

The background is a vibrant, abstract graphic. It features a central bright white light source from which numerous colorful rays emanate, creating a sunburst or starburst effect. The rays transition through a spectrum of colors including yellow, orange, red, and various shades of blue and green. Overlaid on this are several large, semi-transparent, wavy shapes in similar color tones, giving the overall image a sense of motion and energy.

cisco *Live!*

Let's go

#CiscoLive



The bridge to possible

Infrastructure as Code (IAC)

with Catalyst 9000

Story DeWeese, Technical Marketing Engineer

@Story DeWeese

DEVNET-2119

CISCO *Live!*

#CiscoLive

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 June 9, 2023.



<https://ciscolive.ciscoevents.com/ciscolivebot/#DEVNET-2119>

Agenda

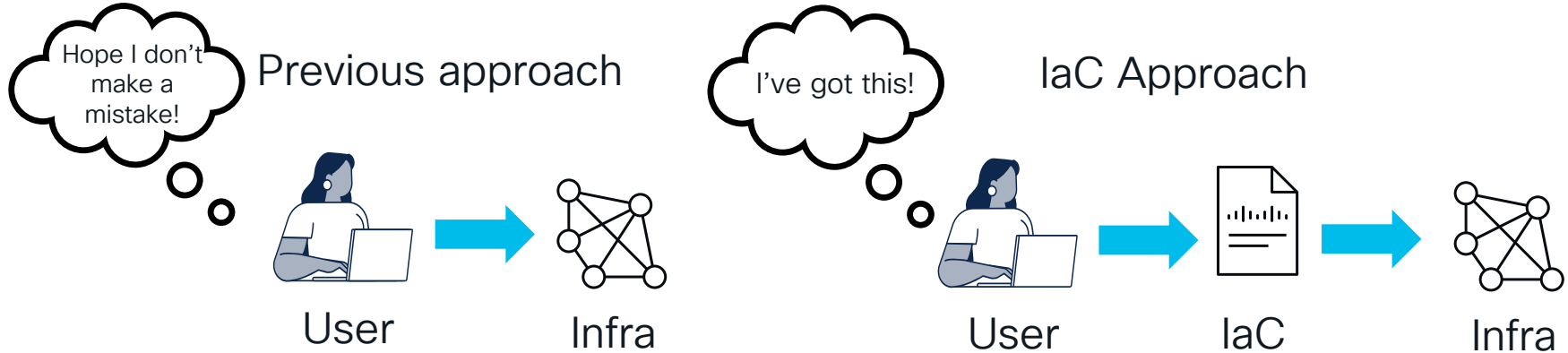
- Introduction to IAC and Terraform
- Using the IOS XE Terraform Provider
- Getting Started
- Use Cases & Demos
- Resources

Intro to AIC and Terraform



What's IaC?

Infrastructure as Code (IaC) is the process of managing changes through code, rather than a manual process



Learn more about IaC here:

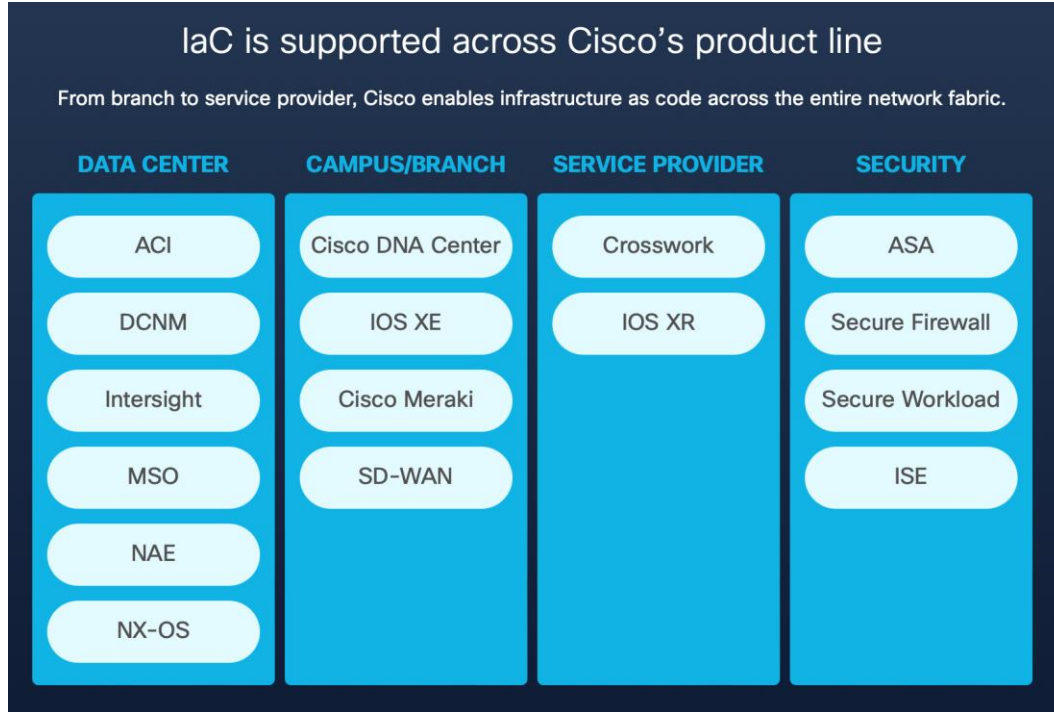
<https://developer.cisco.com/iac/#:~:text=Adopting%20Infrastructure%20as%20Code%20allows,data%20center%20to%20the%20edge>

Terraform One Slide

- **HCL** (HashiCorp Configuration Language) - Establishes the syntax Terraform uses for things like arguments, blocks, literal values, and expressions, and writing plans.
- **Provider** - Plugins responsible for understanding API interactions with other platforms and exposing resources based on their APIs.
- **Data Source** - Allows Terraform to use (read) information defined outside of Terraform. Example: providers, local-only
- **Resource** - Are the most important element in the Terraform language. Each resource block describes one or more infrastructure objects - devices, interfaces, operations.
- **Init** - The command is used to initialize a working directory containing Terraform configuration files. This is the first command that should be run, and it is safe to run this command multiple times. It will install the required providers and modules.
- **Plan** - compares the managed infrastructure state to the configuration, and it determines which changes are necessary. It presents a human-readable summary to the user.
- **Apply** - Makes changes to real infrastructure in order to make it match the desired state. It may use saved plans or creates a new plan and asks for approval.

Ref: <https://www.terraform.io/docs>

Where can I use IaC?



Terraform is...



Open-source Infrastructure as Code (IaC) Software Tool providing a consistent CLI workflow to manage hundreds of cloud services. Terraform codifies cloud APIs into declarative configuration files.

- Cloud Native Tooling circa 2014 from HashiCorp
- Agentless, single binary file
- Zero server-side dependencies

Resources:

Ask IOS XE Terraform Provider Webex space: <https://eurl.io/#PtsT8eJFI>

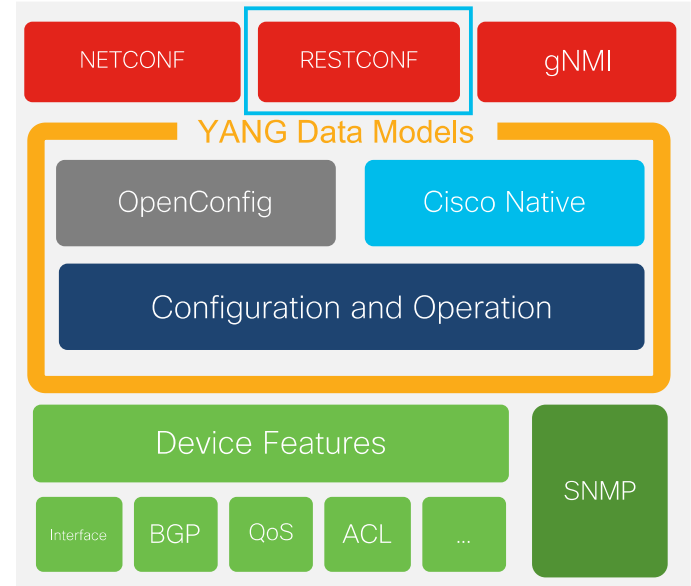
GitHub Provider Examples: <https://github.com/CiscoDevNet/terraform-provider-iosxe/>

Provider Binary: <https://registry.terraform.io/search/providers?namespace=CiscoDevNet>

Go Client: <https://github.com/CiscoDevNet/iosxe-go-client>

Blogs at <https://blogs.cisco.com/tag/terraform>

Terraform uses the RESTCONF API



Using the IOS XE Terraform Provider



IOS XE Programmability integration with Terraform

Terraform is supported on all IOS XE platforms

Phase I: *imperative* for 100% feature coverage (available today) The following features are delivered:

- This Terraform provider is a generic REST resource for IOS XE RESTCONF YANG
- Hashicorp Config Language (HCL) support for management of IOS XE
- RESTCONF operations for PUT/PATCH/POST etc still must be followed for iterative management
- Examples and JSON mappings for top features are shared in GitHub
- Any feature supported by RESTCONF/YANG is supported iteratively by this Terraform provider

| | | |
|-------------------|--------------------|---------------------|
| L3 subinterface | Authentication AAA | IGMP |
| VLAN | MDT | IPsec |
| Voice VLAN Trunk | SPAN and RSPAN | NAT |
| VTP | SNMP | NTP |
| Line | CDP | HSRP |
| ACL | EtherChannel | DHCP |
| RADIUS | OSPF | Ethernet Management |
| Accounting AAA | BGP | Port |
| Authorization AAA | IGMP Proxy | POE |

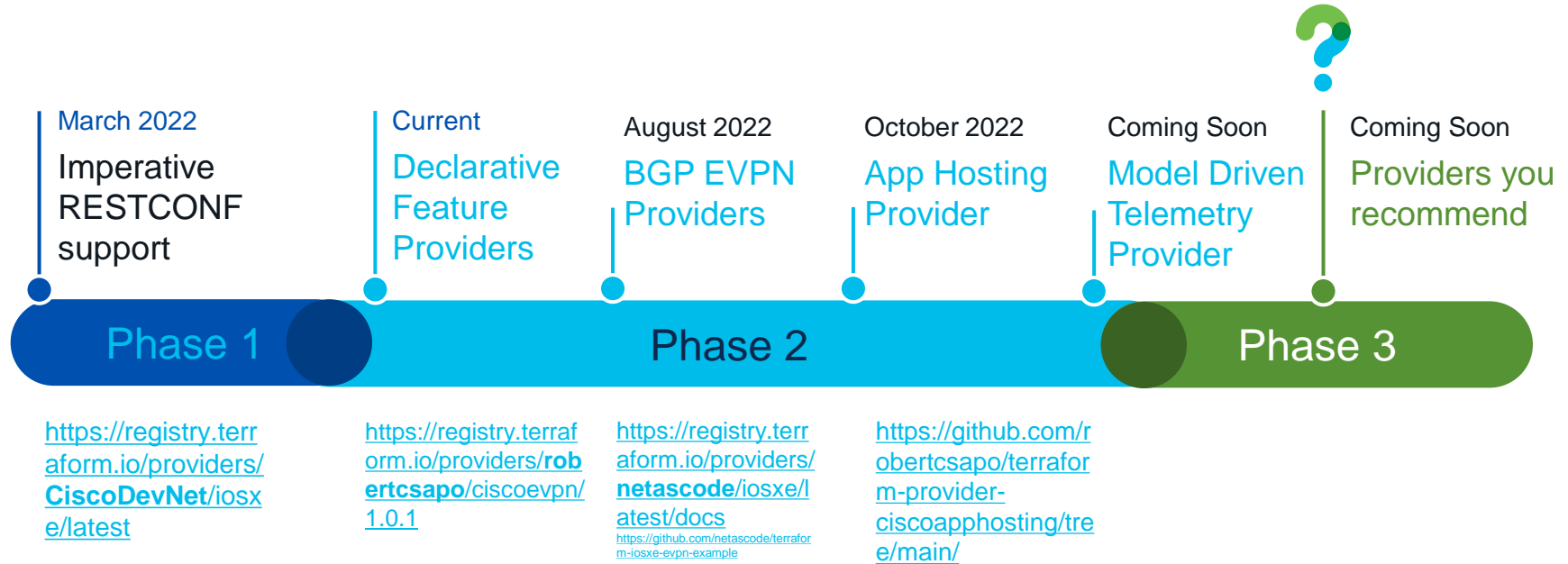
Resources:

GitHub Provider Examples: <https://github.com/CiscoDevNet/terraform-provider-iosxe/>
Provider Binary: <https://registry.terraform.io/search/providers?namespace=CiscoDevNet>
Go Client: <https://github.com/CiscoDevNet/iosxe-go-client>
Blogs at <https://blogs.cisco.com/tag/terraform>



Phase II: new *declarative* features

Evolution of Terraform Provider



Declarative providers leverage the SDK from the Phase 1 imperative provider

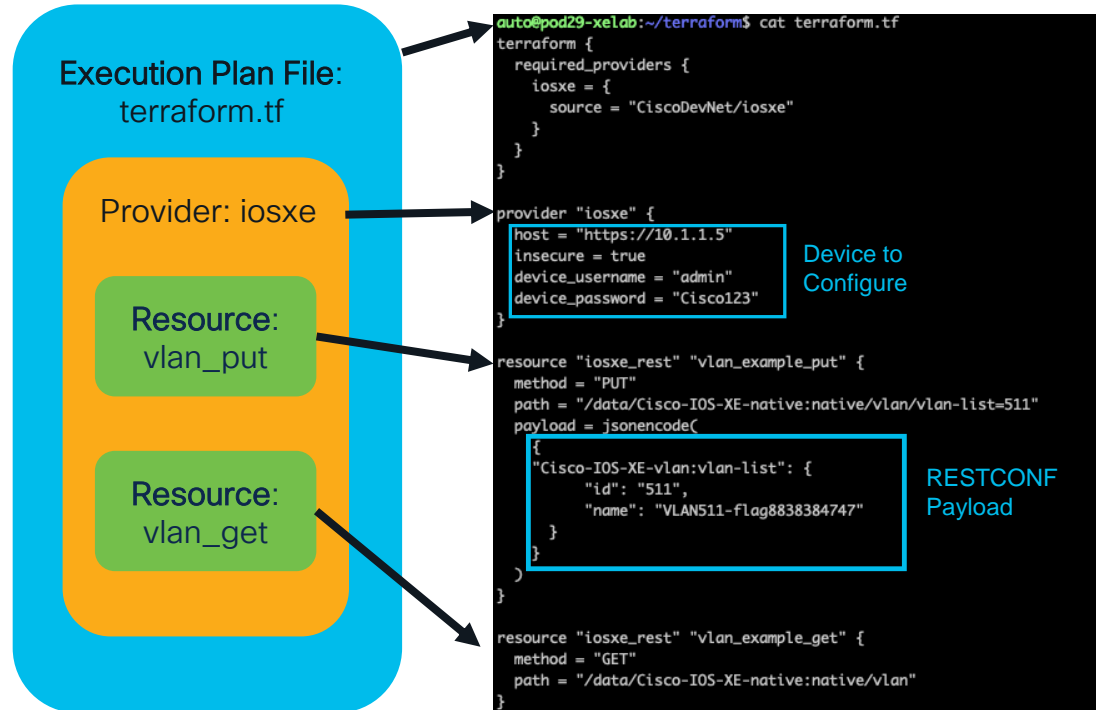
Terraform Terminology

Terraform uses an execution plan file with a provider and resource definitions.

An **execution plan file** defines the provider and resources. It is written in HashiCorp Configuration Language (HCL), similar to JSON, and stored with a .tf extension

A **provider** is a plugin to make a collection of resources accessible

A **resource** (or infrastructure resource) describes one or more infrastructure objects managed by Terraform. With the IOS XE Terraform provider, resources can be considered the same as a configurable feature



Getting Started



Prerequisites: Enable AAA, NETCONF & RESTCONF

```
Cat9k-1#conf t
```

```
Enter configuration commands, one per line.  End with  
CNTL/Z.
```

```
Cat9k-1(config)#aaa new-model
```

Enable AAA

```
Cat9k-1(config)#aaa authentication login default local
```

```
Cat9k-1(config)#aaa authorization exec default local
```

```
Cat9k-1(config)#username admin privilege 15 password cisco
```

```
Cat9k-1(config)#netconf-yang
```

Enable NETCONF

```
Cat9k-1(config)#restconf
```

Enable RESTCONF

Getting Started with Terraform + IOS XE Provider

1. Enabling the RESTCONF API on the switch

```
Switch# conf t  
Switch(config)# restconf
```

2. Install [Terraform](#)

```
$ apt install terraform
```

3. Clone the [IOS XE Terraform Provider](#) GitHub repository

```
$ git clone https://github.com/CiscoDevNet/terraform-provider-  
iosxe
```

4. Apply Terraform VLAN example

```
$ terraform apply acl_and_vlan.tf
```


CLI to YANG

This new CLI addition to “show run | format” brings additional visibility into the YANG modelled configuration, either for NETCONF with XML or JSON with RESTCONF
Easily convert CLI into YANG to re-use in tooling, scripts, and automation and orchestration systems

```
show run | format netconf-xml
show run | format restconf-json
```

```
C9300#show run | format netconf-xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
    <version>17.7</version>
    <memory>
      <free>
        <low-watermark>
          <processor>131752</processor>
        </low-watermark>
      </free>
    </memory>
  </native>
</config>
```

```
C9300#show run | format restconf-json
{
  "data": {
    "Cisco-IOS-XE-native:native": {
      "version": "17.7",
      "memory": {
        "free": {
          "low-watermark": {
            "processor": 131923
          }
        }
      }
    }
  }
}
```

```
C9300#
C9300#show run | i netconf-yang
netconf-yang
C9300#
```

Requires `netconf-yang` Data Model Interfaces to be enabled
CLIs with corresponding native YANG and modeled in show run are returned

DEVNET-2119

17

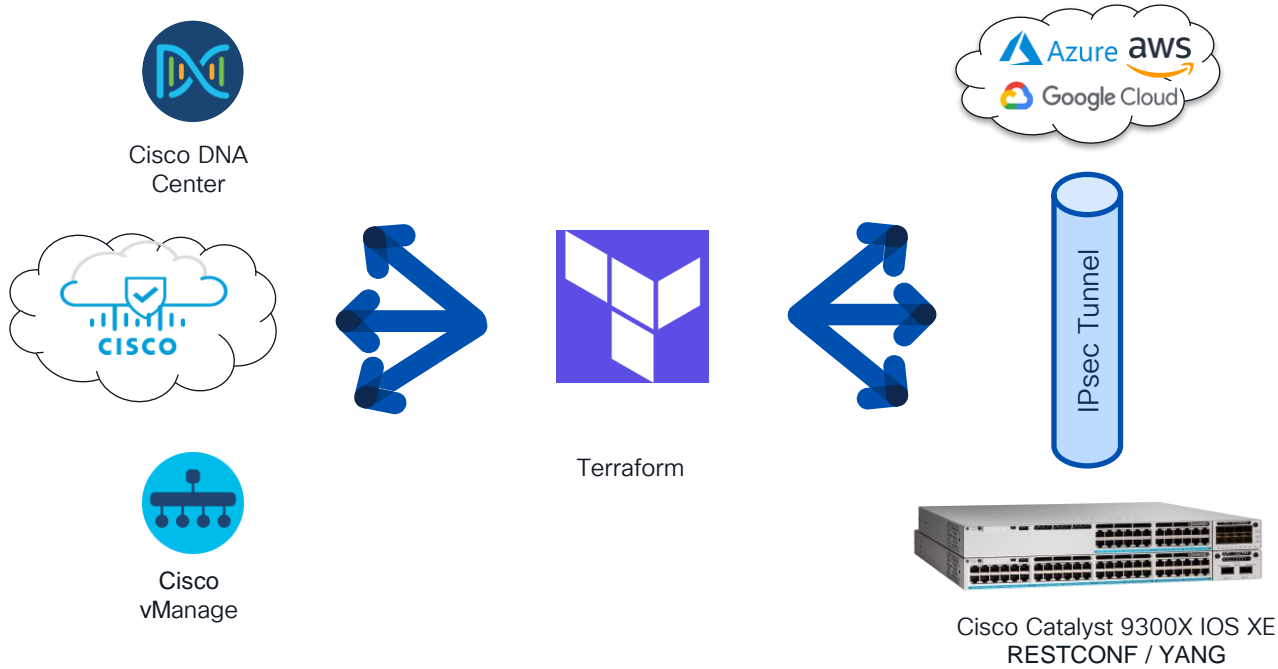
Use Cases & Demos



CISCO *Live!*

https://github.com/CiscoDevNet/terraform-provider-iosxe/tree/main/examples/tutorials/acl_and_vlan

What can Terraform do?



Terraform + Crypto IPsec Demo

The screenshot displays a development environment for a Terraform demo. At the top, four tabs are open: `terraform.tf`, `variables.tf`, `9300X.tfvars`, and `crypto.txt`. The `terraform.tf` tab is active, showing the following HCL code:

```
1 # See preview using: terraform plan -var-file="9300X.tfvars"
2 # Run using: terraform apply -var-file="9300X.tfvars" -auto-approve
3
4 terraform {
5   required_providers {
6     iosxe = {
7       source = "CiscoDevNet/iosxe"
8       version = "0.1.1"
9     }
10  }
11 }
12
13 provider "iosxe" { # variables initialized in variables.tf and values stored in 9300X.tfvars
14   host = var.host_url
15   insecure = var.insecure
16   device_username = var.device_username
17   device_password = var.device_password
18 }
19
20 # crypto all
21 resource "iosxe_rest" "crypto_example_post" {
22   method = "PATCH"
23   path = "/data/Cisco-IOS-XE-native:native/crypto"
24   payload = jsonencode(
25
26 {
27   "Cisco-IOS-XE-native:crypto": {
28     "Cisco-IOS-XE-crypto:ikev2": {
```

Below the code editor, the **TERMINAL** tab is selected, showing a series of commands and their outputs. The prompt is `sdeweese@SDEWEESE-M-KQCG sevt %`. The output shows a list of `9300X-Edge-2#` commands being executed, likely related to the IPsec configuration.

On the right side of the interface, a network diagram is visible, showing a hierarchical structure of network devices and their connections, with various status indicators and labels.

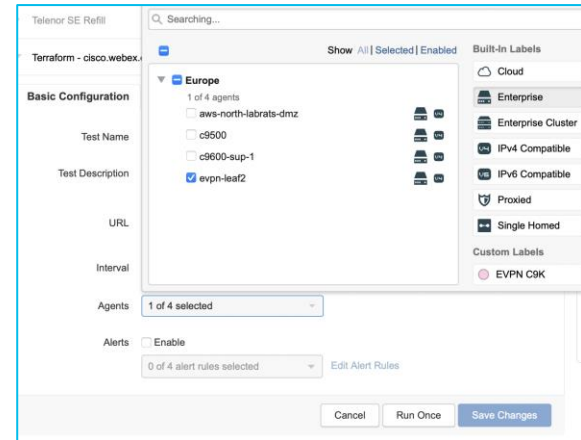
Terraform ThousandEyes lifecycle management

1. Deploy TE agent on switch Catalyst 9000
2. Pass variables including the the Agent ID to the ThousandEyes API
3. Create test and attach the Catalyst 9000 TE Agent ID to the test
4. Trigger test to run

```
terraform {
  required_providers {
    ciscoapphosting = {
      source = "robertcsapo/ciscoapphosting"
      version = "1.0.0"
    }
  }
}

provider "ciscoapphosting" {
  username = var.username
  password = var.password
  insecure = var.insecure
  timeout = var.timeout
}

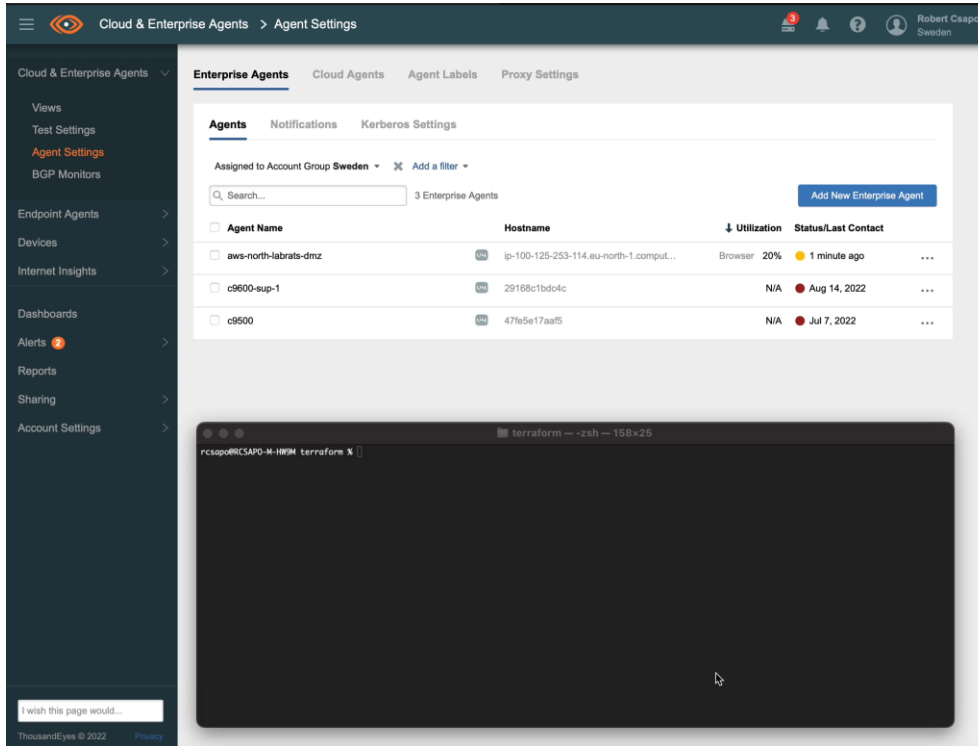
resource "ciscoapphosting_app" "app" {
  host = "127.0.0.1"
  image = "https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-4.2.2.cisco.tar"
  app_gigabit_ethernet = "1/0/1"
  vlan_trunk = false
  vlan = 1
  env = {
    TEAGENT_ACCOUNT_TOKEN = "token"
  }
}
```



<https://github.com/robertcsapo/terraform-provider-ciscoapphosting>

<https://registry.terraform.io/providers/robertcsapo/ciscoapphosting/>

Terraform + ThousandEyes Demo



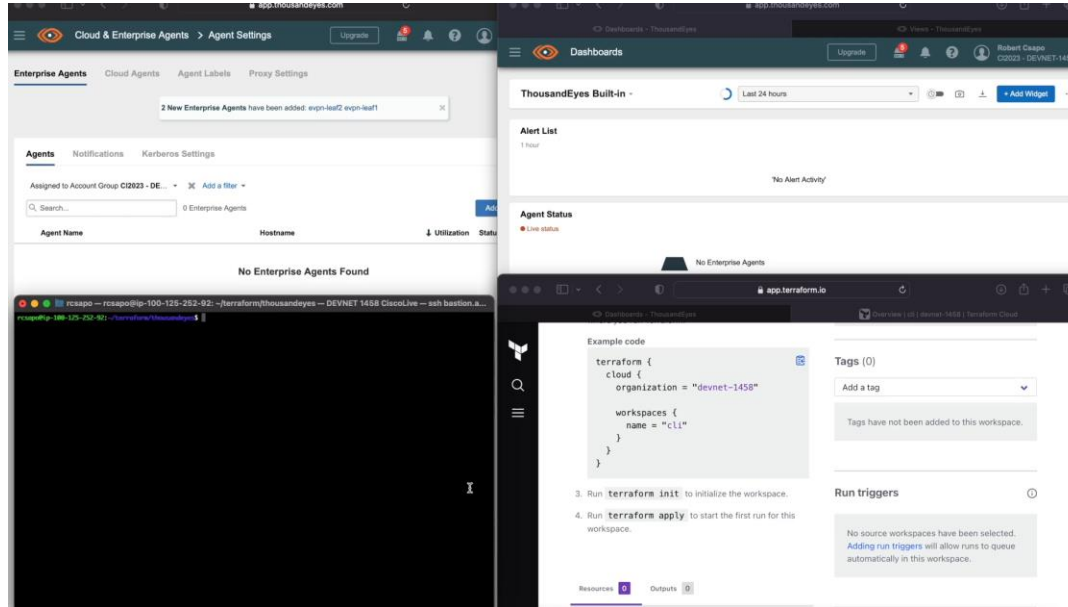
Steps:

1. Test App Hosting Terraform Provider
2. Apply App Hosting Terraform Provider on C9300 switches
3. Verify new Enterprise Agents has been added in ThousandEyes Dashboard
4. Run the current test
5. Destroy Terraform to delete the Enterprise Agents and Test

Multihost example:

<https://github.com/robertcsapo/terraform-provider-ciscoapphosting/tree/main/examples/thousandeyes-multihosts>

Terraform + ThousandEyes Demo #2



Both Declarative providers leverage the ios-xe-go client that was developed in Terraform Phase 1:

EVPN Overlay Module: <https://registry.terraform.io/modules/netascode/evpn-overlay/iosxe/latest>

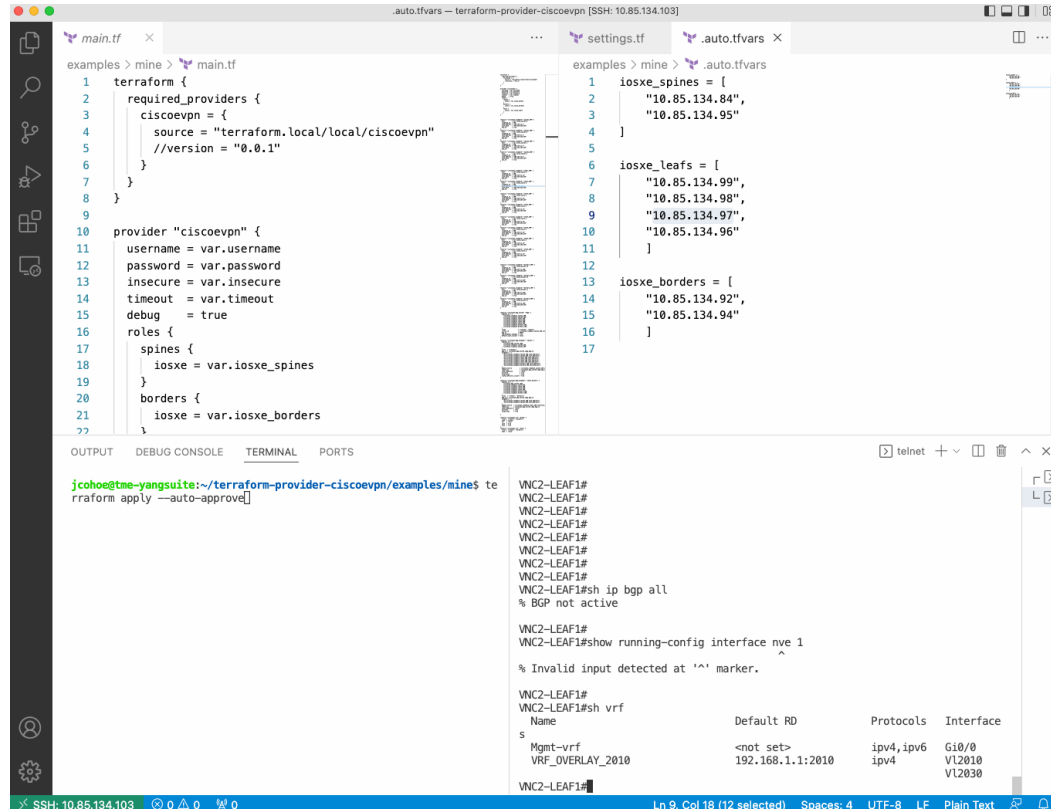
Terraform will perform the following actions:

```
# ciscoevpn_vrf.blue will be destroyed
```

[illegible]

To point this to your own Cat9k fabric, update the `data/inventory.yaml` file accordingly.

EVPN with Terraform Demo



The screenshot displays a Terraform demo environment. The top section shows two Terraform configuration files: `main.tf` and `settings.tf`. The `main.tf` file defines the provider and roles, while `settings.tf` defines the variables for the network configuration.

```
main.tf
1 terraform {
2   required_providers {
3     ciscoevpn = {
4       source = "terraform.local/local/ciscoevpn"
5       //version = "0.0.1"
6     }
7   }
8 }
9
10 provider "ciscoevpn" {
11   username = var.username
12   password = var.password
13   insecure = var.insecure
14   timeout  = var.timeout
15   debug    = true
16   roles {
17     spines {
18       iosxe = var.iosxe_spines
19     }
20     borders {
21       iosxe = var.iosxe_borders
22     }
23   }
24 }
```

```
settings.tf
1 iosxe_spines = [
2   "10.85.134.84",
3   "10.85.134.95"
4 ]
5
6 iosxe_leaves = [
7   "10.85.134.99",
8   "10.85.134.98",
9   "10.85.134.97",
10  "10.85.134.96"
11 ]
12
13 iosxe_borders = [
14   "10.85.134.92",
15   "10.85.134.94"
16 ]
17
```

The bottom section shows a terminal window with the following commands and output:

```
jchooe@tme-yangsuite:~/terraform-provider-ciscoevpn/examples/mine$ terraform apply --auto-approve
```

Output:

```
VNC2-LEAF1#
VNC2-LEAF1#
VNC2-LEAF1#
VNC2-LEAF1#
VNC2-LEAF1#
VNC2-LEAF1#
VNC2-LEAF1#
VNC2-LEAF1#sh ip bgp all
% BGP not active

VNC2-LEAF1#
VNC2-LEAF1#show running-config interface nve 1
^
% Invalid input detected at '^' marker.

VNC2-LEAF1#
VNC2-LEAF1#sh vrf

```

| Name | Default RD | Protocols | Interface |
|------------------|------------------|------------|-----------|
| Mgmt-vrf | <not set> | ipv4, ipv6 | Gi0/0 |
| VRF_OVERLAY_2010 | 192.168.1.1:2010 | ipv4 | VL2010 |
| | | | VL2030 |

Resources

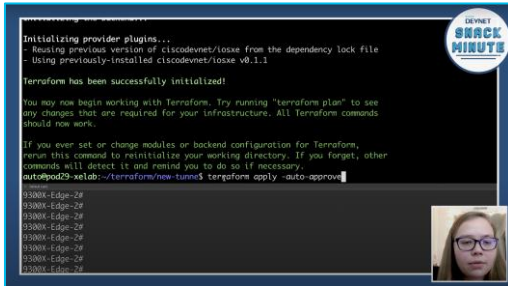
Terraform Blog and Resources

Questions? Join the Ask
IOS XE Terraform Provider
Webex space:

<https://eurl.io/#PtsT8eJFI>



<https://github.com/CiscoDevNet/terraform-provider-iosxe/>
<https://registry.terraform.io/search/providers?namespace=CiscoDevNet>
<https://developer.cisco.com/automation-terraform/>



Demo Create a Crypto Tunnel Video:

<https://www.youtube.com/watch?v=bPS0bhPacDw>



Intro to IOS XE Terraform Provider Video:

https://www.youtube.com/watch?v=GEY_hyXimbA

Introducing Terraform with IOS XE

Code Included

Developer

Automation with Any Tooling on Any Interface

Story DeWeese

Terraform expands into the extensive Cisco IOS XE programmability and automation ecosystem

IOS XE's vast, programmable feature set

The Cisco IOS XE ecosystem is programmatically managed and supports a variety of tooling. This includes Ansible to YANG Suite, pyATS over NETCONF, RESTCONF, gNMI, and even with legacy CLIs. With the addition of the new Cisco IOS XE Terraform provider, we add an additional tool into the IOS XE configuration management toolbox.

<https://blogs.cisco.com/developer/terraformiosxe01>

Cisco IOS XE Programmability Sessions

This week & on demand



January 31, 2023

[Leave a Comment](#)



Developer

Explore Cisco IOS XE Automation at Cisco Live EMEA 2023

[Story DeWeese](#)

Ready to dive deeper into Cisco IOS-XE programmability, automation, and telemetry? And planning to attend Cisco Live EMEA in Amsterdam, February 6 - 10, 2023? Then I hope you'll join us for some great learning sessions. These sessions cover topics ranging from:

- getting started with programmability and automation
- tooling with YANG Suite and Terraform
- gaining hands-on experience with open source solutions for model driven telemetry and machine learning

<https://blogs.cisco.com/developer/iosxecleMEA01>

Fill out your session surveys!



Attendees who fill out a minimum of four session surveys and the overall event survey will get **Cisco Live-branded socks** (while supplies last)!



Attendees will also earn 100 points in the **Cisco Live Challenge** for every survey completed.



These points help you get on the leaderboard and increase your chances of winning daily and grand prizes

Continue your education



- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer 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



The bridge to possible

Thank you

CISCO *Live!*

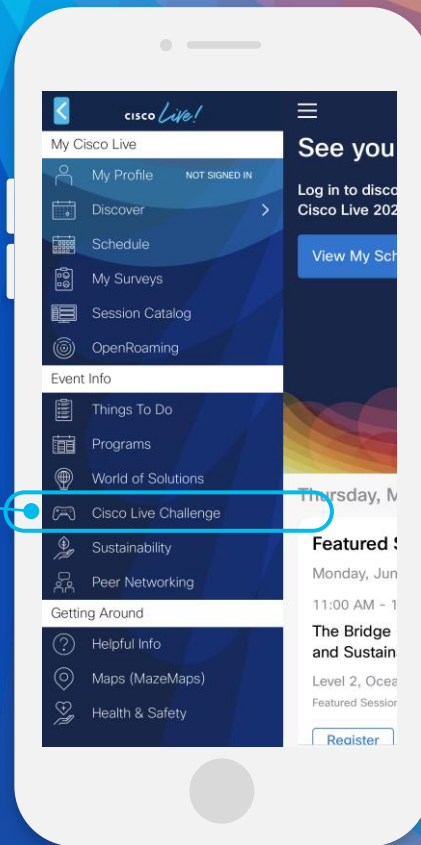
#CiscoLive

Cisco Live Challenge

Gamify your Cisco Live experience!
Get points for attending this session!

How:

- 1 Open the Cisco Events App.
- 2 Click on 'Cisco Live Challenge' in the side menu.
- 3 Click on View Your Badges at the top.
- 4 Click the + at the bottom of the screen and scan the QR code:



The background features a vibrant, multi-colored abstract design. On the left, there are overlapping, wavy bands of color in shades of orange, red, and yellow. On the right, a bright white light source emits a series of colorful rays in shades of blue, green, and yellow, creating a sunburst effect. The overall composition is dynamic and energetic.

cisco *Live!*

Let's go

#CiscoLive