18.1 was installed successfully
Installing 'cisco.dnac:6.13.3' to '/home/parallels/.ansible/collections/ansible_collections/cisco/dnac'
cisco.dnac:6.13.3 was installed successfully
```

Required packages

Ansible Catalyst Center

Collection Modules

- Primary reason they are called collections is because they are a collection of modules
- Modules perform a specific task like create templates, configure interfaces, etc.
- Actively maintained with regular cadence that increases module count and capabilities

A Ansible Community Documentation

☰ Collections in the Cisco Namespace

Cisco.Aci

Cisco.Asa

☰ Cisco.Dnac

Description

Plugin Index

Cisco.Intersight

Cisco.Ios

Cisco.Iosxr

Cisco.Ise

Cisco.Meraki

Cisco.Mso

Cisco.Nxos

Cisco.Ucs

Collections in the Cloud Namespace

Collections in the Cloudscale_ch Namespace

Collections in the Community Namespace

Collections in the Containers Namespace

Collections in the Cyberark Namespace

Collections in the Dellmc Namespace

Collections in the F5networks Namespace

Collections in the Fortinet Namespace

Modules

- `accesspoint_configuration_details_by_task_id_info` module – Infor
- `app_policy_default_info` module – Information module for App Po
- `app_policy_info` module – Information module for App Policy
- `app_policy_intent_create` module – Resource module for App Poli
- `app_policy_queuing_profile` module – Resource module for App P
- `app_policy_queuing_profile_count_info` module – Information mo
- `app_policy_queuing_profile_info` module – Information module fo
- `application_sets` module – Resource module for Application Sets
- `application_sets_count_info` module – Information module for Apj
- `application_sets_info` module – Information module for Applicatio
- `applications` module – Resource module for Applications
- `applications_count_info` module – Information module for Applica
- `applications_health_info` module – Information module for Applic
- `applications_info` module – Information module for Applications
- `assign_device_to_site` module – Resource module for Assign Devi
- `associate_site_to_network_profile` module – Resource module for
- `authentication_import_certificate` module – Resource module for
- `authentication_import_certificate_p12` module – Resource modul
- `authentication_policy_servers_info` module – Information module
- `buildings_planned_access_points_info` module – Information mod
- `business_sda_hostonboarding_ssid_ippool` module – Resource mc
- `business_sda_hostonboarding_ssid_ippool_info` module – Informa
- `business_sda_virtual_network_summary_info` module – Informati
- `business_sda_wireless_controller_create` module – Resource mod
- `business_sda_wireless_controller_delete` module – Resource mod
- `cli_credential` module – Resource module for Cli Credential
- `client_detail_info` module – Information module for Client Detail

<https://docs.ansible.com/ansible/latest/collections/cisco/dnac/index.html>

Ansible Cisco Meraki

Collection Modules (CLI)

Use the CLI to access the module documentation

Grep can be used to filter through all the documentation installed

```
parallels:ansible_test$ ansible-doc -l | grep cisco.meraki.devices
cisco.meraki.devices
cisco.meraki.devices_appliance_performance_info
cisco.meraki.devices_appliance_radio_settings
cisco.meraki.devices_appliance_radio_settings_info
cisco.meraki.devices_appliance_uplinks_settings
cisco.meraki.devices_appliance_uplinks_settings_info
cisco.meraki.devices_appliance_vmx_authentication_token
cisco.meraki.devices_blink_leds
```

The command: `ansible-doc <module_name>` will present the CLI version of the doc. Matches what is on the web.

```
parallels:ansible_test$ ansible-doc cisco.meraki.devices
+ CISCO.MERAKI.DEVICES (/home/parallels/.ansible/collections/ansible_collect
    Manage operation update of the resource devices. Update the
    attributes of a device.

ADDED IN: version 2.16.0 of cisco.meraki

* note: This module has a corresponding action plugin.

OPTIONS (= is mandatory):

- address
  The address of a device.
  default: null
  type: str

- floorPlanId
  The floor plan to associate to this device. Null disassociates
  the device from the floorplan.
  default: null
  type: str

- lat
  The latitude of a device.
  default: null
  type: float

- lng
  The longitude of a device.
  default: null
  type: float

- meraki_action_batch_retry_wait_time
  meraki_action_batch_retry_wait time (integer), action batch
  concurrency error retry wait time
  default: 60
  type: int

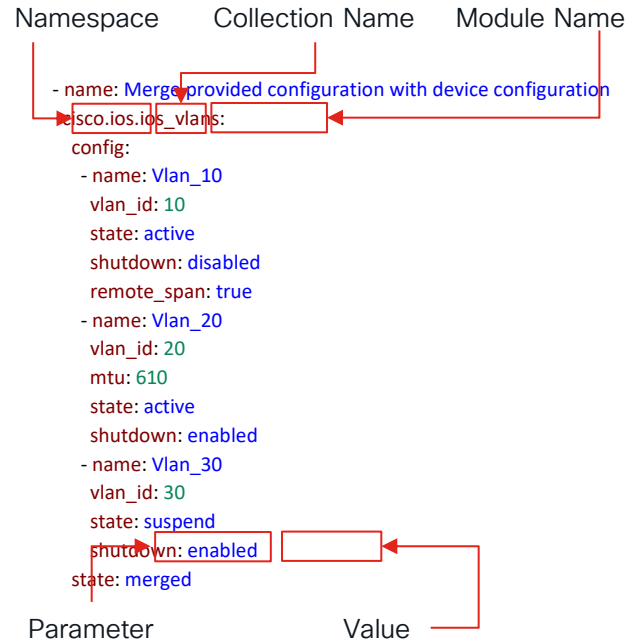
= meraki_api_key
  meraki_api key (string), API key generated in dashboard; can
```

<https://docs.ansible.com/ansible/latest/collections/cisco/meraki/index.html>

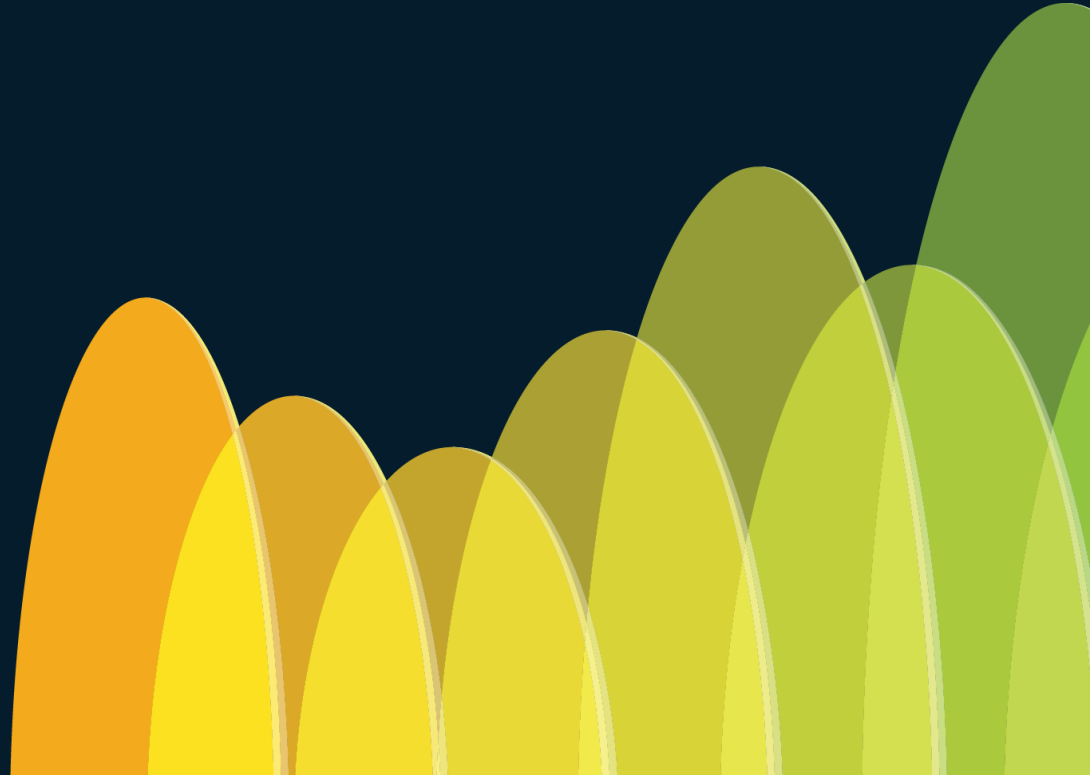
Modules

Used by tasks

- Always use the fully qualified name for the module.
- The modules require values assigned to the parameters that define how you wish to configure IOS XE.
- Documentation provides details as to default values and required values.
- No programming knowledge required. Just data structure build out.



Ansible concepts



Ansible Directory Structure

Best Practice for Growth!



Ansible Data Structures

Yet Another Markup Language

- Human readable data serialization language
- Used in playbooks and inventory files
- Best practice is to use a software focused text editor (e.g. Notepad++) or IDE (e.g. Visual Studio Code) with language assistant support of YAML data structures
- Indentation is very important and a proper editor will simplify this for you



Microsoft VSCode



PyCharm



Notepad++

Ansible Roles

How to do it!

- Roles are content directories that are structured in a conventional way to enable simple reuse
- Roles let you automatically load related vars, files, tasks, handlers and other Ansible artifacts based on a known file structure
- This allows for better data organization in the repository
- You use roles to combine tasks to complete an objective

```
parallels:ansible_test$ ansible-galaxy init ospf_config
- Role ospf_config was created successfully
parallels:ansible_test$ ls -lR
.:
total 4
drwxrwxr-x 10 parallels parallels 4096 May 25 15:39 ospf_config

./ospf_config:
total 36
-rw-rw-r-- 1 parallels parallels 1328 May 25 15:39 README.md
drwxrwxr-x 2 parallels parallels 4096 May 25 15:39 defaults
drwxrwxr-x 2 parallels parallels 4096 May 25 15:39 files
drwxrwxr-x 2 parallels parallels 4096 May 25 15:39 handlers
drwxrwxr-x 2 parallels parallels 4096 May 25 15:39 meta
drwxrwxr-x 2 parallels parallels 4096 May 25 15:39 tasks
drwxrwxr-x 2 parallels parallels 4096 May 25 15:39 templates
drwxrwxr-x 2 parallels parallels 4096 May 25 15:39 tests
drwxrwxr-x 2 parallels parallels 4096 May 25 15:39 vars

./ospf_config/defaults:
total 4
-rw-rw-r-- 1 parallels parallels 36 May 25 15:39 main.yml

./ospf_config/files:
total 0

./ospf_config/handlers:
total 4
-rw-rw-r-- 1 parallels parallels 36 May 25 15:39 main.yml

./ospf_config/meta:
total 4
-rw-rw-r-- 1 parallels parallels 1635 May 25 15:39 main.yml

./ospf_config/tasks:
total 4
-rw-rw-r-- 1 parallels parallels 33 May 25 15:39 main.yml

./ospf_config/templates:
total 0

./ospf_config/tests:
total 8
-rw-rw-r-- 1 parallels parallels 11 May 25 15:39 inventory
-rw-rw-r-- 1 parallels parallels 70 May 25 15:39 test.yml

./ospf_config/vars:
total 4
-rw-rw-r-- 1 parallels parallels 32 May 25 15:39 main.yml
```

```
$ ansible-galaxy init <role-name>
```

Ansible Playbooks

What to do!

- Playbooks define the set of actions that you want Ansible to complete
- Can contain specific tasks or reference roles that contain the tasks
 - Best practice is to use roles!

```
- name: Configure OSPF on IOS XE example  
  hosts: iosxe  
  gather_facts: false
```

roles:

```
- roles/ospf_config
```

Ansible Inventory

Who to do it to!

- Ansible inventory allows you to build data structures that correlate host specific variables
- Allows for grouping and variable inheritance
- Two formats are common: INI and YAML. Best practice is to use YAML.

[iosxe]

10.10.20.175

10.10.20.176

[iosxe:vars]

ansible_network_os=ios

ansible_user=cisco

ansible_password=cisco

ansible_connection=network_cli

Using Environment Variables

- Instead of inserting credentials that are very difficult to remove from a version control system (git) you can use environment variables.
- Set environment variables before ansible-playbook execution

```
[iosxe]
```

```
10.10.20.175
```

```
10.10.20.176
```

```
[iosxe:vars]
```

```
ansible_network_os=ios
```

```
ansible_user='{{ lookup("env", "IOSXE_USERNAME") }}'
```

```
ansible_password='{{ lookup("env", "IOSXE_PASSWORD") }}'
```

```
ansible_connection=network_cli
```

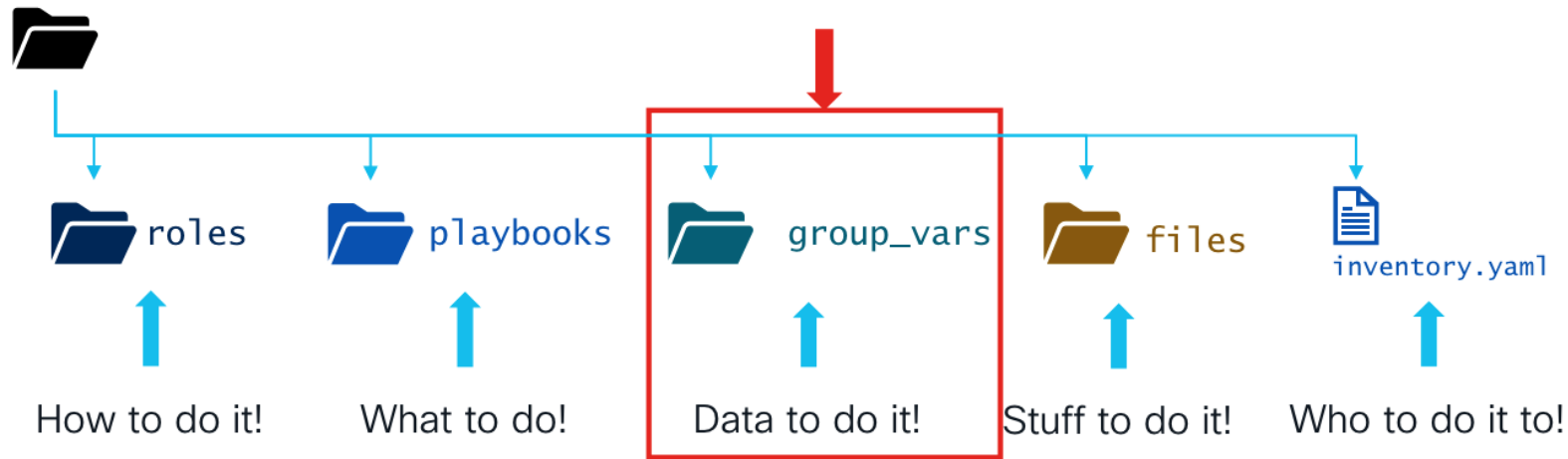
```
export IOSXE_USERNAME=cisco
```

```
export IOSXE_PASSWORD=cisco
```

Better variables

Placement matters!

- Including the variables with the role can result in variable duplication
- A better approach is to move the variables to a location that can be structured with the inventory for better organization



Ansible Variable Precedence

Placement matters!

- Ansible provides variable precedence, which is important when you build your data structure.
- This allows for having default behavior that is then changed by just including the variable in higher precedence.
- Using the **group_vars** folder tied to inventory is very useful.



Higher

- extra vars via CLI (for example, `-e "user=my_user"`)
- include params
- role (and include_role) params
- set_facts / registered vars
- include_vars
- task vars (only for the task)
- block vars (only for tasks in block)
- role vars (defined in `role/vars/main.yml`)
- play vars_files
- play vars_prompt
- play vars
- host facts / cached set_facts
- playbook host_vars/*
- inventory host_vars/*
- inventory file or script host vars
- playbook group_vars/*
- inventory group_vars/*
- inventory group_vars/all
- inventory file or script group vars
- role defaults (defined in `role/defaults/main.yml`)
- command line values (for example, `-u my_user`, these are not variables)

Lower



https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_variables.html

Putting it all together

Ansible-playbook -i inventory/hosts-file playbooks/config_ospf.yaml

playbooks

```
---
- name: Configure OSPF on IOS XE example
  hosts: iosxe
  gather_facts: false

  roles:
    - roles/ospf_config
```

inventory

```
[iosxe]
10.10.20.175
10.10.20.176

[iosxe:vars]
ansible_network_os=ios
ansible_user='{{ lookup("env", "IOSXE_USERNAME") }}'
ansible_password='{{ lookup("env", "IOSXE_PASSWORD") }}'
ansible_connection=network_cli
```

group_vars

```
---
configure_ospf:
  - ospf_process_id: 1
    ospf_network_address: 172.16.252.0
    ospf_wildcard_bits: 0.0.3.255
    ospf_area: 0
  - ospf_process_id: 1
    ospf_network_address: 192.168.0.0
    ospf_wildcard_bits: 0.0.255.255
    ospf_area: 0
```

roles

```
---
- name: Role to remove OSPF configuration and reapply it

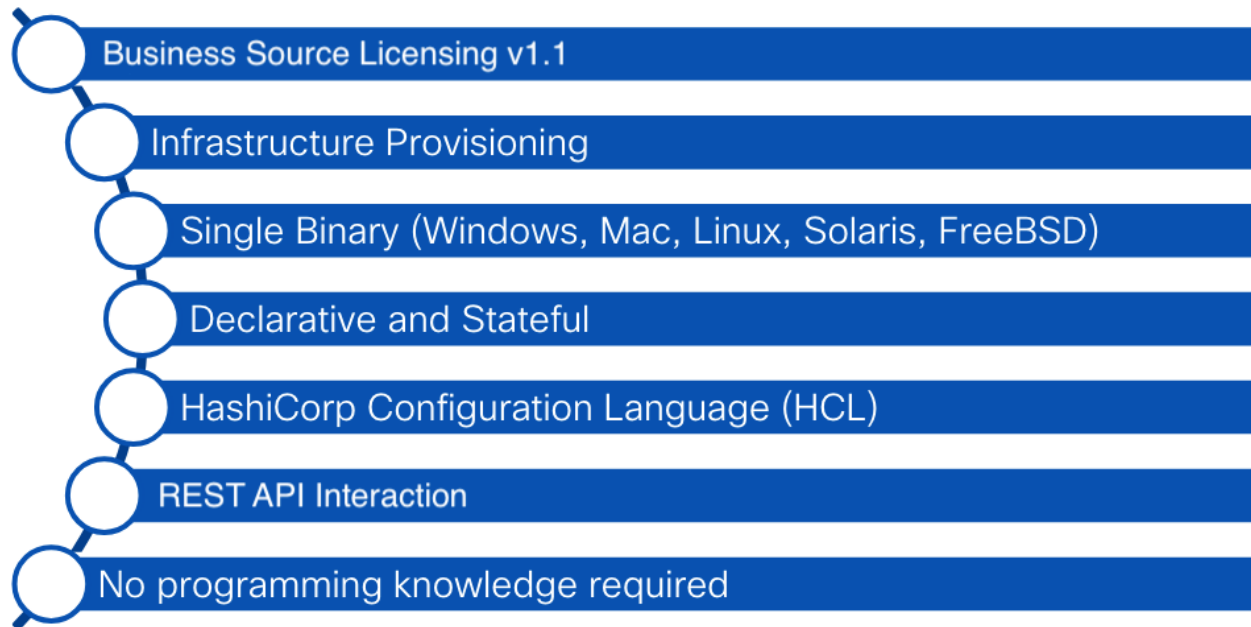
tasks:
  - name: Delete all OSPF processes
    cisco.ios.ios_ospfv2:
      state: deleted

  - name: Configure OSPF V2 on IOS XE
    loop: "{{ configure_ospf }}"
    cisco.ios.ios_ospfv2:
      config:
        processes:
          - process_id: "{{ item.ospf_process_id }}"
            network:
              address: "{{ item.ospf_network_address }}"
              wildcard_bits: "{{ item.ospf_wildcard_bits }}"
              area: "{{ item.ospf_area }}"
      state: present
```

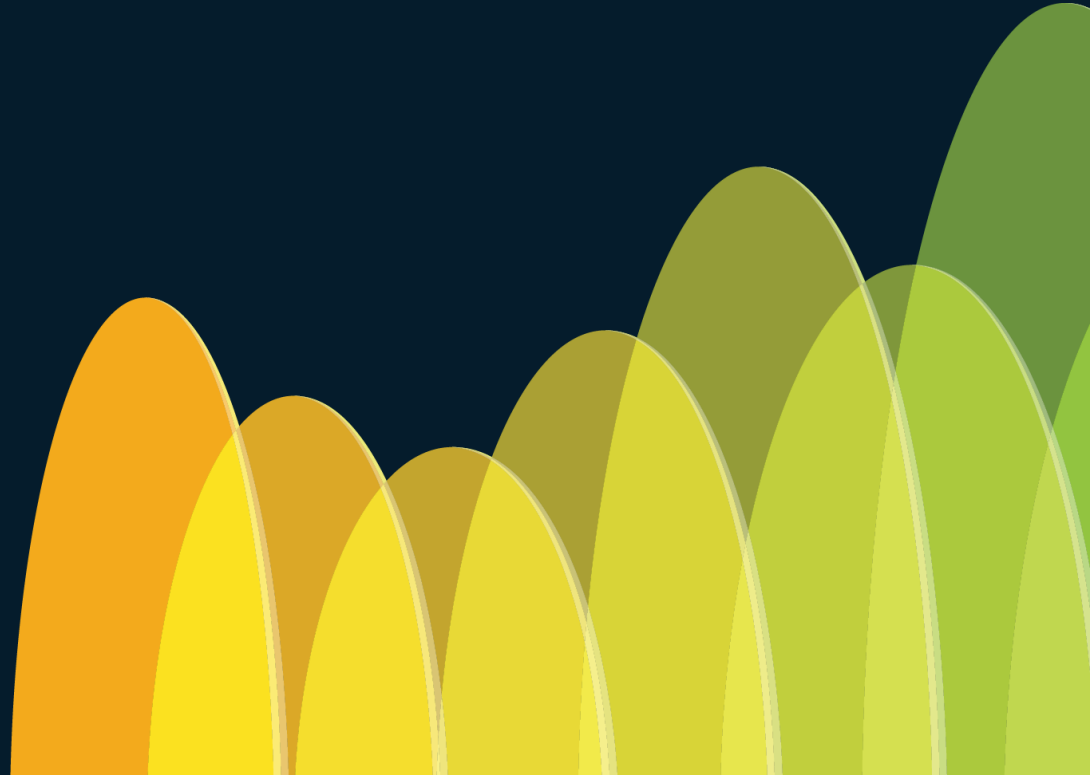
Terraform



What is Terraform?



Terraform Concepts



Terraform

(Version 1.9.7 – latest)

- Runs as single binary (Core)
- Command line executable
 - Reading configuration files
 - State management
 - Graph
 - Plan execution

Installation

<https://developer.hashicorp.com/terraform/install>



Install Terraform

1.9.7 (latest) ▾

macOS

Package manager

```
brew tap hashicorp/tap  
brew install hashicorp/tap/terraform
```

Binary download

AMD64

Version: 1.9.7

Download [↓](#)

ARM64

Version: 1.9.7

Download [↓](#)

Windows

Binary download

386

Version: 1.9.7

Download [↓](#)

AMD64

Version: 1.9.7

Download [↓](#)

Terraform providers

- Terraform binary doesn't know Meraki/Catalyst Center/Catalyst SDWAN/IOS XE
- Relies on specific plugins
 - Downloaded dynamically via initialization (via terraform init command)
- Providers understand API interactions
 - Meraki, Catalyst Center, Catalyst SDWAN and IOS XE API calls

```
terraform {  
  required_providers {  
    iosxe = {  
      source = "CiscoDevNet/iosxe"  
      version = "0.5.5"  
    }  
  }  
}
```

Types of Terraform Providers

<https://registry.terraform.io/namespaces/CiscoDevNet>



Official

Owned & maintained
by HashiCorp

Ex. AWS, Azure, GCP



Partner

Owned & maintained
by partners.

Ex. ACI, MSO, ASA

Community

Published by
individual groups or
maintainers in the
community

```
terraform {  
  required_providers {  
    sdwan = {  
      source = "CiscoDevNet/sdwan"  
      version = "0.3.9"  
    }  
  }  
}
```

```
terraform {  
  required_providers {  
    catalystcenter = {  
      source = "CiscoDevNet/catalystcenter"  
      version = "0.1.9"  
    }  
  }  
}
```

Terraform Provider configuration

Terraform block → terraform {
 required_version ">= 1.5.0" ← Terraform version
Provider definition → required_providers {
 iosxe = {
 source = "CiscoDevNet/iosxe" ← Registry namespace
 version = "0.5.5" ← Provider version
 }
}

Provider block → provider "iosxe" {
 username = "developer" ← Authentication
 password = "C1sco12345" ← Authentication
RESTCONF URL → url = "https://10.10.20.48"
 insecure = true ← HTTP API request
}

Terraform Resources & Data Sources

Resources

- Specific to a given provider
- apply/destroy/modifies infrastructure
- Accepts arguments
- Describes your intent for infrastructure

```
resource "iosxe_interface_vlan" "vlan_10" {  
  name          = 10  
  autostate     = false  
  description   = "Production data VLAN"  
  shutdown     = false  
}
```

Data Sources

- Allow data to be fetched or computed for use elsewhere in Terraform configuration
- Terraform apply/destroy does not modify data source

```
data "iosxe_interface_ethernet" "gi3" {  
  type      = "GigabitEthernet"  
  name      = "3"  
}  
  
data "iosxe_interface_vlan" "vlan_10" {  
  name      = 10  
}
```

Terraform Plans/Configuration Files

- Collection of HCL instructions
 - What you want to provision (intent)
- .tf extension
 - Terraform scans directory
 - Directory that Terraform is run in
 - Can be a singular file – main.tf
 - Can be broken up into smaller *.tf files

Terraform Configuration Example

Used in state file

Resource type
(From Provider)

Name of the resource
Must be unique

```
resource "catalystcenter_building" "main_building" {  
  name           = "Main Building"  
  parent_name    = "Global/${catalystcenter_area_san_jose.name}"  
  latitude       = 37.338  
  longitude      = -121.832  
}  
  
resource "catalystcenter_floor" "ground_floor" {  
  name           = "Ground Floor"  
  parent_name    =  
"${catalystcenter_building.main_building.parent_name}/${catalystcenter_building.main_building.name}"  
  rf_model       = "Drywall Office Only"  
  width          = 30.5  
  length         = 50.5  
  height         = 3.5  
}
```

Terraform Data Source Example

Resource type
(From Provider)

Name of the resource
Must be unique

```
data "sdwan_cisco_logging_feature_template" "Factory_Default_Cisco_Logging_Template" {  
  name  
    = "Factory_Default_Cisco_Logging_Template"  
}
```

```
data "sdwan_cedge_aaa_feature_template" "Factory_Default_AAA_CISCO_Template" {  
  name  
    = "Factory_Default_AAA_CISCO_Template"  
}
```

```
resource "sdwan_feature_device_template" "TF_device_template" {  
  name  
    = "TF_DT"  
  description  
    = "My device template via terraform. Using new and available feature templates."  
  device_type  
    = "vedge-C8000V"  
  device_role  
    = "sdwan-edge"  
  general_templates  
    = [{  
      id  
        = sdwan_cisco_system_feature_template.TF_system_template.id  
      type  
        = sdwan_cisco_system_feature_template.TF_system_template.template_type  
      version  
        = sdwan_cisco_system_feature_template.TF_system_template.version  
    },
```

```
    { id  
      type  
    },  
    { id  
      type  
    }  
  ]  
}
```

Terraform Registry – Documentation

meraki

OverviewDocumentationUSE PROVIDER

MERAKI DOCUMENTATION

Filter

meraki provider

Resources

meraki_administered_licensing_subscription_subscriptions_bind

meraki_administered_licensing_subscription_subscriptions_claim

meraki_administered_licensing_subscription_subscriptions_claim_key_validate

meraki_devices

meraki_devices_appliance_radio_settings

meraki_devices_appliance_uplinks_settings

meraki_devices_appliance_vmx_authentication_token

meraki_devices_blink_leds

meraki_devices_camera_custom_analytics

meraki_devices_camera_generate_snapshot

meraki_devices_camera_quality_and_retention

meraki_devices_camera_sense

meraki_devices_camera_video_settings

meraki_devices_camera_wireless_profiles

meraki_devices (Resource)

ON THIS PAGE

Example Usage

Schema

Required

Optional

Report an issue

Example Usage

```
resource "meraki_devices" "example" {  
  
  lat   = 37.4180951010362  
  lng   = -122.098531723022  
  name  = "My AP"  
  serial = "string"  
  tags  = ["recently-added"]  
}  
  
output "meraki_devices_example" {  
  value = meraki_devices.example  
}
```

Schema

Required

- `serial` (String) Serial number of the device

Optional

- `address` (String) Physical address of the device
- `floor_plan_id` (String) The floor plan to associate to this device. null disassociates the device from the floorplan.

Terraform State

- Terraform is stateful
 - Tracks objects it builds (terraform.tfstate)
 - Source of everything it knows about
- Stored inside working directory
- Can use backend – AWS, Terraform Cloud
- Do not modify state file directly

```
{
  "version": 4,
  "terraform_version": "1.8.1",
  "serial": 1,
  "lineage": "f92f5848-b96b-acf6-1e92-10972bb456bb",
  "outputs": {},
  "resources": [
    {
      "mode": "data",
      "type": "iosxe_restconf",
      "name": "example",
      "provider": "provider[\"registry.terraform.io/cisco/devnet/iosxe\"]",
      "instances": [
        {
          "schema_version": 0,
          "attributes": {
            "attributes": {
              "poe-module": "[\n  {\n    \"module\": 1,\n    \"available-power\": \"370.0\",\n    \"used-power\": \"0.0\",\n    \"remaining-power\": \"370.0\",\n    \"chassis-num\": 1,\n    \"power-pool-idx\": 1,\n    \"num-ports\": 24,\n    \"used-ports\"
```

Brownfield Infrastructure

- Import Infrastructure for Terraform to manage
 - Import CLI command
 - import blocks (version 1.5)

```
terraform import iosxe_static_route.static_route "Cisco-IOS-XE-native:native/ip/route/ip-route-interface-forwarding-list=10.20.0.0,255.255.255.0"
```

```
import {  
  id = "Cisco-IOS-XE-native:native/interface/GigabitEthernet=3"  
  to = iosxe_interface_ethernet.gi3  
}  
  
import {  
  id = "Cisco-IOS-XE-native:native/interface/Vlan=10"  
  to = iosxe_interface_vlan.vlan10  
}
```

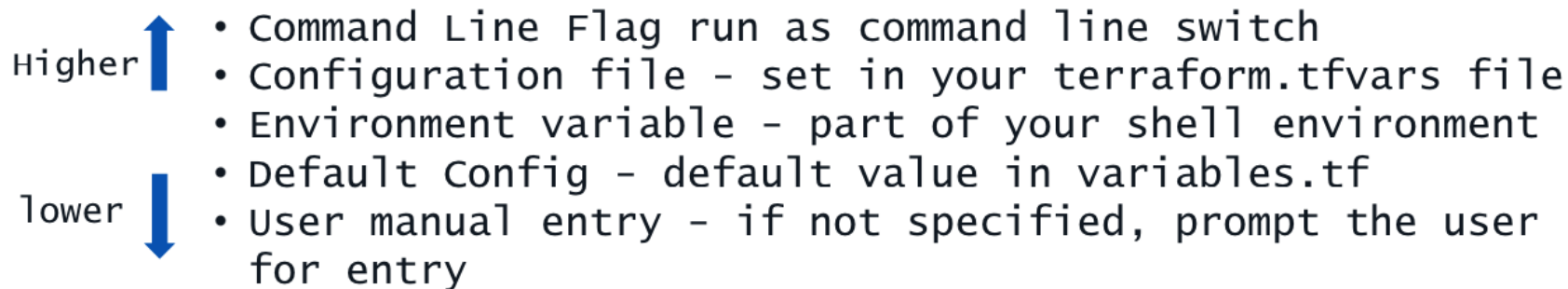
Variables in Terraform

- Value substitution – makes code reusable

Can be defined	Variable Types	Complex Types
<ul style="list-style-type: none">• CLI (<code>-var=</code> <code>-var-file=</code>)• <code>variables.tf</code><ul style="list-style-type: none">• Default config**• <code>terraform.tfvars</code>	<ul style="list-style-type: none">• String• Number• Bool• Any (default)	<ul style="list-style-type: none">• List• Map• Object

Terraform Variable Precedence

- Variables have precedence
- Variables can be set, but overridden



<https://developer.hashicorp.com/terraform/language/values/variables>

Terraform – CLI commands

terraform init

- Download and Install plugins for configured providers
- Must initialize before plan/apply
- Creates a provider “lock” file

terraform plan

- Scans the current directory for the configuration (.tf & .tfvars extensions)
- Determines what actions are necessary to achieve the desired state
- Preview your changes – no changes made

terraform apply

- Scans the current directory for the configuration (.tf & .tfvars files)
- Preview your changes (can bypass with -auto-approve)
- Applies the configuration to targets (upon approval “yes”)

terraform destroy

- Scans the state file for what to “destroy”
- Preview your deletions
- Infrastructure is destroyed
- Can be specific with “-target”

Terraform – CLI commands

`terraform fmt`

- Formats Terraform configuration files in directory

`terraform show`

- Show the state file in a readable format
- Can also read a specific state file (path)

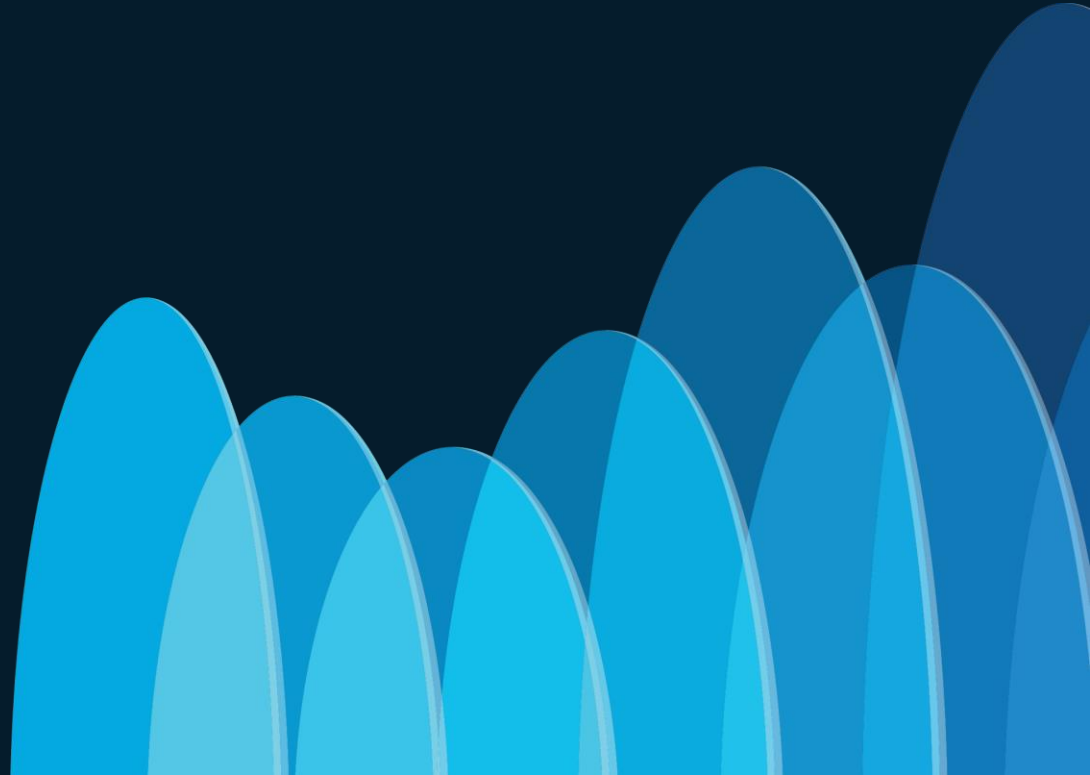
`terraform state`

- Advanced State Management
- **show <resource>** – shows a particular resource
- **list** – lists all resources in current state file
- **rm <instance>** – remove an instance from the state file
- **mv** – move an item. Good for renaming resources

`terraform validate`

- Verifies correctness of Terraform configuration files (*.tf)
- Checks syntax
- Can be used to solve configuration errors

Ansible and Terraform Comparison



Ansible/Terraform comparison



Source	Open Source	Business Source Licensing
IaC Type	Configuration Management	Provisioning
Language Type	Procedural	Declarative
Stateful	No	Yes
Written in	Python	Go
Cisco commitment	Yes	Yes

Cisco Ansible Collections

Collections in the Cisco Namespace

These are the collections with docs hosted on docs.ansible.com in the **cisco** namespace.

- [cisco.aci](#)
- [cisco.asa](#)
- [cisco.dnac](#)
- [cisco.intersight](#)
- [cisco.ios](#)
- [cisco.iosxr](#)
- [cisco.ise](#)
- [cisco.meraki](#)
- [cisco.mso](#)
- [cisco.nxos](#)
- [cisco.ucs](#)

Cisco Terraform Providers

 aci by: CiscoDevNet	 intersight by: CiscoDevNet	 mso by: CiscoDevNet
 fmc by: CiscoDevNet	 ciscoasa by: CiscoDevNet	 dcnm by: CiscoDevNet
 iosxe by: CiscoDevNet	 nxos by: CiscoDevNet	 sdwan by: CiscoDevNet
 iosxr by: CiscoDevNet	 cdo by: CiscoDevNet	 cm12 by: CiscoDevNet
 tetration by: CiscoDevNet	 ise by: CiscoDevNet	 ciscomcd by: CiscoDevNet
 catalystcenter by: CiscoDevNet	 nso by: CiscoDevNet	 secureworkload by: CiscoDevNet

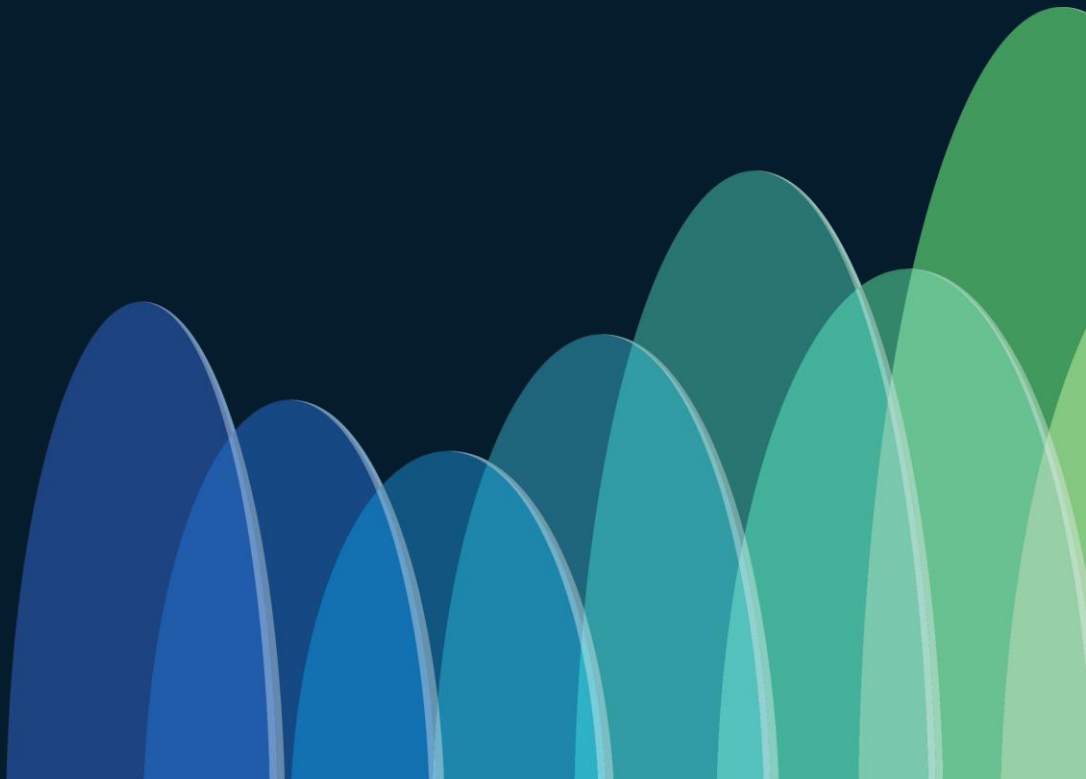
Next Steps



Infrastructure as Code with Terraform and Ansible

- Install and test Terraform and Ansible
 - Available for most platforms
 - Which one works better for you?
 - What are you currently using?
- Think big...start small
 - Automate the simple, then build into more complex tasks
- Ease of writing Infrastructure as Code with Terraform and Ansible
 - No special programming skills needed
- Ansible Modules/Terraform Resources for most common tasks
- Robust REST APIs make automation easy and scalable

Resources



More information – Ansible/Terraform

- <https://www.terraform.io/>
- <https://www.ansible.com/>
- <https://developer.cisco.com/automation-terraform/>
- <https://registry.terraform.io/namespaces/CiscoDevNet>
- <https://docs.ansible.com/ansible/latest/collections/cisco/index.html>

Keynote Deep Dives

Wednesday
10:30am – 11:30am



Experiences Amplified:
How AI Can Fuel Better Employee and Customer Experiences

Level 1
Room 106



Smart, Secure, Seamless:
Transforming Experiences with Next-Generation Networking

Level 2
Room 204



Harness a Bold New Era:
Transform Data Centre and Service Provider Connectivity

Level 2
Room 203



Securing User to Application and Everything in Between

Level 2
Melbourne Room 2



Unlocking Digital Resilience through Unified Observability

The HUB
Centre Stage

Complete Your Session Evaluations



Complete a minimum of 4 session surveys and the Overall Event Survey to claim a **Cisco Live T-Shirt**.



Complete your surveys in the **Cisco Live mobile app**.



Continue your education

- Visit the Cisco Stand 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

Contact me at: @aidevnet



Thank you

CISCO *Live!*

#CiscoLiveAPJC

CISCO *Live!*

GO BEYOND

#CiscoLiveAPJC