

The background features a vibrant, multi-colored abstract design. On the left, there are horizontal, wavy bands of color in shades of red, orange, yellow, and green. On the right, a bright white light source emits a series of sharp, triangular rays in various shades of blue and cyan, creating a sunburst effect that spreads across the right half of the image.

cisco *Live!*

Let's go

#CiscoLiveAPJC



The bridge to possible

Cisco IOS XE & Terraform

How to become an IOS XE Terraform Expert

Jeremy Cohoe, Technical Marketing
@jeremycohoe
DEVNET-2476

CISCO *Live!*

#CiscoLiveAPJC

Infrastructure as Code (IAC) allows us to automate provisioning and management of Cisco Catalyst 9000 devices more efficiently. In this session, automation tooling will be used to programmatically deploy BGP EVPN network fabric. Infrastructure as Code concepts will be demonstrated on the Catalyst 9000 using Terraform. Both enterprise campus and cloud-native use cases and examples will be shared. You'll leave this session with ideas to help you get started with Infrastructure as Code, understand benefits of IAC, and understand more about the Terraform tooling. With IAC and Cisco IOS XE, you can automate any tooling on any interface.

Cisco Webex App

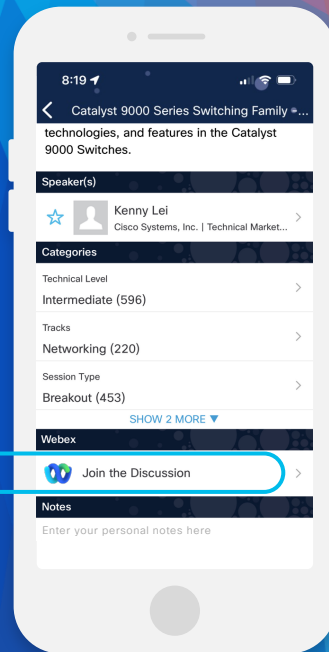
Questions?

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 December 22, 2023.



<https://ciscolive.ciscoevents.com/ciscolivebot/#DEVNET-2476>

Agenda

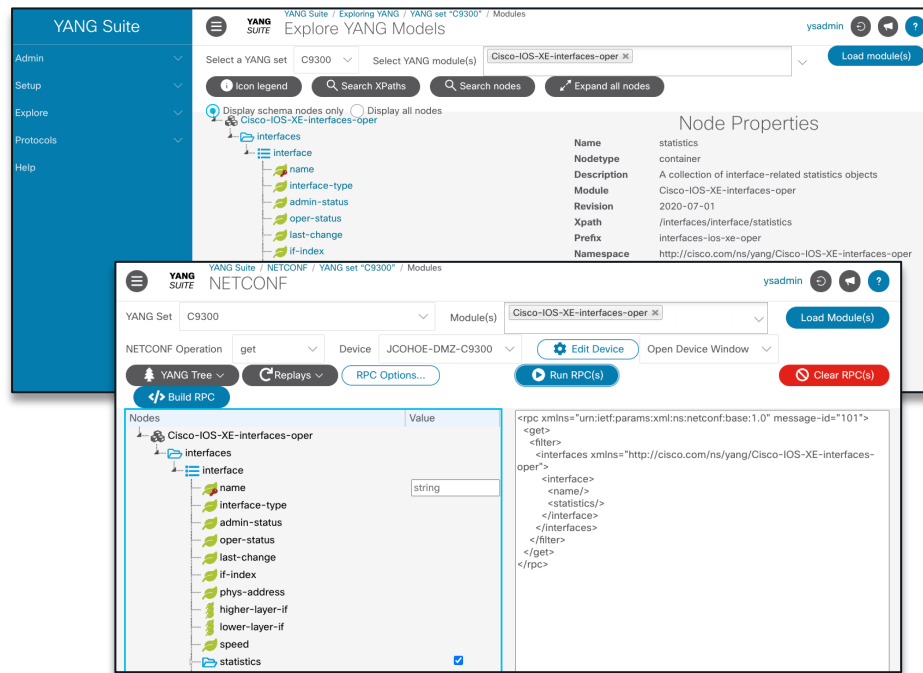
- Introduction
- Cisco YANG Suite
- IOS XE Capabilities
- Terraform's IOS XE Integration
- Use Case Examples
- Conclusion

Cisco YANG Suite

Cisco YANG Suite



- YANG API Testing and Validation Environment
- Construct and test YANG based APIs over NETCONF, RESTCONF, gRPC and gNMI
- IOS XE / IOS XR / NX OS platforms



Now Generally Available !

developer.cisco.com/yangsuite

github.com/CiscoDevNet/yangsuite

YANG Suite

- Initial Release: YANG Suite + core plugins
- Second release: RESTCONF, gNMI
- Next release: June 2023
 1. gRPC Dial-Out with TLS Support
 2. SNMP OID to YANG XPATH
 3. NETCONF/RESTCONF/gNMI to Ansible YAML
 4. Pip install support + Docker

<https://github.com/CiscoDevNet/yangsuite/>

<https://developer.cisco.com/yangsuite/>

<https://eurl.io/#MaW78CeIS> YANG Suite General (external)

Cisco YANG Suite

Admin



Setup



Analytics



Datasets and diffs

SNMP to YANG Mapping

YANG coverage

Explore



YANG

Protocols



gNMI

gRPC telemetry

NETCONF

RESTCONF

Test Manager



Help

Terraform uses the RESTCONF API on IOS XE



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.

Resources:

Ask IOS XE Terraform Provider Webex space:

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

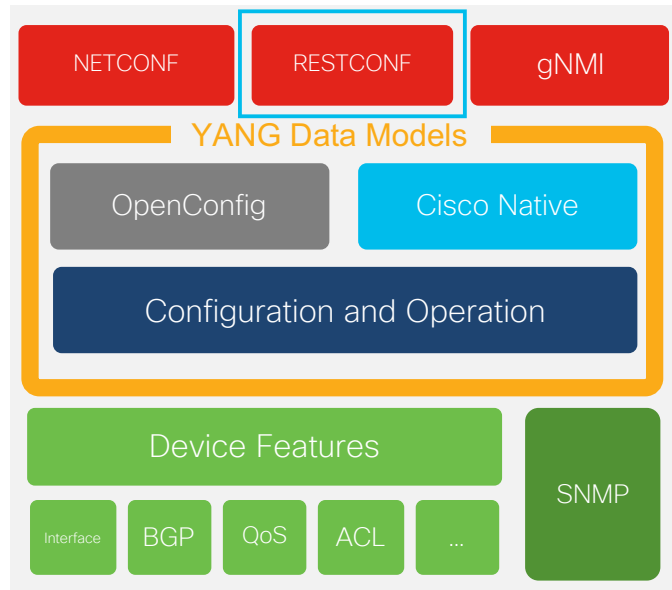
GitHub Repository:

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

Provider Documentation:

<https://registry.terraform.io/providers/CiscoDevNet/iosxe/latest>

Terraform uses the RESTCONF API



IOS XE

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

CLI to YANG

This CLI “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

YANG model for CLI execution

Any configure CLI can now be sent within the YANG payload

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
```

```
  message-id="101">
```

```
  <config-ios-cli-rpc xmlns=http://cisco.com/ns/yang/Cisco-IOS-XE-cli-rpc
```

```
    <config-clis>
```

```
      interface Loopback111
```

```
        description configured-via-CLI-YANG
```

```
        no shutdown
```

```
      </config-clis>
```

```
    </config-ios-cli-rpc>
```

```
  </rpc>]]>]]>
```

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
```

```
  message-id="101">
```

```
  <config-ios-cli-trans xmlns=http://cisco.com/ns/yang/Cisco-IOS-XE-cli-rpc
```

```
    <clis>
```

```
      interface Loopback111
```

```
        description configured-via-CONFED-YANG
```

```
        no shutdown
```

```
    </clis>
```

```
  </config-ios-cli-trans>
```

```
</rpc>]]>]]>
```



clis

“cli rpc” sends CLI to the IOS parser

This is similar to configuring CLI on the VTY

Directly into running-config, then synchronized to ConfD

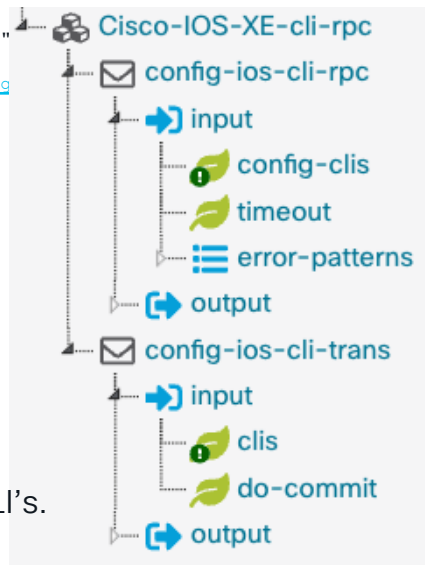


config-clis

“transitional cli rpc” sends a list of CLI to ConfD

This is similar to sending edit-config RPCs corresponding to the CLI's.

Synchronized from ConfD into the CLI running-config



<https://github.com/YangModels/yang/blob/main/vendor/cisco/xe/1791/Cisco-IOS-XE-cli-rpc.yang>

CLI RPC: Operational examples

Use the CLI RPC to send un-structured CLI over YANG to:

- Configure single feature: conf t
- Configure multiple features
- Privileged EXEC commands: clock/clear
- Show commands: show ver
- Show commands with redirect: show ver | redirect tftp

Config-IOS-CLI-RPC example payloads:

```
hostname jcohoe

int lo99
ip address 192.169.254.1 255.255.255.0

no hostname
no int lo99
hostname jcohoe-c9300

do clock set 16:20:59 sep 12 2023

do sh ver | redirect flash:shver.txt
do sh tech | redirect tftp://10.1.1.3/sh-tech.txt
```

```
Sending:

#336
<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:7c8e9760-222d-4b4b-
  <config-cli>

do sh ver | redirect flash:shver.txt

</config-cli>
</config-ios-cli-rpc>
</nc:rpc>

##

Received message from host

<?xml version="1.0" ?>
<rpc-reply message-id="urn:uuid:7c8e9760-222d-4b4b-a94d-f49043eab97f" xmlns="urn:ietf:params:xml:ns:
  <result xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-cli-rpc">RPC request successful</result>

</rpc-reply>
```

Terraform resource utilizing the CLI RPC

<https://registry.terraform.io/providers/CiscoDevNet/iosxe/latest/docs/resources/cli>

iosxe_cli (Resource)

This resource is used to configure arbitrary CLI commands. This should be considered a last resort in case YANG models are not available, as it cannot read the state and therefore cannot reconcile changes.

Example Usage

```
resource "iosxe_cli" "example" {  
  cli = <<-EOT  
  interface Loopback123  
  description configured-via-restconf-cli  
  EOT  
}
```

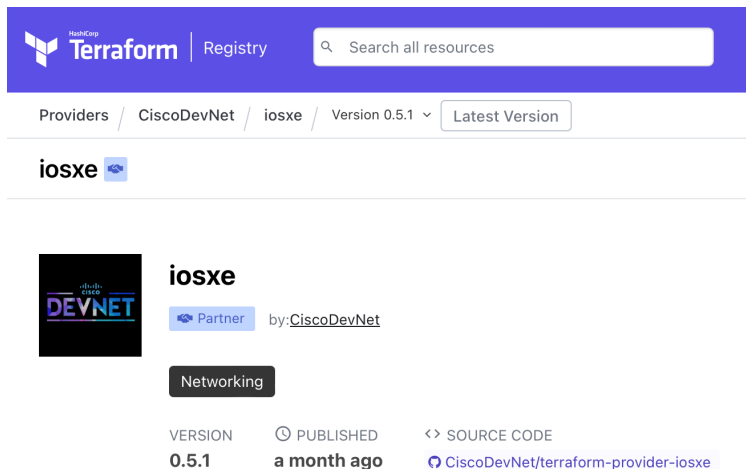
Terraform on IOS XE

IOS XE Terraform Provider

Documentation and details about the provider are available on the Hashicorp Registry

<https://registry.terraform.io/providers/CiscoDevNet/iosxe/latest>

Source code is in the GitHub Repository: <https://github.com/CiscoDevNet/terraform-provider-iosxe/>



The screenshot shows the HashiCorp Terraform Registry interface. At the top, there's a search bar and navigation links for Providers, CiscoDevNet, and iosxe. The current version is 0.5.1, with a link to the latest version. Below the navigation, the provider name 'iosxe' is displayed with a Cisco DevNet logo. A 'Partner' badge and 'by: CiscoDevNet' are shown. A 'Networking' tag is present. The version information shows 'VERSION 0.5.1' and 'PUBLISHED a month ago'. A link to the 'SOURCE CODE' is provided, pointing to the GitHub repository.

iosxe provider

> Guides

> AAA

> BFD

> BGP

> CTS

> Crypto

> EVPN

> General

> Interface

> MDT

> MPLS

> Management

> Multicast

> OSPF

> QoS

> Routing

> Switching

> System

> VRF

▼ General

▼ Resources

iosxe_cli

iosxe_restconf

iosxe_save_config

▼ Data Sources

iosxe_restconf

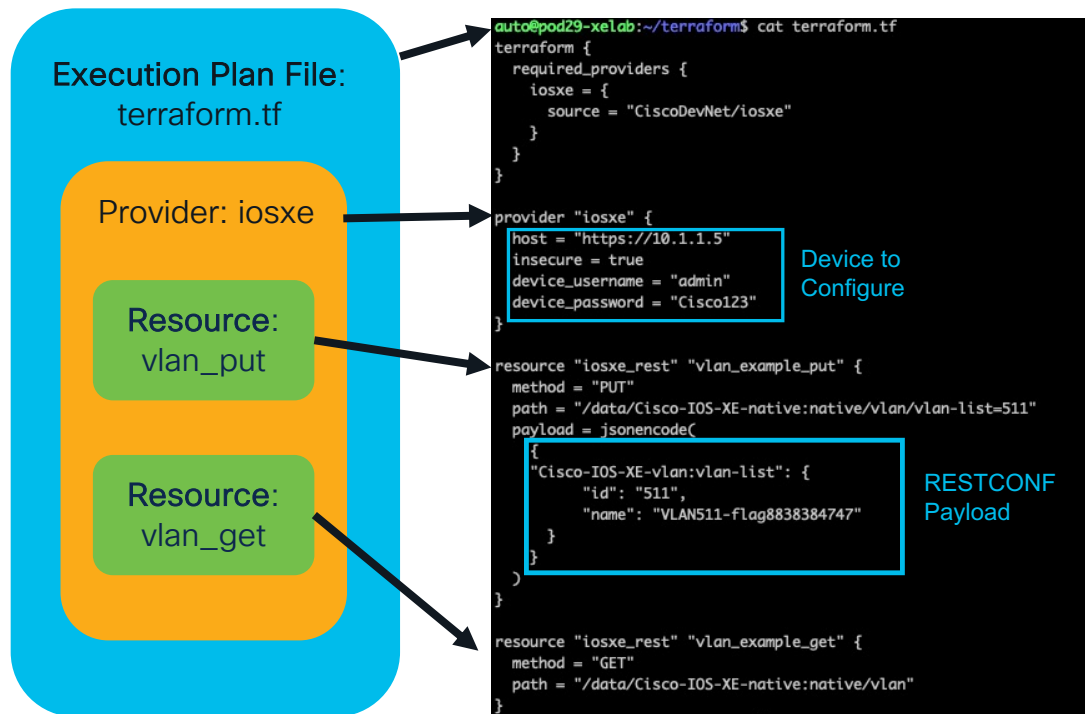
Terraform Terminology

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 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
```

Use Case Examples



Terraform for Model Driven Telemetry

Enable gRPC Dial-Out telemetry subscriptions for the POE and Basic Device Monitoring use cases

```
auto@pod22-xelab:~/ndt_examples$ terraform plan -var-file="var.tfvars"
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# iosxe_mdt_subscription.cpu_subs["100"] will be created
+ resource "iosxe_mdt_subscription" "cpu_subs" {
  + encoding      = "encode-kvgpb"
  + filter_xpath  = "/process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds"
  + id            = (known after apply)
  + receivers     = [
    + {
      + address = "10.1.1.3"
      + port    = 57500
      + protocol = "grpc-tcp"
    },
  ]
  + stream        = "yang-push"
  + subscription_id = 100
  + update_policy_periodic = 3000
}

# iosxe_mdt_subscription.cpu_subs["101"] will be created
+ resource "iosxe_mdt_subscription" "cpu_subs" {
  + encoding      = "encode-kvgpb"
  + filter_xpath  = "/process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/one-minute"
  + id            = (known after apply)
  + receivers     = [
    + {
      + address = "10.1.1.3"
      + port    = 57500
      + protocol = "grpc-tcp"
    },
  ]
  + stream        = "yang-push"
  + subscription_id = 101
  + update_policy_periodic = 3000
}
```

Lab Guide Steps:

1. Introduction
2. IOS XE CLI pre-req
3. Terraform Install & headers, device variables
4. Variables and config for MDT subscription
5. Terraform configuration for device monitoring
6. TF Workflow: init, plan, apply & destroy
7. Validation with TF and CLI
8. Conclusion

https://registry.terraform.io/providers/CiscoDevNet/iosxe/latest/docs/resources/mdt_subscription
https://registry.terraform.io/providers/CiscoDevNet/iosxe/latest/docs/data-sources/mdt_subscription
<https://github.com/jeremycohoe/cisco-ios-xe-panda-lab-terraform>

iosxe_mdt_subscription (Resource)

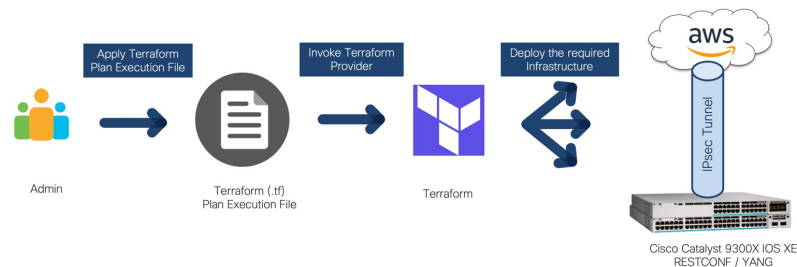
This resource can manage the MDT Subscription configuration.

Example Usage

```
resource "iosxe_mdt_subscription" "example" {
  subscription_id = 101
  stream          = "yang-notif-native"
  encoding        = "encode-kvgpb"
  source_vrf      = "Mgmt-vrf"
  source_address  = "1.2.3.4"
  update_policy_on_change = true
  filter_xpath    = "/ios-events-ios-xe-oper"
  receivers = [
    {
      address = "5.6.7.8"
      port    = 57600
      protocol = "grpc-tcp"
    }
  ]
}
```

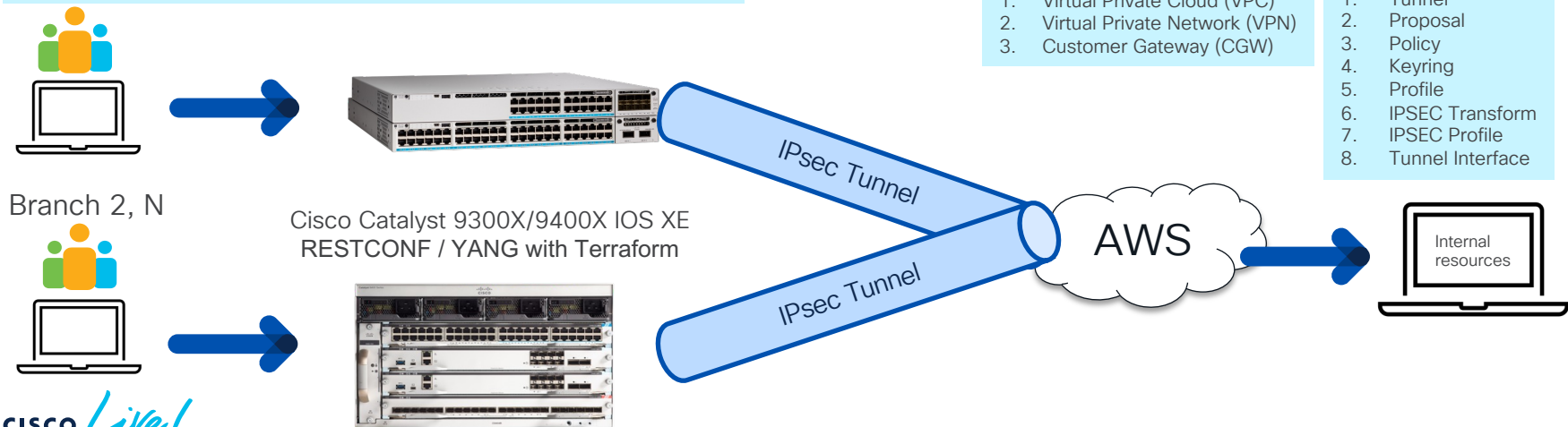
IPsec tunnel + cloud automation with Terraform

1. Terraform configures the IPsec tunnel between the 9300X and the cloud service where the internal resources are available
2. Terraform also manages the cloud-native resources including certificate key management and IP subnetting
3. Connections between VPC, VPN, CG Wand device certificates, tunnels, and interfaces are created



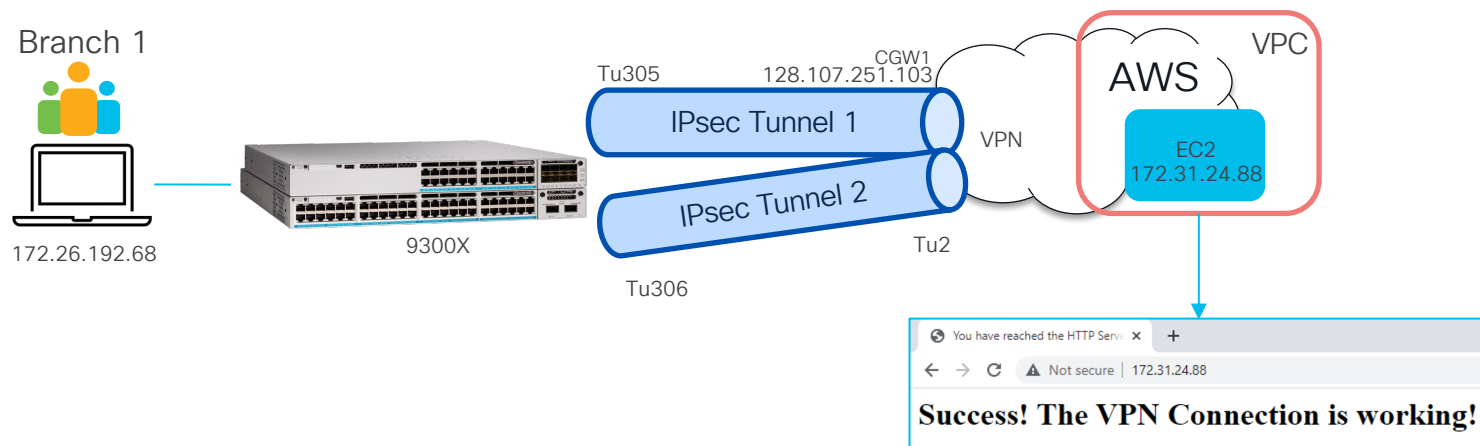
1. Virtual Private Cloud (VPC)
2. Virtual Private Network (VPN)
3. Customer Gateway (CGW)

1. Tunnel
2. Proposal
3. Policy
4. Keyring
5. Profile
6. IPSEC Transform
7. IPSEC Profile
8. Tunnel Interface



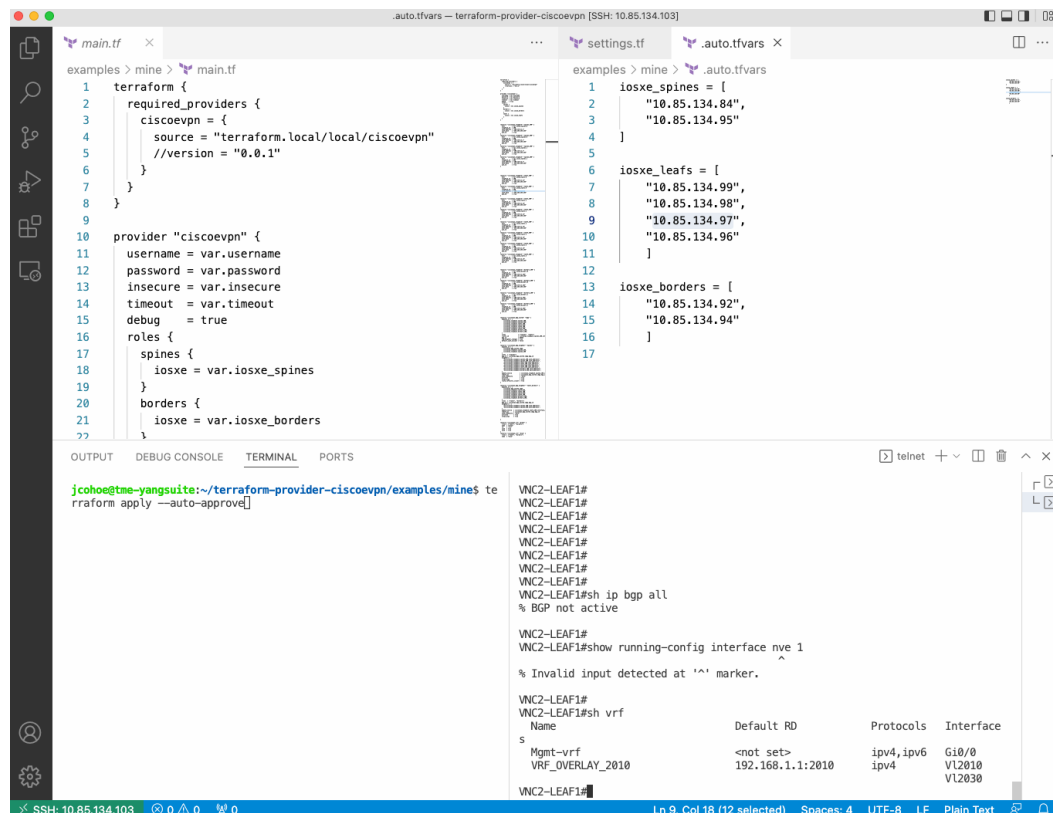
Topology

YouTube video coming soon!



<https://github.com/sdeweese/terraform-ipsec-tunnel-to-cloud>

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 required providers and roles for the EVPN setup. The `settings.tf` file defines the variables for the EVPN setup, including the list of spine and leaf IP addresses.

```
main.tf
1 terraform {
2   required_providers {
3     ciscoevpn = {
4       source = "terraform.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:

```
jchoe@tme-yangsuite:~/terraform-provider-ciscoevpn/examples/mine$ terraform apply --auto-approve
VNC2-LEAF1#
VNC2-LEAF1#
VNC2-LEAF1#
VNC2-LEAF1#
VNC2-LEAF1#
VNC2-LEAF1#
VNC2-LEAF1#
VNC2-LEAF1#sh ip bgp all
% BGP not active

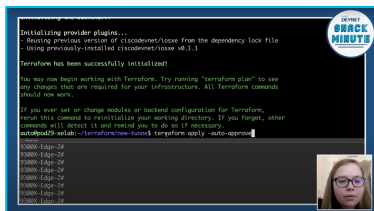
VNC2-LEAF1#
VNC2-LEAF1#show running-config interface nve 1
^
% Invalid input detected at '^' marker.

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       V12010
V12030                               V12030
```

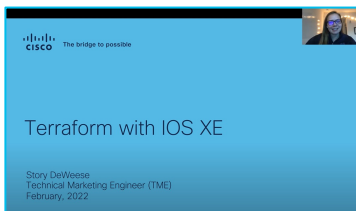
Resources

Blog and Resources: Terraform

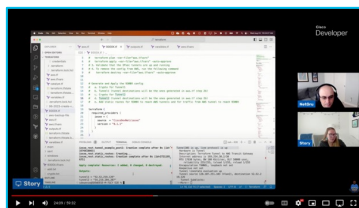
<https://github.com/CiscoDevNet/terraform-provider-iosxe/>
<https://registry.terraform.io/search/providers?namespace=CiscoDevNet>



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



Crypto IPsec Live Stream with DevRel:
<https://www.youtube.com/watch?v=WkgDIE0HpXs>

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



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.

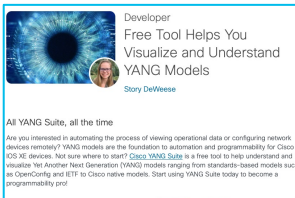
<https://blogs.cisco.com/developer/terraformiosxe01>

YANG Suite Resources

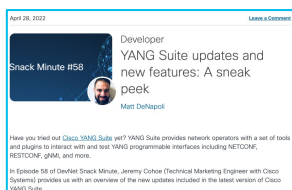
Blogs



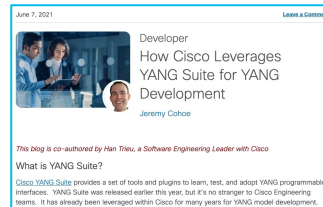
<https://blogs.cisco.com/developer/363-yangsuite-01>



<https://blogs.cisco.com/developer/er/yangallthetime01>

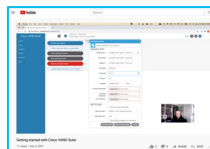


<https://blogs.cisco.com/developer/2023yangsuiteupdatesfeature/s01>

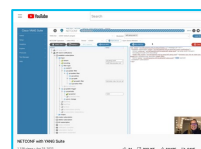


<https://blogs.cisco.com/developer/er/leverageyangsuite01?dtid=osscdc000283>

YouTube Videos



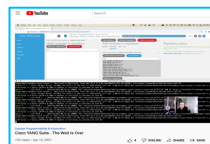
<https://youtu.be/smrhIL5Ayz0>



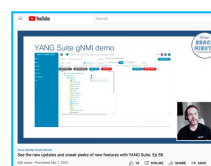
<https://www.youtube.com/watch?v=dTun33611JA>



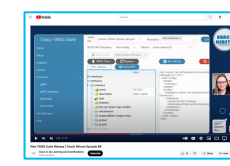
<https://www.youtube.com/watch?v=3zmNDfn8b38>



<https://www.youtube.com/watch?v=PkbAOzZ1vNk>



<https://www.youtube.com/watch?v=soyWPr0fJ0s>



https://www.youtube.com/watch?v=zVSO09_6rAU

Additional Resources

<https://github.com/CiscoDevNet/yangsuite/>
<https://developer.cisco.com/yangsuite/>
<https://eurl.io/#MaW78CeIS> YANG Suite General (external)

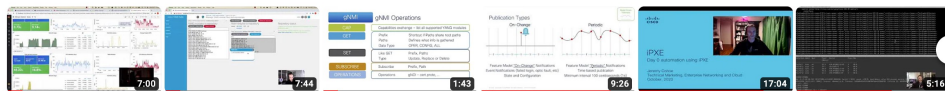
Cisco Programmability on YouTube

<https://www.youtube.com/jeremycohoe1>

<https://www.youtube.com/ciscoenprogrammability>

<https://www.youtube.com/ciscocatalysttv>

Uploads ▶ Play all



Using PowerShell and PowerCLI to create 10 clon... 280 views • 2 years ago

Getting started with Cisco YANG Suite 5K views • 2 years ago

Cisco Catalyst IOS XE - gNMI gNOI cert.proto - quick dem... 265 views • 2 years ago

Cisco IOS XE gRPC Model-Driven Telemetry (MDT) wit... 1.8K views • 2 years ago

IPXE - Cisco IOS XE Programmability and... 851 views • 2 years ago

Cisco Catalyst IOS XE 9300 demo of the NETCONF API L... 1.2K views • 3 years ago

HOME VIDEOS PLAYLISTS COMMUNITY CHANNELS ABOUT 🔍

Videos ▶ Play all



Cisco IOS XE Programmability and... 64 views • 1 month ago

Cisco YANG Suite release 3 35 views • 5 months ago

NETCONF with YANG Suite 117 views • 1 year ago

Cisco IOS XE Terraform provider introduction... 177 views • 1 year ago

Cisco Catalyst 9300 IOS XE 17.5 GNMII GNOI OS proto... 225 views • 2 years ago

Cisco DNA Center Client Proximity API and Event Use Case 157 views • 2 years ago



Cisco Catalyst TV

@CiscoCatalystTV 1.63K subscribers 74 videos

Welcome to Cisco Catalyst TV! This channel is all about Cisco Catalyst Pla...

Subscribed

HOME VIDEOS PLAYLISTS COMMUNITY CHANNELS ABOUT 🔍

Videos ▶ Play all



URL Filtering on Catalyst 8000V 28 views • 3 days ago

Catalyst PoE High Availability 81 views • 2 weeks ago

Introducing Catalyst 9500X 60L4D Campus Core 52 views • 2 weeks ago

Introducing the Catalyst 9200CX Series Switches 348 views • 1 month ago

IOS XE 17.11.1 Catalyst Switching Highlights 219 views • 1 month ago

IOS-XE Power Telemetry and Automation 188 views • 2 months ago

Catalyst Switching ▶ Play all



Cloud Security on Catalyst 9000 300 views • 2 years ago

Application Hosting with Catalyst 9000 Demo 465 views • 2 years ago

CVK Precision Timing PTPv2 IEEE1588-2008 490 views • 2 years ago

Cisco UPOE+ and PoE Analytics 340 views • 2 years ago

Encrypted Traffic Analytics 242 views • 2 years ago

ThousandEyes with Catalyst 9000 Switching - Overview 928 views • 1 year ago



Did you know?

You can have a
one-on-one session with
a technical expert!

Visit Meet the Expert in The HUB
to meet, greet, whiteboard & gain
insights about your unique questions
with the best of the best.



Meet the Expert Opening Hours:

Tuesday	3:00pm – 7:00pm
Wednesday	11:15am – 7:00pm
Thursday	9:30am – 4:00pm
Friday	10:30am – 1:30pm

Session Surveys

We would love to know your feedback on this session!

- Complete a minimum of four session surveys and the overall event surveys to claim a Cisco Live T-Shirt



Continue your education



- Visit the Cisco Showcase 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



The bridge to possible

Thank you

CISCO *Live!*

#CiscoLiveAPJC

The background is a vibrant, abstract graphic. On the left, there are overlapping, wavy shapes in shades of red, orange, 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 color palette is a rainbow spectrum.

cisco *Live!*

Let's go

#CiscoLiveAPJC