The bridge to possible

Terraform with Cisco IOS XE

Explore the Cisco IOS XE Terraform Provider & Beyond!

Speaker name
Speaker title

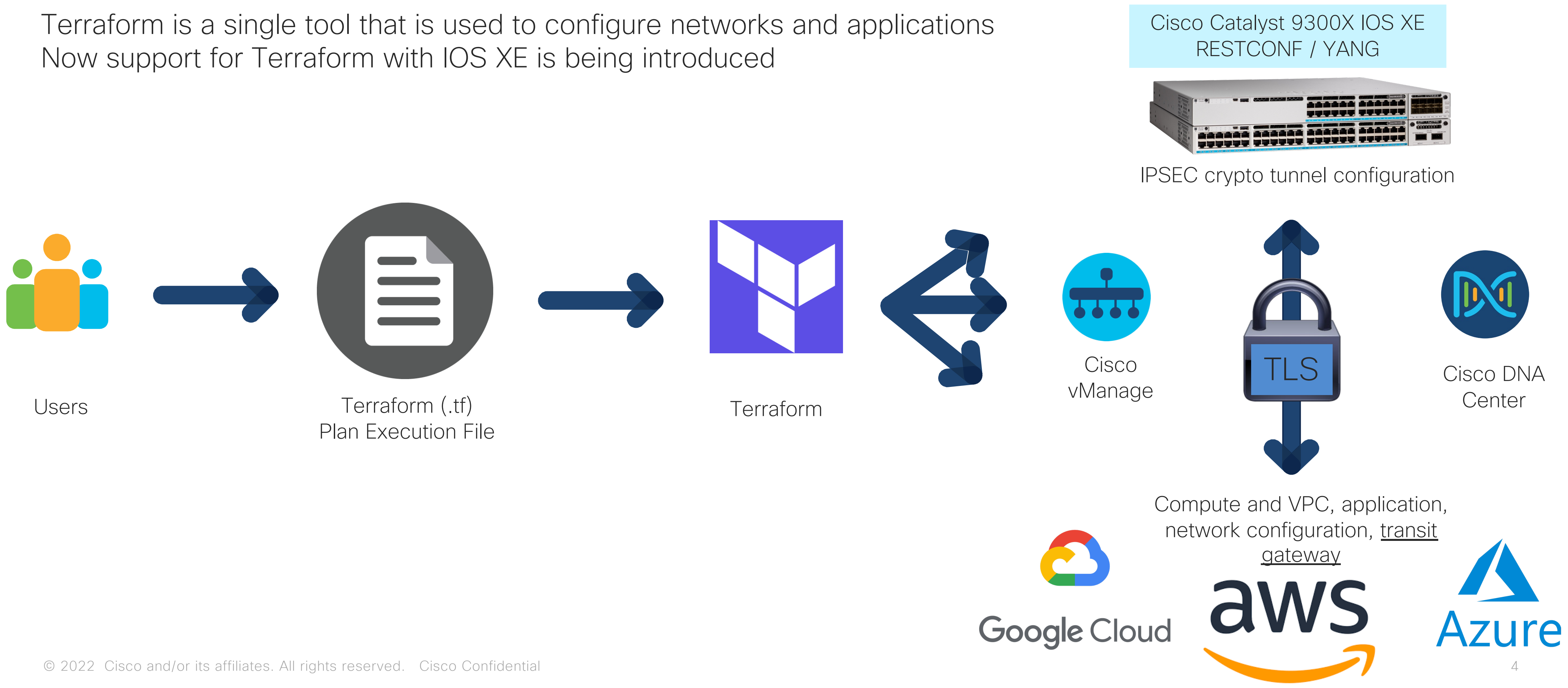
Overview

- Intro to Terraform
- Getting Started
- Demos
  - ACL & VLAN
  - IPsec
  - App Hosting with ThousandEyes
- Evolution and Adoption
- Troubleshooting & Resources

Intro to Terraform

Why Terraform ?

Terraform is a single tool that is used to configure networks and applications
Now support for Terraform with IOS XE is being introduced

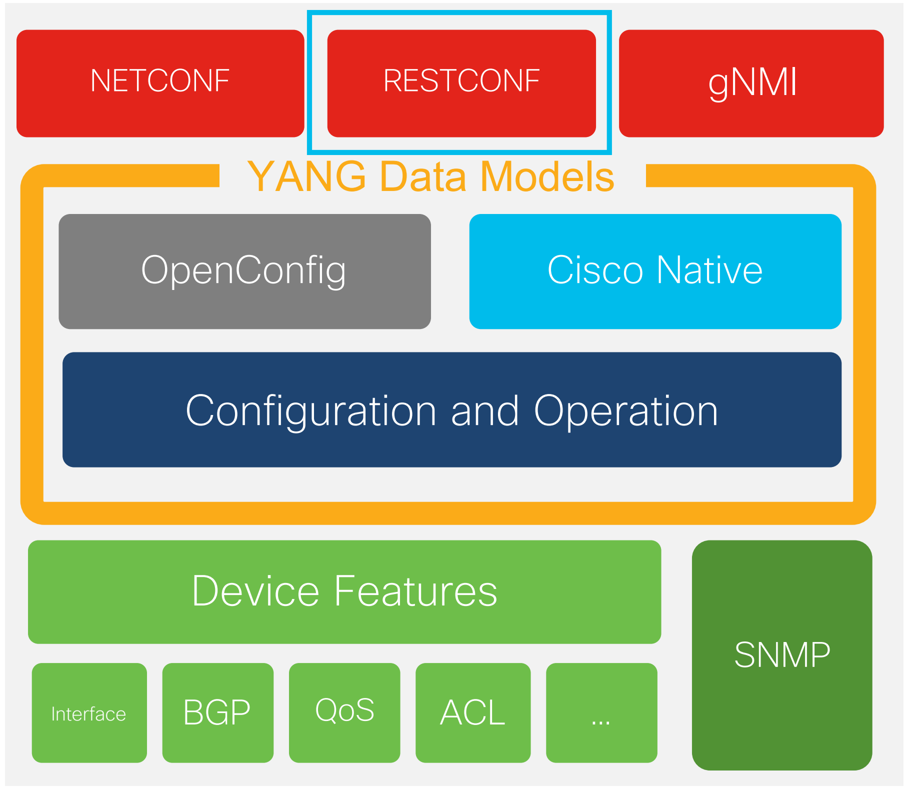

Cisco Catalyst 9300X IOS XE
RESTCONF / YANG
IPSEC crypto tunnel configuration
Users
Terraform (.tf)
Plan Execution File
Terraform
Cisco vManage
TLS
Cisco DNA Center
Compute and VPC, application, network configuration, transit gateway
Google Cloud
aws
Azure

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

Terraform uses the RESTCONF API



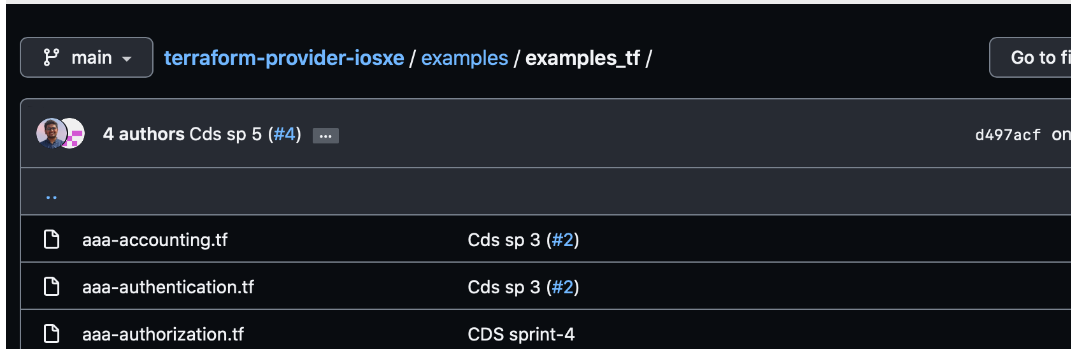
IOS XE Programmability integration with Terraform

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


main terraform-provider-iosxe / examples / examples_tf /

4 authors Cds sp 5 (#4) d497acf

..

aaa-accounting.tf	Cds sp 3 (#2)
aaa-authentication.tf	Cds sp 3 (#2)
aaa-authorization.tf	CDS sprint-4

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

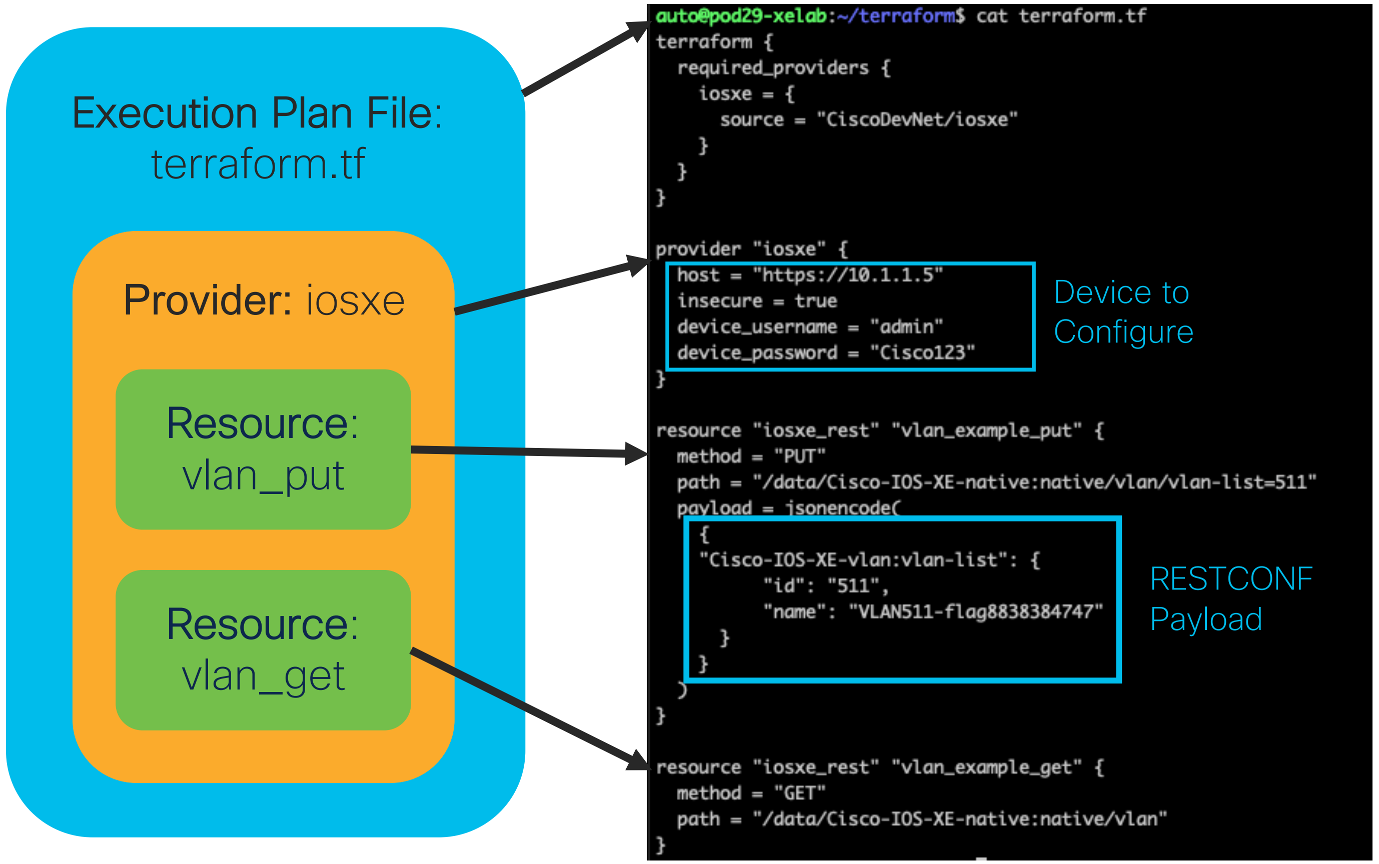
Terraform Terminology

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"
}
```

Example Terraform file

```
terraform {
  required_providers {
    iosxe = {
      source = "local.plugin/ciscodevnet/iosxe"
    }
  }
}

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

provider "iosxe" {
  host = "https://128.107.251.88"
  insecure = true
  device_username = "netadmin"
  device_password = "C1sc0dna"
}

# crypto all
resource "iosxe_rest" "crypto_example_post" {
  method = "PATCH"
  path = "/data/Cisco-IOS-XE-native:native/crypto"
  payload = jsonencode(

{
    "Cisco-IOS-XE-native:crypto": {
       "Cisco-IOS-XE-crypto:ikev2": {
          "keyring": [
            {
              "name": "aws_tgw_bgp_2_backup",
              "peer": [
                {
                  "name": "aws_tgw_bgp_2_backup",
                  "address": {
                    "ipv4": {
                      "ipv4-address": "0.0.0.0",
                      "ipv4-mask": "0.0.0.0"
                    }
                  },
                  "pre-shared-key": {
                    "key": "uNZptlnyDbRUFZxXRBImilyYouoDmLVb"
                  }
                }
              ]
            }
          ],
          "policy": [
            {
              "name": "aws_tgw_bgp_2_backup",
              "match": {
                "fvrf": {
                  "any": [null]
                }
              },
              "proposal": [
                {
                  "proposals": "aws_tgw_bgp_2_backup"
                }
              ]
            }
          ],
          "profile": [
            {
              "name": "aws_tgw_bgp_2_backup",
              "authentication": {
                "local": {
                  "pre-share": {
                  }
                },
                "remote": {
                  "pre-share": {
                  }
                }
              },
              "config-exchange": {
                "request-1": false
              },
              "dpd": {
                "interval": 10,
                "retry": 2,
                "query": "periodic"
              },
              "identity": {
                "local": {
                  "address": "128.107.251.88"
                }
              },
              "keyring": {
                "local": {
                  "name": "aws_tgw_bgp_2_backup"
                }
              },
              "match": {
                "identity": {
                  "remote": {
                    "address": {
                      "ipv4": [
                        {
                          "ipv4-address": "52.52.2.74",
                          "ipv4-mask": "255.255.255.0"
                        }
                      ]
                    }
                  }
                }
              }
            }
          ],
          "proposal": [
            {
              "name": "aws_tgw_bgp_2_backup",
              "encryption": {
                "aes-cbc-256": [null]
              },
              "group": {
                "fourteen": [null],
                "nineteen": [null],
                "twenty": [null]
              },
              "integrity": {
                "sha1": [null]
              }
            }
          ]
        },
        "Cisco-IOS-XE-crypto:ipsec": {
          "transform-set": [
            {
              "tag": "aws_tgw_bgp_2_backup",
              "esp": "esp-aes",
              "esp-hmac": "esp-sha-hmac",
              "mode": {
                "tunnel-choice": [null]
              }
            }
          ],
          "profile": [
            {
              "name": "aws_tgw_bgp_2_backup",
              "set": {
                "ikev2-profile": "aws_tgw_bgp_2_backup"
              }
            }
          ]
        }
    }
}
  )
}


# Create Tunnel 303
resource "iosxe_rest" "tunnel_example_post" {
  method = "POST"
  path = "/data/Cisco-IOS-XE-native:native/interface"
  payload = jsonencode(
    {
     "Cisco-IOS-XE-native:Tunnel": {
    "name": 303,
    "description": "##Tunnel to AWS TGW##",
    "ip": {
      "address": {
        "primary": {
          "address": "169.254.26.254",
          "mask": "255.255.255.252"
        }
      }
    },
    "Cisco-IOS-XE-tunnel:tunnel": {
      "source": "Vlan2",
      "destination-config": {
        "ipv4": "52.52.2.74"
      },
      "mode": {
        "ipsec": {
          "ipv4": {
          }
        }
      },
      "protection": {
        "Cisco-IOS-XE-crypto:ipsec": {
          "profile-option": {
            "name": "aws_tgw_bgp_2_backup"
          }
        }
      }
    }
  }
    }
  )
}
```

Template Terraform File

```
terraform {
  required_providers {
    iosxe = {
      source = "local.plugin/ciscodevnet/iosxe"
    }
  }
}

provider "iosxe" {
  host = ”<DEVICE_IP>"
  insecure = true
  device_username = ”<DEVICE_USERNAME>"
  device_password = ”<DEVICE_PASSWORD>”
}

# feature to configure
resource "iosxe_rest" "example_patch" {
  method = "PATCH"
  path = ”<PATH>” # example: "/data/Cisco-IOS-XE-native:native/crypto”
  payload = jsonencode(
   <RESPONSE_FOR_FEATURE_USING_FORMAT_RESTCONF-JSON> # example:  "Cisco-IOS-XE-native:crypto": { … }
)
}
```

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
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

Cat9k-1(config)#restconf
```

Enable AAA

Enable NETCONF

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
```

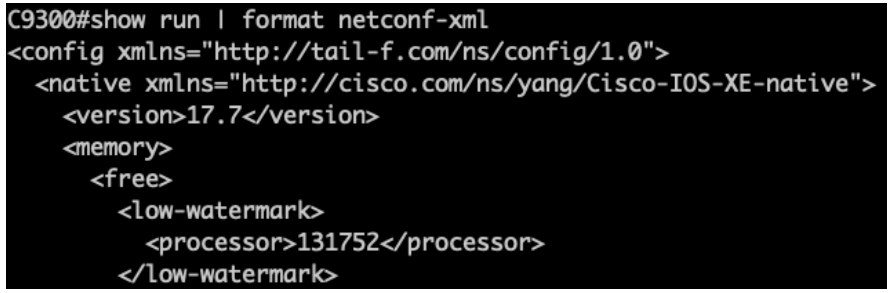
4. Apply Terraform VLAN example

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

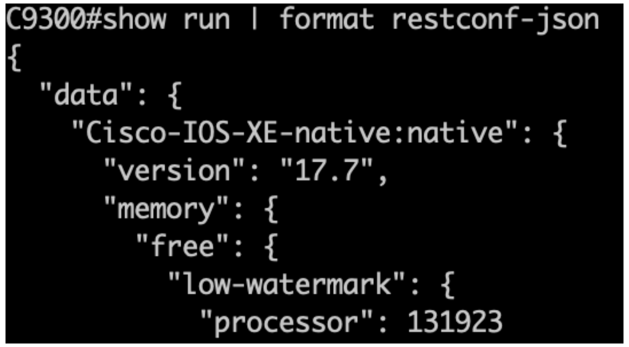
CLI to YANG

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

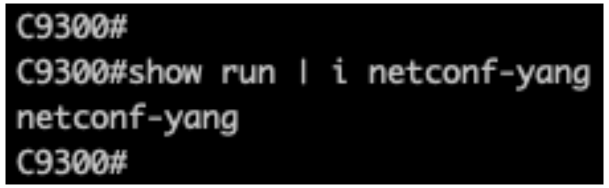
```
show run | format netconf-xml
show run | format restconf-json
```



```
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>
```



```
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#
```

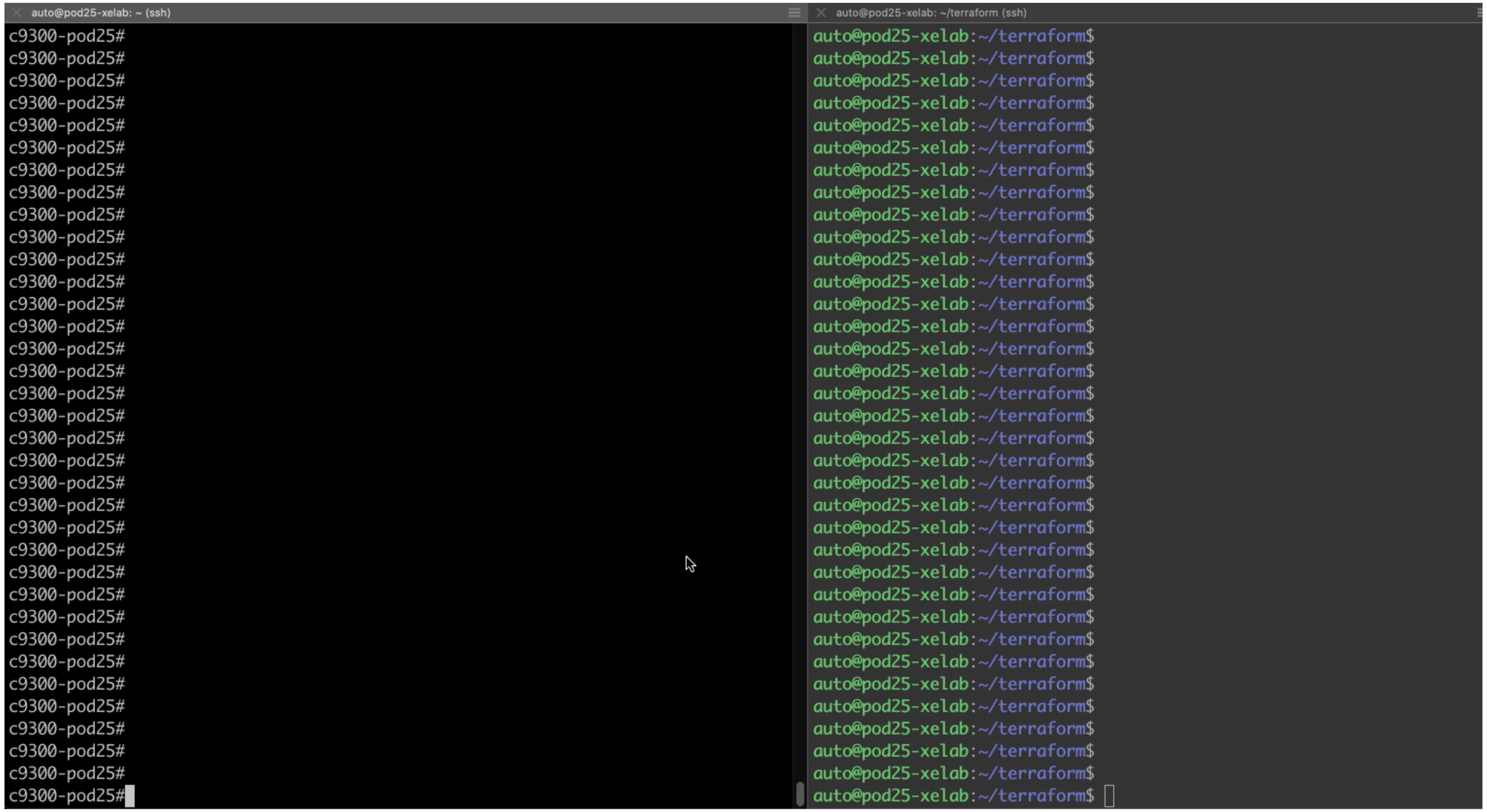
LTROPS-1836 13

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

Demos

- ACL & VLAN
- IPsec
- App Hosting with ThousandEyes
- EVPN

Configure ACL & VLAN



auto@pod25-xelab: ~ (ssh)

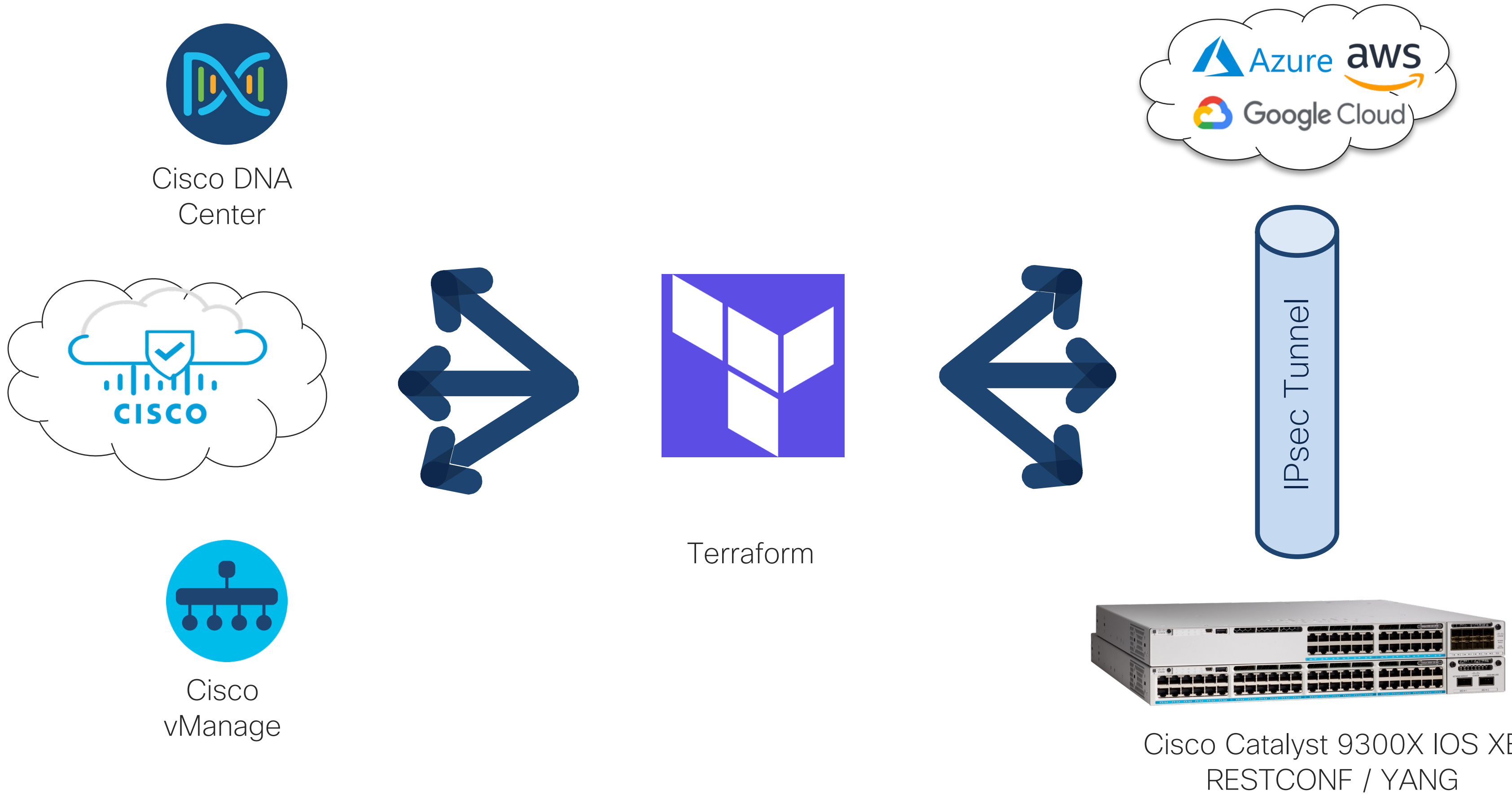
```
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
c9300-pod25#
```

auto@pod25-xelab: ~/terraform (ssh)

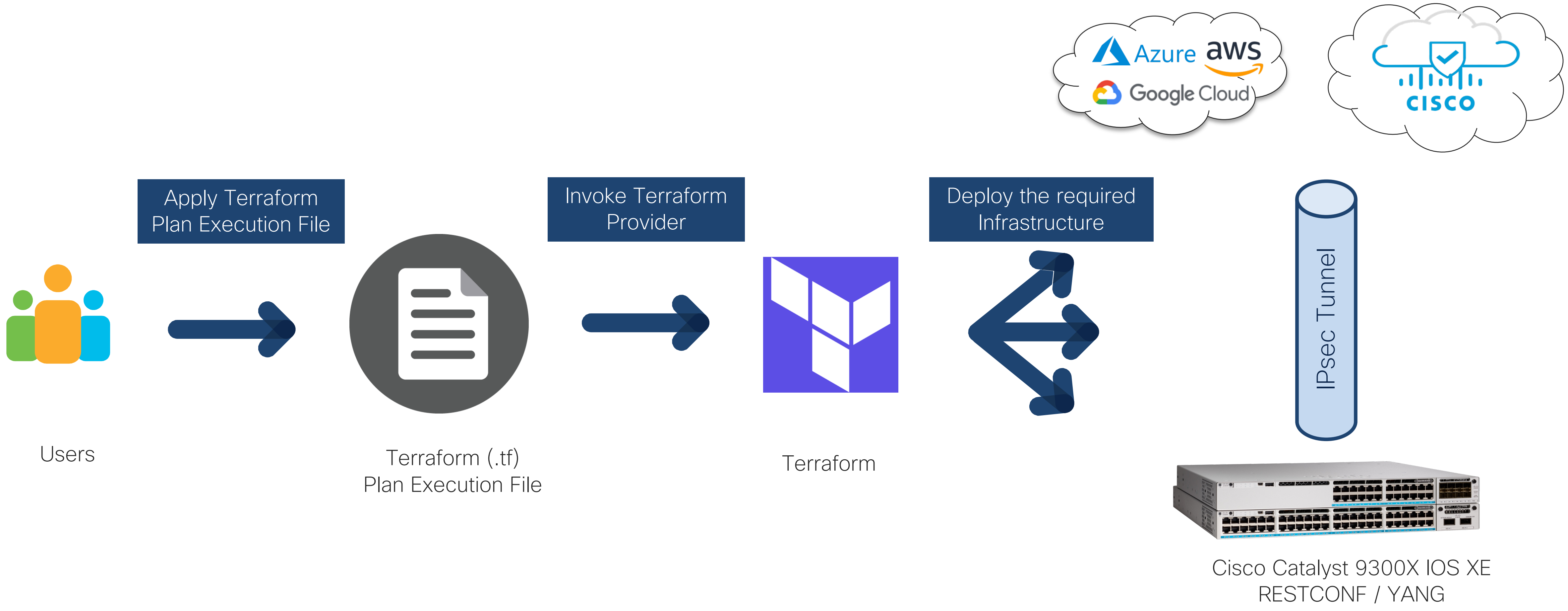
```
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
auto@pod25-xelab:~/terraform$
```

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

What can Terraform do?

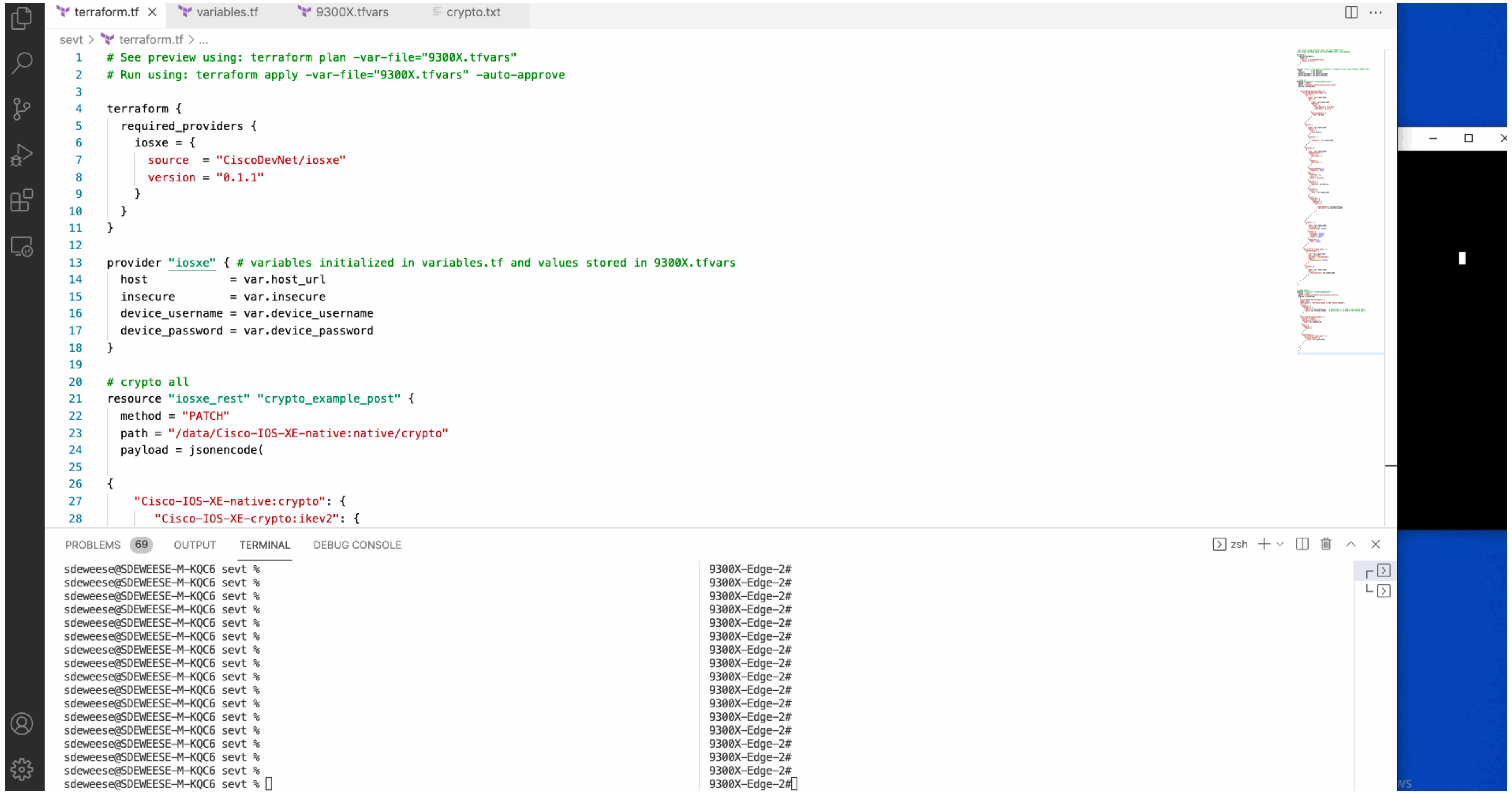


Crypto IPsec



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

Terraform + Crypto IPsec Demo



```
# See preview using: terraform plan -var-file="9300X.tfvars"
# Run using: terraform apply -var-file="9300X.tfvars" -auto-approve

terraform {
  required_providers {
    iosxe = {
      source  = "CiscoDevNet/iosxe"
      version = "0.1.1"
    }
  }
}

provider "iosxe" { # variables initialized in variables.tf and values stored in 9300X.tfvars
  host            = var.host_url
  insecure        = var.insecure
  device_username = var.device_username
  device_password = var.device_password
}

# crypto all
resource "iosxe_rest" "crypto_example_post" {
  method = "PATCH"
  path = "/data/Cisco-IOS-XE-native:native/crypto"
  payload = jsonencode(

{
    "Cisco-IOS-XE-native:crypto": {
      "Cisco-IOS-XE-crypto:ikev2": {
```

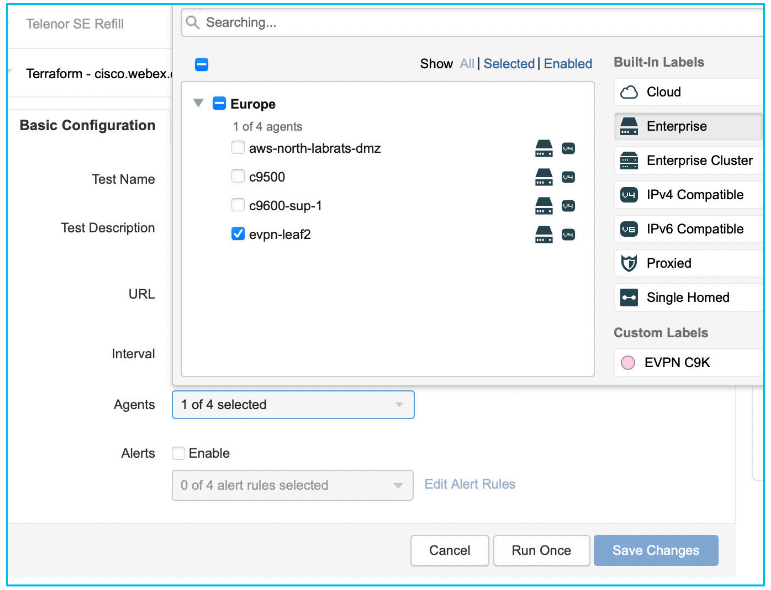
```
sdeweese@SDEWEESE-M-KQC6 sevt %
sdeweese@SDEWEESE-M-KQC6 sevt %
sdeweese@SDEWEESE-M-KQC6 sevt %
sdeweese@SDEWEESE-M-KQC6 sevt %
sdeweese@SDEWEESE-M-KQC6 sevt %
sdeweese@SDEWEESE-M-KQC6 sevt %
sdeweese@SDEWEESE-M-KQC6 sevt %
sdeweese@SDEWEESE-M-KQC6 sevt %
sdeweese@SDEWEESE-M-KQC6 sevt %
sdeweese@SDEWEESE-M-KQC6 sevt %
sdeweese@SDEWEESE-M-KQC6 sevt %
sdeweese@SDEWEESE-M-KQC6 sevt %
sdeweese@SDEWEESE-M-KQC6 sevt %
sdeweese@SDEWEESE-M-KQC6 sevt %
sdeweese@SDEWEESE-M-KQC6 sevt %
sdeweese@SDEWEESE-M-KQC6 sevt %
sdeweese@SDEWEESE-M-KQC6 sevt %
```

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

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"
  }
}
```



Telenor SE Refill

Terraform - cisco.webex.

Basic Configuration

Test Name

Test Description

URL

Interval

Agents: 1 of 4 selected

Alerts: Enable

0 of 4 alert rules selected — Edit Alert Rules

Searching...

Show All | Selected | Enabled

Europe — 1 of 4 agents
- aws-north-labrats-dmz
- c9500
- c9600-sup-1
- [✓] evpn-leaf2

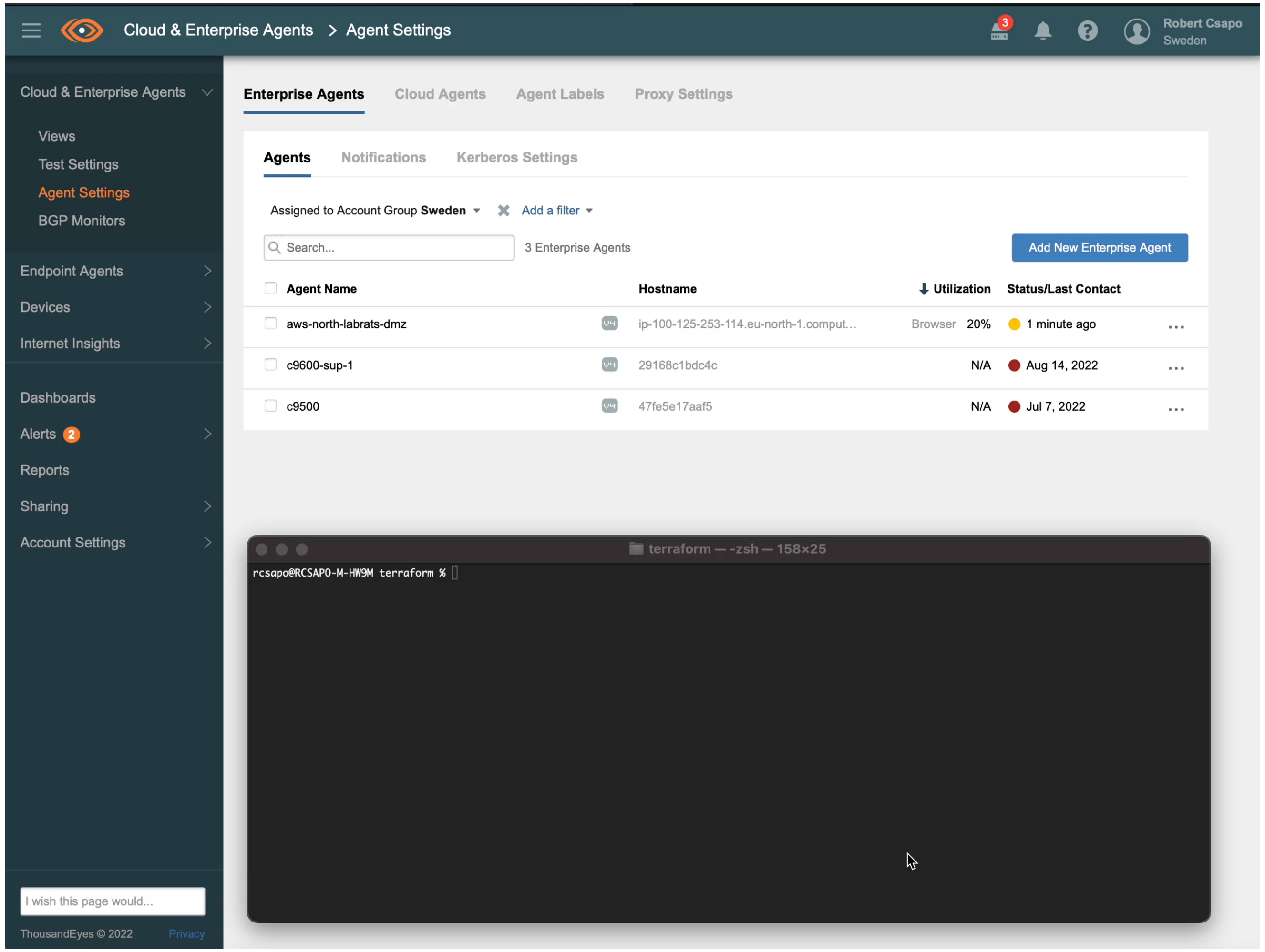
Built-In Labels
- Cloud
- Enterprise
- Enterprise Cluster
- IPv4 Compatible
- IPv6 Compatible
- Proxied
- Single Homed

Custom Labels
- EVPN C9K

Cancel | Run Once | Save Changes

https://github.com/robertcsapo/terraform-provider-ciscoapphosting
https://registry.terraform.io/providers/robertcsapo/ciscoapphosting/

Terraform + ThousandEyes Demo



Cloud & Enterprise Agents > Agent Settings

Robert Csapo
Sweden

Enterprise Agents | Cloud Agents | Agent Labels | Proxy Settings

Agents | Notifications | Kerberos Settings

Assigned to Account Group **Sweden** · Add a filter

3 Enterprise Agents

Agent Name	Hostname	Utilization	Status/Last Contact
aws-north-labrats-dmz	ip-100-125-253-114.eu-north-1.comput...	Browser 20%	● 1 minute ago
c9600-sup-1	29168c1bdc4c	N/A	● Aug 14, 2022
c9500	47fe5e17aaf5	N/A	● Jul 7, 2022

```
terraform — -zsh — 158×25
rcsapo@RCSAPO-M-HW9M terraform % 
```

Steps:

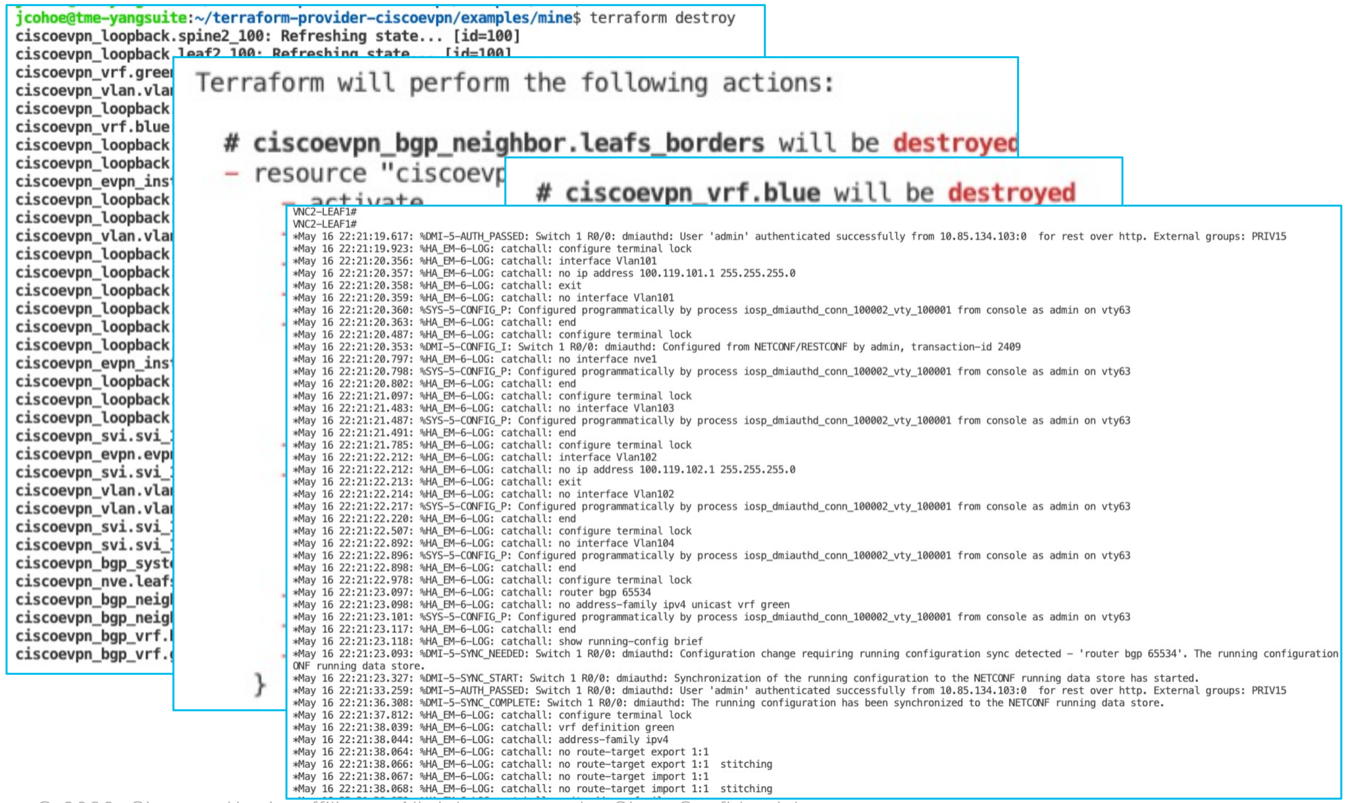
1. Test App Hosting Terraform Provider
2. Apply App Hosting Terraform Provider on C9300 switches
3. Verify new Enterprise Agents has been added in ThousandEyes Dashboard
4. Run the current test
5. Destroy Terraform to delete the Enterprise Agents and Test

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

Declarative EVPN management with Terraform

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

Robert: https://github.com/robertcsapo/terraform-provider-ciscoevpn/
https://registry.terraform.io/providers/robertcsapo/ciscoevpn/latest
Daniel: https://github.com/netascode/terraform-iosxe-evpn-example
EVPN OSPF Underlay Module: https://registry.terraform.io/modules/netascode/evpn-ospf-underlay/iosxe/latest
EVPN Overlay Module: https://registry.terraform.io/modules/netascode/evpn-overlay/iosxe/latest

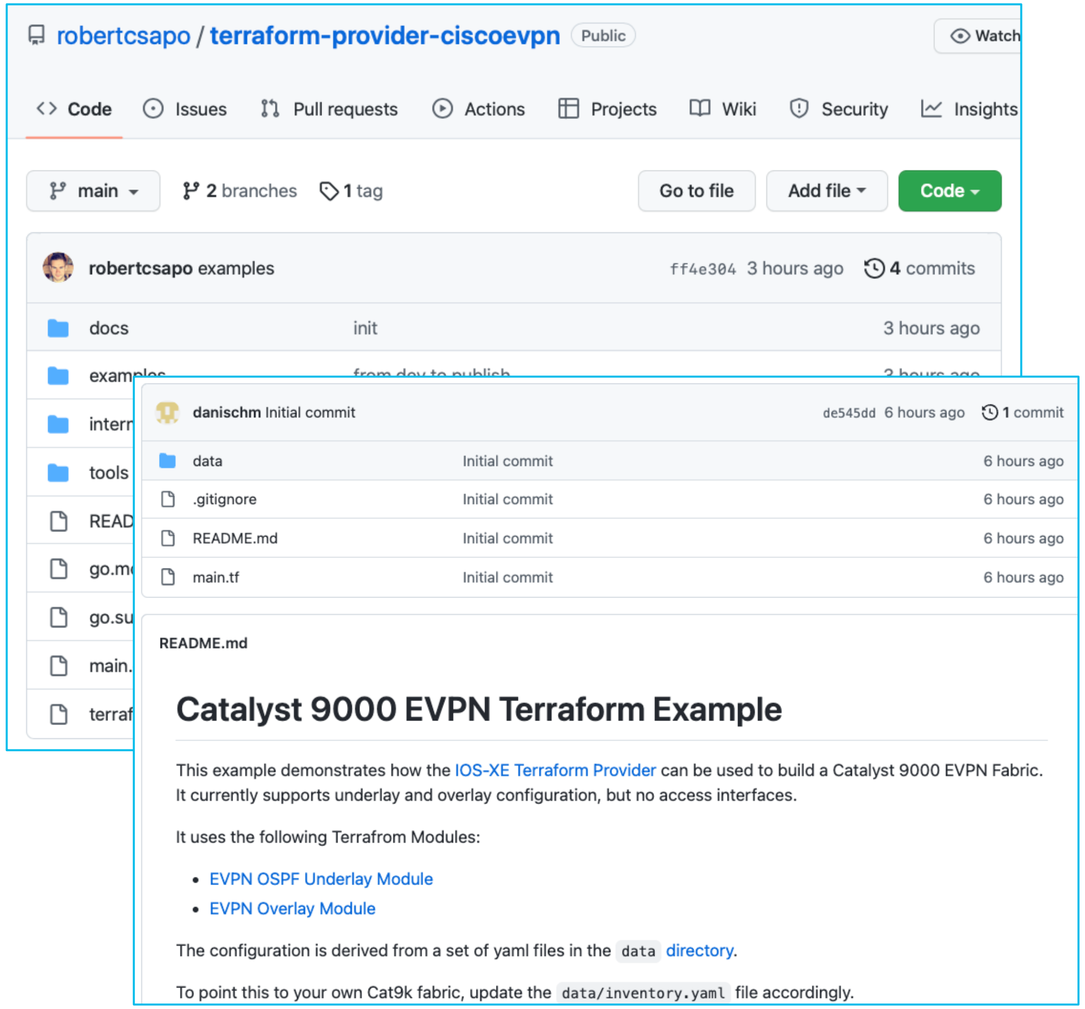


```
jcohoe@tme-yangsuite:~/terraform-provider-ciscoevpn/examples/mine$ terraform destroy
ciscoevpn_loopback.spine2_100: Refreshing state... [id=100]
ciscoevpn_loopback.leaf2_100: Refreshing state... [id=100]
ciscoevpn_vrf.green
ciscoevpn_vlan.vla
ciscoevpn_loopback
ciscoevpn_vrf.blue
ciscoevpn_loopback
ciscoevpn_loopback
ciscoevpn_evpn_ins
ciscoevpn_loopback
ciscoevpn_loopback
ciscoevpn_vlan.vla
ciscoevpn_loopback
ciscoevpn_loopback
ciscoevpn_loopback
ciscoevpn_loopback
ciscoevpn_loopback
ciscoevpn_loopback
ciscoevpn_evpn_ins
ciscoevpn_loopback
ciscoevpn_loopback
ciscoevpn_loopback
ciscoevpn_svi.svi_
ciscoevpn_evpn.evp
ciscoevpn_svi.svi_
ciscoevpn_vlan.vla
ciscoevpn_vlan.vla
ciscoevpn_svi.svi_
ciscoevpn_svi.svi_
ciscoevpn_bgp_syst
ciscoevpn_nve.leaf
ciscoevpn_bgp_neig
ciscoevpn_bgp_neig
ciscoevpn_bgp_vrf.
ciscoevpn_bgp_vrf.
```

```
Terraform will perform the following actions:

  # ciscoevpn_bgp_neighbor.leafs_borders will be destroyed
  - resource "ciscoevp
      - activate
    }
```

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

```
VNC2-LEAF1#
VNC2-LEAF1#
*May 16 22:21:19.617: %DMI-5-AUTH_PASSED: Switch 1 R0/0: dmiauthd: User 'admin' authenticated successfully from 10.85.134.103:0  for rest over http. External groups: PRIV15
*May 16 22:21:19.923: %HA_EM-6-LOG: catchall: configure terminal lock
*May 16 22:21:20.356: %HA_EM-6-LOG: catchall: interface Vlan101
*May 16 22:21:20.357: %HA_EM-6-LOG: catchall: no ip address 100.119.101.1 255.255.255.0
*May 16 22:21:20.358: %HA_EM-6-LOG: catchall: exit
*May 16 22:21:20.359: %HA_EM-6-LOG: catchall: no interface Vlan101
*May 16 22:21:20.360: %SYS-5-CONFIG_P: Configured programmatically by process iosp_dmiauthd_conn_100002_vty_100001 from console as admin on vty63
*May 16 22:21:20.363: %HA_EM-6-LOG: catchall: end
*May 16 22:21:20.487: %HA_EM-6-LOG: catchall: configure terminal lock
*May 16 22:21:20.353: %DMI-5-CONFIG_I: Switch 1 R0/0: dmiauthd: Configured from NETCONF/RESTCONF by admin, transaction-id 2409
*May 16 22:21:20.797: %HA_EM-6-LOG: catchall: no interface nve1
*May 16 22:21:20.798: %SYS-5-CONFIG_P: Configured programmatically by process iosp_dmiauthd_conn_100002_vty_100001 from console as admin on vty63
*May 16 22:21:20.802: %HA_EM-6-LOG: catchall: end
*May 16 22:21:21.097: %HA_EM-6-LOG: catchall: configure terminal lock
*May 16 22:21:21.483: %HA_EM-6-LOG: catchall: no interface Vlan103
*May 16 22:21:21.487: %SYS-5-CONFIG_P: Configured programmatically by process iosp_dmiauthd_conn_100002_vty_100001 from console as admin on vty63
*May 16 22:21:21.491: %HA_EM-6-LOG: catchall: end
*May 16 22:21:21.785: %HA_EM-6-LOG: catchall: configure terminal lock
*May 16 22:21:22.212: %HA_EM-6-LOG: catchall: interface Vlan102
*May 16 22:21:22.212: %HA_EM-6-LOG: catchall: no ip address 100.119.102.1 255.255.255.0
*May 16 22:21:22.213: %HA_EM-6-LOG: catchall: exit
*May 16 22:21:22.214: %HA_EM-6-LOG: catchall: no interface Vlan102
*May 16 22:21:22.217: %SYS-5-CONFIG_P: Configured programmatically by process iosp_dmiauthd_conn_100002_vty_100001 from console as admin on vty63
*May 16 22:21:22.220: %HA_EM-6-LOG: catchall: end
*May 16 22:21:22.507: %HA_EM-6-LOG: catchall: configure terminal lock
*May 16 22:21:22.892: %HA_EM-6-LOG: catchall: no interface Vlan104
*May 16 22:21:22.896: %SYS-5-CONFIG_P: Configured programmatically by process iosp_dmiauthd_conn_100002_vty_100001 from console as admin on vty63
*May 16 22:21:22.898: %HA_EM-6-LOG: catchall: end
*May 16 22:21:22.978: %HA_EM-6-LOG: catchall: configure terminal lock
*May 16 22:21:23.097: %HA_EM-6-LOG: catchall: router bgp 65534
*May 16 22:21:23.098: %HA_EM-6-LOG: catchall: no address-family ipv4 unicast vrf green
*May 16 22:21:23.101: %SYS-5-CONFIG_P: Configured programmatically by process iosp_dmiauthd_conn_100002_vty_100001 from console as admin on vty63
*May 16 22:21:23.117: %HA_EM-6-LOG: catchall: end
*May 16 22:21:23.118: %HA_EM-6-LOG: catchall: show running-config brief
*May 16 22:21:23.093: %DMI-5-SYNC_NEEDED: Switch 1 R0/0: dmiauthd: Configuration change requiring running configuration sync detected - 'router bgp 65534'. The running configuration
ONF running data store.
*May 16 22:21:23.327: %DMI-5-SYNC_START: Switch 1 R0/0: dmiauthd: Synchronization of the running configuration to the NETCONF running data store has started.
*May 16 22:21:33.259: %DMI-5-AUTH_PASSED: Switch 1 R0/0: dmiauthd: User 'admin' authenticated successfully from 10.85.134.103:0  for rest over http. External groups: PRIV15
*May 16 22:21:36.308: %DMI-5-SYNC_COMPLETE: Switch 1 R0/0: dmiauthd: The running configuration has been synchronized to the NETCONF running data store.
*May 16 22:21:37.812: %HA_EM-6-LOG: catchall: configure terminal lock
*May 16 22:21:38.039: %HA_EM-6-LOG: catchall: vrf definition green
*May 16 22:21:38.044: %HA_EM-6-LOG: catchall: address-family ipv4
*May 16 22:21:38.064: %HA_EM-6-LOG: catchall: no route-target export 1:1
*May 16 22:21:38.066: %HA_EM-6-LOG: catchall: no route-target export 1:1  stitching
*May 16 22:21:38.067: %HA_EM-6-LOG: catchall: no route-target import 1:1
*May 16 22:21:38.068: %HA_EM-6-LOG: catchall: no route-target import 1:1  stitching
```



robertcsapo examples	ff4e304 3 hours ago	4 commits
docs	init	3 hours ago
examples	from dev to publish	3 hours ago
intern		
tools		
READ		
go.m		
go.su		
main.		
terraf		

danischm Initial commit	de545dd 6 hours ago	1 commit
data	Initial commit	6 hours ago
.gitignore	Initial commit	6 hours ago
README.md	Initial commit	6 hours ago
main.tf	Initial commit	6 hours ago

README.md

Catalyst 9000 EVPN Terraform Example

This example demonstrates how the IOS-XE Terraform Provider can be used to build a Catalyst 9000 EVPN Fabric. It currently supports underlay and overlay configuration, but no access interfaces.

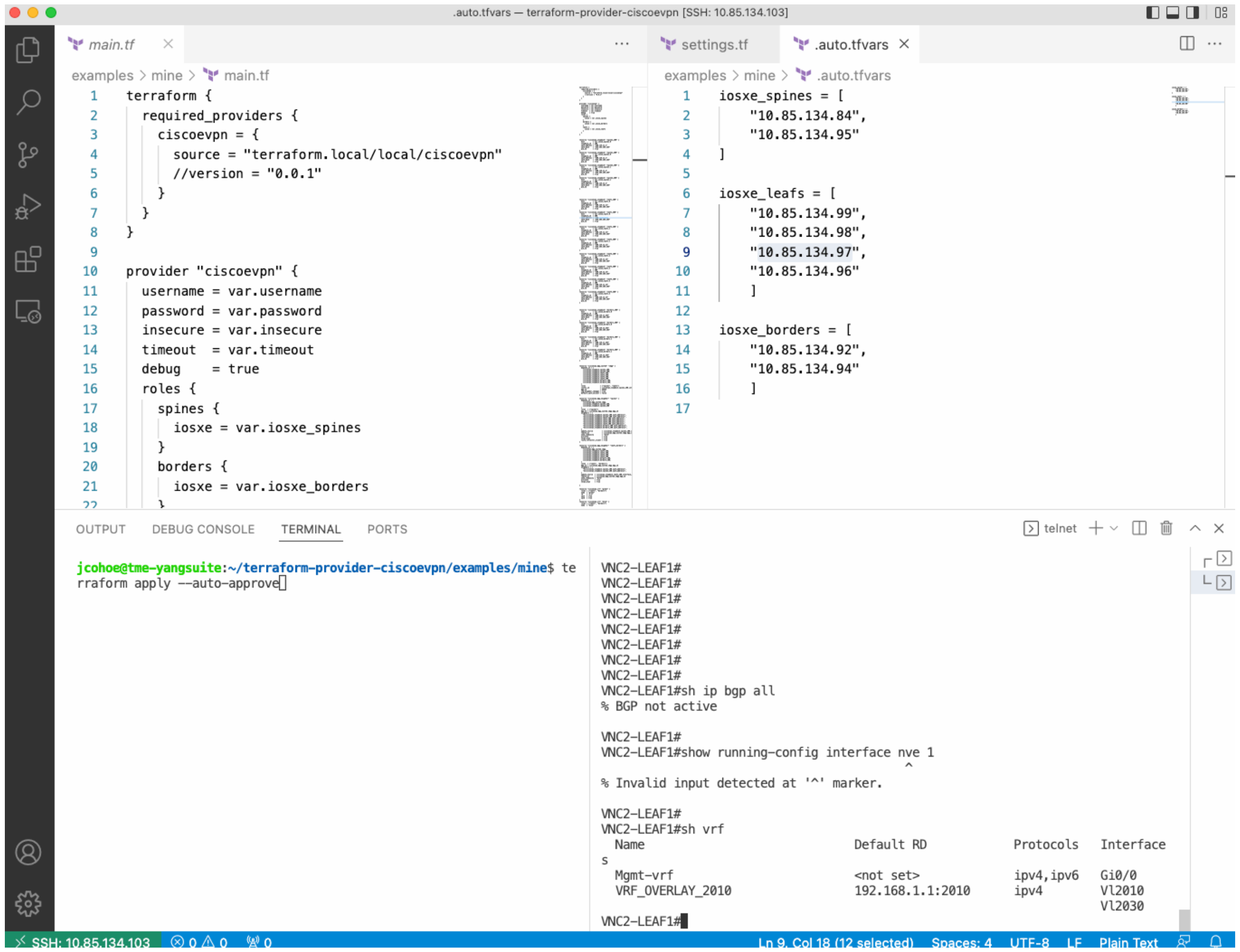
It uses the following Terrafrom Modules:

- EVPN OSPF Underlay Module
- EVPN Overlay Module

The configuration is derived from a set of yaml files in the `data` directory.

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

EVPN with Terraform Demo


.auto.tfvars — terraform-provider-ciscoevpn [SSH: 10.85.134.103]
main.tf
settings.tf
.auto.tfvars
examples > mine > main.tf
examples > mine > .auto.tfvars
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
telnet
SSH: 10.85.134.103
Ln 9, Col 18 (12 selected)
Spaces: 4
UTF-8
LF
Plain Text

```
terraform {
  required_providers {
    ciscoevpn = {
      source = "terraform.local/local/ciscoevpn"
      //version = "0.0.1"
    }
  }
}

provider "ciscoevpn" {
  username = var.username
  password = var.password
  insecure = var.insecure
  timeout  = var.timeout
  debug    = true
  roles {
    spines {
      iosxe = var.iosxe_spines
    }
    borders {
      iosxe = var.iosxe_borders
```

```
iosxe_spines = [
    "10.85.134.84",
    "10.85.134.95"
]

iosxe_leafs = [
    "10.85.134.99",
    "10.85.134.98",
    "10.85.134.97",
    "10.85.134.96"
    ]

iosxe_borders = [
    "10.85.134.92",
    "10.85.134.94"
    ]
```

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

```
VNC2-LEAF1#
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
s
  Mgmt-vrf                         <not set>             ipv4,ipv6   Gi0/0
  VRF_OVERLAY_2010                 192.168.1.1:2010      ipv4        Vl2010
                                                                     Vl2030

VNC2-LEAF1#
```

Evolution & Adoption

Evolution of Terraform Provider

March 2022
Imperative RESTCONF support

Current
Declarative Feature Providers

August 2022
BGP EVPN Providers

October 2022
App Hosting Provider

Coming Soon
Model Driven Telemetry Provider

Coming Soon
Providers you recommend

Phase 1 — Phase 2 — Phase 3

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

https://registry.terraform.io/providers/robertcsapo/ciscoevpn/1.0.1

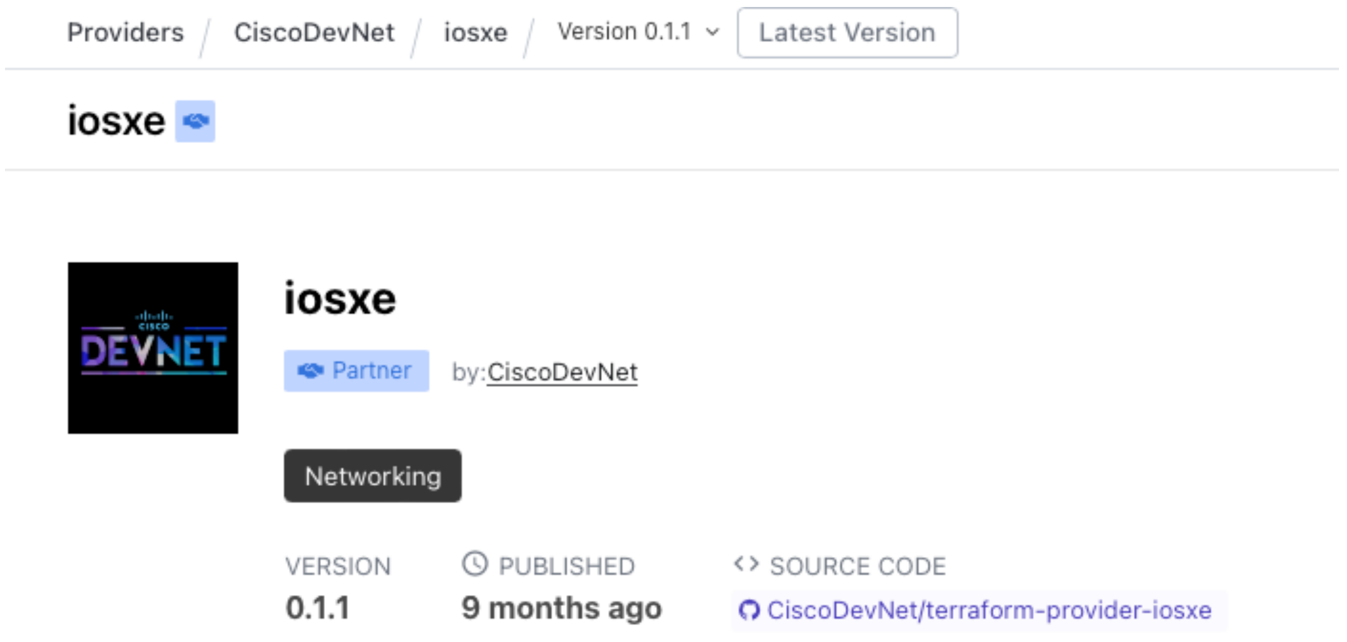
https://registry.terraform.io/providers/netascode/iosxe/latest/docs
https://github.com/netascode/terraform-iosxe-evpn-example

https://github.com/robertcsapo/terraform-provider-ciscoapphosting/tree/main/

Declarative providers leverage the SDK from the Phase 1 imperative provider

Terraform use and adoption

We continue to see increased adoption of the IOS XE terraform resources



Providers / CiscoDevNet / iosxe / Version 0.1.1 Latest Version

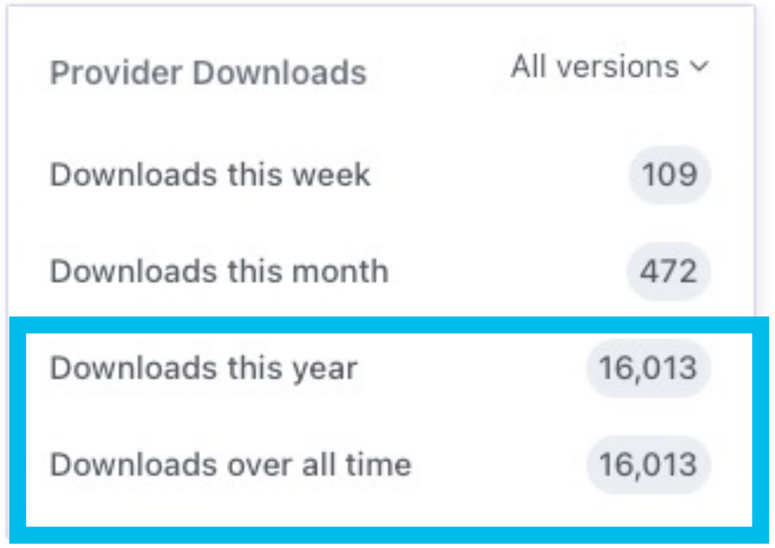
iosxe

iosxe

Partner by:CiscoDevNet

Networking

VERSION	PUBLISHED	SOURCE CODE
0.1.1	9 months ago	CiscoDevNet/terraform-provider-iosxe



Provider Downloads All versions

Downloads this week	109
Downloads this month	472
Downloads this year	16,013
Downloads over all time	16,013

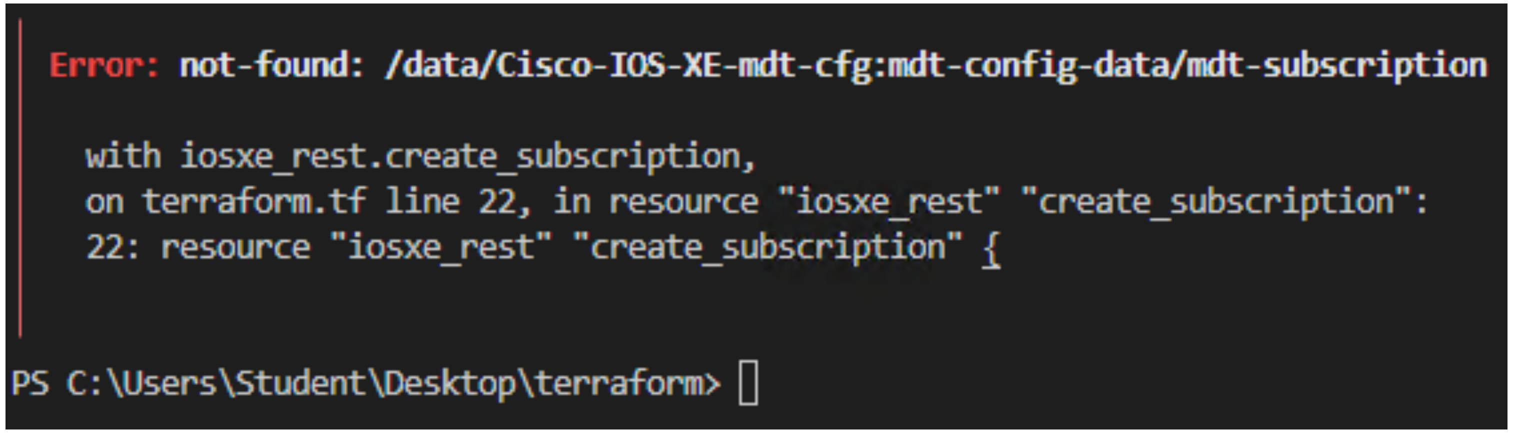
* As of 11/30/22

https://registry.terraform.io/providers/CiscoDevNet/iosxe/0.1.1

Troubleshooting & Resources

Troubleshooting

If the following error is found, it may be caused by the device not having restconf enabled



```
Error: not-found: /data/Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription

  with iosxe_rest.create_subscription,
  on terraform.tf line 22, in resource "iosxe_rest" "create_subscription":
  22: resource "iosxe_rest" "create_subscription" {


PS C:\Users\Student\Desktop\terraform> 
```

To fix this, simply add restconf in global config mode

```
Device(conf)# restconf
```

Cisco

DEVNET

Start Now

Videos and Tutorials

Sandbox

Learning Lab

Automation and Code Exchange

Learning and Certifications

Community and Study Groups

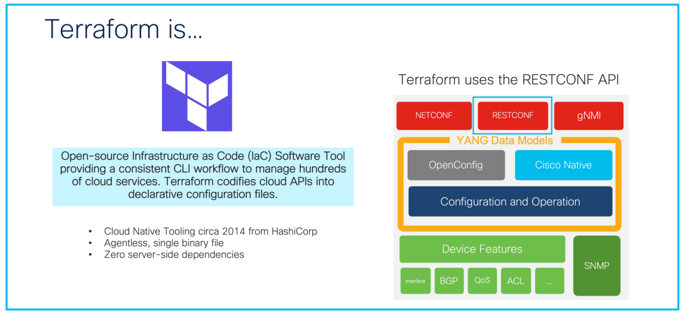
developer.cisco.com

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

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/#PtsT8eJFI



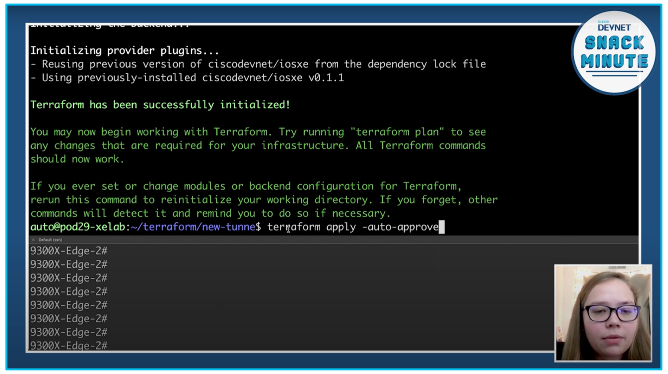
Terraform is…

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

Terraform uses the RESTCONF API

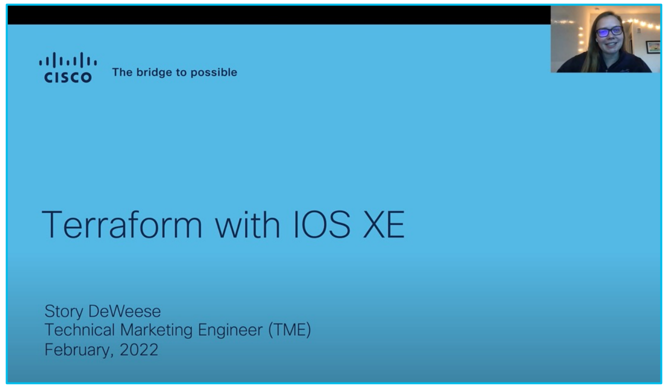
https://salesconnect.cisco.com/#/content-detail/fa072157-b099-494b-8ec5-2522c6ab2bf6



```
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



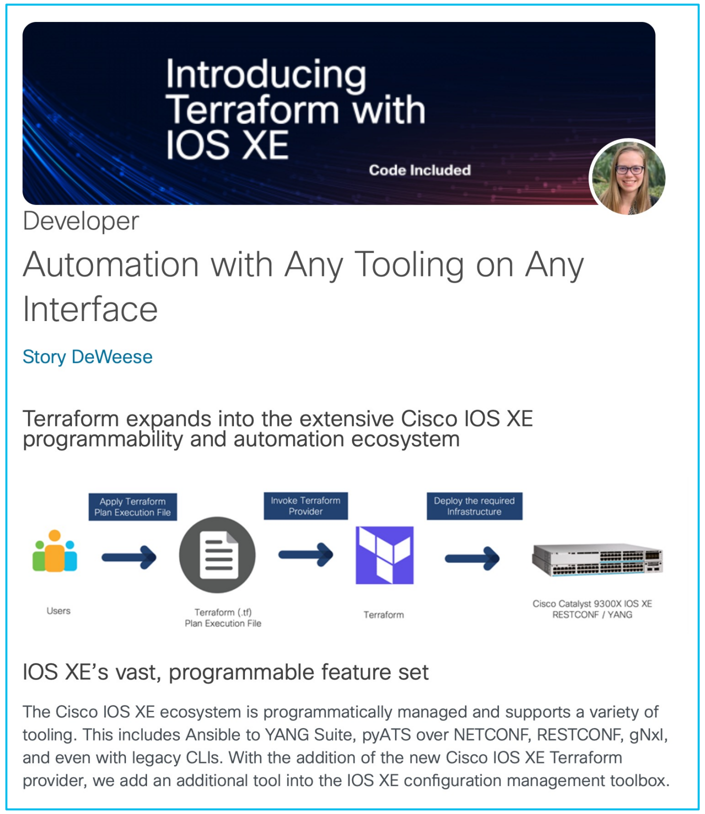
CISCO The bridge to possible

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

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



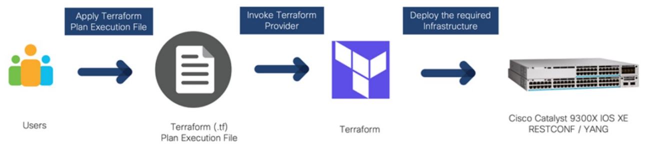
Introducing Terraform with IOS XE
Code Included

Developer

Automation with Any Tooling on Any Interface

Story DeWeese

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



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

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 cisco

The bridge to possible