Let’s go

cisco The bridge to possible

Cisco IOS XE & Terraform

How to become an IOS XE Terraform Expert

Story DeWeese, Technical Marketing
@StoryDeWeese

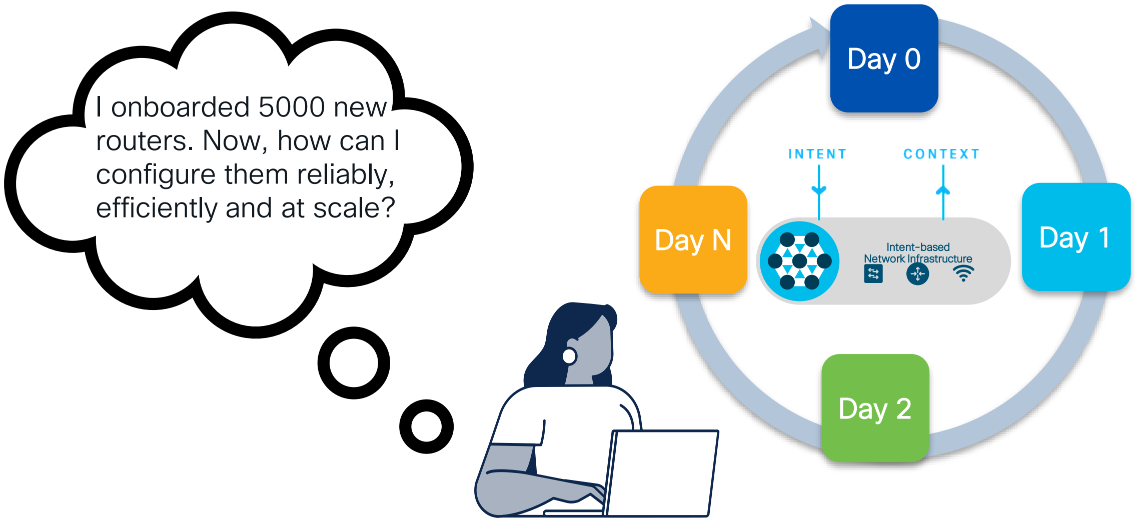
cisco Live!

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.

Agenda

- Programmability and Automation
- YANG Suite
- IOS XE Capabilities
- Terraform IOS XE Integration
- Use Case Examples
- Conclusion

Cisco IOS XE Programmability & Automation Lifecycle

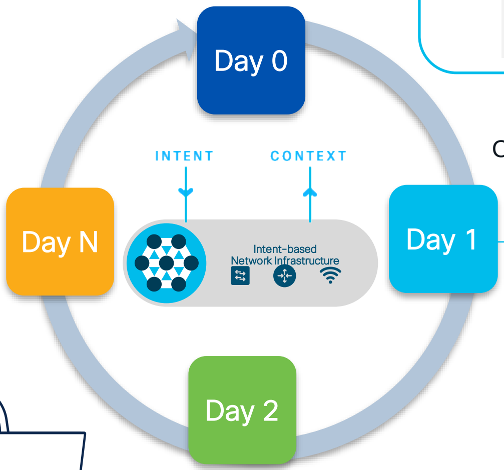

I onboarded 5000 new routers. Now, how can I configure them reliably, efficiently and at scale?
Day 0
INTENT
CONTEXT
Day N
Intent-based Network Infrastructure
Day 1
Day 2

Network Engineer

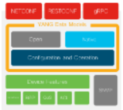
Cisco IOS XE Programmability & Automation Lifecycle

I can use a tooling such as Ansible or Terraform to configure my 5000 routers at once

Network Engineer



Model Driven Programmability



Device Configuration

Network Configuration Protocol (NETCONF), RESTCONF, gNMI

YANG "native" Data Models, OpenConfig,

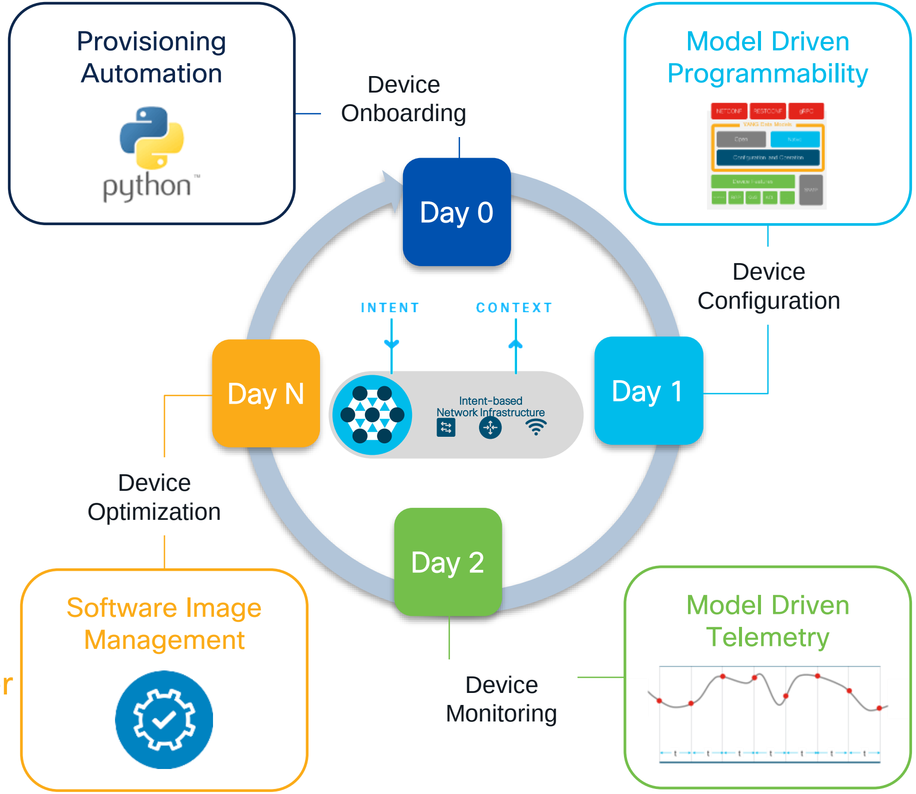
YANG Suite, Terraform, Ansible, pyATS tooling

IOS XE Programmability & Automation Lifecycle

Pre-boot Execution Environment (iPXE)

Zero Touch Provisioning

VM Automation



Provisioning Automation

Device Onboarding

Day 0

Model Driven Programmability

Device Configuration

Day 1

INTENT

CONTEXT

Intent-based Network Infrastructure

Day N

Device Optimization

Software Image Management

Day 2

Device Monitoring

Model Driven Telemetry

Network Configuration Protocol (NETCONF), RESTCONF, gNMI

YANG Data Models, OpenConfig, and YANG Suite tooling

YANG Suite, Terraform, Ansible, pyATS

gNOI cert/os/reset proto

Guest Shell + NETCONF

CentOS 8 Python 3

Application Hosting with Docker

CLI to XML

TIG_MDT container + examples

YANG On-Change support

gRPC Dial-Out + DNS + TLS

gNMI/NETCONF Dial-In

Model Driven Programmability Interface Comparison

Network architecture, security posture and policy, YANG data modules, tools and language preferences are some considerations when leveraging the various MDP interfaces

	NETCONF	RESTCONF	gNMI
Minimum IOS XE Version	16.6	16.7	16.8
Recommended Version	17.6	17.6	17.7
Default Port	830	443	9339
Operations	<get>,<get-config>,<edit-config>,<establish-subscription>	GET, POST, PUT, PATCH, DELETE	GET, SET, SUBSCRIBE
Encoding	XML	XML or JSON	RFC7951 JSON_IETF
Security	SSH + PKI certificate or password	HTTPS user/pass	TLS certificate with user authentication
Transport Protocol	SSH	HTTPS	HTTP/2
Tooling	YANG Suite, ncclient, Netconf-console	YANG Suite*, Postman, python	YANG Suite*, gnmic, gnmi_cli
Content	YANG	YANG	YANG + Protobuf

IOS XE Programmability and Telemetry “Stack”

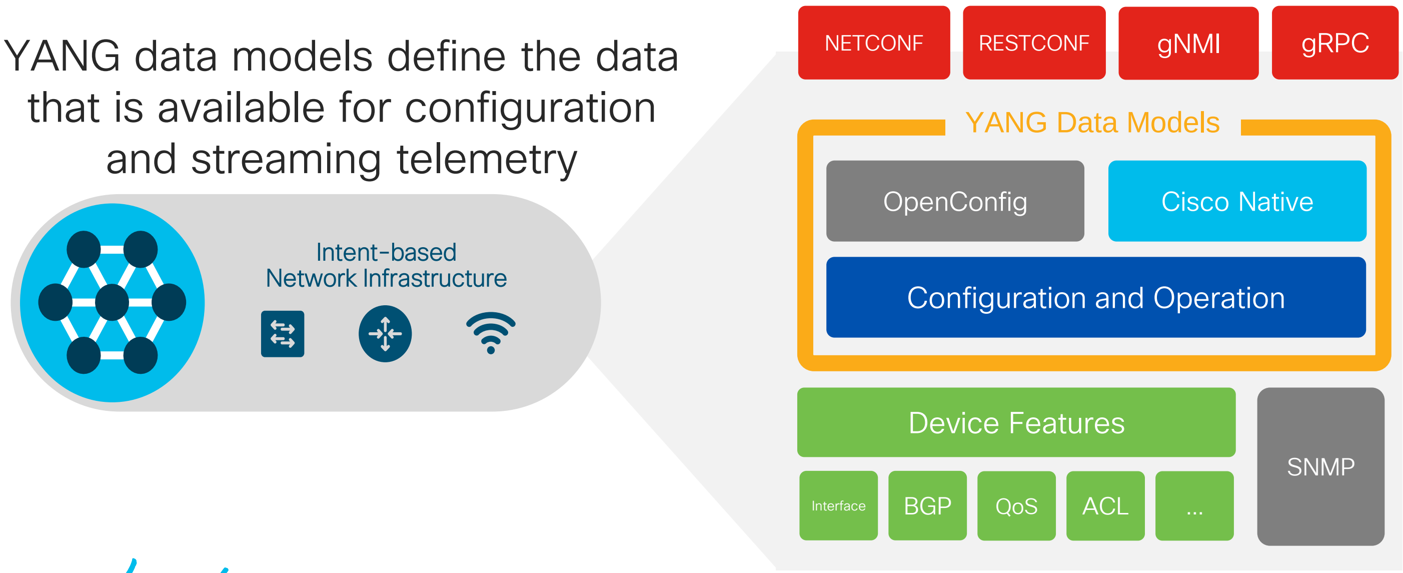
CLI

SNMP

WebUI

The NETCONF, RETCONF, gNMI and gRPC are programmatic interfaces that provide additional methods for interfacing with the IOS XE device – Just like the CLI, SNMP, and WebUI is used for configuration changes and operational metrics so can the programmatic interfaces of NETCONF, RESTCONF, gNMI, and gRPC.

YANG data models define the data that is available for configuration and streaming telemetry

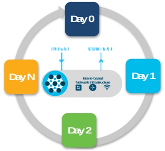


IETF

OPENCONFIG

IEEE

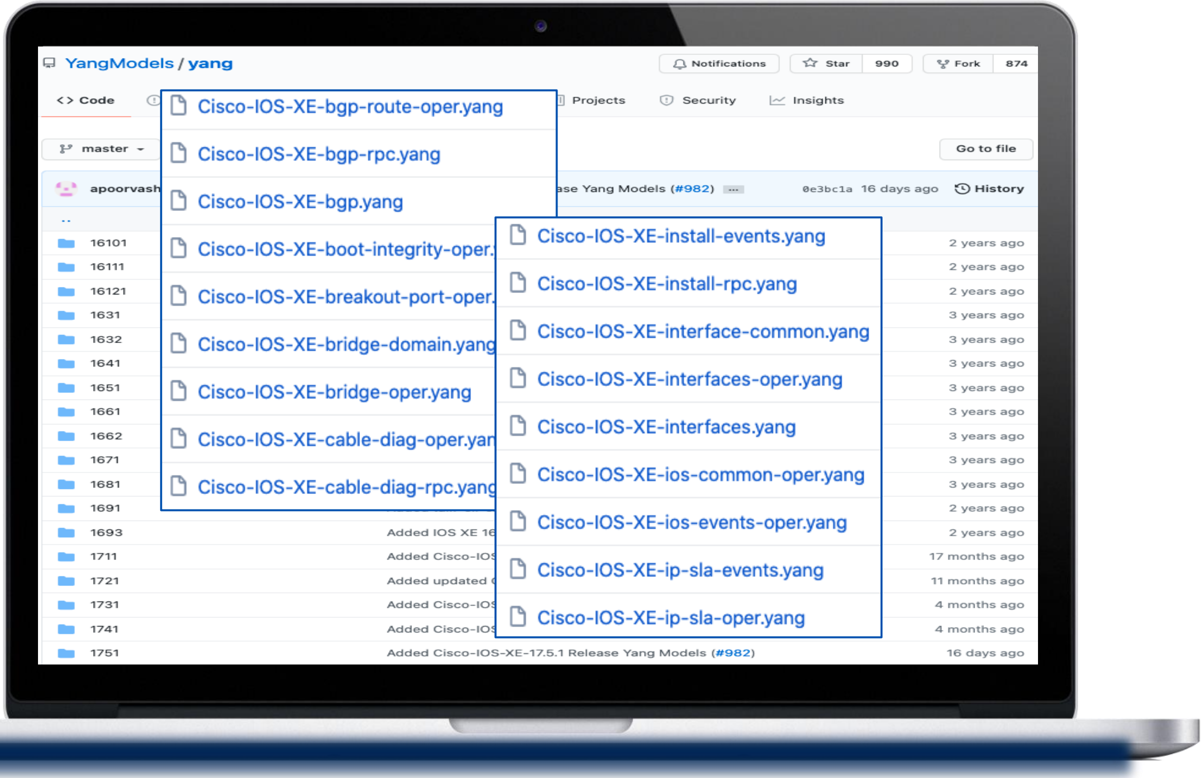
CISCO



IOS XE - YANG Model Coverage on GitHub

RFC7950 states that “YANG is a data modeling language used to model configuration data, state data, Remote Procedure Calls, and notifications for network management protocols”

YANG module name.yang	Description
Cisco-IOS-XE-native	running-config
Cisco-IOS-XE-{feature}-cfg	Feature configuration
Cisco-IOS-XE-{feature}-oper	Feature operational data
Cisco-IOS-XE-{feature}-rpc	Actions
Cisco-evpn-service	EVPN service abstraction
OpenConfig-{feature}	abstraction for config & oper



YangModels / yang

apoorvash ... ase Yang Models (#982) ... 0e3bc1a 16 days ago

..		
16101		2 years ago
16111		2 years ago
16121		2 years ago
1631		3 years ago
1632		3 years ago
1641		3 years ago
1651		3 years ago
1661		3 years ago
1662		3 years ago
1671		3 years ago
1681		3 years ago
1691		2 years ago
1693	Added IOS XE 16	2 years ago
1711	Added Cisco-IOS	17 months ago
1721	Added updated	11 months ago
1731	Added Cisco-IOS	4 months ago
1741	Added Cisco-IOS	4 months ago
1751	Added Cisco-IOS-XE-17.5.1 Release Yang Models (#982)	16 days ago

- Cisco-IOS-XE-bgp-route-oper.yang
- Cisco-IOS-XE-bgp-rpc.yang
- Cisco-IOS-XE-bgp.yang
- Cisco-IOS-XE-boot-integrity-oper.
- Cisco-IOS-XE-breakout-port-oper.
- Cisco-IOS-XE-bridge-domain.yang
- Cisco-IOS-XE-bridge-oper.yang
- Cisco-IOS-XE-cable-diag-oper.yan
- Cisco-IOS-XE-cable-diag-rpc.yang

- Cisco-IOS-XE-install-events.yang
- Cisco-IOS-XE-install-rpc.yang
- Cisco-IOS-XE-interface-common.yang
- Cisco-IOS-XE-interfaces-oper.yang
- Cisco-IOS-XE-interfaces.yang
- Cisco-IOS-XE-ios-common-oper.yang
- Cisco-IOS-XE-ios-events-oper.yang
- Cisco-IOS-XE-ip-sla-events.yang
- Cisco-IOS-XE-ip-sla-oper.yang

https://github.com/YangModels/yang/tree/master/vendor/cisco/xe

Programmability and Automation

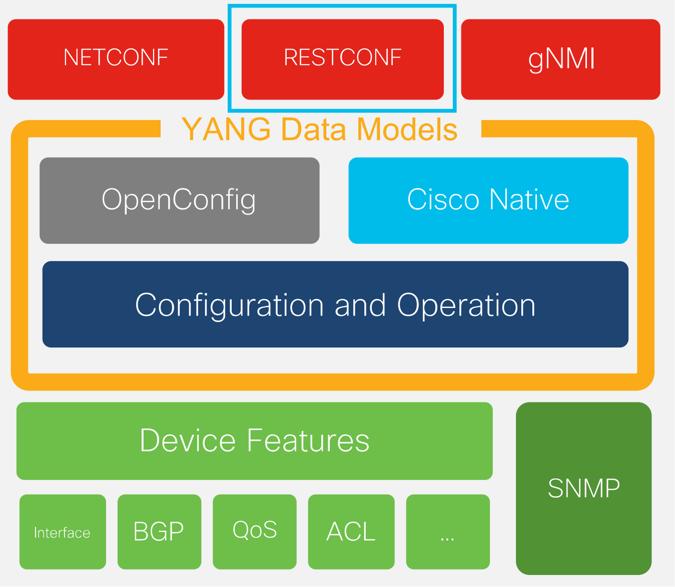
Terraform uses the RESTCONF API on IOS XE

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

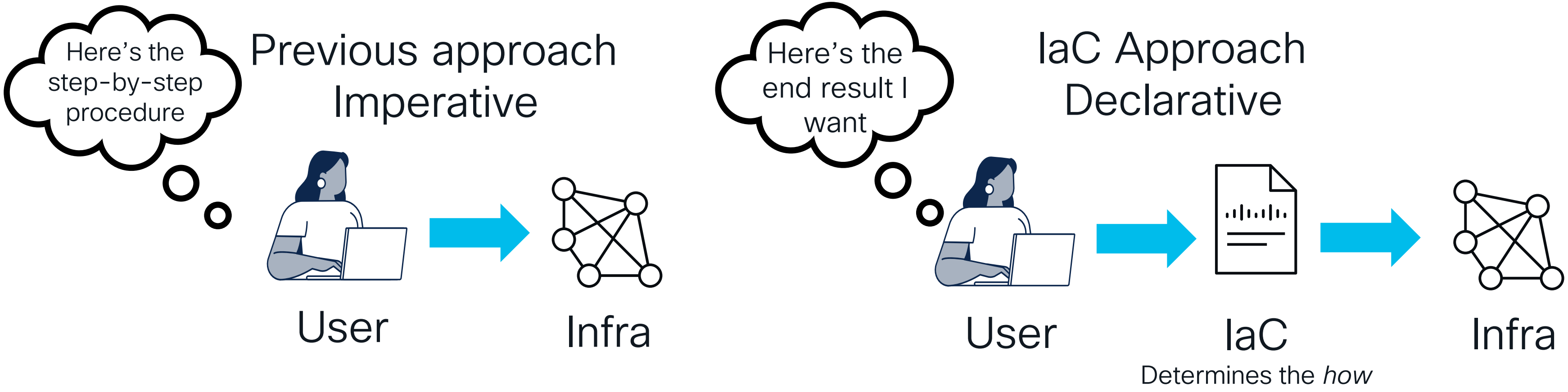


NETCONF
RESTCONF
gNMI
YANG Data Models
OpenConfig
Cisco Native
Configuration and Operation
Device Features
Interface
BGP
QoS
ACL
...
SNMP

Terraform image source: https://registry.terraform.io/

What’s IaC?

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



Here’s the step-by-step procedure

Previous approach

Imperative

User

Infra

Here’s the end result I want

IaC Approach

Declarative

User

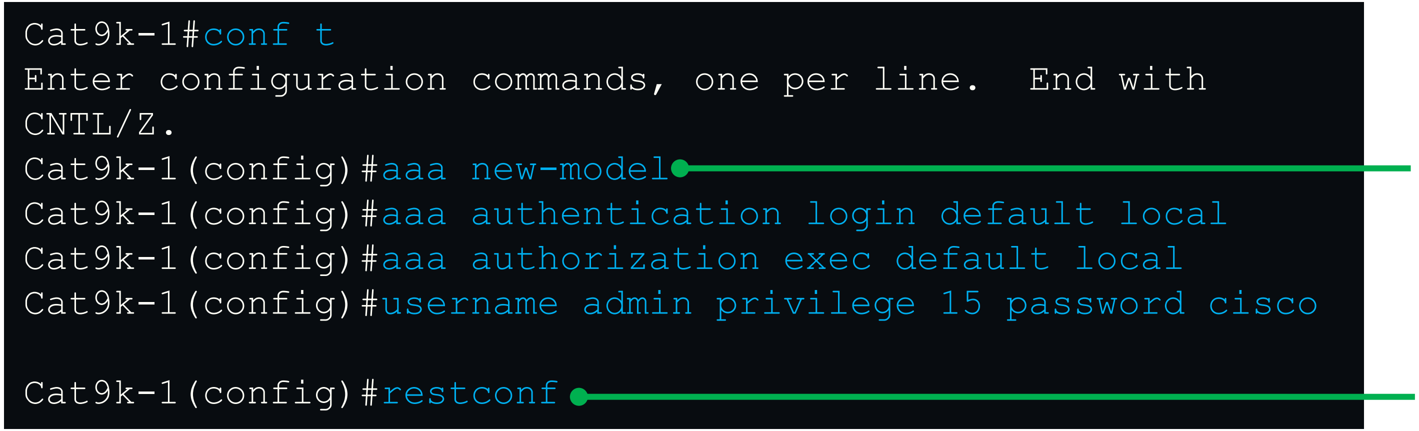
IaC

Determines the *how*

Infra

Learn more about IaC here:
https://developer.cisco.com/iac/#:~:text=Adopting%20Infrastructure%20as%20Code%20allows,data%20center%20to%20the%20edge

IOS XE API Pre-Req



```
Cat9k-1#conf t
Enter configuration commands, one per line.  End with
CNTL/Z.
Cat9k-1(config)#aaa new-model
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)#restconf
```

Enable AAA

Enable RESTCONF

This is the most basic example ☺
There are additional configuration knobs for ACL, AAA, RBAC, etc

Terraform on IOS XE

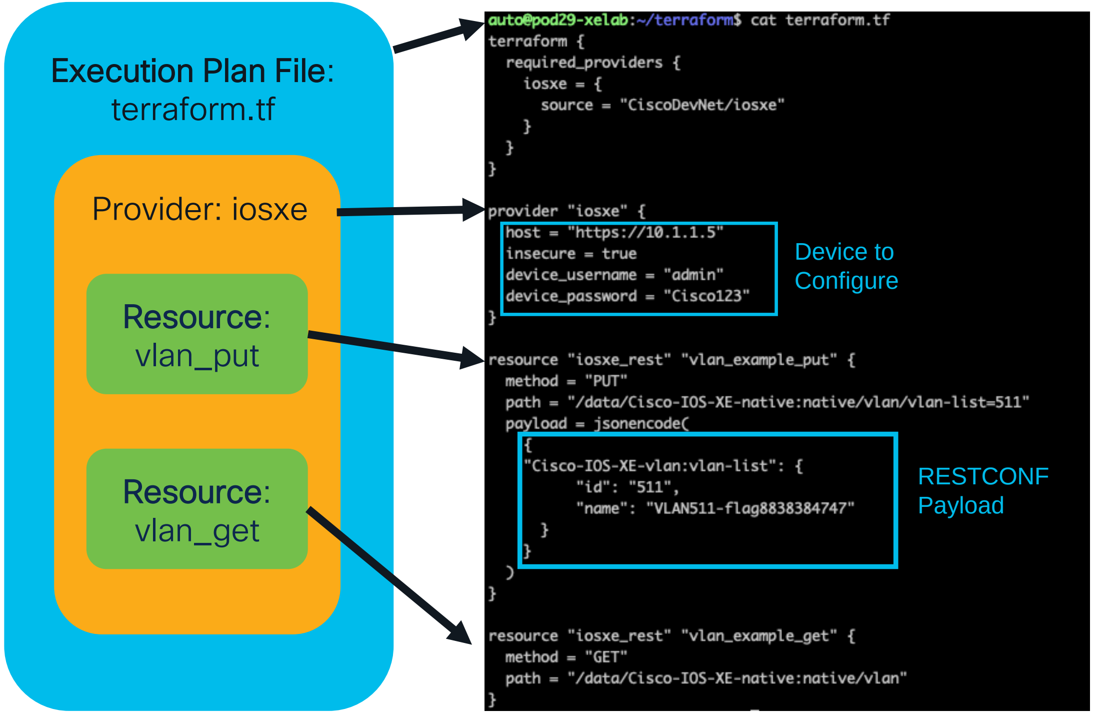
Terraform Terminology

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



```
auto@pod29-xelab:~/terraform$ cat terraform.tf
terraform {
  required_providers {
    iosxe = {
      source = "CiscoDevNet/iosxe"
    }
  }
}

provider "iosxe" {
  host = "https://10.1.1.5"
  insecure = true
  device_username = "admin"
  device_password = "Cisco123"
}

resource "iosxe_rest" "vlan_example_put" {
  method = "PUT"
  path = "/data/Cisco-IOS-XE-native:native/vlan/vlan-list=511"
  payload = jsonencode(
    {
    "Cisco-IOS-XE-vlan:vlan-list": {
          "id": "511",
          "name": "VLAN511-flag8838384747"
      }
    }
  )
}

resource "iosxe_rest" "vlan_example_get" {
  method = "GET"
  path = "/data/Cisco-IOS-XE-native:native/vlan"
}
```

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
… or …
$ https://github.com/jeremycohoe/cisco-ios-xe-dcloud-terraform
```

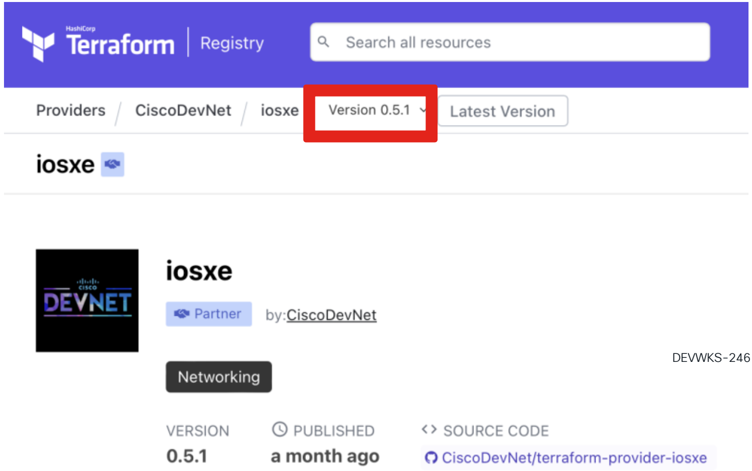
4. Apply Terraform VLAN example

```
$ terraform init
$ terraform plan
$ terraform apply acl_and_vlan.tf
```

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/



Providers / CiscoDevNet / iosxe / Version 0.5.1 Latest Version

iosxe

iosxe

Partner by:CiscoDevNet

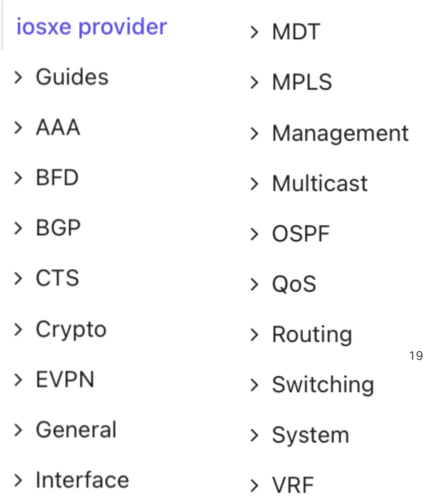
Networking

VERSION 0.5.1

PUBLISHED a month ago

SOURCE CODE CiscoDevNet/terraform-provider-iosxe

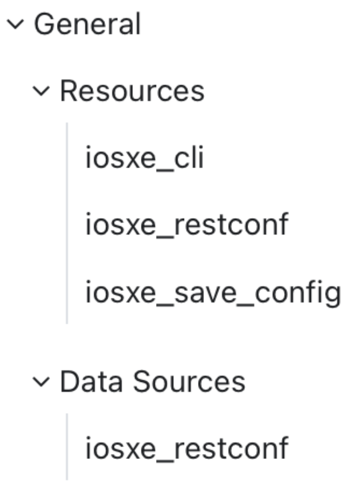
DEVWKS-2461



iosxe provider

- Guides
- AAA
- BFD
- BGP
- CTS
- Crypto
- EVPN
- General
- Interface
- MDT
- MPLS
- Management
- Multicast
- OSPF
- QoS
- Routing
- Switching
- System
- VRF

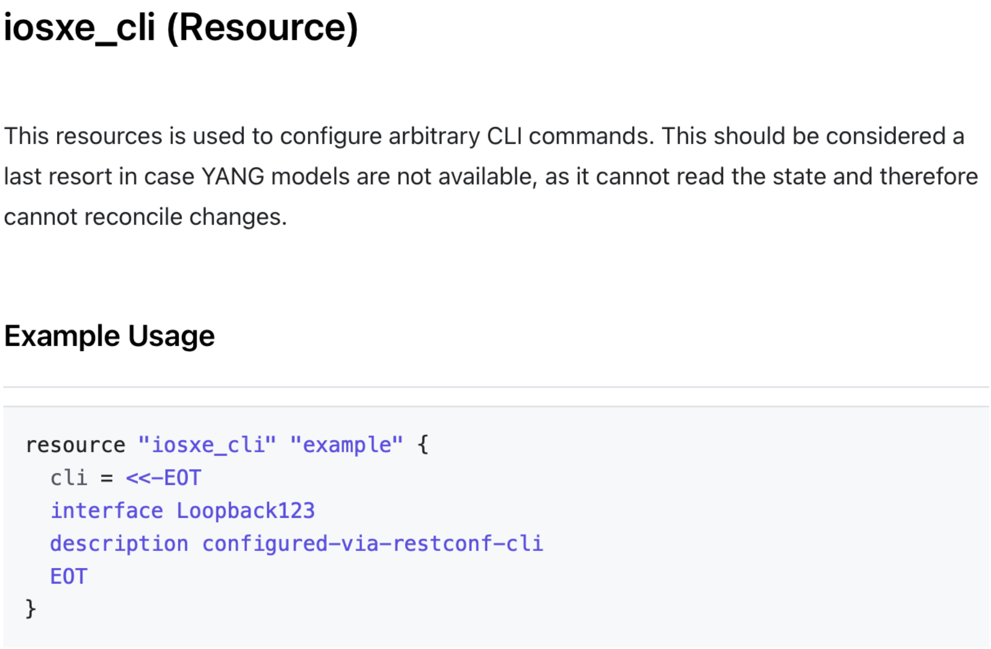
19



- General
  - Resources
    - iosxe_cli
    - iosxe_restconf
    - iosxe_save_config
  - Data Sources
    - iosxe_restconf

Terraform resource utilizing the CLI RPC

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



iosxe_cli (Resource)

This resources 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
}
```

What’s different about Terraform?

State-FUL

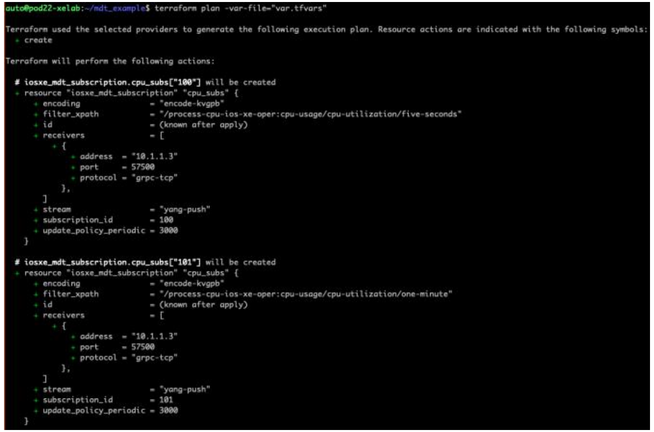
- Checks the current configuration before making changes to help avoid any potential conflicts
- Quick to remove config, no need to create a separate file

```
$ terraform destroy
```

Use Case Examples

Terraform for Model Driven Telemetry

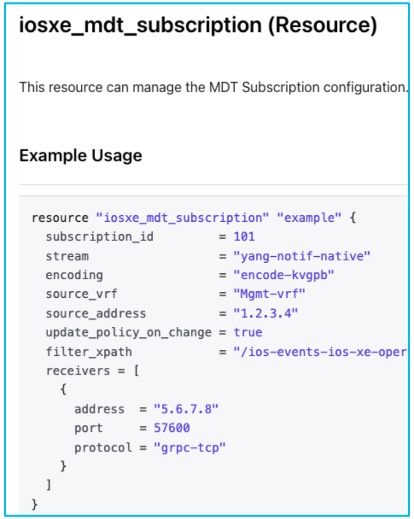
Enable gRPC Dial-Out telemetry subscriptions for the POE and Basic Device Monitoring use cases



```
auto@pod22-xelab:~/mdt_example$ 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



iosxe_mdt_subscription (Resource)

This resource can manage the MDT Subscription configuration.

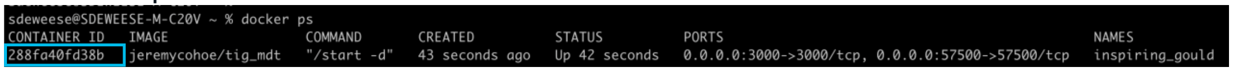
Example Usage

```
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"
    }
  ]
}
```

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

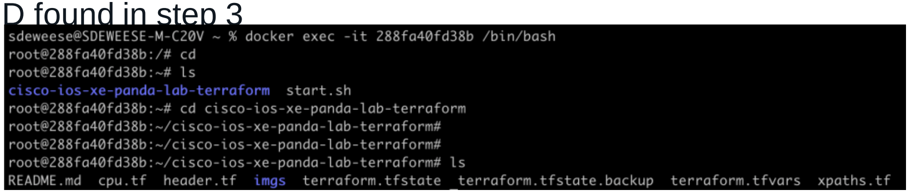
Use the Docker Container with the Terraform files included!

1. Ensure Docker is installed
2. Run the following commands
    1. docker pull jeremycohoe/tig_mdt
    2. docker run -ti -p 3000:3000 -p 57500:57500 jeremycohoe/tig_mdt
3. Identify the container ID
    1. docker ps



```
sdeweese@SDEWEESE-M-C20V ~ % docker ps
CONTAINER ID   IMAGE               COMMAND         CREATED          STATUS           PORTS                                              NAMES
288fa40fd38b   jeremycohoe/tig_mdt "/start -d"     43 seconds ago   Up 42 seconds    0.0.0.0:3000->3000/tcp, 0.0.0.0:57500->57500/tcp   inspiring_gould
```

4. Enter into the container
    1. docker exec -it CONTAINER_ID /bin/bash
        1. Note: replace CONTAINER_ID with the ID found in step 3



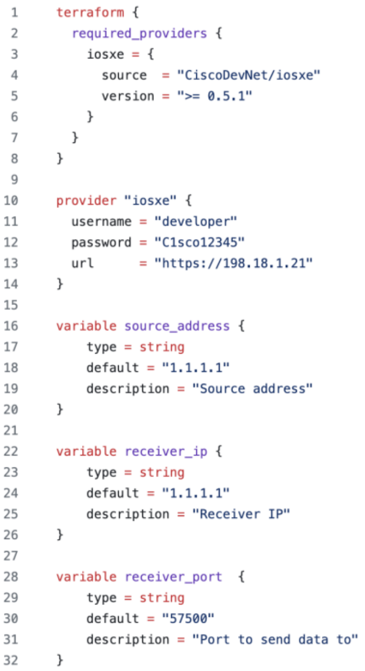
```
sdeweese@SDEWEESE-M-C20V ~ % docker exec -it 288fa40fd38b /bin/bash
root@288fa40fd38b:/# cd
root@288fa40fd38b:~# ls
cisco-ios-xe-panda-lab-terraform  start.sh
root@288fa40fd38b:~# cd cisco-ios-xe-panda-lab-terraform
root@288fa40fd38b:~/cisco-ios-xe-panda-lab-terraform#
root@288fa40fd38b:~/cisco-ios-xe-panda-lab-terraform#
root@288fa40fd38b:~/cisco-ios-xe-panda-lab-terraform# ls
README.md  cpu.tf  header.tf  imgs  terraform.tfstate  terraform.tfstate.backup  terraform.tfvars  xpaths.tf
```

5. Navigate to the correct folder
    1. cd
    2. cd cisco-ios-xe-panda-lab-terraform
6. Modify the device credentials in the header.tf file

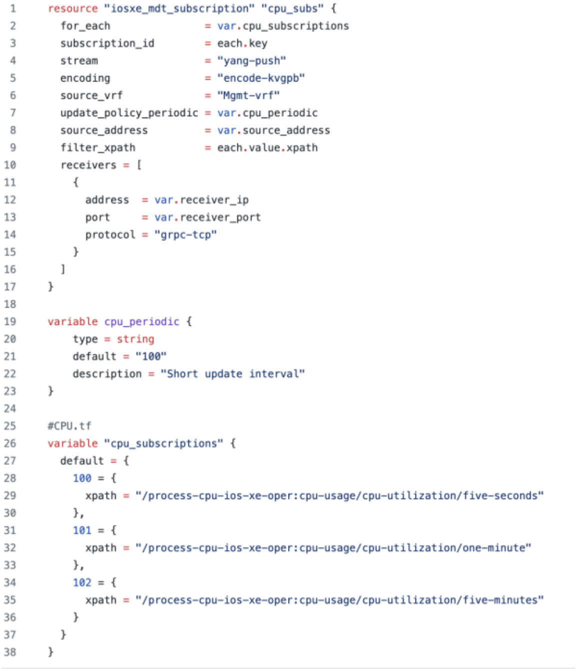
```
provider "iosxe" {
        username = "admin" ← replace with Cisco IOS XE device username
        password = "XXXXXXXX" ← replace with Cisco IOS XE device password
        url = "https://your-switch-hostname-or-ip" ← replace with Cisco IOS XE device hostname or I
}
```

7. Configure the Cisco IOS XE device using Terraform

Example TF file



```
terraform {
  required_providers {
    iosxe = {
      source  = "CiscoDevNet/iosxe"
      version = ">= 0.5.1"
    }
  }
}

provider "iosxe" {
  username = "developer"
  password = "C1sco12345"
  url      = "https://198.18.1.21"
}

variable source_address {
    type = string
    default = "1.1.1.1"
    description = "Source address"
}

variable receiver_ip {
    type = string
    default = "1.1.1.1"
    description = "Receiver IP"
}

variable receiver_port  {
    type = string
    default = "57500"
    description = "Port to send data to"
}
```

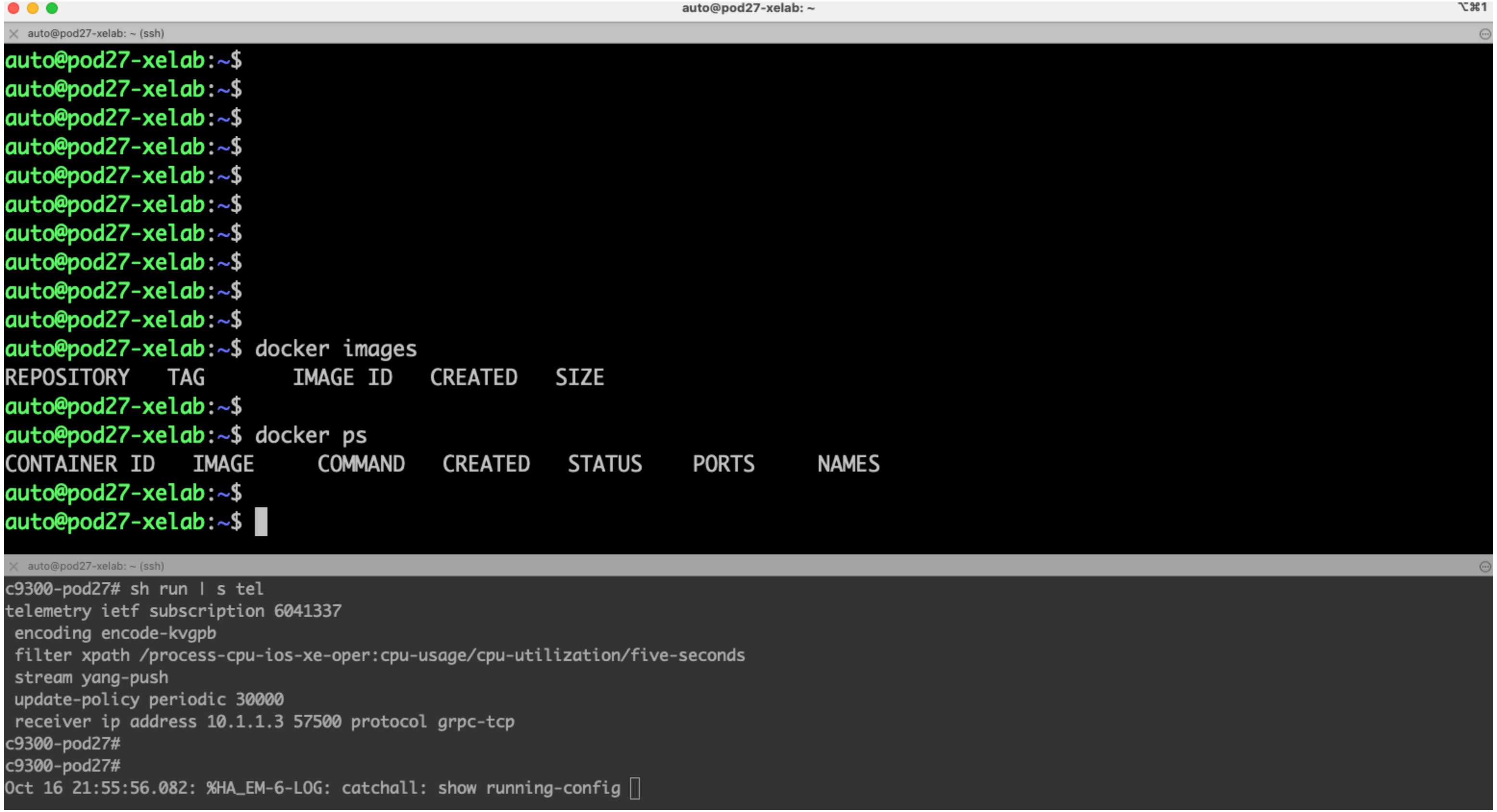


```
resource "iosxe_mdt_subscription" "cpu_subs" {
  for_each                = var.cpu_subscriptions
  subscription_id         = each.key
  stream                  = "yang-push"
  encoding                = "encode-kvgpb"
  source_vrf              = "Mgmt-vrf"
  update_policy_periodic = var.cpu_periodic
  source_address          = var.source_address
  filter_xpath            = each.value.xpath
  receivers = [
    {
      address  = var.receiver_ip
      port     = var.receiver_port
      protocol = "grpc-tcp"
    }
  ]
}

variable cpu_periodic {
    type = string
    default = "100"
    description = "Short update interval"
}

#CPU.tf
variable "cpu_subscriptions" {
  default = {
    100 = {
      xpath = "/process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds"
    },
    101 = {
      xpath = "/process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/one-minute"
    },
    102 = {
      xpath = "/process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-minutes"
    }
  }
}
```

Creates three MDT subscriptions for various CPU data

https://github.com/jeremycohoe/cisco-ios-xe-dcloud-terraform/tree/main

DEMO – install docker container and use Terraform files to configure telemetry subscriptions



```
auto@pod27-xelab:~$
auto@pod27-xelab:~$
auto@pod27-xelab:~$
auto@pod27-xelab:~$
auto@pod27-xelab:~$
auto@pod27-xelab:~$
auto@pod27-xelab:~$
auto@pod27-xelab:~$
auto@pod27-xelab:~$
auto@pod27-xelab:~$
auto@pod27-xelab:~$ docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
auto@pod27-xelab:~$
auto@pod27-xelab:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
auto@pod27-xelab:~$
auto@pod27-xelab:~$
```

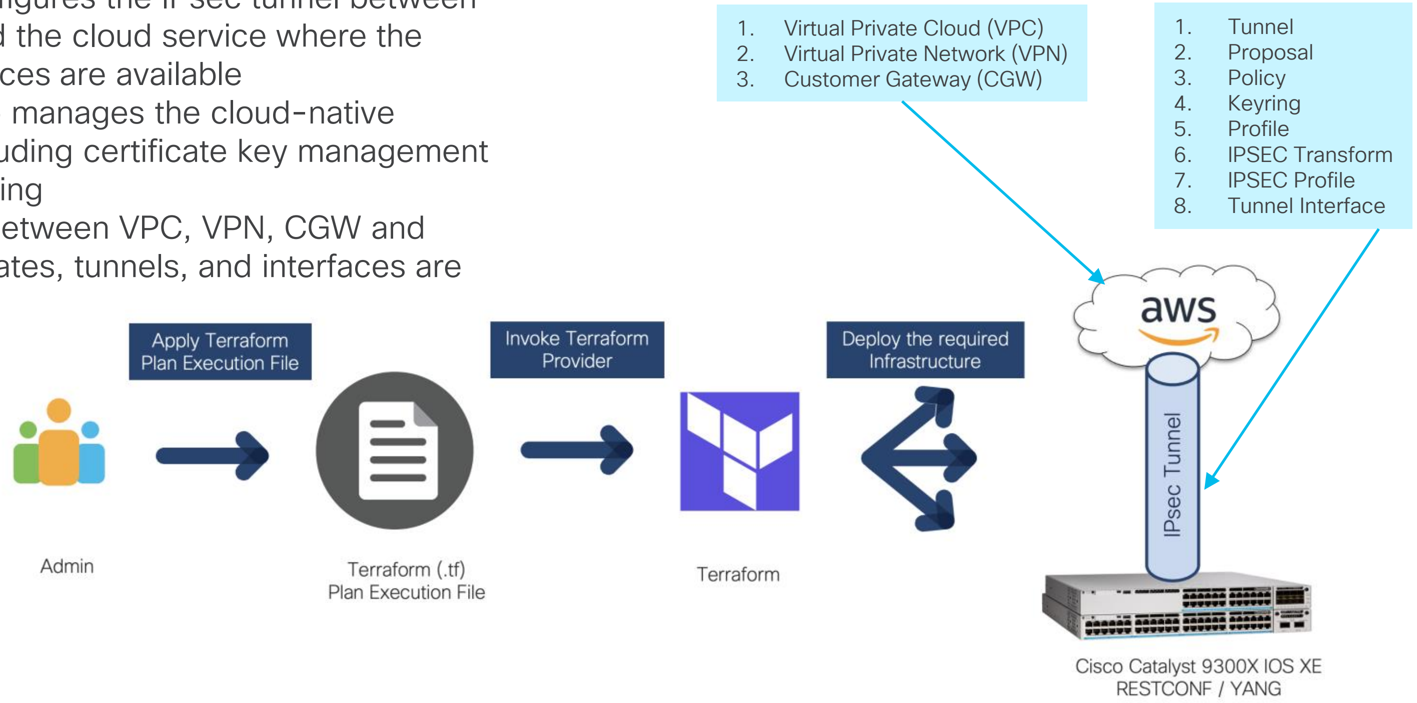
```
c9300-pod27# sh run | s tel
telemetry ietf subscription 6041337
 encoding encode-kvgpb
 filter xpath /process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds
 stream yang-push
 update-policy periodic 30000
 receiver ip address 10.1.1.3 57500 protocol grpc-tcp
c9300-pod27#
c9300-pod27#
Oct 16 21:55:56.082: %HA_EM-6-LOG: catchall: show running-config
```

IPsec tunnel + cloud automation with Terraform

Day 0

YouTube video coming soon!

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, CGW and device certificates, tunnels, and interfaces are created



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

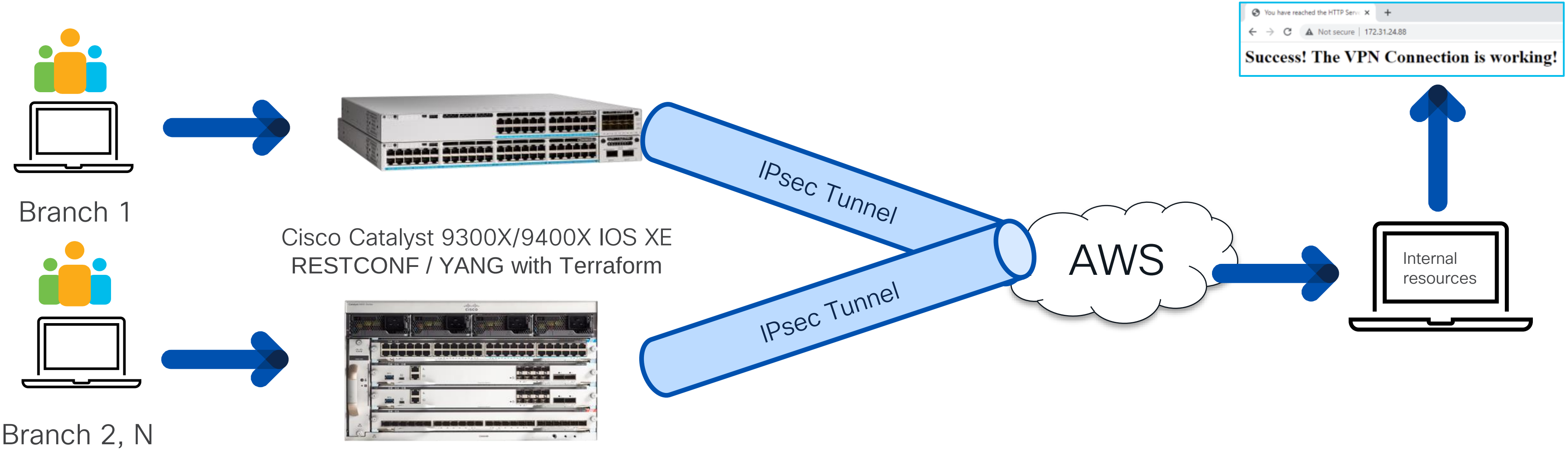
1. Tunnel
2. Proposal
3. Policy
4. Keyring
5. Profile
6. IPSEC Transform
7. IPSEC Profile
8. Tunnel Interface

Apply Terraform Plan Execution File
Invoke Terraform Provider
Deploy the required Infrastructure
aws
IPsec Tunnel
Admin
Terraform (.tf) Plan Execution File
Terraform
Cisco Catalyst 9300X IOS XE RESTCONF / YANG

IPsec Tunnel for employees

Day 1

YouTube video coming soon!



Success! The VPN Connection is working!

Cisco Catalyst 9300X/9400X IOS XE
RESTCONF / YANG with Terraform

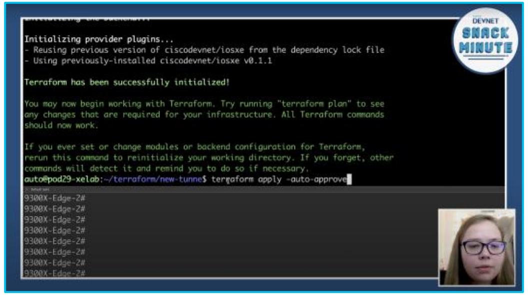
https://github.com/sdeweese/terraform-ipsec-tunnel-to-cloud

Resources

Blog and Resources: Terraform

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

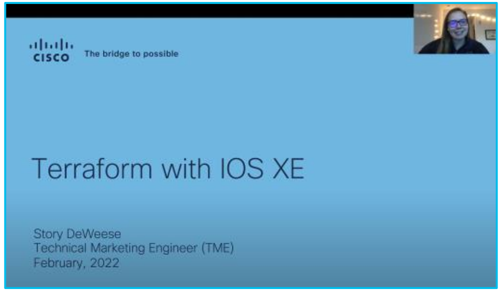
Questions? Join the Ask IOS XE Terraform Provider Webex space:
https://eurl.io/#PtsT8eJFl



```
Initializing provider plugins...
- Reusing previous version of ciscodevnet/iosxe from the dependency lock file
- Using previously-installed ciscodevnet/iosxe v0.1.1

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
auto@pod29-xelab:~/terraform/new-tunne$ terraform apply -auto-approve
```

```
9300X-Edge-2#
9300X-Edge-2#
9300X-Edge-2#
9300X-Edge-2#
9300X-Edge-2#
9300X-Edge-2#
9300X-Edge-2#
9300X-Edge-2#
```

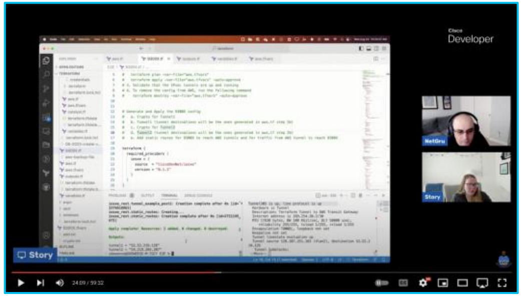
Demo Create a Crypto Tunnel Video:
https://www.youtube.com/watch?v=bPS0bhPacDw



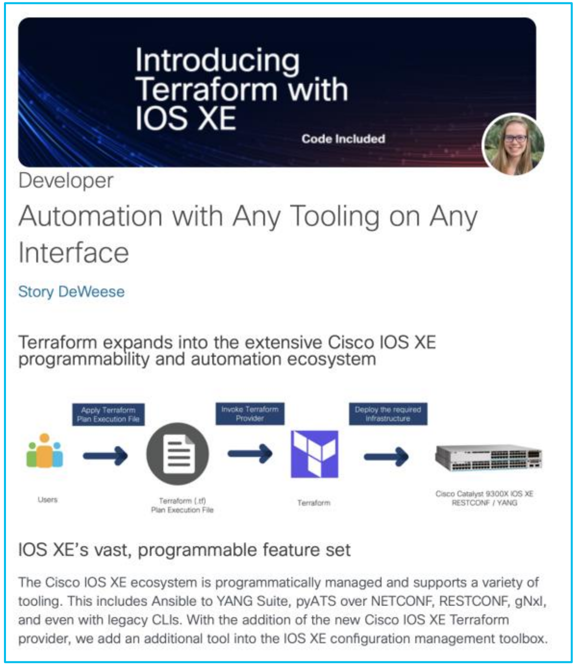
Terraform with IOS XE

Story DeWeese
Technical Marketing Engineer (TME)
February, 2022

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=WkgDlE0HpXs


Introducing Terraform with IOS XE
Code Included
Apply Terraform Plan Execution File
Invoke Terraform Provider
Deploy the required infrastructure
Users
Terraform (.tf) Plan Execution File
Terraform
Cisco Catalyst 9300X IOS XE RESTCONF / YANG

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, gNxI, 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

dCloud Programmability

https://dcloud.cisco.com
“Cisco Catalyst 9000 IOS XE Programmability & Automation Lab v1”
https://dcloud2.cisco.com/demo/catalyst-9000-ios-xe-programmability-automation-lab-v1

Use Cases:

EVPN:

Ansible with CLI deployment of EVPN solutions
EVPN management over RESTCONF/YANG with Postman
Declarative EVPN fabric management with Terraform

Model Driven Telemetry

Telemetry configuration with CLI and YANG Suite
Collection with TIG_MDT container and tooling

YANG Programmability

YANG Suite tooling and integrations to YANG API’s
Ansible integrations

Tooling and Integrations

YANG Suite

- NETCONF/RESTCONF/gNMI API
  - Ansible integration
- NETCONF/gNMI Dial-In Telemetry
- gRPC Dial-Out Telemetry receiver

Telemetry

- TIG stack in Docker
- Grafana dashboard for device health

Postman / RESTCONF

- EVPN fabric API calls

Terraform/RESTCONF

- Declarative EVPN fabric management

Ansible

- EVPN solution enablement using CLI

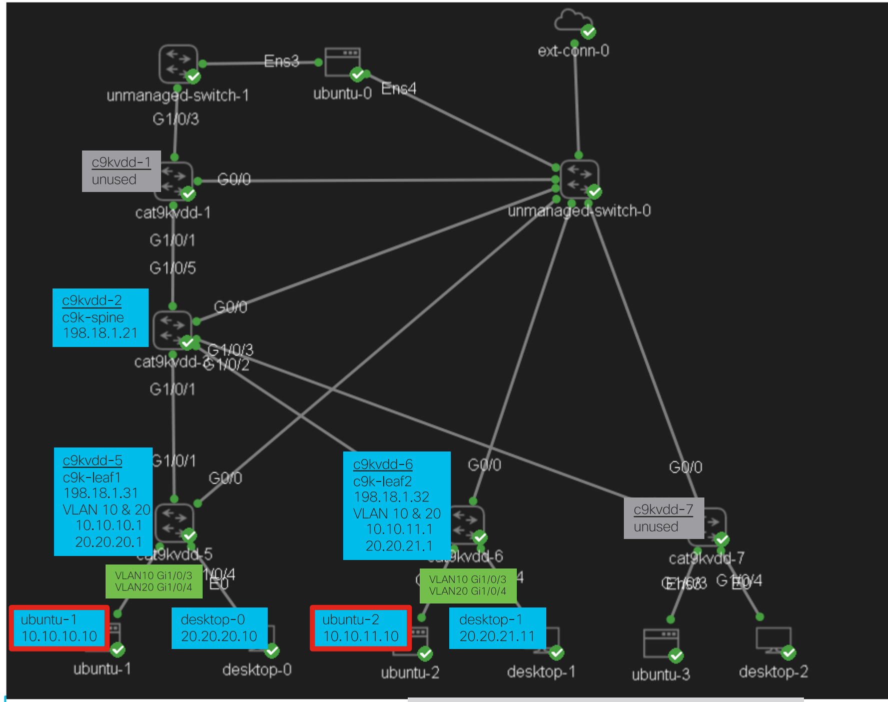
Ubuntu VM Details:

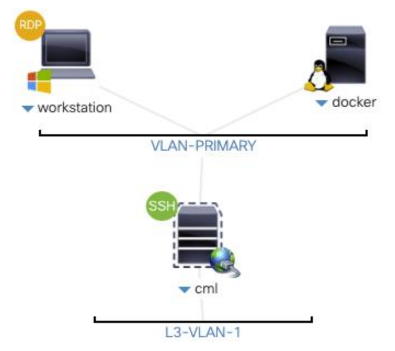
Syslog receiver from all switches
TFTP config backup
See slide

Windows VM Details

VS Code
Terraform @ folder
Ansible @ folder
Chrome browser
YANG Suite, Grafana
Bash/PS/Cmd shells
SSH into C9K or Ubuntu
Postman
Workspace for EVPN

3x C9K Virtual VM’s





VLAN1
c9k-spine
IP: 198.18.1.21
developer / C1sco12345
c9k-leaf1
IP: 198.18.1.31
developer / C1sco12345
c9k-leaf2
IP: 198.18.1.32
developer / C1sco12345
c9kvdd-1 - unconfigured

c9kvdd-7 - unconfigured

Cisco Catalyst IOS XE Programmability Sessions at Cisco Live Europe Amsterdam 2024

Monday Feb 5

11AM DEVLIT-2083: Automate Cisco IOS XE Device Configuration Using Terraform (Story)

1:30PM CISCOU-1024: Model Driven Telemetry with Cisco IOS XE made Simple (Jeremy)

3:30PM DEVLIT-1965: Automation using multiple API's in the IOS XE Device Programmability Lab (Jeremy)

Tuesday Feb 6

11:30AM DEVNET-1441: Catalyst 9000 Virtual topology simulation and configuration management solutions (Ama+Jeremy)

11:30AM DEVLIT-2787: Explore and visualize YANG models with YANG Suite (Story)

1:30PM DEVWKS-2270: Implementing Cisco IOS XE Model Driven Telemetry with Telegraf, InfluxDB, and Grafana (Jeremy)

1:30PM CISCOU-2013: Explore Cisco IOS XE Automation Tooling and Use Cases (Story)

4:30PM DEVLIT-2062: Getting Started with Secure Zero Touch Provisioning (Story)

Wednesday Feb 7

1:30PM DEVNET-2464: How to Become a Cisco IOS XE Terraform Expert (Story)

4:30PM CISCOU-2021: gNMI Oh My! Automation with the gRPC micro-services on IOS XE (Jeremy)

Thursday Feb 8

2:30PM BRKOPS-2455: Infrastructure as Code with Cisco Catalyst 9000 Virtual (Ama+Jeremy)

2:30PM DEVNET-1468: Programmability, Automation Model Driven Telemetry on Cisco IOS XE with a dash of YANG Suite (Story)

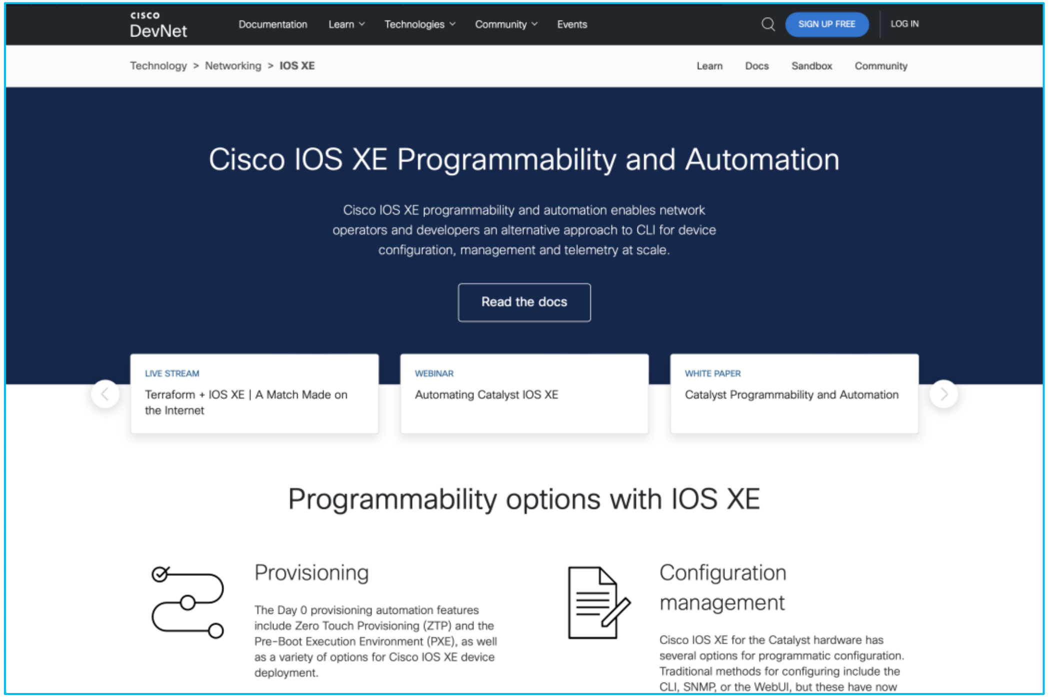
Friday Feb 9

Learn more about sessions this week and on-demand: https://blogs.cisco.com/developer/iosxeciscoliveemea2024

Session Levels:
Beginner
Intermediate

Cisco DevNet Site

The one-stop-shop for Cisco IOS XE Programmability resources including videos, white papers, labs and more!



Cisco IOS XE Programmability and Automation

Cisco IOS XE programmability and automation enables network operators and developers an alternative approach to CLI for device configuration, management and telemetry at scale.

Read the docs

LIVE STREAM
Terraform + IOS XE | A Match Made on the Internet

WEBINAR
Automating Catalyst IOS XE

WHITE PAPER
Catalyst Programmability and Automation

Programmability options with IOS XE

Provisioning

The Day 0 provisioning automation features include Zero Touch Provisioning (ZTP) and the Pre-Boot Execution Environment (PXE), as well as a variety of options for Cisco IOS XE device deployment.

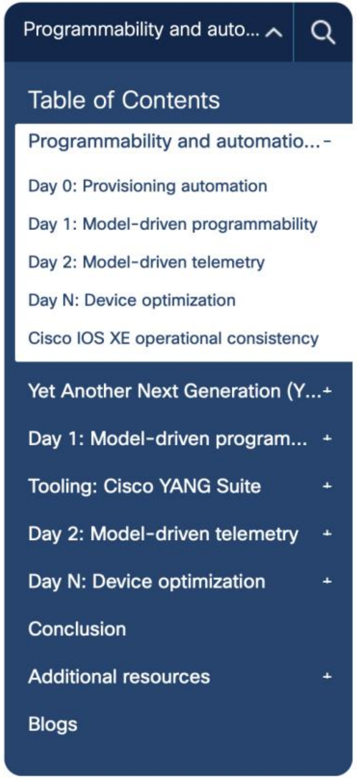
Configuration management

Cisco IOS XE for the Catalyst hardware has several options for programmatic configuration. Traditional methods for configuring include the CLI, SNMP, or the WebUI, but these have now

https://developer.cisco.com/iosxe/

http://cs.co/apiwp

API White Paper



Programmability and auto...

Table of Contents

Programmability and automatio... -

Day 0: Provisioning automation

Day 1: Model-driven programmability

Day 2: Model-driven telemetry

Day N: Device optimization

Cisco IOS XE operational consistency

Yet Another Next Generation (Y... +

Day 1: Model-driven program... +

Tooling: Cisco YANG Suite +

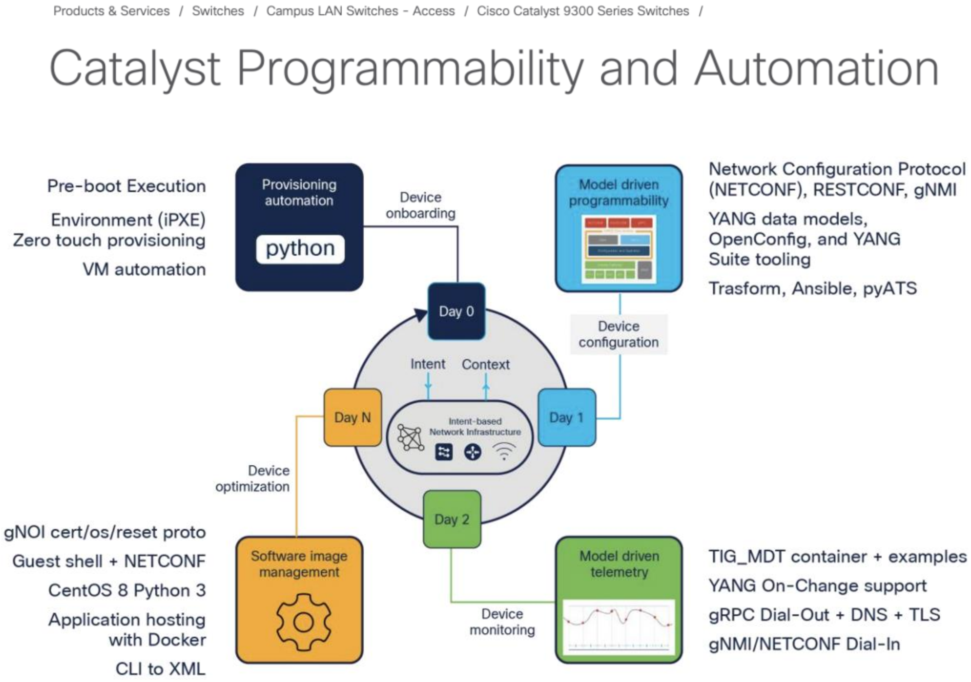
Day 2: Model-driven telemetry +

Day N: Device optimization +

Conclusion

Additional resources +

Blogs



Products & Services / Switches / Campus LAN Switches - Access / Cisco Catalyst 9300 Series Switches /

Catalyst Programmability and Automation

Pre-boot Execution Environment (iPXE)
Zero touch provisioning
VM automation

Network Configuration Protocol (NETCONF), RESTCONF, gNMI
YANG data models, OpenConfig, and YANG Suite tooling
Trasform, Ansible, pyATS

gNOI cert/os/reset proto
Guest shell + NETCONF
CentOS 8 Python 3
Application hosting with Docker
CLI to XML

TIG_MDT container + examples
YANG On-Change support
gRPC Dial-Out + DNS + TLS
gNMI/NETCONF Dial-In

Website: https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9300-series-switches/nb-06-catalyst-programmability-automation-wp.html

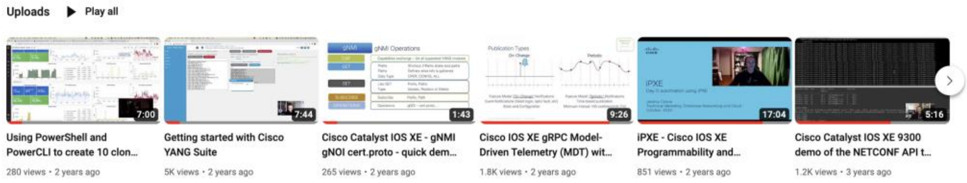
PDF: https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9300-series-switches/nb-06-catalyst-programmability-automation-wp.pdf

Webinar with live demos & examples: https://www.youtube.com/watch?v=LdcK5PnPu2I

http://cs.co/apiwppdf

Cisco Programmability on YouTube

https://www.youtube.com/jeremycohoe1
https://www.youtube.com/ciscoenprogrammability
https://www.youtube.com/ciscocatalysttv



Using PowerShell and PowerCLI to create 10 clon... (7:00) — 280 views • 2 years ago

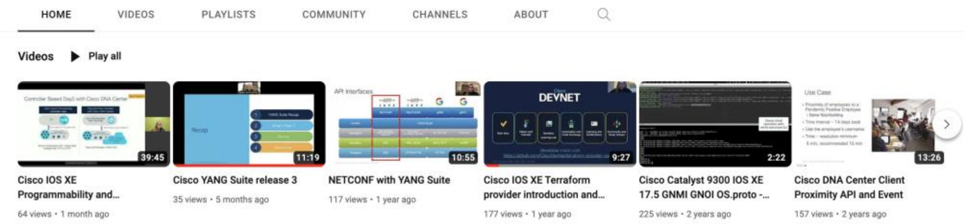
Getting started with Cisco YANG Suite (7:44) — 5K views • 2 years ago

Cisco Catalyst IOS XE - gNMI gNOI cert.proto - quick dem... (1:43) — 265 views • 2 years ago

Cisco IOS XE gRPC Model-Driven Telemetry (MDT) wit... (9:26) — 1.8K views • 2 years ago

iPXE - Cisco IOS XE Programmability and... (17:04) — 851 views • 2 years ago

Cisco Catalyst IOS XE 9300 demo of the NETCONF API t... (5:16) — 1.2K views • 3 years ago



Cisco IOS XE Programmability and... (39:45) — 64 views • 1 month ago

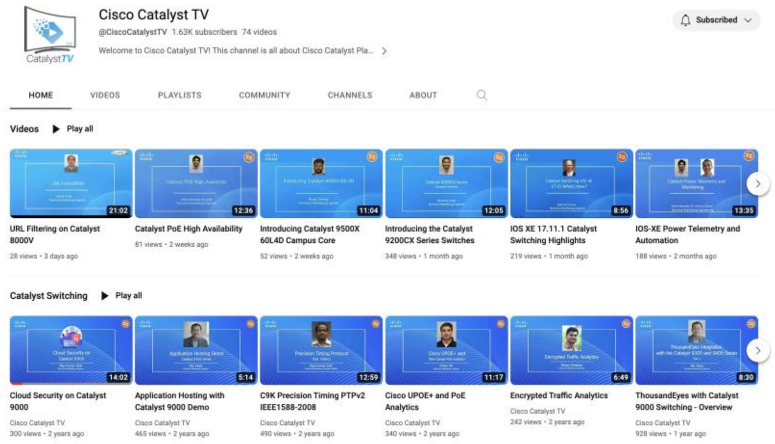
Cisco YANG Suite release 3 (11:19) — 35 views • 5 months ago

NETCONF with YANG Suite (10:55) — 117 views • 1 year ago

Cisco IOS XE Terraform provider introduction and... (9:27) — 177 views • 1 year ago

Cisco Catalyst 9300 IOS XE 17.5 GNMI GNOI OS.proto -... (2:22) — 225 views • 2 years ago

Cisco DNA Center Client Proximity API and Event (13:26) — 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...

URL Filtering on Catalyst 8000V (21:02) — 28 views • 3 days ago

Catalyst PoE High Availability (12:36) — 81 views • 2 weeks ago

Introducing Catalyst 9500X 60L4D Campus Core (11:04) — 52 views • 2 weeks ago

Introducing the Catalyst 9200CX Series Switches (12:05) — 348 views • 1 month ago

IOS XE 17.11.1 Catalyst Switching Highlights (8:56) — 219 views • 1 month ago

IOS-XE Power Telemetry and Automation (13:35) — 188 views • 2 months ago

Cloud Security on Catalyst 9000 (14:02) — Cisco Catalyst TV — 300 views • 2 years ago

Application Hosting with Catalyst 9000 Demo (5:14) — Cisco Catalyst TV — 465 views • 2 years ago

C9K Precision Timing PTPv2 IEEE1588-2008 (12:59) — Cisco Catalyst TV — 490 views • 2 years ago

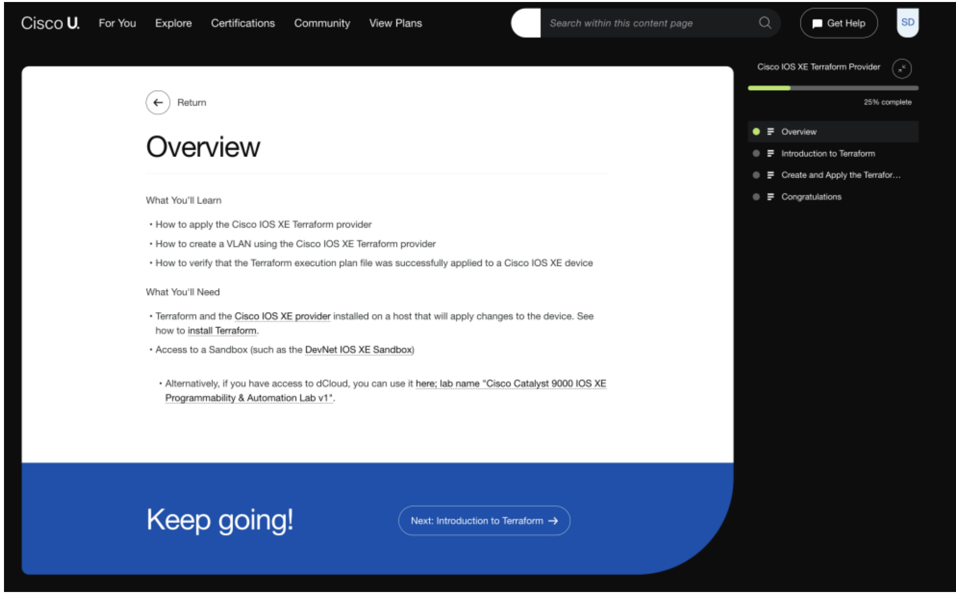
Cisco UPOE+ and PoE Analytics (11:17) — Cisco Catalyst TV — 340 views • 2 years ago

Encrypted Traffic Analytics (6:49) — Cisco Catalyst TV — 242 views • 2 years ago

ThousandEyes with Catalyst 9000 Switching - Overview (8:30) — Cisco Catalyst TV — 928 views • 1 year ago

Cisco U modules

- ZTP: https://ondemandelearning.cisco.com/apollo-alpha/tc-iosxe-ztp/pages/1
- Terraform: https://ondemandelearning.cisco.com/apollo-alpha/tc-terraform-ios-xe/pages/1
- YANG Suite + NETCONF: https://ondemandelearning.cisco.com/apollo-alpha/tc-yangsuite-netconf/pages/1
- YANG Suite + RESTCONF: https://ondemandelearning.cisco.com/apollo-alpha/tc-yangsuite-restconf/pages/1



Cisco U.
For You
Explore
Certifications
Community
View Plans
Search within this content page
Get Help
SD
Cisco IOS XE Terraform Provider
25% complete
Overview
Introduction to Terraform
Create and Apply the Terrafor...
Congratulations
Return

Overview

What You'll Learn

- How to apply the Cisco IOS XE Terraform provider
- How to create a VLAN using the Cisco IOS XE Terraform provider
- How to verify that the Terraform execution plan file was successfully applied to a Cisco IOS XE device

What You'll Need

- Terraform and the Cisco IOS XE provider installed on a host that will apply changes to the device. See how to install Terraform.
- Access to a Sandbox (such as the DevNet IOS XE Sandbox)
  - Alternatively, if you have access to dCloud, you can use it here; lab name "Cisco Catalyst 9000 IOS XE Programmability & Automation Lab v1".

Keep going!

CISCO
DevNet

WE ARE CISCO DEVNET 10

Continue to Learn, Code and Build with Cisco DevNet!

Get access to an exclusive learning module filled with digital learning opportunities on topics including Cisco Networking and more.

Scan QR Code to get started.

The bridge to possible

Thank you

cisco Live!

Let’s go