

CISCO *Live!*

GO BEYOND

#CiscoLiveAPJC



Network Automation Simplified: Harnessing the Power of Terraform and Ansible with Meraki

Hyeyeon Choi Solutions Engineer

DEVNET-1293

CISCO *Live!*

#CiscoLiveAPJC

Cisco Webex App

Questions?

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

How

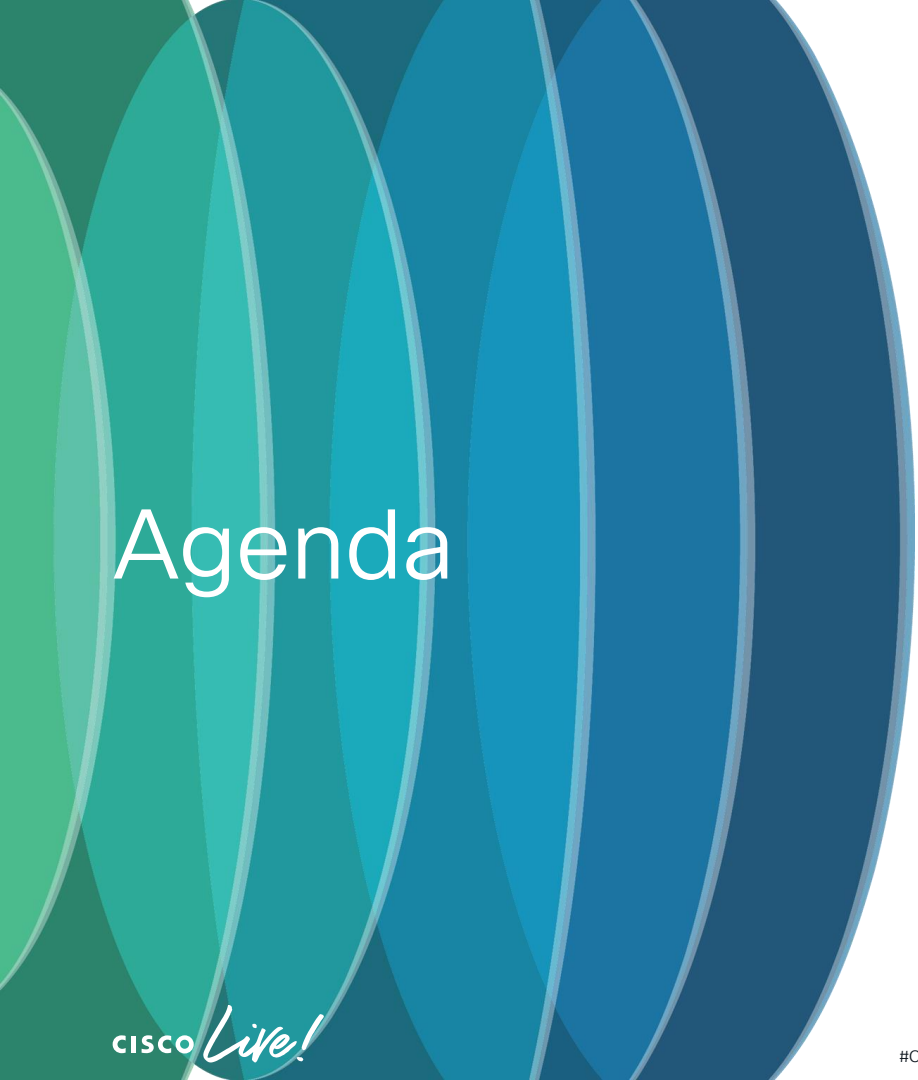
- 1 Find this session in the Cisco Live Mobile App
- 2 Click “Join the Discussion”
- 3 Install the Webex App or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until November 15, 2024.



<https://ciscolive.ciscoevents.com/ciscolivebot/#DEVNET-1293>

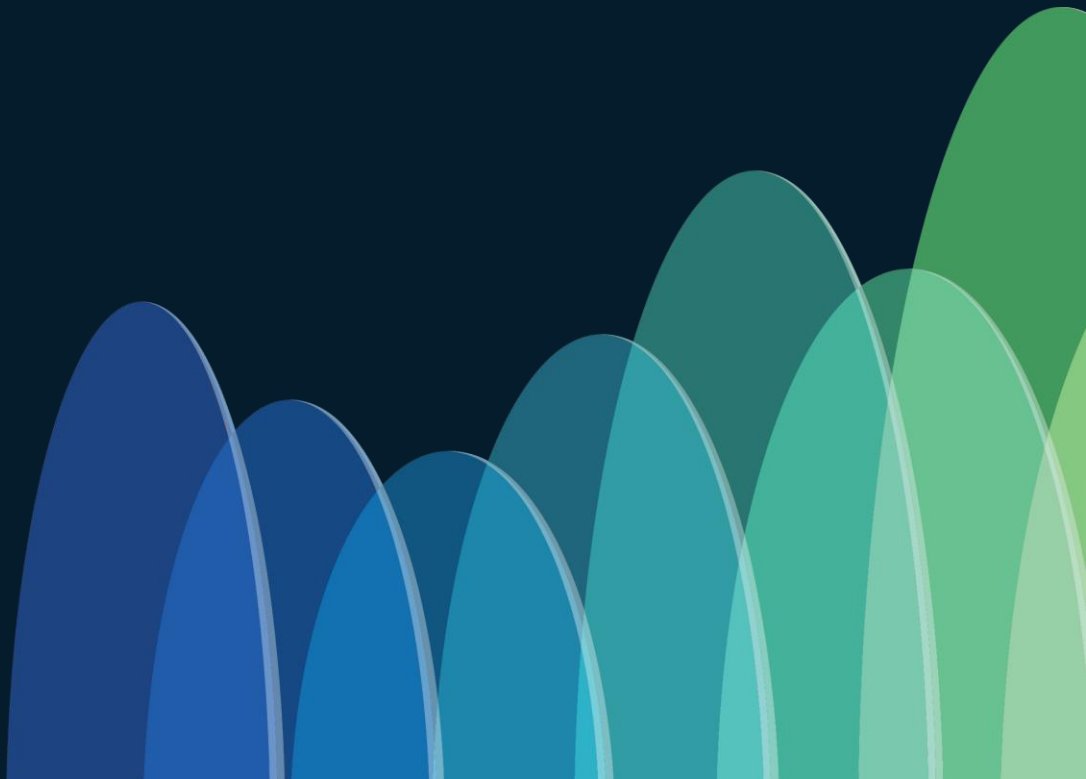




Agenda

- Meraki platform overview
- Why do you need automation
- Terraform and Ansible for Meraki
- Terraform vs Ansible
- Recap and next steps

Meraki platform overview



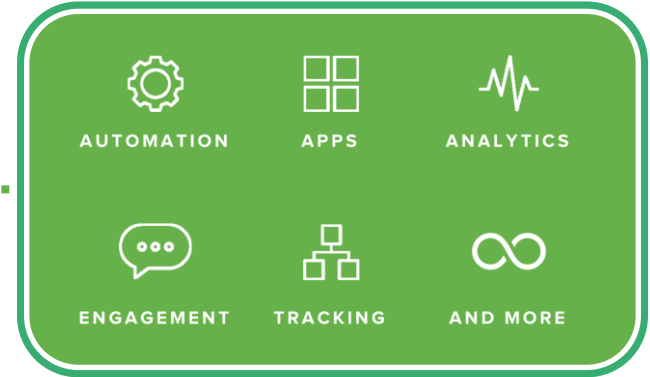
Meraki Platform Architecture

OUT-OF-THE-BOX
MANAGEMENT & ANALYTICS



{ API }

DIGITAL BUSINESS
POWERED BY MERAKI



Meraki API Services



{APIs}

DASHBOARD API

- Programmability
- Automation
- Monitoring
- Reporting
- Data Insights

WEBHOOK API

- Event Stream
- Automation Trigger

LOCATION STREAMING API

- Wayfinding
- Asset Tracking
- Location Analytics

CAPTIVE PORTAL API

- Guest Wi-Fi
- Secure Onboarding

MV SENSE API

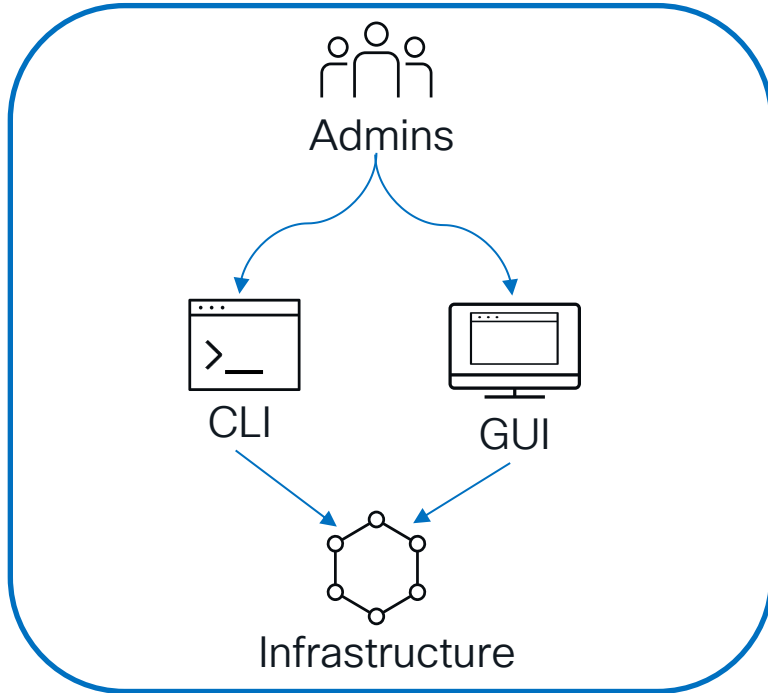
- Real time (4Hz) Data Stream
- Historical time-series via REST
- Current snapshot

Why do you need automation

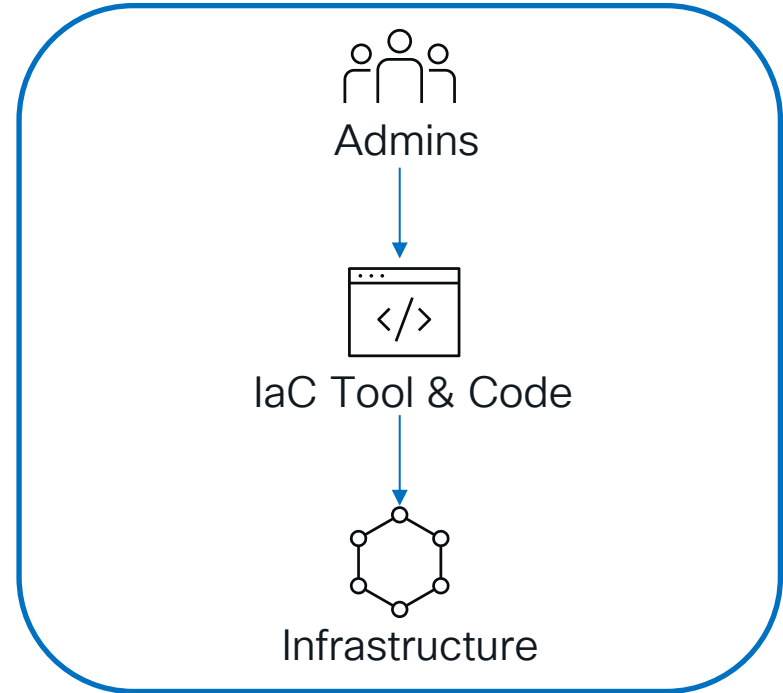


Manual vs Automation

Manual



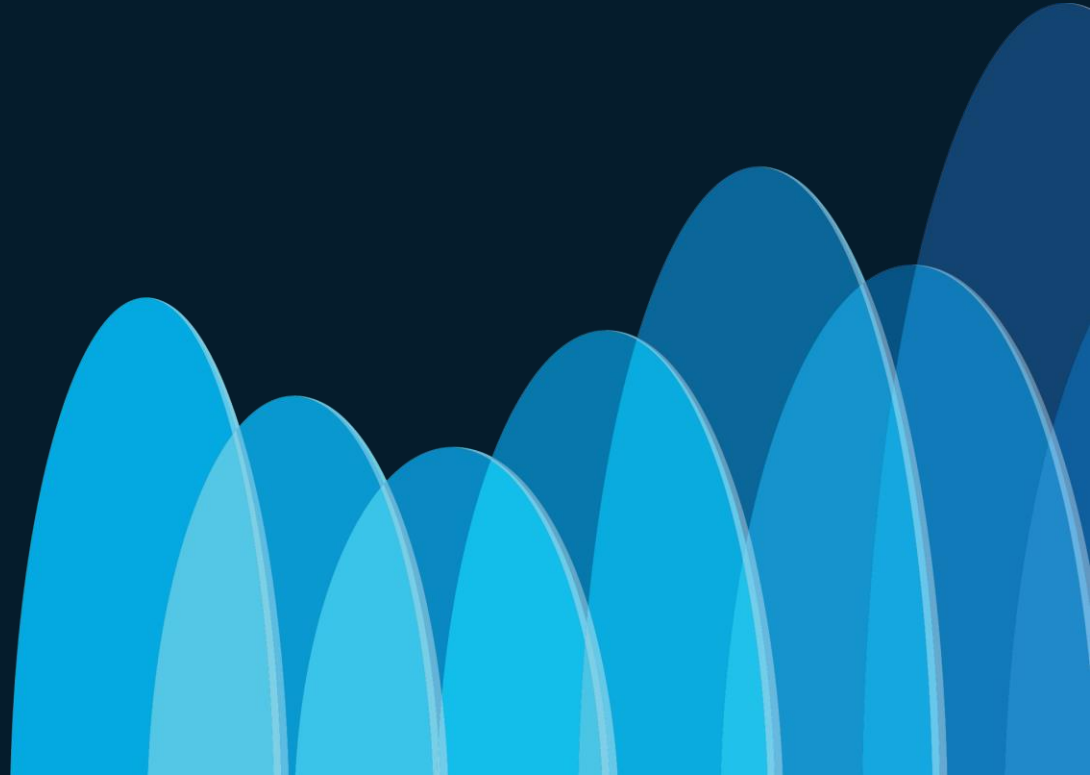
Automation



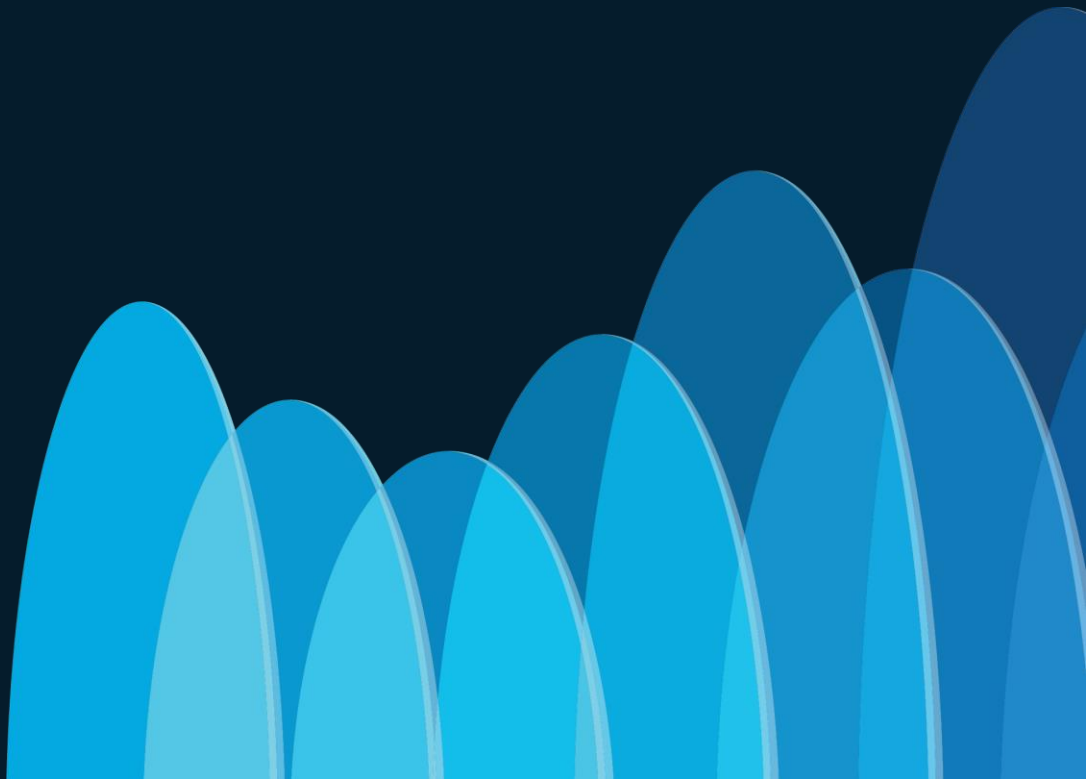
Benefits of Automation

- ✓ Increased Efficiency by Automating routine tasks
- ✓ Boost speed and responsiveness
- ✓ Achieve scalability
- ✓ Reduce errors
- ✓ Ensure infrastructure consistency
- ✓ Minimize time to resolve

Terraform and Ansible for Meraki



Terraform

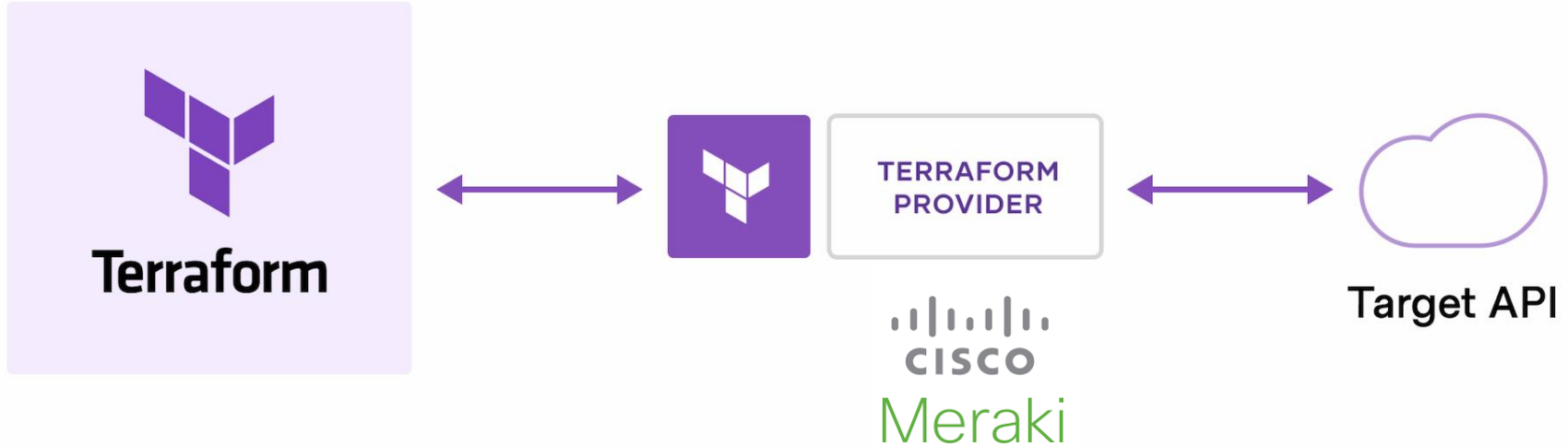


What is Terraform?



- infrastructure-as-code software tool created by HashiCorp
- Used to be Open Source – Not anymore!
- Infrastructure Provisioning (build, change and version)
- Agentless
- Declarative and Stateful
- Meraki APIs supported
- No Special Programming Skills required

How does Terraform work?



Cisco Meraki Terraform Provider

- Developed by Cisco / Meraki
- Plugins that allow interaction with remote APIs and platforms
- Adds the resource types and data sources that the Terraform can use
- As our APIs evolve the provider will be maintained.

```
terraform {  
  required_providers {  
    meraki = {  
      source = "cisco-open/meraki"  
      version = "0.2.12-alpha"  
    }  
  }  
}  
  
provider "meraki" {  
  # Configuration options  
}
```

Cisco Meraki Terraform Provider

 Registry

Search all resources

Browse ▾ Publish ▾ Sign-in 

Use HCP Terraform for free 

Providers / cisco-open / meraki / Version 0.2.11-alpha ▾ Latest Version

meraki Overview Documentation [USE PROVIDER ▾](#)**meraki**
 Partner by [cisco-open](#)
Networking

VERSION
0.2.11-alpha

 PUBLISHED
a month ago

 SOURCE CODE
[cisco-open/terraform-provider-meraki](#)

Provider Downloads All versions ▾

Downloads this week 1,408

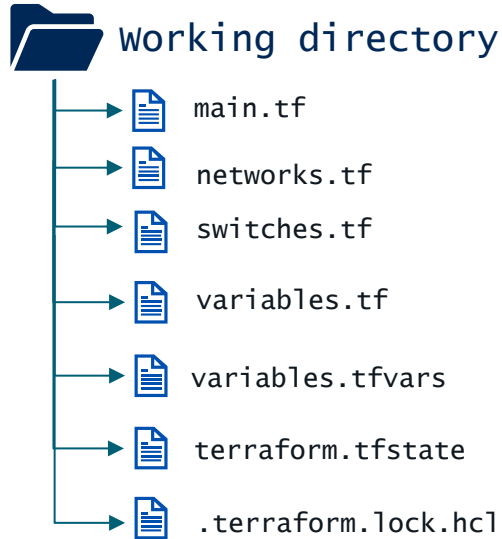
Downloads this month 2,923

Downloads this year 9,689

Downloads over all time 9,689

<https://registry.terraform.io/providers/cisco-open/meraki/latest>

Terraform Files



- Automatic Execution Order
- Declarative (define “what” you want, then Terraform takes care of “how”)
- Modularity
- `variables.tf` declares the variables.
- `.tfvars` assigns actual values to those variables
- Maintains the current state of infrastructure in a `.tfstate` file

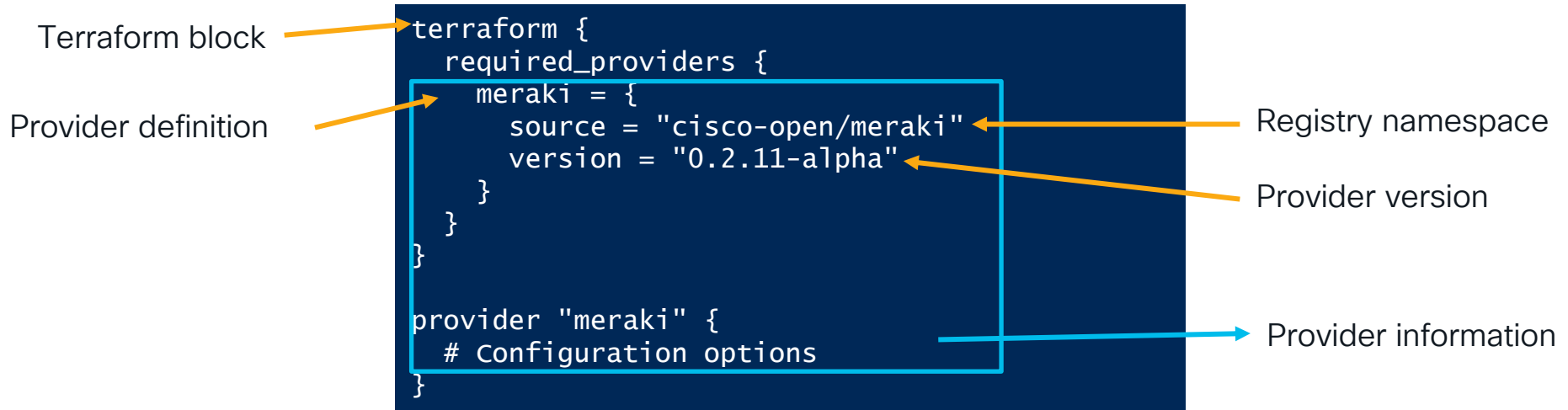
HashiCorp Configuration Language (HCL)

- Based on HCL2 toolkit
- Designed for Infrastructure Provisioning
- Built around
 - Blocks – Resources/Data Sources/Modules
 - Parameters/Arguments
 - Blocks open "{" and "}"

```
Block_Type Block_Label {  
  # this is a comment  
  // This is also a comment  
  /*  
    This is a comment too  
  */  
  parameter1 = value  
  parameter2 = value  
}
```

Terraform Block Configuration

- Configures Terraform behavior
- Specify the Providers you want



Terraform State

```
{
  "version": 4,
  "terraform_version": "1.9.5",
  "serial": 1,
  "lineage": "6ab93088-4015-bd36-4146-a6e7bfd0c2fd",
  "outputs": {
    "meraki_organizations_example": {
      "value": [
        {
          "cloud": {
            "region": {
              "name": "Asia"
            }
          },
          "id": "12345678",
          "licensing": {
            "model": "co-term"
          },
          "name": "Henie",
          "url": "https://nxxx.meraki.com/o/8ZtF-c/manage/organization/overview"
        }
      ]
    }
  }
}
```

- Terraform is stateful
 - Stored by default in “terraform.tfstate”
 - Determines which changes to make to your infrastructure
 - Prior to any operation, Terraform does a refresh to update the state with the real infrastructure
- **Avoid modifying the state file directly**
 - State locking can prevent concurrent operations from being applied by different users or processes

Cisco Meraki Terraform Provider

The screenshot shows the Terraform Registry page for the Cisco Meraki Provider. The header is purple with the Terraform logo, a search bar, and navigation links for Browse, Publish, Sign-in, and a button to 'Use HCP Terraform for free'. Below the header, the breadcrumb trail is 'Providers / cisco-open / meraki / Version 0.2.11-alpha', with 'Latest Version' selected. The main content area is titled 'meraki' and has tabs for 'Overview' and 'Documentation', with a 'USE PROVIDER' button. The 'Documentation' tab is active, showing a search filter and a list of documentation items. The 'meraki provider' item is highlighted with a green box. Below it are links for 'Resources' and 'Data Sources'. The 'Schema' section is also visible, showing a required variable 'meraki_dashboard_api_key'.

Providers / cisco-open / meraki / Version 0.2.11-alpha ▾ Latest Version

meraki 

Overview Documentation **USE PROVIDER ▾**

MERAKI DOCUMENTATION

Filter

meraki provider

- > Resources
- > Data Sources

meraki Provider

Schema

Required

- `meraki_dashboard_api_key` (String, Sensitive) Cisco meraki_dashboard_api_key to authenticate. If not set, it uses the MERAKI_DASHBOARD_API_KEY environment variable.

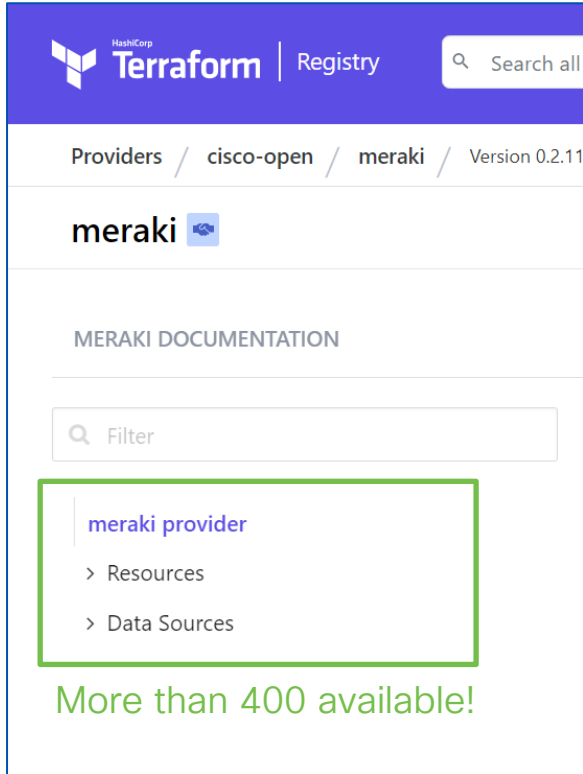
ON THIS PAGE

- Schema

Report an issue 

<https://registry.terraform.io/providers/cisco-open/meraki/latest/docs>

Cisco Meraki Terraform Provider



The screenshot shows the Terraform Registry page for the Cisco Meraki provider. The header includes the HashiCorp Terraform logo and a search bar. The breadcrumb trail is "Providers / cisco-open / meraki / Version 0.2.11". The main heading is "meraki" with a small icon. Below it is a section for "MERAKI DOCUMENTATION" with a filter input. A green box highlights the "meraki provider" section, which lists "Resources" and "Data Sources". Below this, a green banner states "More than 400 available!"

HashiCorp Terraform | Registry

Providers / cisco-open / meraki / Version 0.2.11

meraki

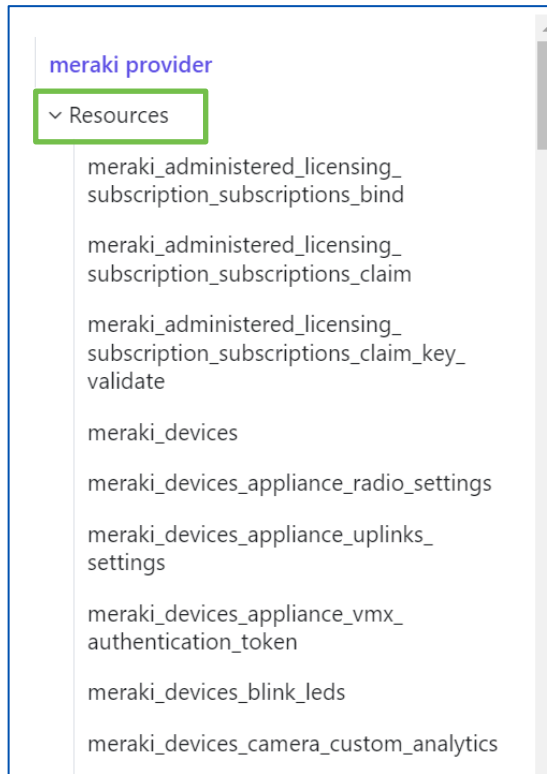
MERAKI DOCUMENTATION

Filter

meraki provider

- > Resources
- > Data Sources

More than 400 available!

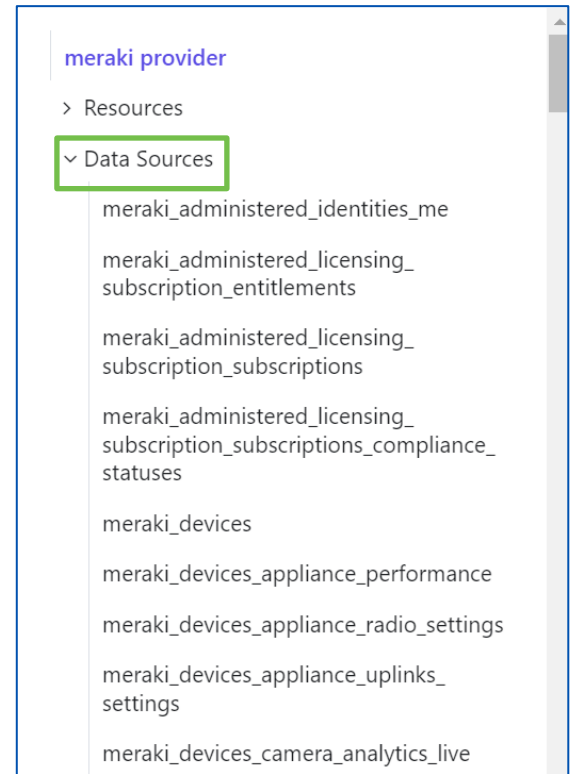


This screenshot shows the "Resources" list for the meraki provider. A green box highlights the "Resources" header. The list includes: meraki_administered_licensing_subscription_subscriptions_bind, meraki_administered_licensing_subscription_subscriptions_claim, meraki_administered_licensing_subscription_subscriptions_claim_key_validate, meraki_devices, meraki_devices_appliance_radio_settings, meraki_devices_appliance_uplinks_settings, meraki_devices_appliance_vmx_authentication_token, meraki_devices_blink_leds, and meraki_devices_camera_custom_analytics.

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



This screenshot shows the "Data Sources" list for the meraki provider. A green box highlights the "Data Sources" header. The list includes: meraki_administered_identities_me, meraki_administered_licensing_subscription_entitlements, meraki_administered_licensing_subscription_subscriptions, meraki_administered_licensing_subscription_subscriptions_compliance_statuses, meraki_devices, meraki_devices_appliance_performance, meraki_devices_appliance_radio_settings, meraki_devices_appliance_uplinks_settings, and meraki_devices_camera_analytics_live.

meraki provider

Resources

Data Sources

- meraki_administered_identities_me
- meraki_administered_licensing_subscription_entitlements
- meraki_administered_licensing_subscription_subscriptions
- meraki_administered_licensing_subscription_subscriptions_compliance_statuses
- meraki_devices
- meraki_devices_appliance_performance
- meraki_devices_appliance_radio_settings
- meraki_devices_appliance_uplinks_settings
- meraki_devices_camera_analytics_live

Resources and Data-sources

- Important to understand how to find the right resource for the task.
- The resources are named based on their path, which can be found on the Dashboard API documentation site.

Operation for Network Switch Setings	
API	Resource
<code>/networks/{networkId}/switch/settings</code>	<code>meraki_networks_switch_settings</code>

Operation for Device Camera Video Settings	
API	Resource
<code>/devices/{serial}/camera/video/settings</code>	<code>meraki_devices_camera_video_settings</code>

Terraform Resource Example

Used in state file

Resource type
(From Provider)

Name of the resource
Must be unique

```
resource "meraki_networks_wireless_ssids" "my_8021x_ssid" {  
  network_id      = "L_12341234"  
  number          = 10  
  auth_mode       = "8021x-radius"  
  name            = "ACME"  
  ip_assignment_mode = "Bridge mode"  
  default_vlan_id = "100"  
  radius_servers  = [{  
    host = "1.2.3.4"  
    port = 1812  
    secret = "SuperSecretPassword"  
  }]  
}
```

Resource Body

Arguments the
resource support

Terraform Data Source Example

Data Source type
(From Provider)

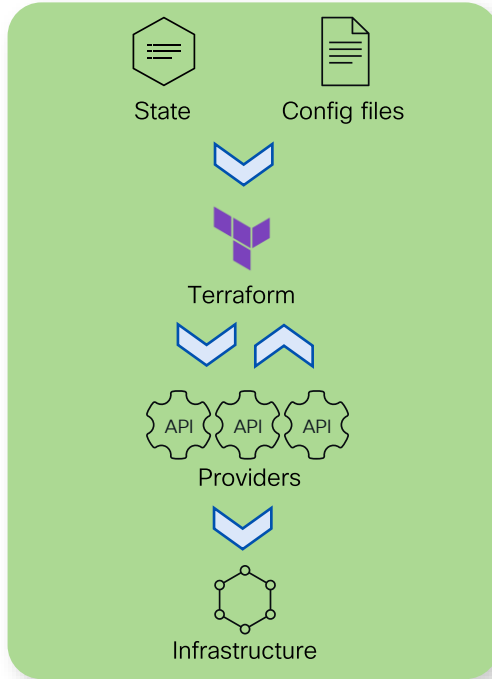
Data Source name
Must be unique

```
data "meraki_networks" "CLM_network_info" {  
  config_template_id      = "string"  
  ending_before           = "string"  
  is_bound_to_config_template = false  
  organization_id         = "string"  
  per_page                = 1  
  starting_after          = "string"  
  tags                    = ["string"]  
  tags_filter_type        = "string"  
}
```

```
output "meraki_networks_example" {  
  value = data.meraki_networks.CLM_network_info.item  
}
```

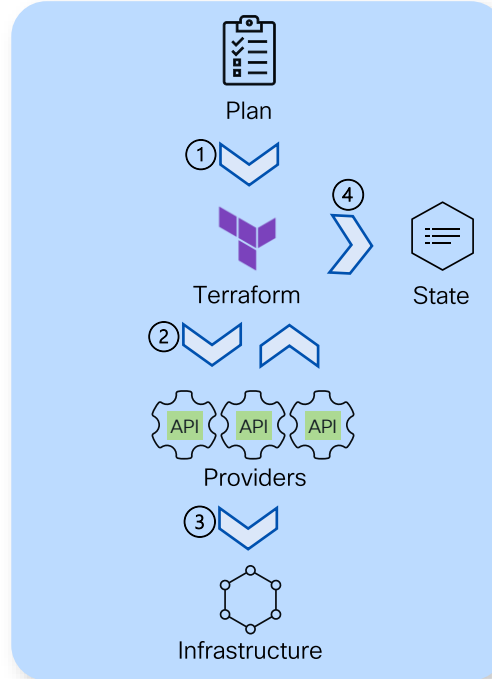
displays data from data sources and allows reuse in configurations or by external tools.

Terraform workflow



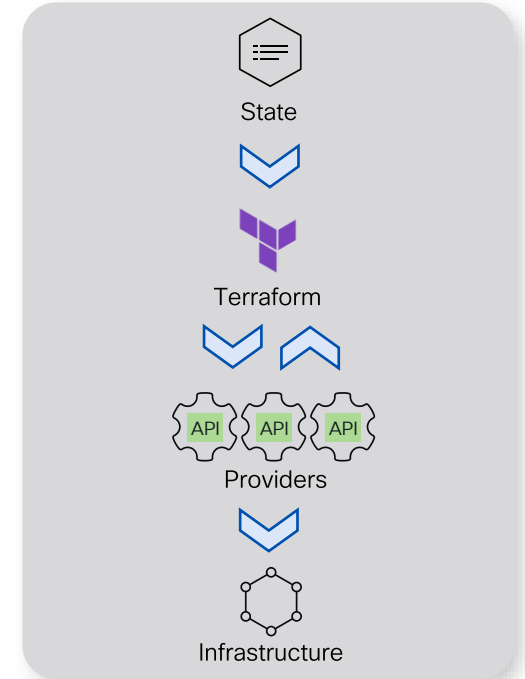
Plan

Check for the current state of infrastructure



Apply

Create, modify or delete infrastructure



Destroy

Delete infrastructure

Terraform – CLI commands

terraform init

- Download and Installs 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 extension)
- Determines what actions are necessary to achieve the desired state
- Preview your changes – no changes made (Dry Run)

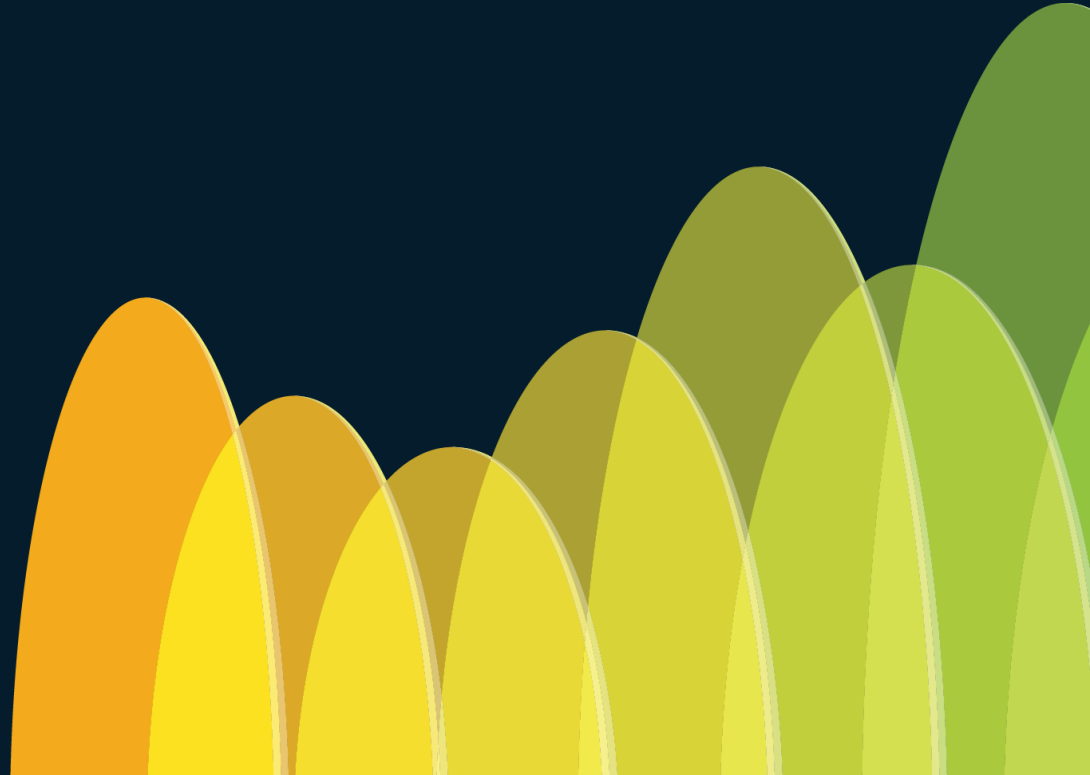
terraform apply

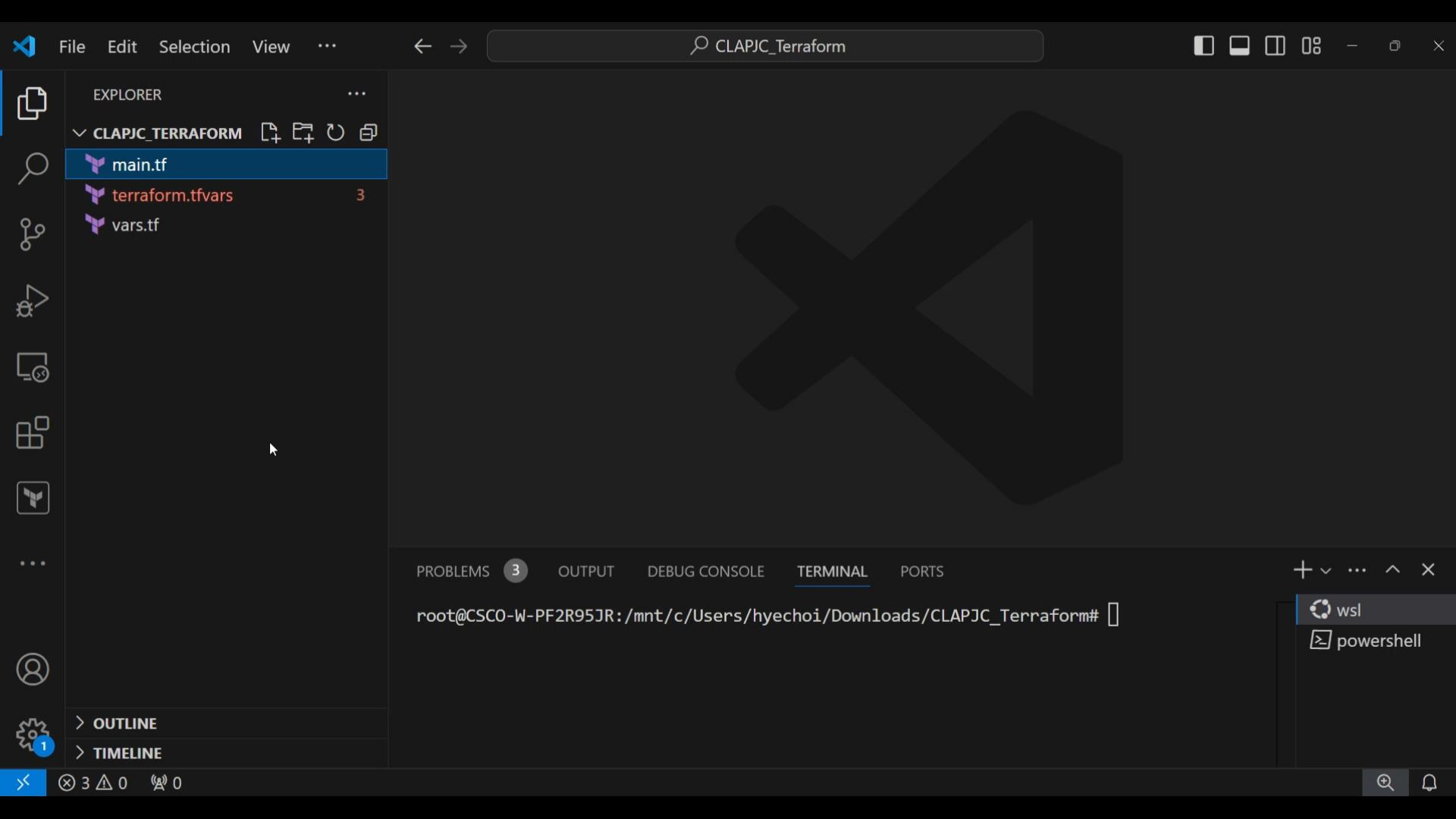
- Scans the current directory for the configuration (.tf & .tfvars extension)
- Preview your changes (can bypass with -auto-approve)
- Applies the configuration to targets (upon approval “yes”)

terraform destroy

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

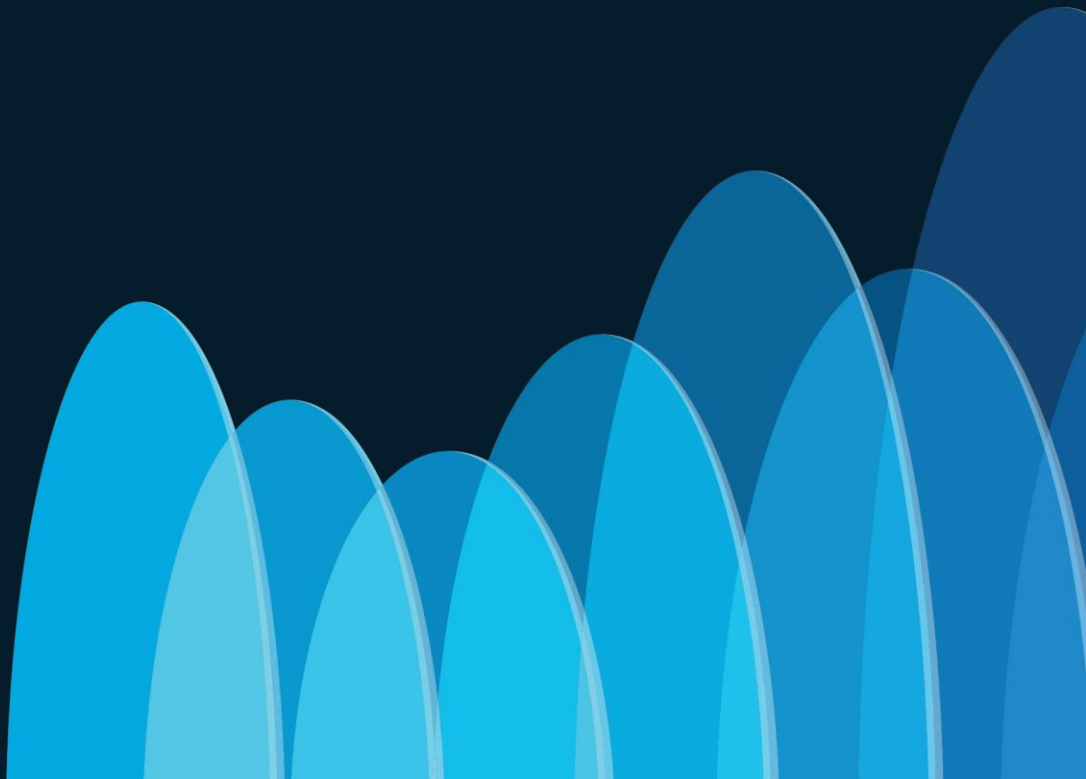
Let's see it in
action





Ansible

CISCO *Live!*



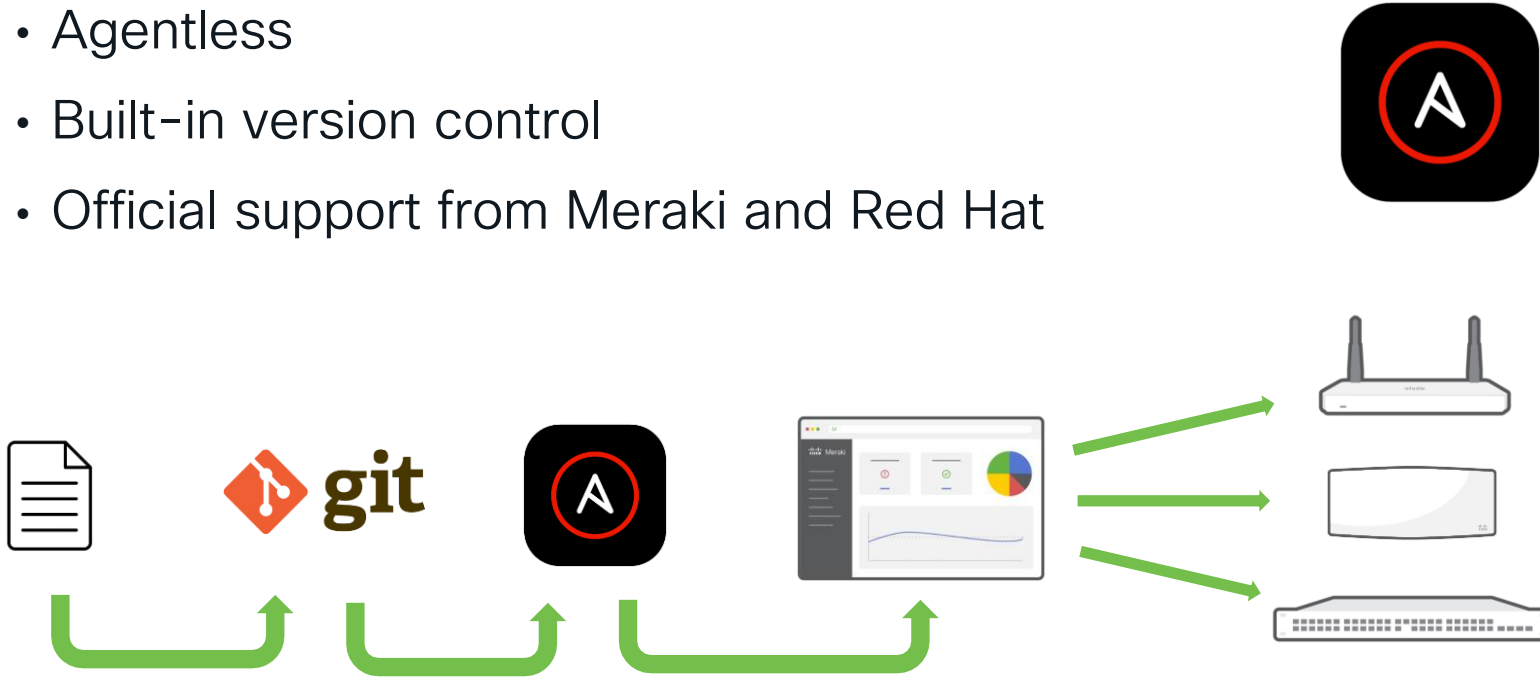
What is Ansible?



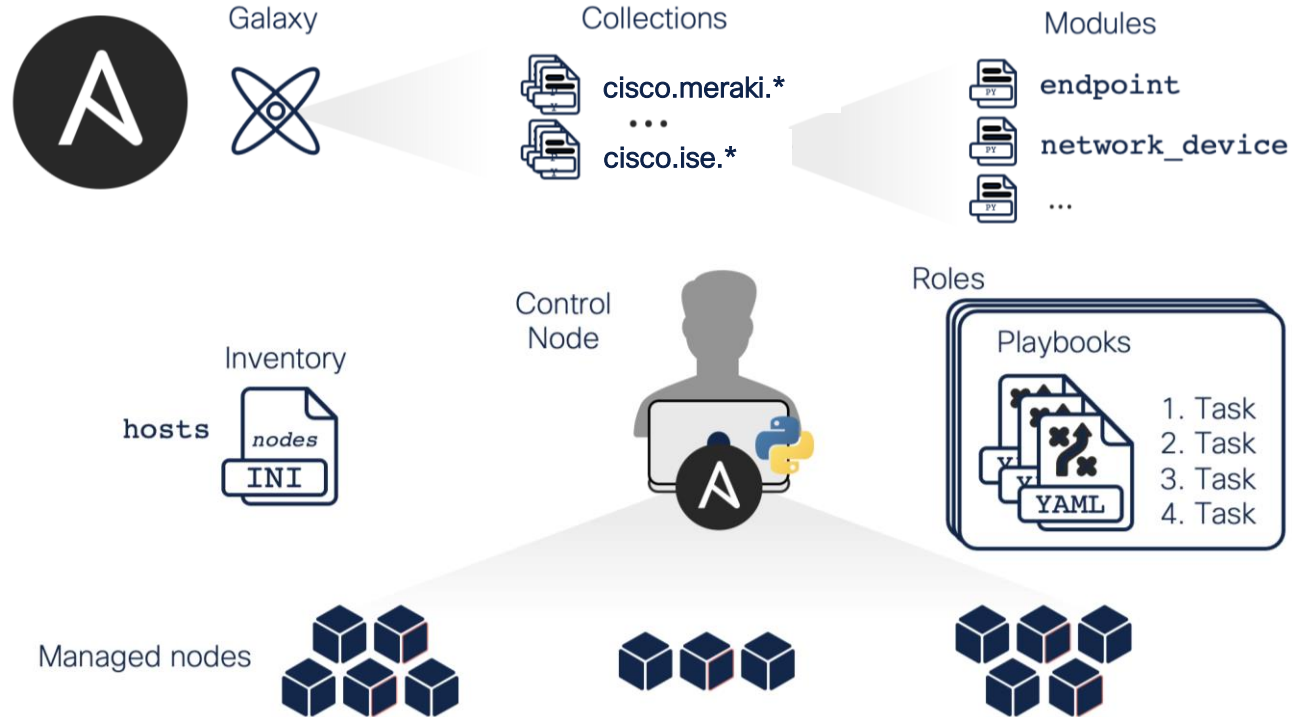
- Automation / Configuration / Orchestration tool
- Open Source
- Agentless Push Model
- Produces the same results no matter how many times it is executed
- No programming knowledge required
- Requires only data-structure manipulation knowledge
- Meraki REST API interaction

How does Ansible work?

- Agentless
- Built-in version control
- Official support from Meraki and Red Hat



Ansible Concepts



Ansible Collections – Cisco Namespace

 Ansible Community Documentation

BLOGANSIBLE COMMUNITY FORUMDOCUMENTATION

☰ Collections 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

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 Dellemc Namespace

Collections in the F5networks Namespace

 / [Collection Index](#) / Collections in the Cisco Namespace

This is the **latest** (stable) Ansible community documentation. For Red Hat Ansible Automation Platform subscriptions, see [Life Cycle](#) for version details.

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](#)

 Previous

Next 

© Copyright Ansible project contributors. Last updated on Sep 11, 2024.

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

cisco *Live!*

Ansible Cisco.Meraki Collection

More than 500 modules

Modules

- `administered_identities_me_info` module – Information module for `administered_identities_me`
- `administered_licensing_subscription_entitlements_info` module – Information module for `administered_licensing_subscription_entitlements`
- `administered_licensing_subscription_subscriptions_bind` module – Resource module for `administered_licensing_subscription_subscriptions_bind`
- `administered_licensing_subscription_subscriptions_claim` module – Resource module for `administered_licensing_subscription_subscriptions_claim`
- `administered_licensing_subscription_subscriptions_claim_key_validate` module – Resource module for `administered_licensing_subscription_subscriptions_claimkey_validate`
- `administered_licensing_subscription_subscriptions_compliance_statuses_info` module – Information module for `administered_licensing_subscription_subscriptions_compliance_statuses`
- `administered_licensing_subscription_subscriptions_info` module – Information module for `administered_licensing_subscription_subscriptions`
- `devices` module – Resource module for `devices`
- `devices_appliance_performance_info` module – Information module for `devices_appliance_performance`
- `devices_appliance_radio_settings` module – Resource module for `devices_appliance_radio_settings`
- `devices_appliance_radio_settings_info` module – Information module for `devices_appliance_radio_settings`
- `devices_appliance_uplinks_settings` module – Resource module for `devices_appliance_uplinks_settings`
- `devices_appliance_uplinks_settings_info` module – Information module for `devices_appliance_uplinks_settings`
- `devices_appliance_vmx_authentication_token` module – Resource module for `devices_appliance_vmx_authenticationtoken`
- `devices_blink_leds` module – Resource module for `devices_blinkleds`
- `devices_camera_analytics_live_info` module – Information module for `devices_camera_analytics_live`
- `devices_camera_custom_analytics` module – Resource module for `devices_camera_customanalytics`
- `devices_camera_custom_analytics_info` module – Information module for `devices_camera_customanalytics`

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

Cisco.Meraki collection module example

 Ansible Community Documentation

[BLOG](#)

[ANSIBLE COMMUNITY FORUM](#)

[DOCUMENTATION](#)

[Galaxy User Guide](#)

[Galaxy Developer Guide](#)

[REFERENCE & APPENDICES](#)

Collection Index

Collections in the Amazon Namespace

Collections in the Ansible Namespace

Collections in the Arista Namespace

Collections in the Awx Namespace

Collections in the Azure Namespace

Collections in the Check_point Namespace

Collections in the Chocolatey Namespace

Collections in the Cisco Namespace

- Cisco.Aci
- Cisco.Asa
- Cisco.Dnac
- Cisco.Intersight
- Cisco.Ios
- Cisco.Iosxr
- Cisco.Ise
- Cisco.Meraki
 - Description
 - Plugin Index
 - Cisco.Mso

cisco.meraki.devices_switch_ports module – Resource module for devices _switch _ports

Note

This module is part of the [cisco.meraki collection](#) (version 2.18.3).

You might already have this collection installed if you are using the `ansible` package. It is not included in `ansible-core`. To check whether it is installed, run `ansible-galaxy collection list`.

To install it, use: `ansible-galaxy collection install cisco.meraki`. You need further requirements to be able to use this module, see [Requirements](#) for details.

To use it in a playbook, specify: `cisco.meraki.devices_switch_ports`.

New in *cisco.meraki* 2.16.0

- Synopsis
- Requirements
- Parameters
- Notes
- See Also
- Examples
- Return Values

Synopsis

- Manage operation update of the resource devices _switch _ports.
- Update a switch port.

- Manage operation update of the resource devices _switch _ports.
- Update a switch port.

https://docs.ansible.com/ansible/latest/collections/cisco/meraki/devices_switch_ports_module.html#ansible-collections-cisco-meraki-devices-switch-ports-module

cisco *Live!*

#CiscoLiveAPJC

DEVNET-1293

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

37

Cisco.Meraki Module

- Important to understand how to find the right module for the task.
- The modules are named based on their path, which can be found on the Dashboard API documentation site.

Operation for Network Switch Setings	
API	module
/networks/{networkId}/switch/settings	cisco.meraki.networks_switch_settings

Operation for Device Camera Video Settings	
API	module
/devices/{serial}/camera/video/settings	cisco.meraki.devices_camera_video_settings

Ansible Taxonomy

Collection: a set of Roles, playbooks, modules, ...

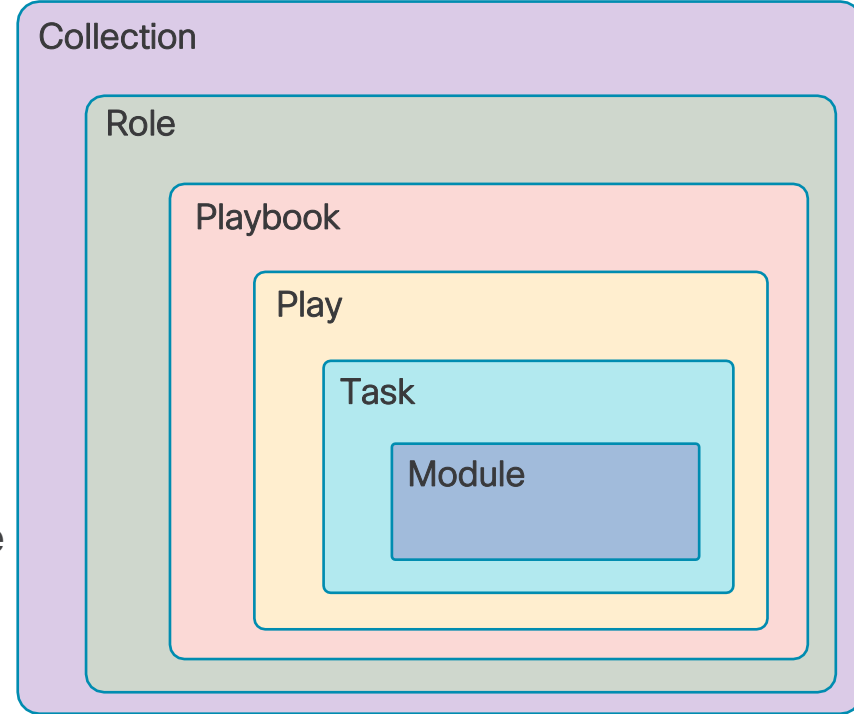
Role: a set of Playbooks

Playbook: repeatable standard config

Play: a set of tasks

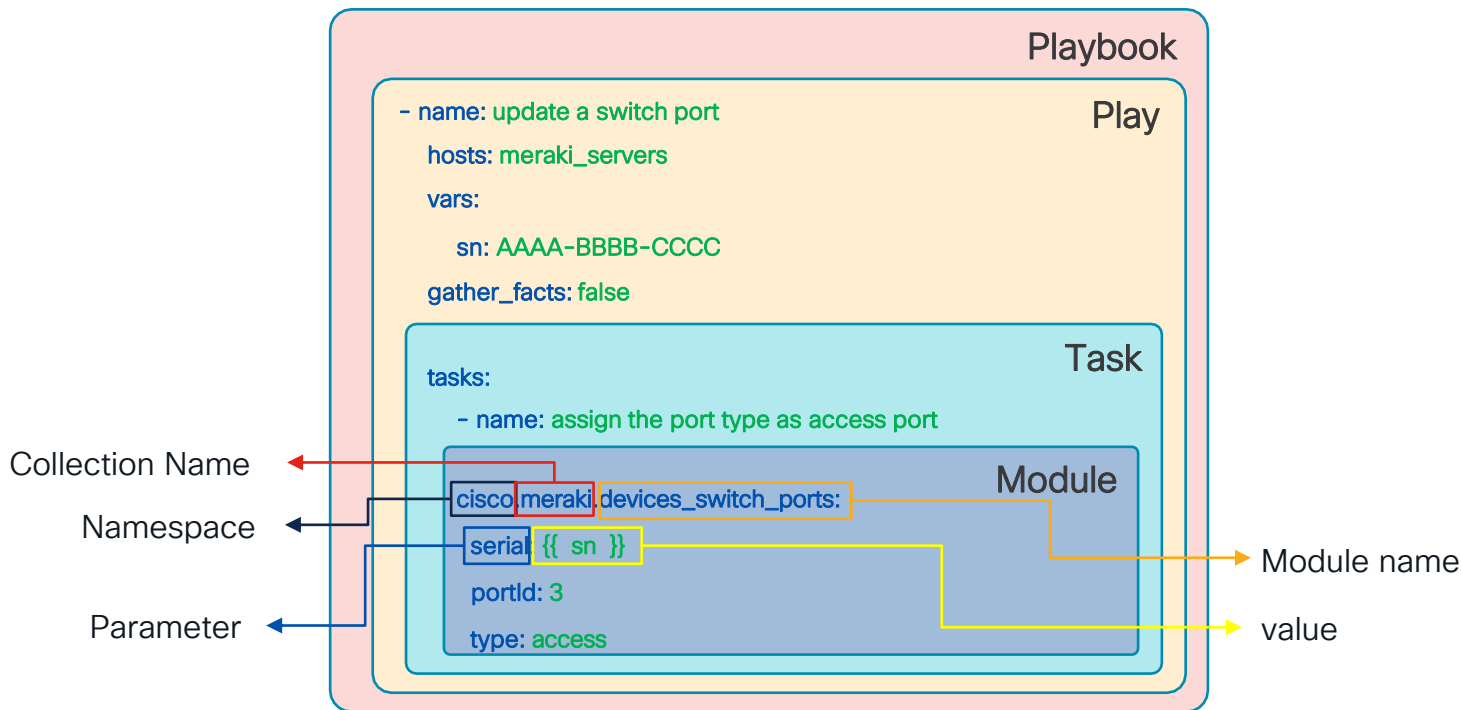
Task: single action that references a module

Module: reusable, standalone operation



Playbook

- Playbooks define the set of actions that you want Ansible to complete



Using variables

- You can define variables in your playbooks, in your inventory, in reusable files or roles, or at the command line.

Defining variables in a playbook

```
---
- name : play
  vars:
    org_id: 123456
    time_zone: "Asia/Seoul"
  tasks:
    . . .
```

Variables in a separate file

```
org_id: 123456
time_zone: "Asia/Seoul" vars.yml
```

```
---
- name: Manage Meraki Network and Devices
  hosts: localhost
  gather_facts: false
  vars_files:
    - vars.yml
  tasks:
    - name: Create a new network in Meraki
      cisco.meraki.networks:
        org_id: "{{ org_id }}"
        time_zone: "{{ time_zone }}"
```

Variables referencing

- Ansible uses Jinja2 templating to enable dynamic expressions and access to variables and facts.
- Jinja2 variables use by double curly brackets `{{ }}`.

```
---  
- name: Manage Meraki Network and Devices  
  hosts: localhost  
  gather_facts: false  
  vars_files:  
    - vars.yml  
  tasks:  
    - name: Create a new network in Meraki  
      cisco.meraki.networks:  
        org_id: "{{ org_id }}"  
        time_zone: "{{ time_zone }}"
```

`{{ <variable reference> }}`

Ansible Data Structures (YAML)

- YAML : YAML Ain't Markup Language
- Human-friendly Data Serialization Language
- Used in plays, playbooks and inventory files
- Indentaion-based: very important!
- Supports Multiple Data Types:
 - scalars (strings, numbers, booleans), lists, and dictionaries (key-value pairs)

Basic YAML syntax

```
---  
key1 : value1  
key2 : value2
```

Indicates the beginning of a
new YAML document

```
---  
ntp1 : 10.10.10.1  
ntp2 : 10.10.10.2  
ntp3 : 10.10.10.3
```

Key value mapping

```
---  
key1 : quote string  
key2 : "char like : or "  
Key3 : 'single quote'
```

Quote can be omitted for strings
Can use ' or "

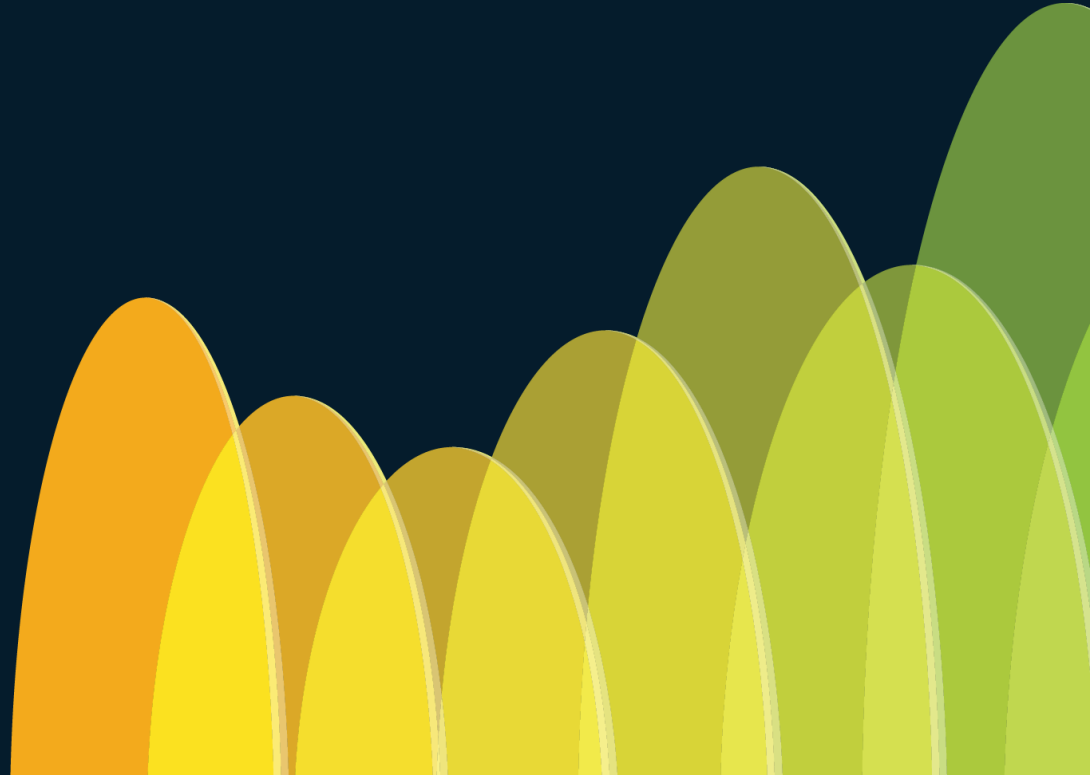
```
---  
- GigabitEthernet1/0/1  
- GigabitEthernet1/0/2  
- GigabitEthernet1/0/3
```

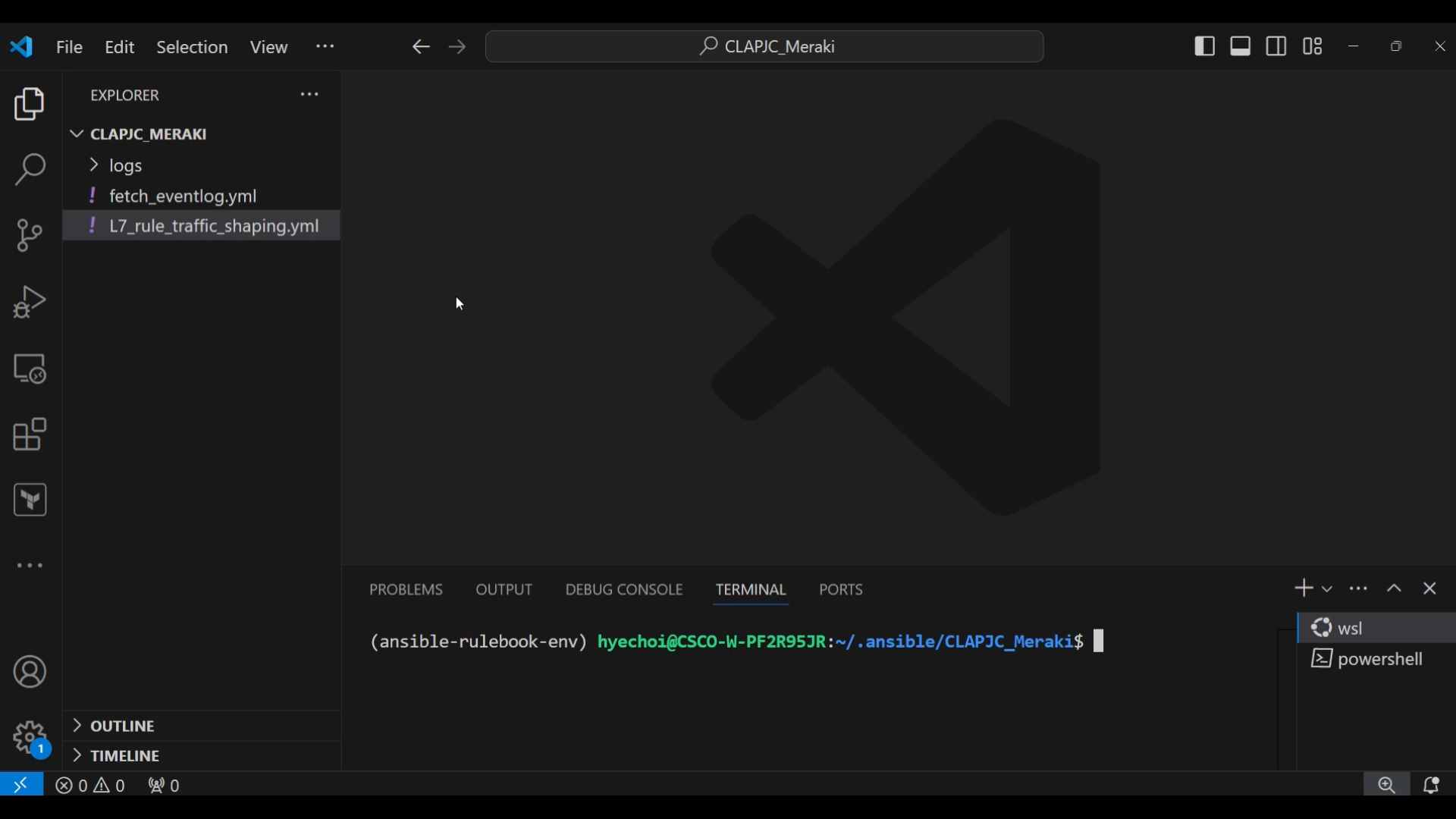
Lists are separated by a dash

```
---  
person1:  
  name: Hyeyeon Choi  
  age: 36  
  gender: female
```

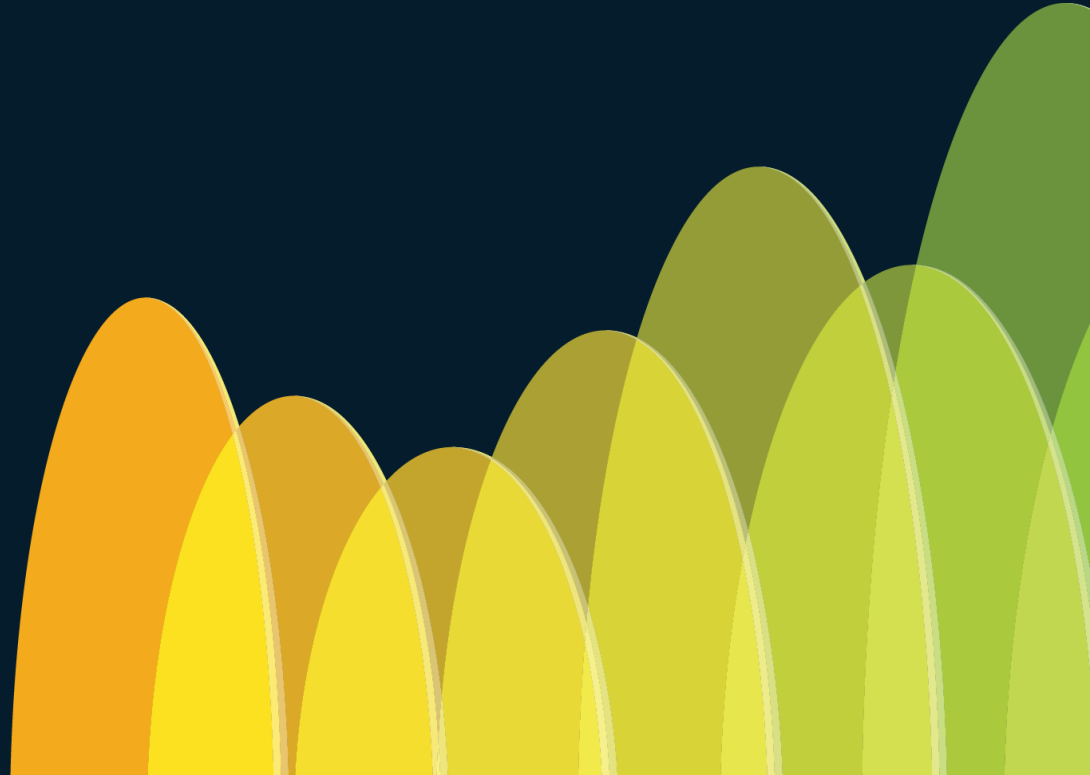
Indentation required in
nested structures
(2 or 4 space)

Let's see it in
action





Terraform vs Ansible



Ansible/Terraform comparison



laC Type	Configuration Management	Orchestration
Communication	SSH / API	API
Agent	Agentless	Agentless
Language Type	Procedural	Declarative
Stateful	No	Yes
Provisioning	Limited support for infrastructure provisioning	Specializes in infrastructure provisioning
Written in	Python	Go
Command line operation	Yes	Yes

Recap and next steps



Let's Review

- Ask yourself, why do you need automation
- Define and standardize your processes
- Select appropriate tools for your purpose
- Ansible works well for imperative and mutable use cases
- Terraform is great for deploying and destroying immutable objects
- No special programming skills required

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



Thank you

CISCO *Live!*

#CiscoLiveAPJC

CISCO *Live!*

GO BEYOND

#CiscoLiveAPJC