

The background features a vibrant, abstract design with a color gradient from dark blue on the left to bright yellow and white on the right. The design consists of overlapping, wavy horizontal bands and a radial pattern of lines emanating from a bright white point on the right side, creating a sense of motion and energy.

CISCO *Live!*

Let's go



The bridge to possible

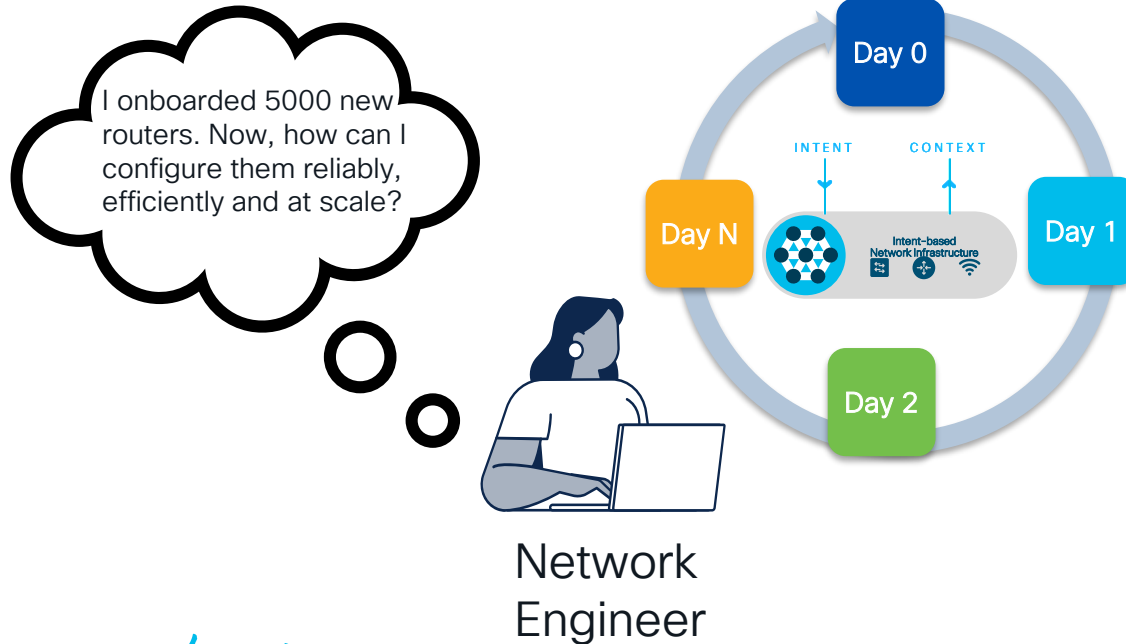
Automate Cisco IOS XE Device Configuration Using Terraform

Story DeWeese, Technical Marketing
@StoryDeWeese

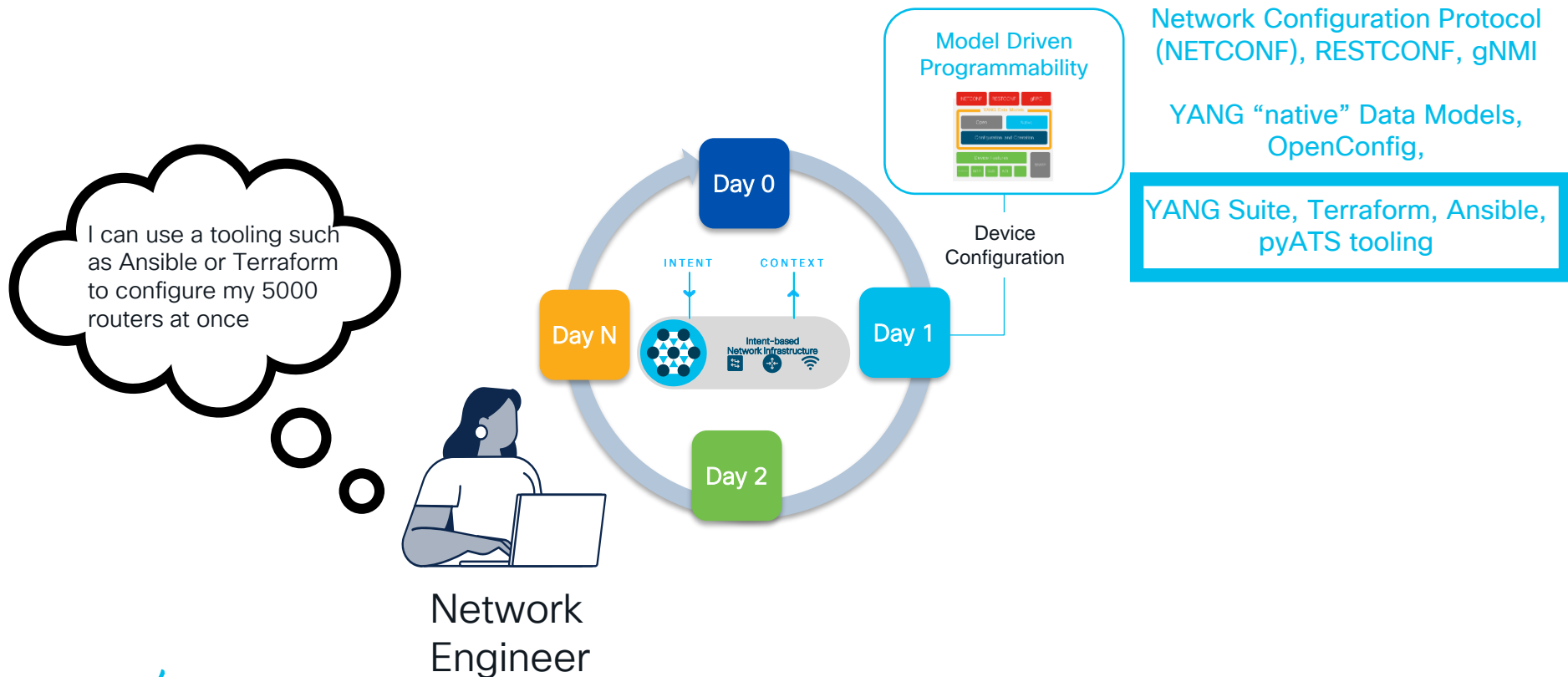
Agenda

- Terraform
- Use Cases & Demos
- Resources

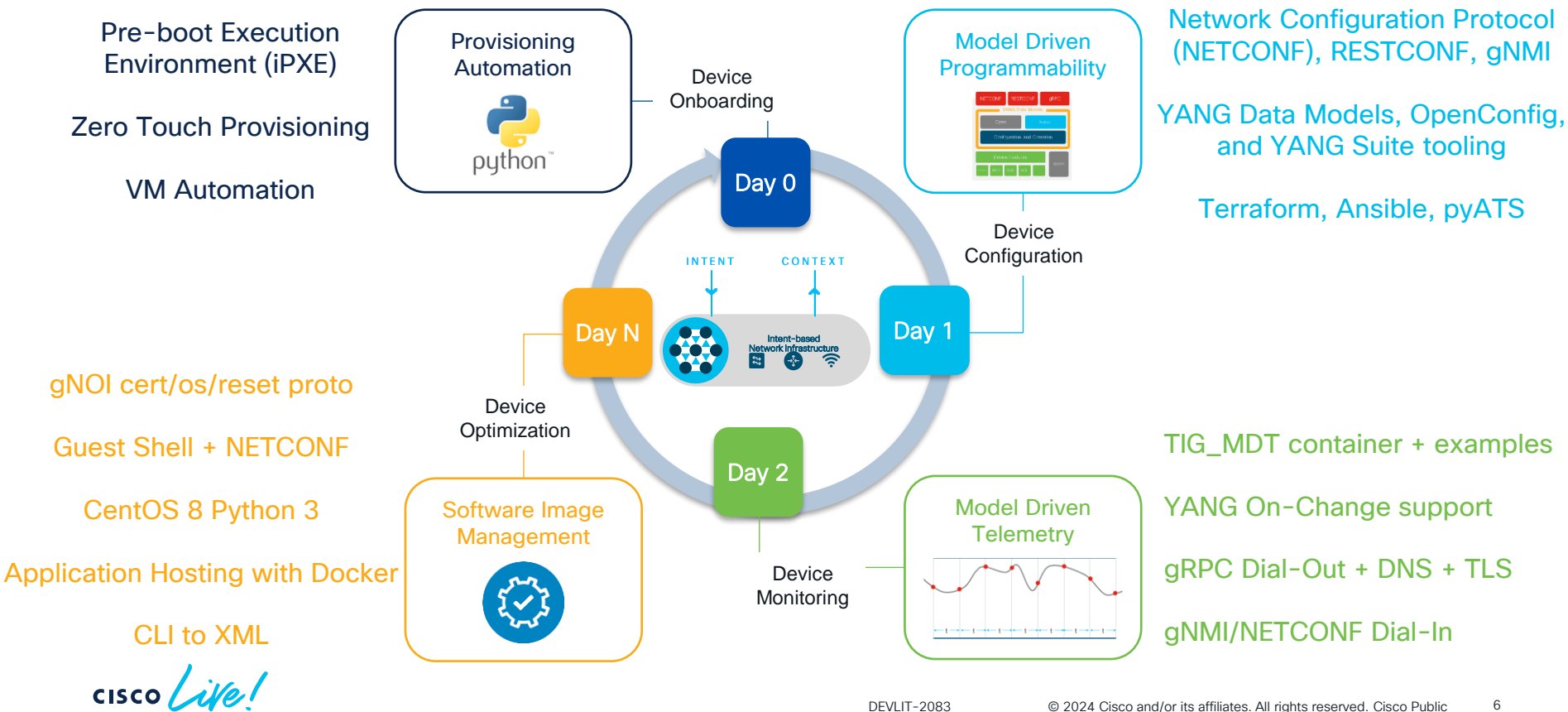
Cisco IOS XE Programmability & Automation Lifecycle



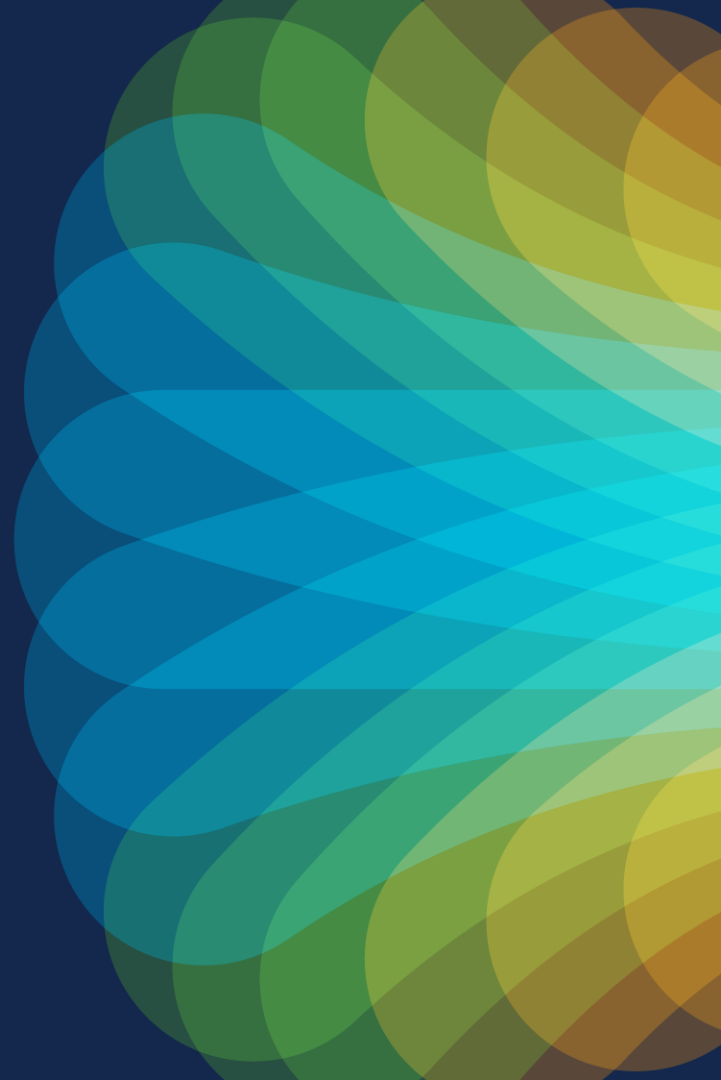
Cisco IOS XE Programmability & Automation Lifecycle



IOS XE Programmability & Automation Lifecycle



Terraform



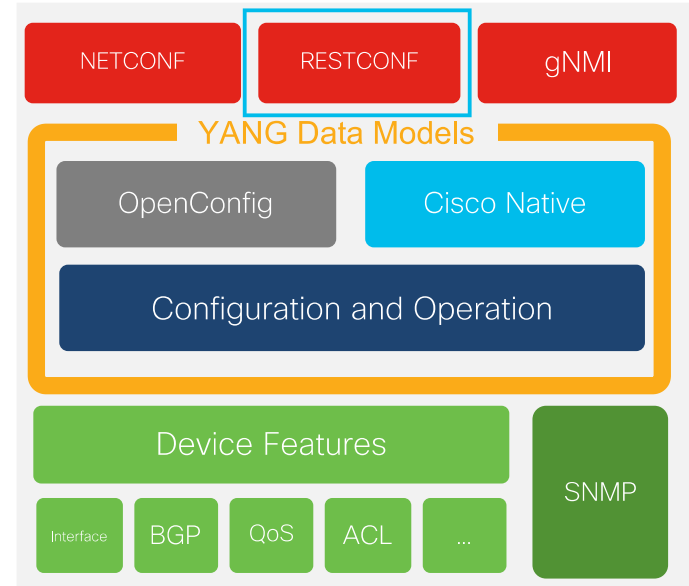
Terraform is...



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.

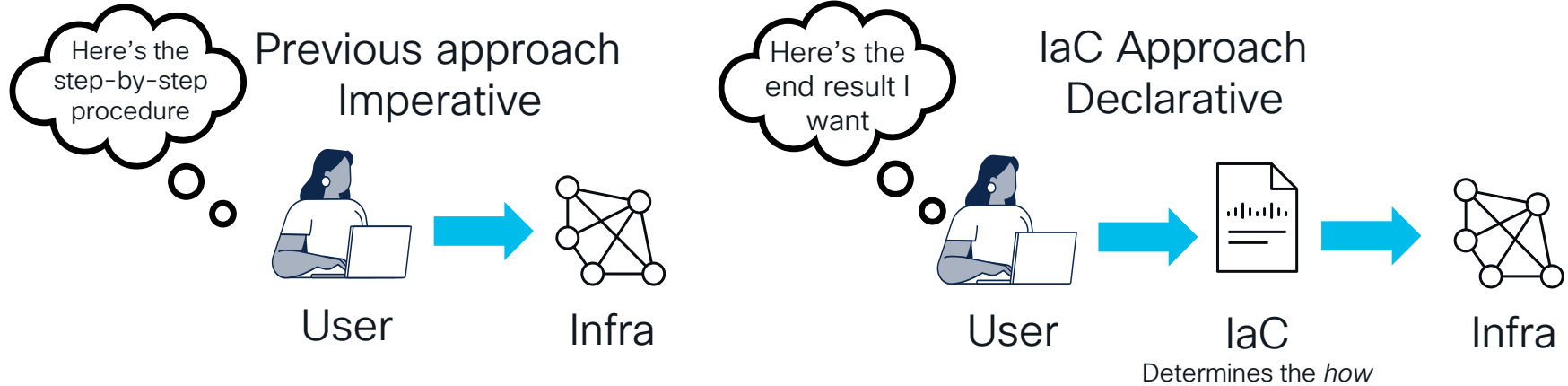
- Cloud Native Tooling circa 2014 from HashiCorp
- Agentless, single binary file
- Zero server-side dependencies

Terraform uses the RESTCONF API



What's IaC?

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



Learn more about IaC here:

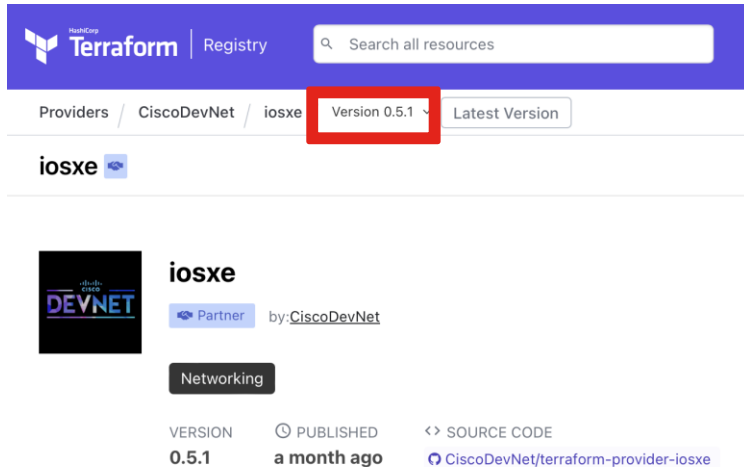
<https://developer.cisco.com/iac/#:~:text=Adopting%20Infrastructure%20as%20Code%20allows,data%20center%20to%20the%20edge>

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/>



HashiCorp Terraform Registry

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](https://github.com/CiscoDevNet/terraform-provider-iosxe/)

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 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  
}
```

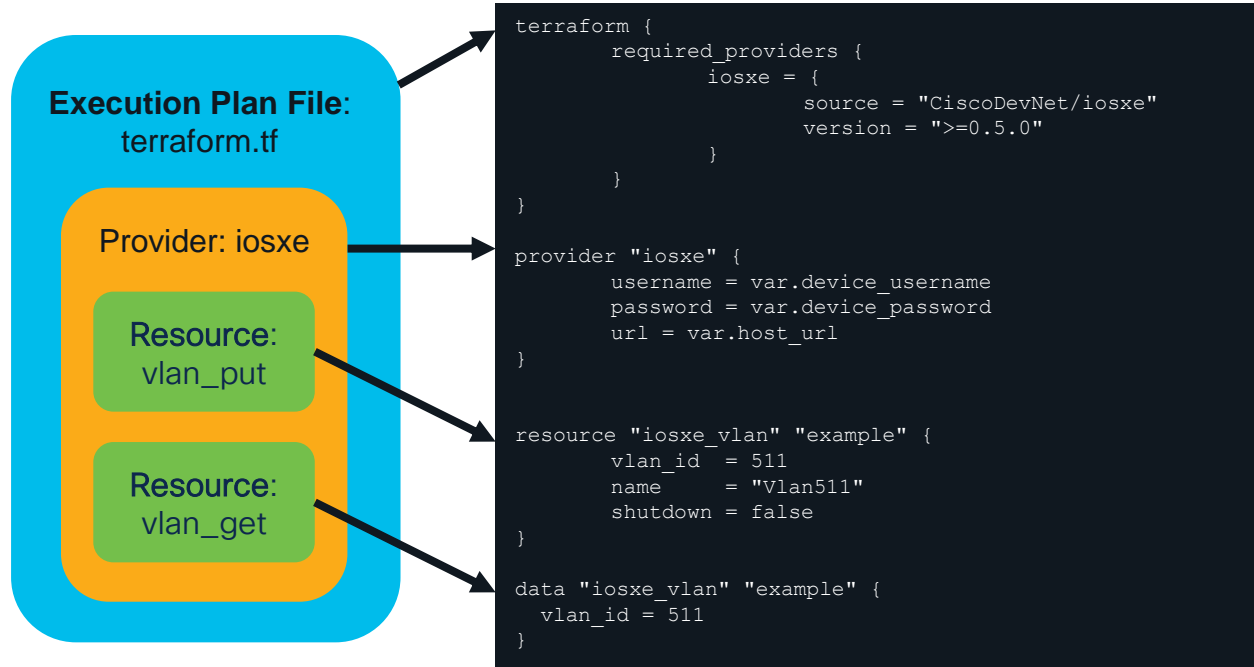
Terraform Terminology

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



CLI to YANG

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

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

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

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

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

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

Use Cases & Demos

Terraform for Model Driven Telemetry

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

```
aut@pod22-xelab:~/mdt_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

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>

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
```

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

5. Navigate to the correct folder

1. `cd`
2. `cd cisco-ios-xe-panda-lab-terraform`

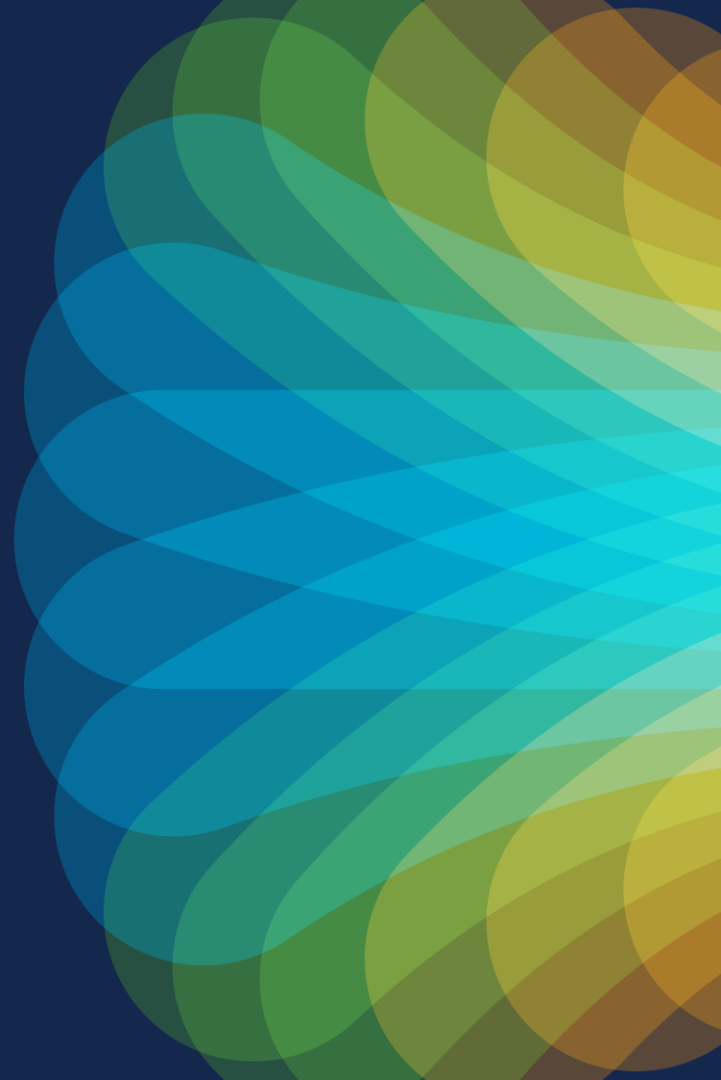
6. Modify the device credentials in the header.tf file

```
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 IP
}
```

```
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
```

7. Configure the Cisco IOS XE device using Terraform

Resources



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

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

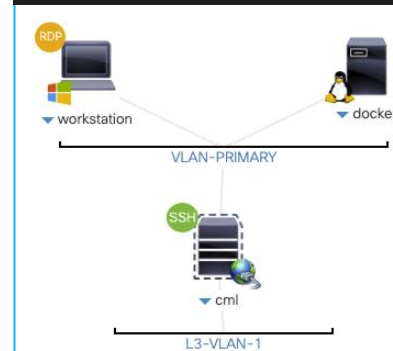
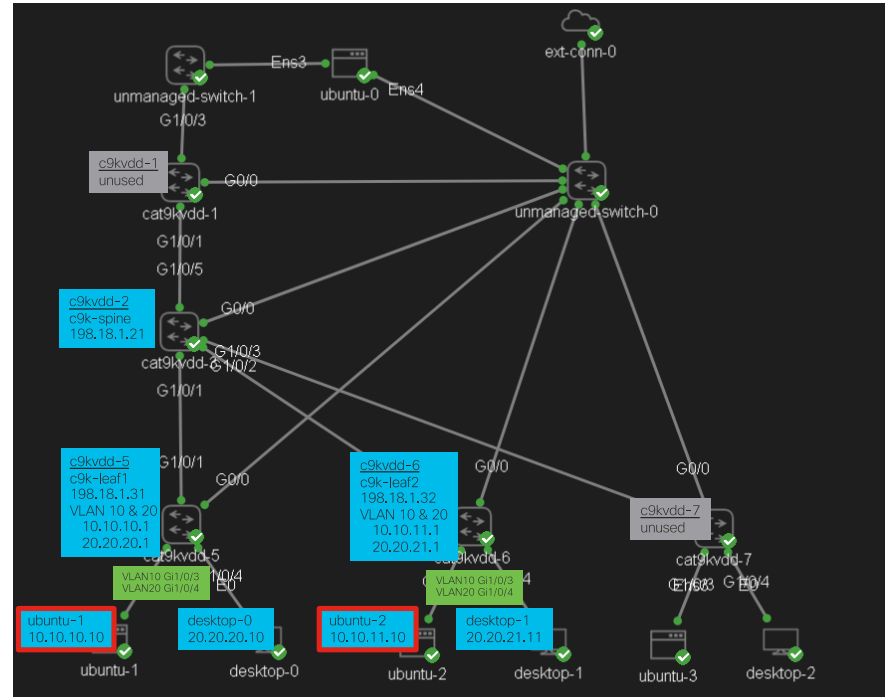
Bash/PS/Cmd shells
SSH into C9K or Ubuntu

Postman
Workspace for EVPN

3x C9K Virtual VM's

Ansible

- EVPN solution enablement using CLI



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

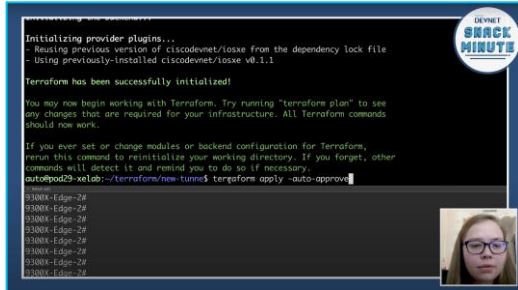
c9kvdd-1 - unconfigured

c9kvdd-7 - unconfigured

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>



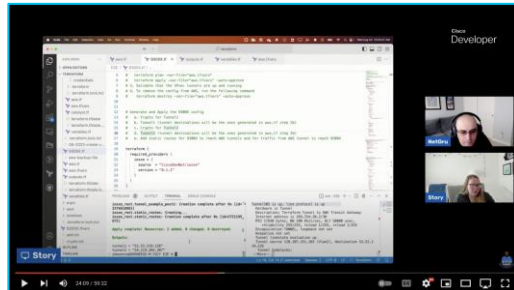
Demo Create a Crypto Tunnel Video:

<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>

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

Users → Apply Terraform Plan Execution File → Invoke Terraform Provider → Deploy the required Infrastructure → Cisco Catalyst 9300K 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, gNMI, and even with legacy CLIs. With the addition of the new Cisco IOS XE Terraform provider, we add an additional tool into the IOS XE configuration management toolbox.

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

Cisco 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)

Session Levels:
Beginner
Intermediate

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 **DEWWKS-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

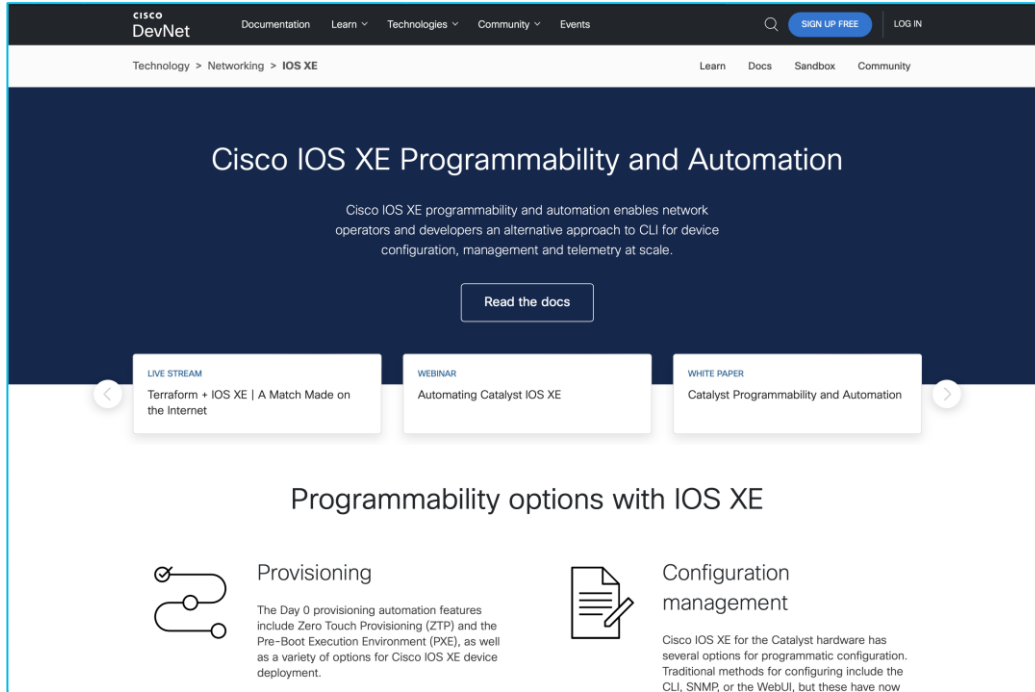


Learn more about sessions this week and on-demand:

<https://blogs.cisco.com/developer/osxeciscoliveemea2024>

Cisco DevNet Site

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



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

White Paper – API

Programmability and auto... ^ Q

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 ^

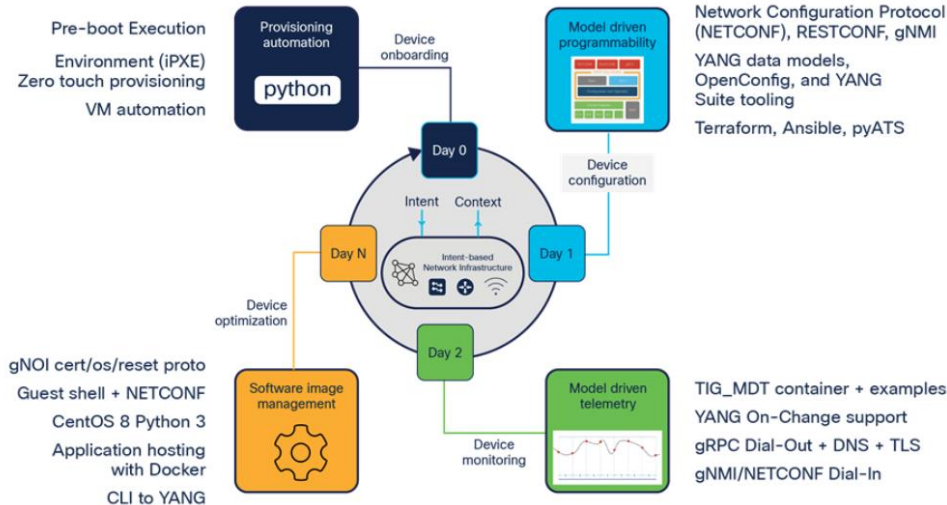
Conclusion

Additional resources ^

Blogs

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

Catalyst Programmability and Automation



<http://cs.co/apiwp>



<http://cs.co/apiwppdf>

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>

MDT White Paper coming soon!

Cisco IOS XE Programmability – Booksprint Book

<http://cs.co/programmabilitybook> OR <https://www.cisco.com/c/dam/en/us/products/collateral/enterprise-networks/nb-06-ios-xe-prog-ebook-cte-en.pdf>

Cisco IOS XE Programmability Automating Device Lifecycle Management

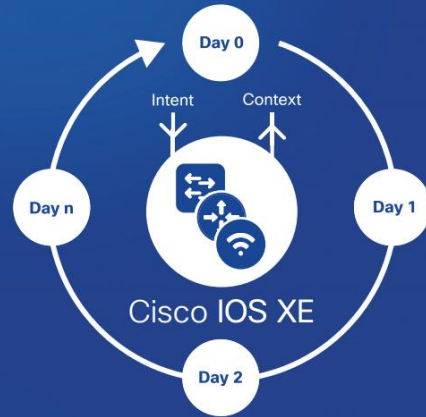


Table of Contents

Authors	Telemetry
Acknowledgments	Overview
About this Book	Operational Data
Introduction	Flow Data
Why Programmability Matters	Use Cases
Lifecycle of Network Device Operations	Subscription Tools
Use Cases	Data Collectors
Operational Approaches	Python
Next Steps	Overview
General Concepts	Python WebUI Sandbox
Cisco IOS XE	On-Box Python
What is Programmability?	Advanced On-Box Python
Application Programming Interfaces (APIs)	Common Issues
Programming Languages	Guest Shell
Structured Data	Introduction
Data Encoding Formats	Security
Day 0 Device Onboarding	Configuration and Updates
Introduction	Resource Allocation
Zero-Touch Provisioning (ZTP) Scenarios	Use Cases
Basic ZTP Workflows	Next Steps
Advanced ZTP Workflows	Application Hosting
Considerations	Introduction
Next Steps	Cisco Application-Hosting Framework
YANG	Containers and Virtual Machines
Overview	Use Case
YANG Concepts	Next Steps
YANG Native vs Open Data Models	Controllers
YANG Data Model Highlights	Introduction
YANG Tools	Common Controllers
Network Device APIs	Why Use a Controller?
Overview	DevOps and NetDevOps
NETCONF	Introduction
RESTCONF	Continuous Integration and Delivery
Comparison of NETCONF and	DevOps Tools
RESTCONF	Next Steps
Next Steps	Appendices
	Additional Resources
	Acronyms

Programmability Configuration Guide

Book Table of Contents

- Preface
- New and Changed Information
- ✓ Provisioning
 - Zero-Touch Provisioning
 - IPXE
- ✓ Shells and Scripting
 - Guest Shell
 - Python API
 - EEM Python Module
- ✓ Model-Driven Programmability
 - NETCONF Protocol
 - RESTCONF Protocol
 - NETCONF and RESTCONF Service-Level ACLs
 - gNMI Protocol
 - gRPC Network Operations Interface
 - Model Based AAA
 - Model-Driven Telemetry
 - In-Service Model Update
- ✓ Application Hosting
 - Application Hosting
- ✓ OpenFlow
 - OpenFlow
 - High Availability in OpenFlow Mode



https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/1713/b_1713_programmability_cg.html



The bridge to possible

Thank you

CISCO *Live!*

The background of the slide is a vibrant, abstract graphic. It features a series of overlapping, wavy bands of color in shades of red, orange, yellow, green, and blue, creating a sense of movement and energy. On the right side, there is a bright, multi-colored sunburst or starburst effect, with rays of light radiating outwards in various colors including yellow, orange, red, and blue. The overall composition is dynamic and visually appealing.

cisco *Live!*

Let's go