

CISCO *Live!*

GO BEYOND

#CiscoLiveAPJC



Programmability, Automation Model Driven Telemetry on Cisco IOS XE with a dash of YANG Suite

Story DeWeese, Technical Marketing
@StoryDeWeese
DEVNET-1283

Cisco Webex App

Questions?

Use Cisco Webex App to chat with the speaker after the session

How

- 1 Find this session in the Cisco Live Mobile App
- 2 Click “Join the Discussion”
- 3 Install the Webex App or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until November 15, 2024.

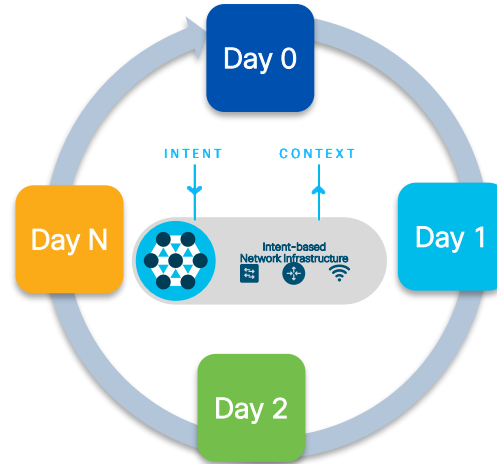


<https://ciscolive.ciscoevents.com/ciscolivebot/#DEVNET-1283>



Agenda

Intro to Programmability and Automation



Tooling

Resources

Why do we use programmability?

Programmability solutions for complex networks



Efficient

Managing infrastructure as code while lowering OPEX



Consistent

Repeatable and precise intent-based outcomes



Scalable

Reduce time and effort with large network automation requirements



Secure

Secure APIs and integrations



Seamless

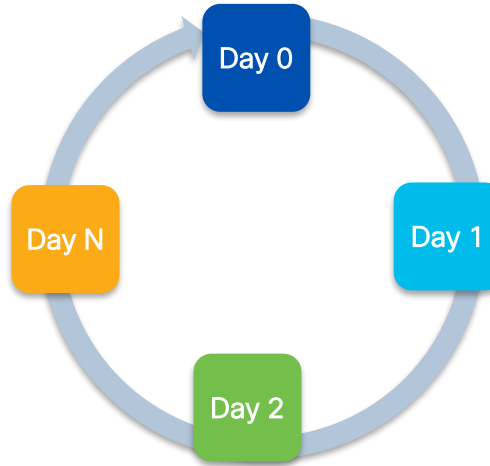
Create value integrating with 3rd party platforms



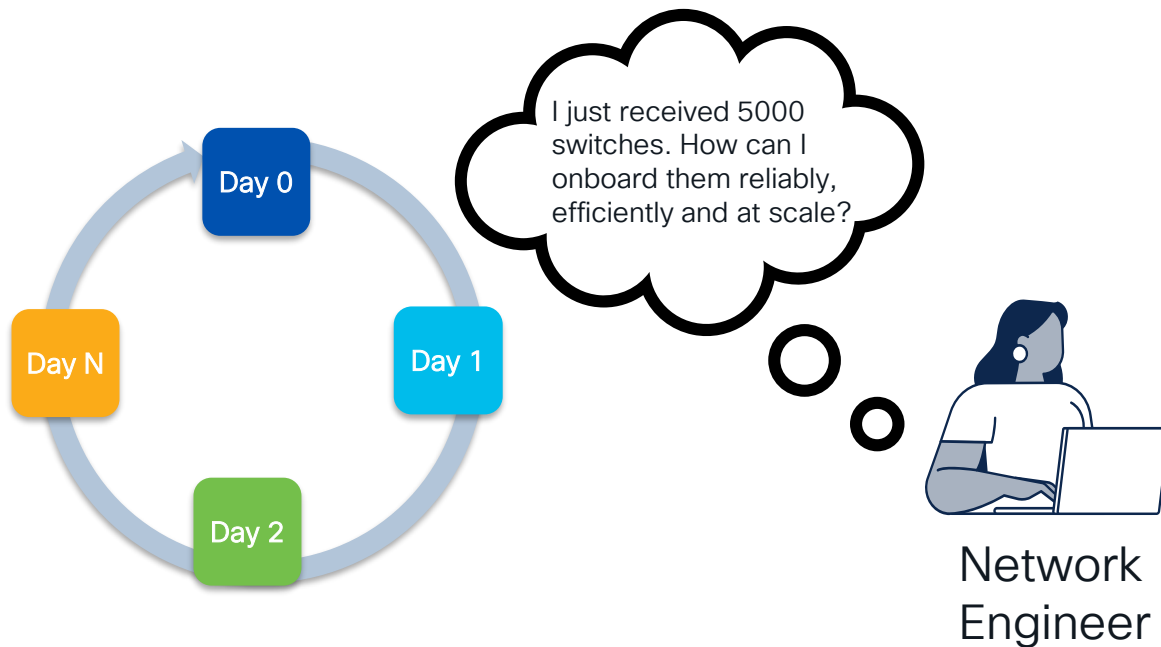
Smart

Real time event notifications, AI/ML capabilities, and streaming telemetry

Cisco IOS XE Programmability & Automation Lifecycle



Cisco IOS XE Programmability & Automation Lifecycle



Cisco IOS XE Programmability & Automation Lifecycle

Pre-boot Execution
Environment (iPXE)

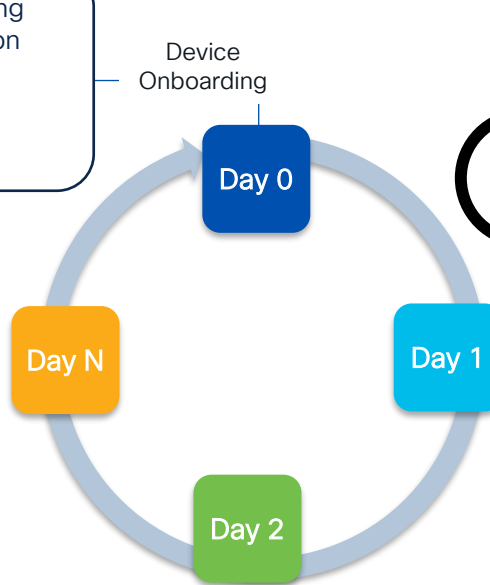
RFC8572 Secure Zero
Touch Provisioning ZTP

VM Automation

Provisioning
Automation



Device
Onboarding

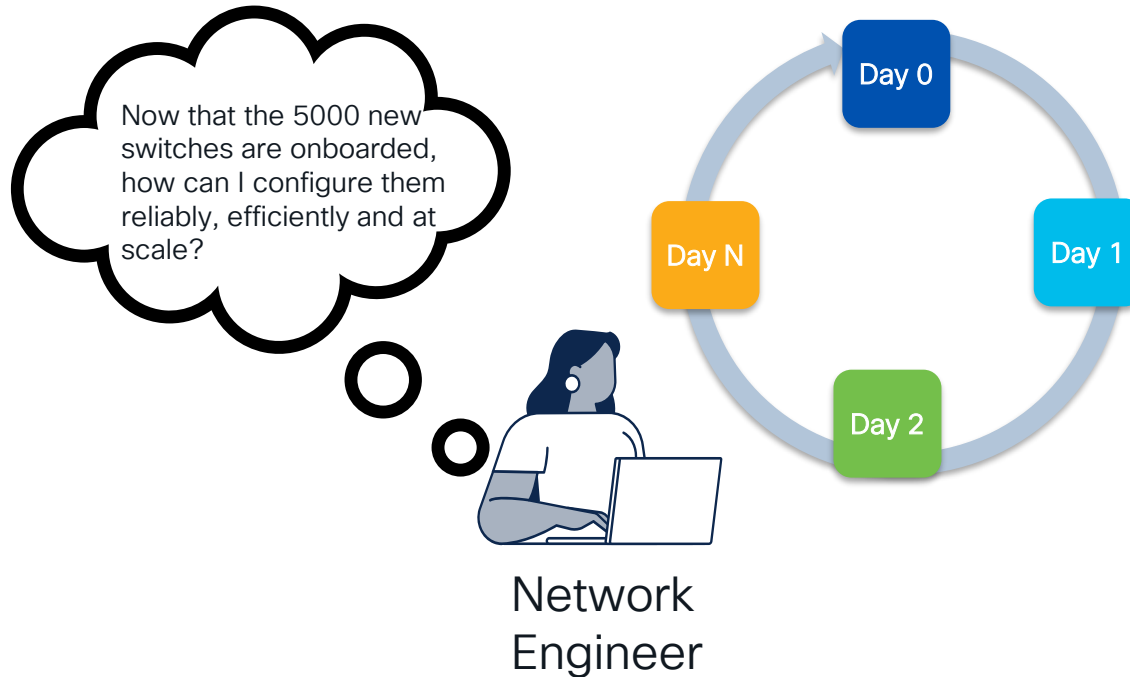


Nice! All 5000 switches are
onboarded using Secure
Zero Touch Provisioning

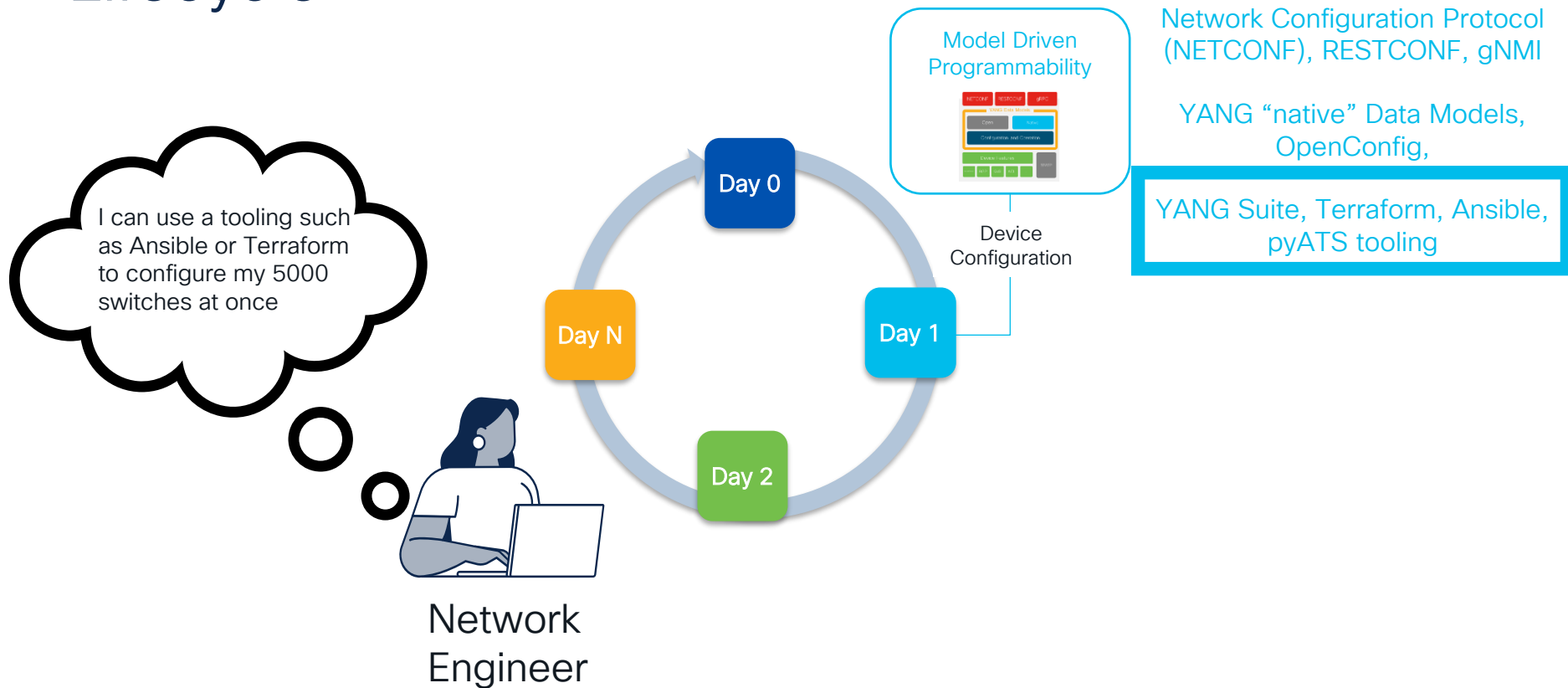


Network
Engineer

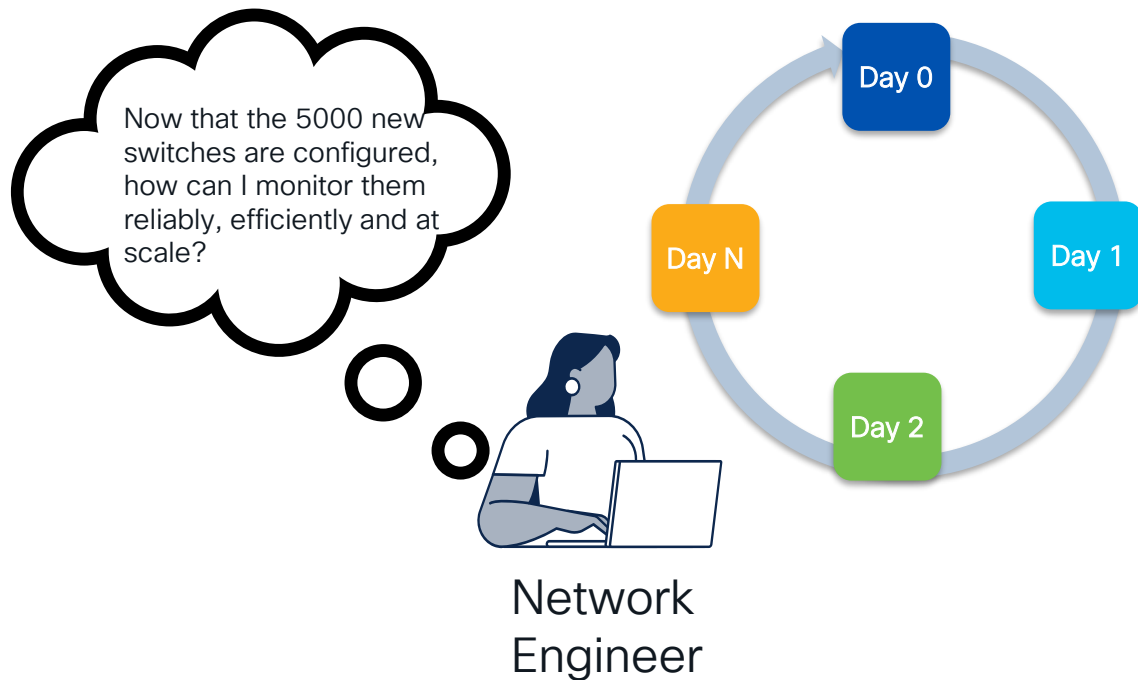
Cisco IOS XE Programmability & Automation Lifecycle



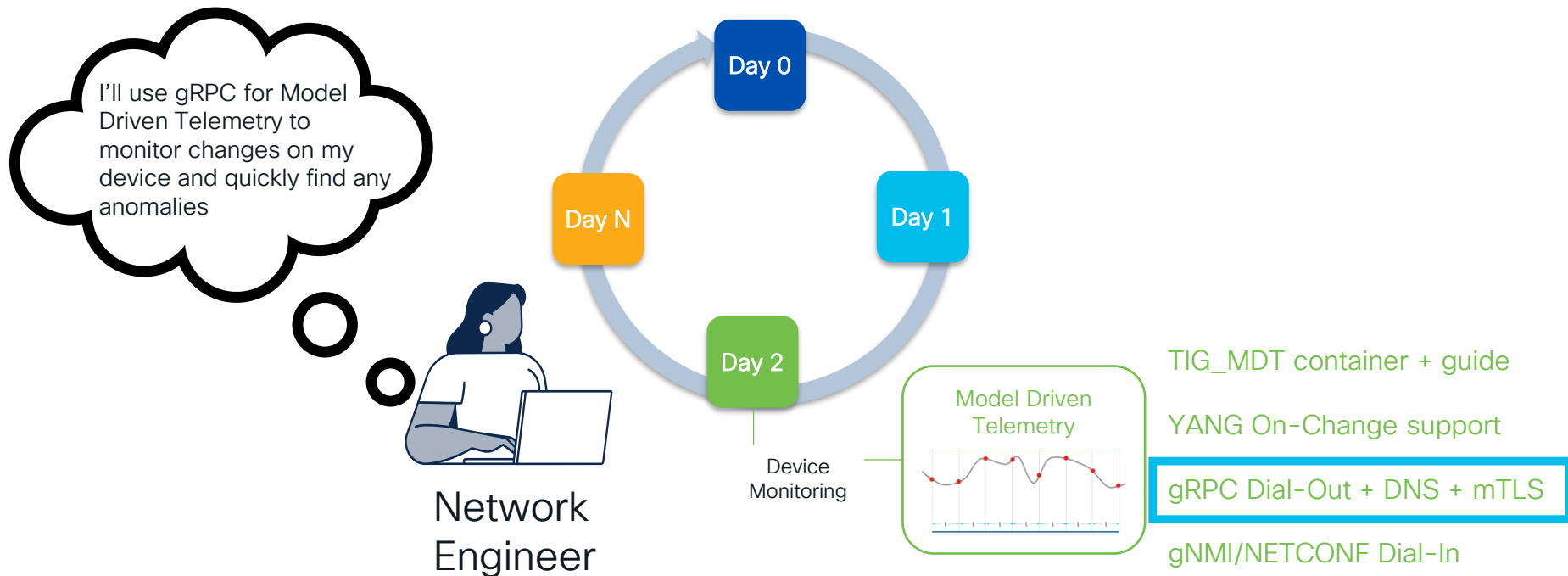
Cisco IOS XE Programmability & Automation Lifecycle



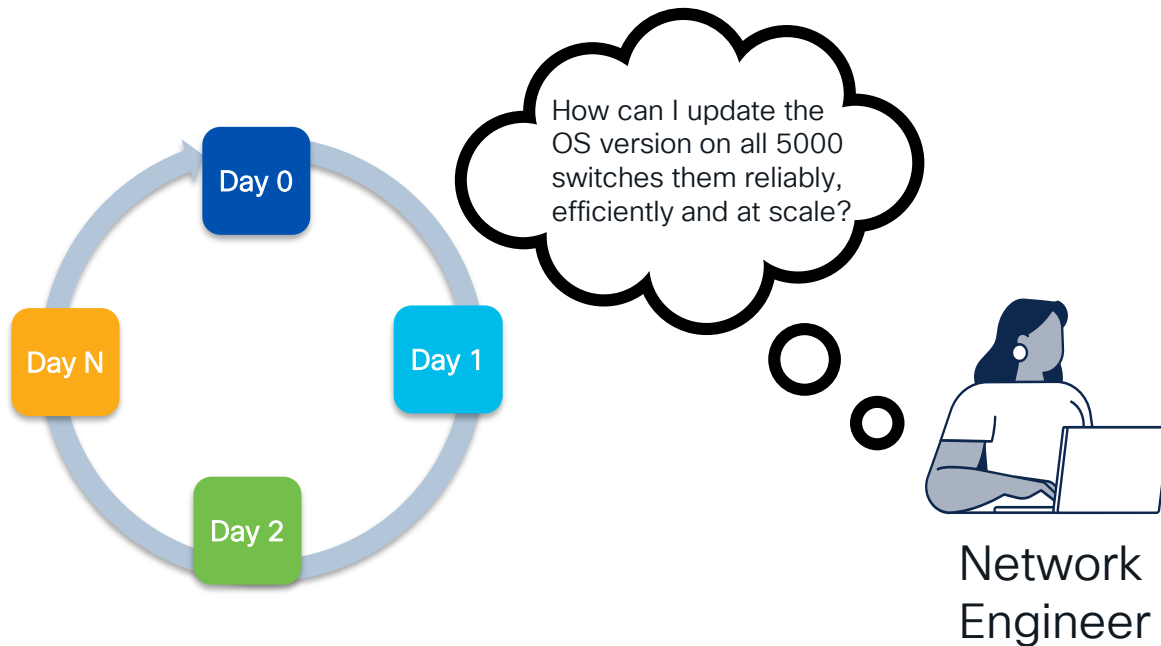
Cisco IOS XE Programmability & Automation Lifecycle



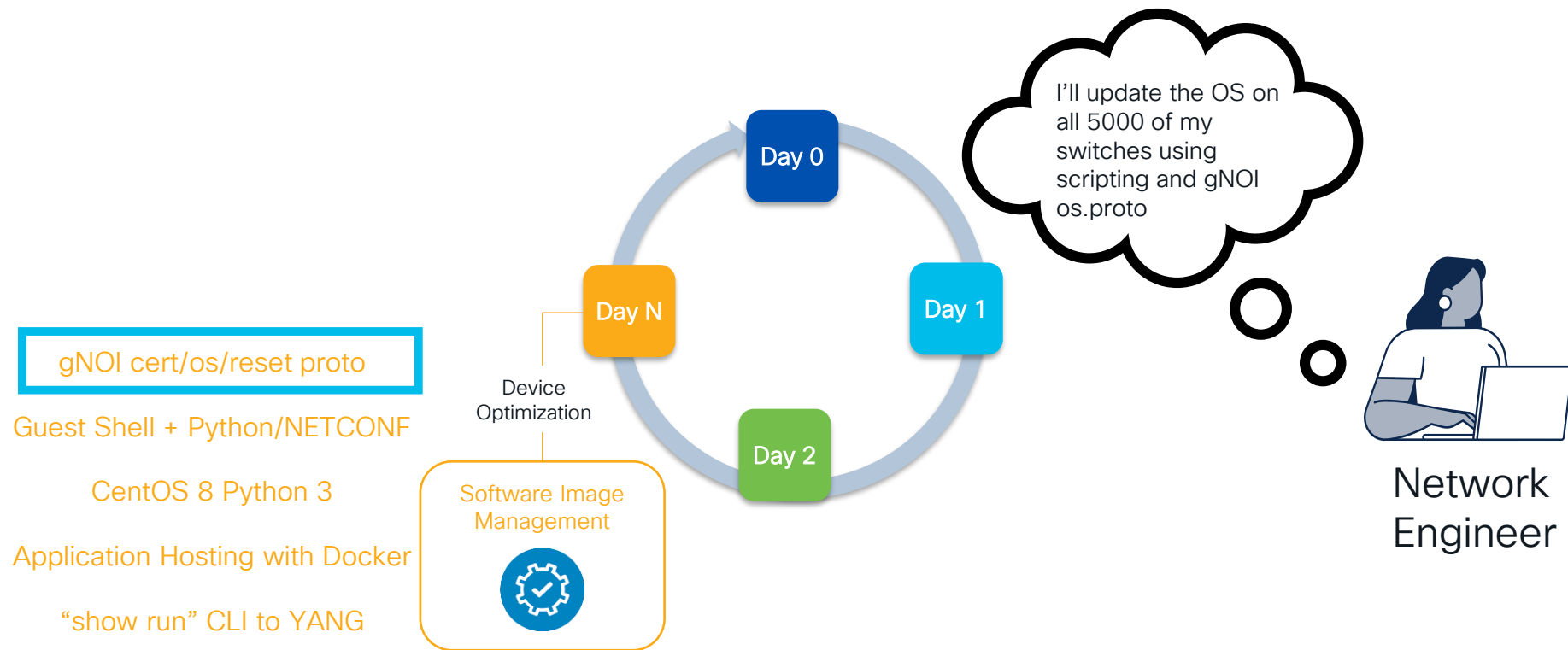
Cisco IOS XE Programmability & Automation Lifecycle



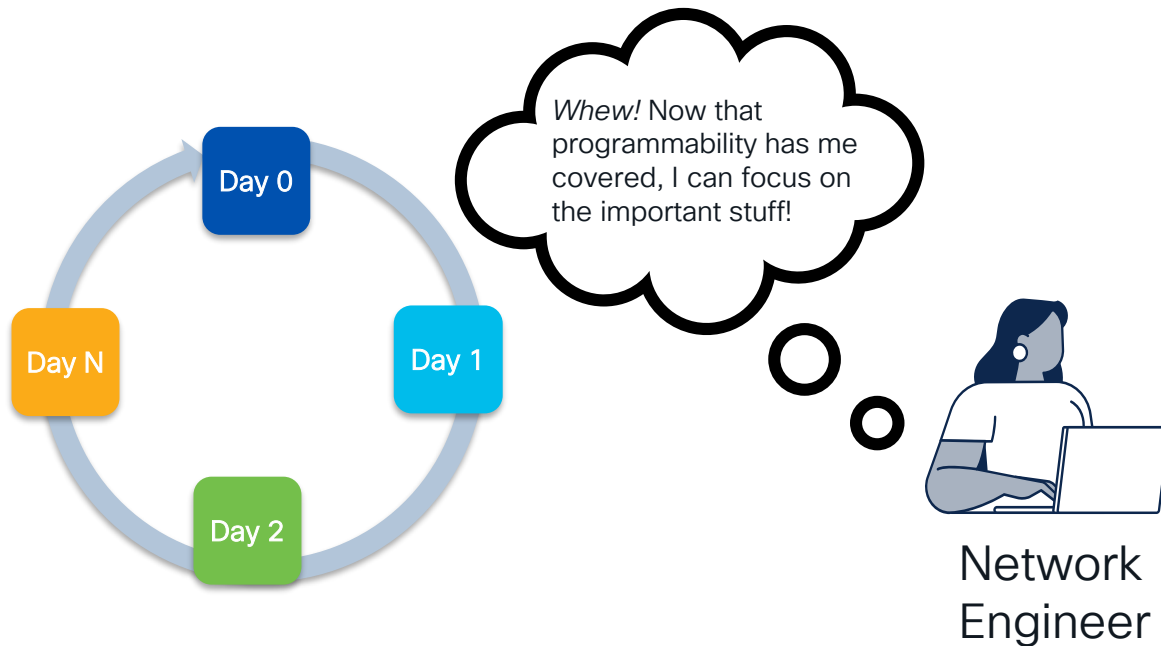
Cisco IOS XE Programmability & Automation Lifecycle



Cisco IOS XE Programmability & Automation Lifecycle



Cisco IOS XE Programmability & Automation Lifecycle



Cisco IOS XE Programmability & Automation Lifecycle

Pre-boot Execution Environment (iPXE)

RFC8572 Secure Zero Touch Provisioning ZTP

VM Automation

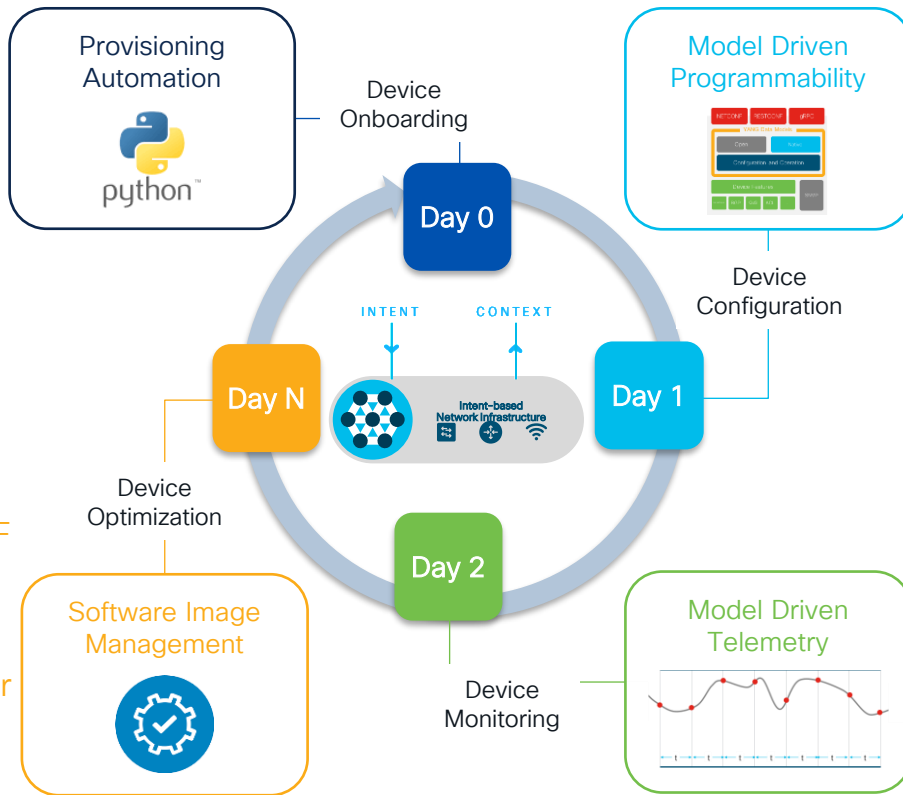
gNOI cert/os/reset proto

Guest Shell + Python/NETCONF

CentOS 8 Python 3

Application Hosting with Docker

“show run” CLI to YANG



Network Configuration Protocol (NETCONF), RESTCONF, gNMI

YANG “native” Data Models, OpenConfig,

YANG Suite, Terraform, Ansible, pyATS tooling

TIG_MDT container + guide

YANG On-Change support

gRPC Dial-Out + DNS + mTLS

gNMI/NETCONF Dial-In

IOS XE programmability and Telemetry “Stack”

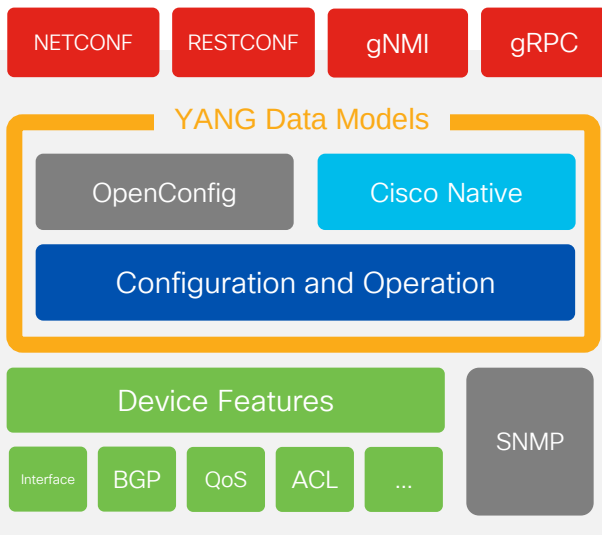
CLI

SNMP

WebUI

The NETCONF, RESTCONF, 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



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

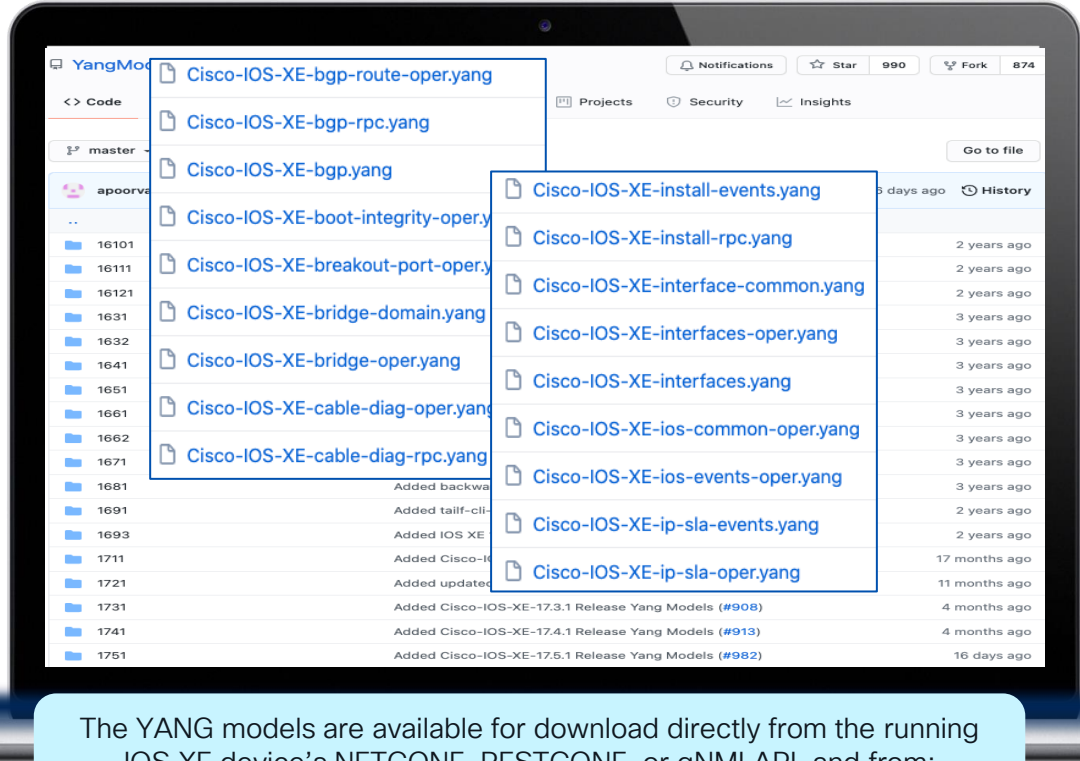
	NETCONF	RESTCONF	gNMI
Minimum IOS XE Version	16.6	16.7	16.8
Recommended Version	17.6	17.6	17.7
Default Port	830	443	9339
Operations	<get>,<get-config>,<edit-config>,<establish-subscription>	GET, POST, PUT, PATCH, DELETE	GET, SET, SUBSCRIBE
Encoding	XML	XML or JSON	RFC7951 JSON_IETF
Security	SSH + PKI 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



Cisco IOS XE – YANG models on GitHub

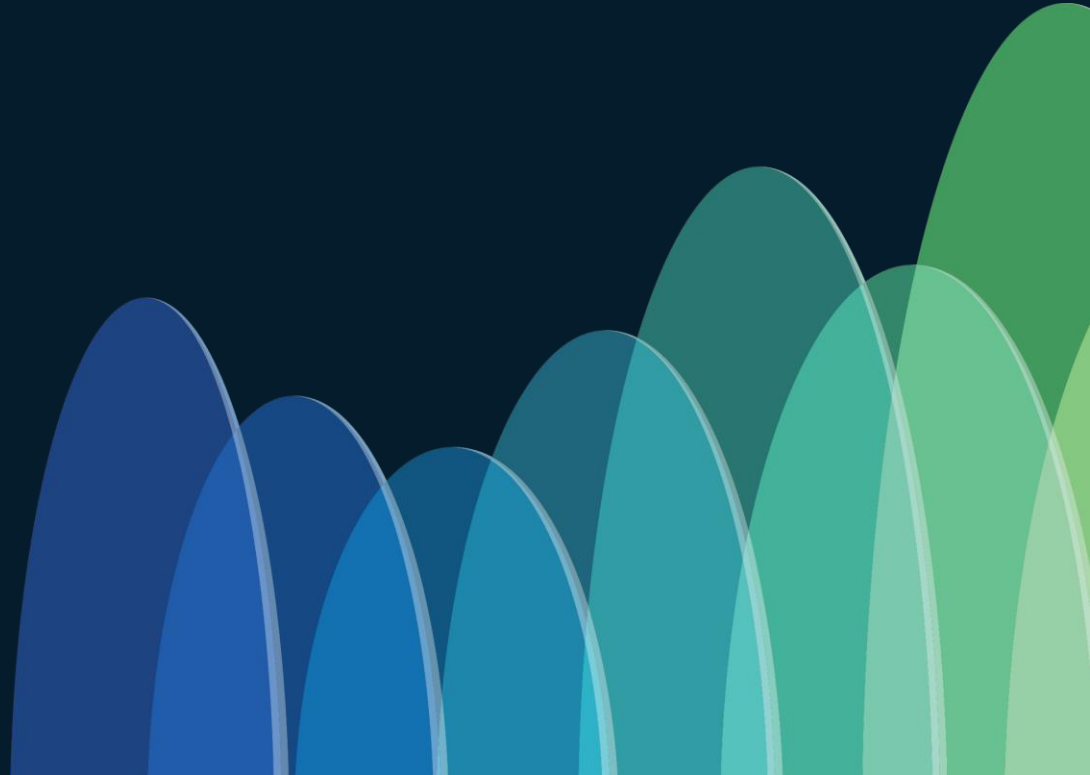
- RFC 7950 YANG data modelling language are the API definitions for IOS XE
- The YANG modules are available for download from the API and are also published on Github.com
- Notable modules are listed below for the running-config, feature oper, actions and event notifications

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-IOS-XE-{feature}-events	Telemetry Events
Cisco-evpn-service	EVPN service abstraction
OpenConfig-{feature}	abstraction for config & oper



The YANG models are available for download directly from the running IOS XE device's NETCONF, RESTCONF, or gNMI API, and from:
<https://github.com/YangModels/yang/tree/main/vendor/cisco/xe>

Day 0

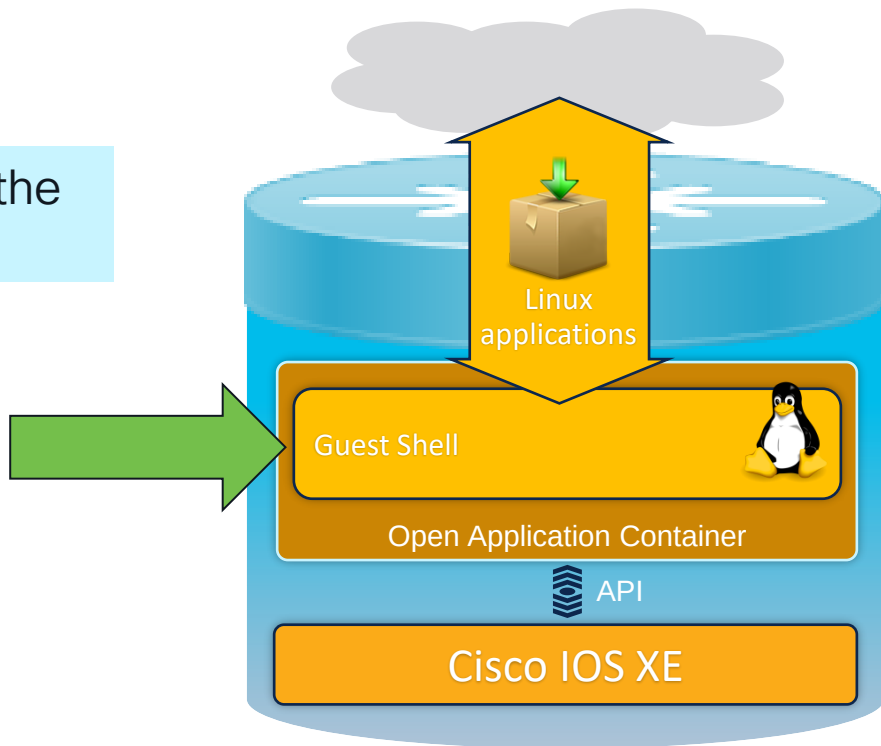


Linux Shell Environment on Cisco IOS XE devices

Cisco IOS XE devices can access the on-box Guest Shell using API calls

This Linux shell can be used for

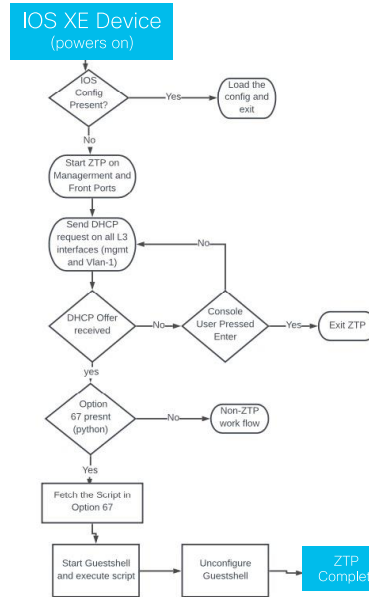
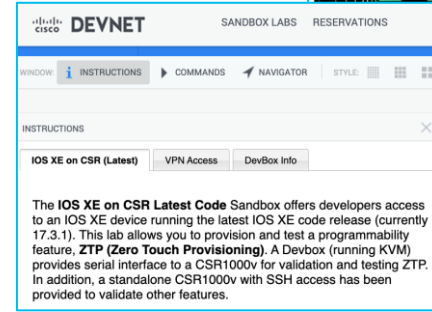
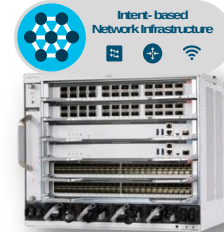
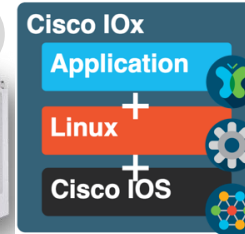
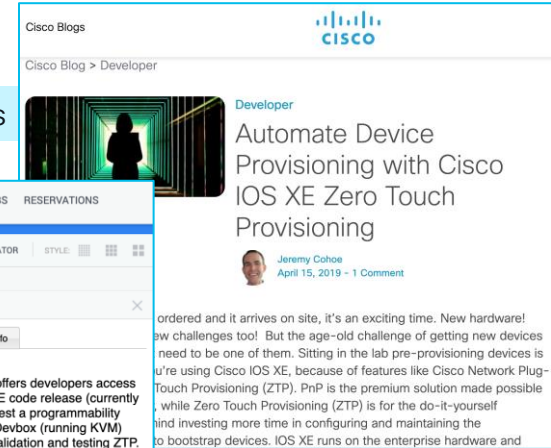
- Scripting
- Hosting Applications
- Zero Touch Provisioning
- & so much more!



Day 0 - Classic ZTP Overview

The Day 0 ZTP features are used to automatically configure and provision network devices

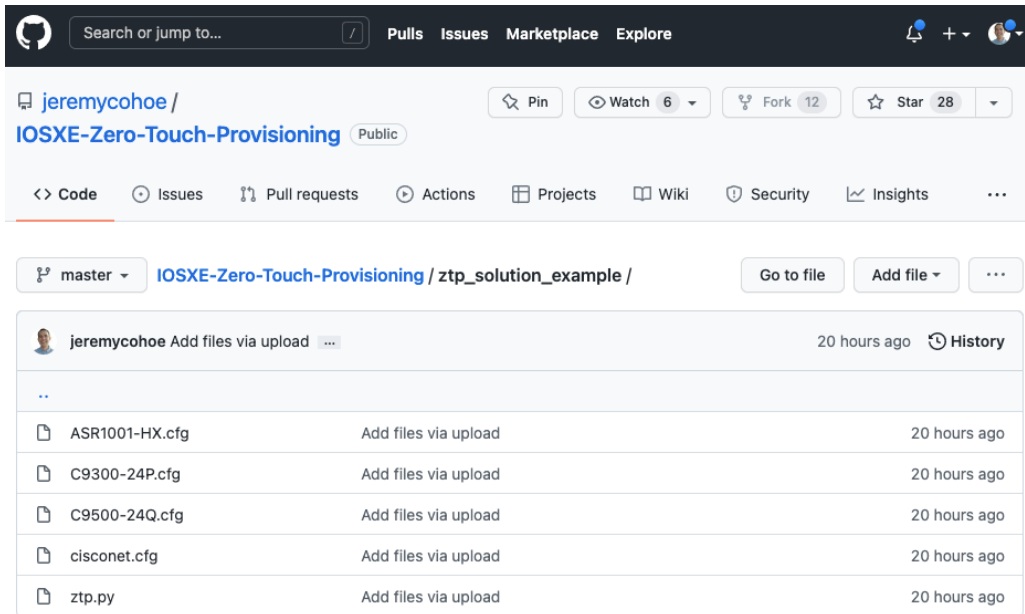
1. When an IOS XE device boots and no configuration is present, the device will issue a DHCP request on the management port and on the front panel port.
2. If the DHCP response contains option 67 then ZTP is initiated and the device will retrieve and execute the python script from within the Guest Shell
3. Guest Shell is started and networking is automatically configured



<https://www.youtube.com/watch?v=EAXnftG6odg>
<https://blogs.cisco.com/developer/device-provisioning-with-ios-xe-zero-touch-provisioning>
<https://devnetsandbox.cisco.com/DevNet>
Secure ZTP blog <https://blogs.cisco.com/developer/secureztp01>

ZTP customer use case – code samples

https://github.com/jeremycohoe/IOSXE-Zero-Touch-Provisioning/tree/master/ztp_solution_example



The screenshot shows the GitHub repository page for 'jeremycohoe/IOSXE-Zero-Touch-Provisioning'. The repository is public and has 28 stars, 12 forks, and 6 watchers. The 'ztp_solution_example' directory is selected, showing a list of files: ASR1001-HX.cfg, C9300-24P.cfg, C9500-24Q.cfg, ciscoet.cfg, and ztp.py. Each file has an 'Add files via upload' button and a timestamp of '20 hours ago'.



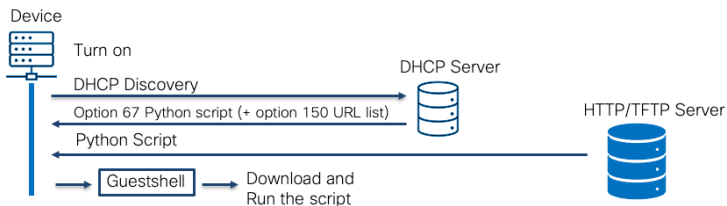
```
1
2 # Importing cli module
3 from cli import configure, cli, configurep, executep
4 import re
5 import time
6 import urllib
7 import sys
8 import logging
9 import os
10 from logging.handlers import RotatingFileHandler
11 import subprocess
12
13 software_mappings = {
14     'C9300-24P': {
15         'software_image': 'cat9k_iosxe.17.06.01.SPA.bin',
16         'software_version': '17.06.01',
17         'software_md5_checksum': 'fdb9c92bae37f9138d0ee6761afe2919'
18     },
19     'C9500-24Q': {
20         'software_image': 'cat9k_iosxe.17.06.01.SPA.bin',
21         'software_version': '17.06.01',
22         'software_md5_checksum': 'fdb9c92bae37f9138d0ee6761afe2919'
23     },
24     'ASR1001-HX': {
25         'software_image': 'asr1000-universalk9.17.05.01a.SPA.bin',
26         'software_version': '17.05.01a',
27         'software_md5_checksum': '0e4b1fc1448f8ee289634a41f75dc215'
28     }
29 }
30
31 http_server = '10.85.134.66'
32 log_to_file = True
33 release_set = ['17.05', '17.06']
34
35 def main():
36
37     try:
38         print ('##### STARTING ZTP SCRIPT #####\n')
39         # switch to enable/disable persistent logger
40         if(log_to_file == True):
41             filepath = create_logfile()
42             configure_logger(filepath)
```

RFC8572 Secure ZTP

RFC details: <https://www.rfc-editor.org/rfc/rfc8572.html>

1. Conveyed Information: used to encode the redirect information and onboarding information (switch config)
2. Ownership Certificate: used by a device to verify the signature over the conveyed information
3. Ownership Voucher: used to verify a device owner as defined by the manufacturer (from the MASA)

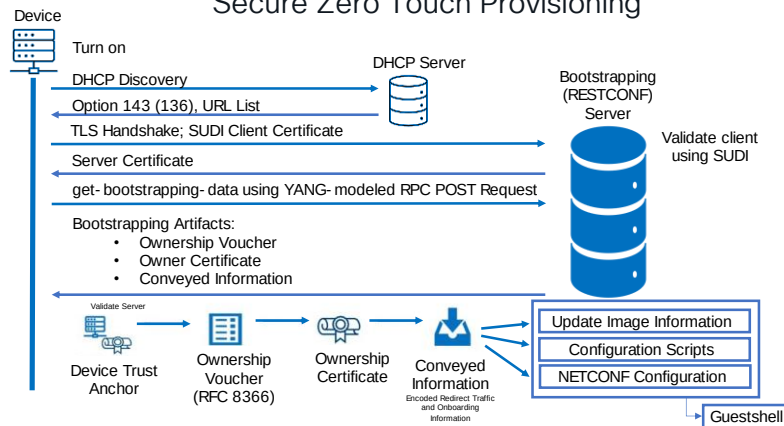
Classic Zero Touch Provisioning



Some security requirements for classic ZTP are resolved using Secure ZTP:

- Management system needs to validate the device
- Device needs to validate the server
- Device must validate the data is what server sent

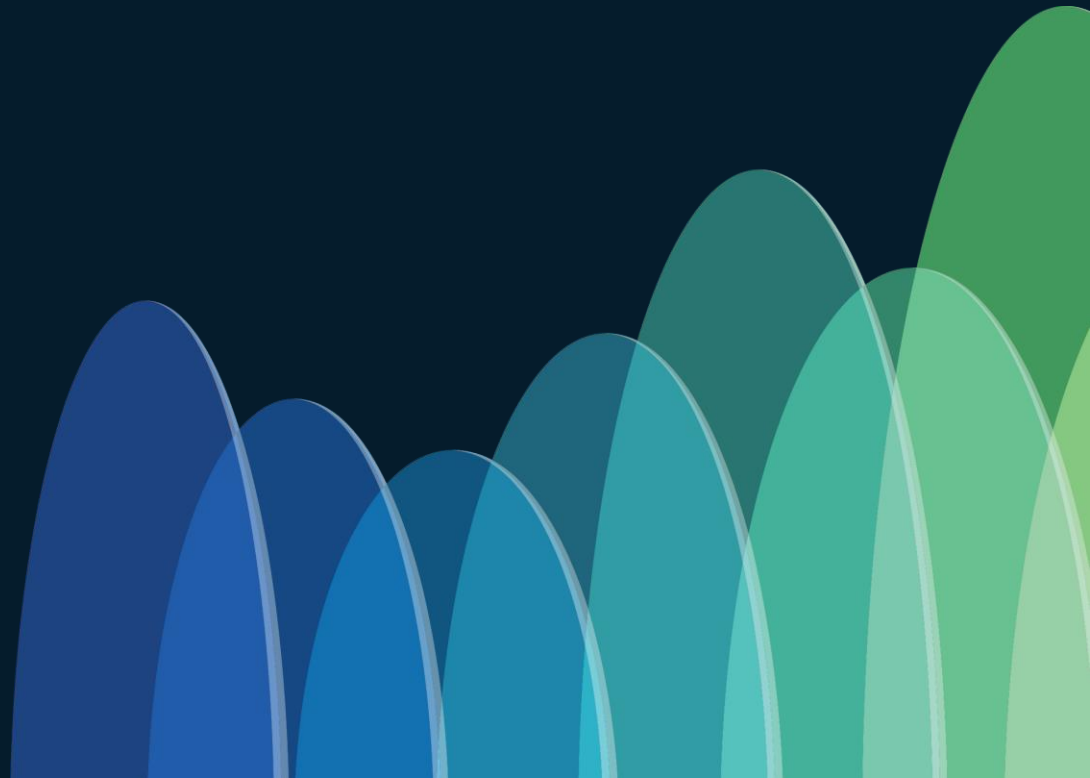
Secure Zero Touch Provisioning



As part of the SZTP RFC, the device supports image upgrade as part of the conveyed information

Day 1

CISCO *Live!*



Cisco YANG Suite



YANG API Testing and Validation Environment

Construct and test YANG based APIs over
NETCONF, RESTCONF, gRPC and gNMI

IOS XE / IOS XR / NX OS platforms

developer.cisco.com/yangsuite

github.com/CiscoDevNet/yangsuite

Get hands-on using the new learning lab!

<https://developer.cisco.com/learning/labs/intro-yangsuite/>

Docker container innovation 1 container



Build NETCONF XML Payload to SET Hostname

The screenshot displays the Cisco YANG Suite interface for configuring a NETCONF XML payload. The left sidebar shows the navigation menu with options: Admin, Setup, Analytics, Explore, Protocols, Test Manager, and Help. The main workspace is titled "YANG Suite / NETCONF / YANG set 'c9300-default-yangset' / Modules".

Key interface elements include:

- YANG Set:** c9300-default-yangset
- Module(s):** Cisco-IOS-XE-native
- NETCONF Operation:** edit-config
- Device:** C9300
- Buttons:** Edit Device, Open Device Window, Run RPC(s), Clear RPC(s), Load Module(s), RPC Options..., Build RPC
- YANG Tree:** A hierarchical tree view of the configuration nodes. The "hostname" node is selected, and its value is "netconf-pro".
- XML Payload:** The right pane shows the generated NETCONF XML payload:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <hostname>netconf-pro</hostname>
      </native>
    </config>
  </edit-config>
</rpc>
```

YANG Suite RESTCONF Demo

Cisco YANG Suite

Admin

Setup

Analytics

Explore

Protocols

Test Manager

Help

YANG Suite / Help

Help: Welcome to Cisco YANG Suite!

admin

Welcome to Cisco YANG Suite!

YANG Suite is a set of tools related to YANG models ([RFC 6020](#), [RFC 7950](#)) and related technologies such as NETCONF ([RFC 6241](#)).

It provides a modular infrastructure which various YANG application plugins can be used.

Please check [developer.cisco.com/yangsuite](#) for information, learning labs, and announcements.

Docker-based Installation

The `yangsuite/docker/start_yang_suite.sh` script performs the following:

- Prompts for username, password, and email which will be the superuser to yangsuite.
- Gives the choice of creating test SSL/TLS certificate and key.
- Creates an environment file needed for the yangsuite docker container install.
- Runs `docker-compose up`.

YANG Suite Documentation

Search docs

- Welcome to Cisco YANG Suite!
- Device Profiles
 - Managing device profiles
 - Setting up YANG Suite TLS/SSL
- File Manager
 - Constructing and populating a YANG module repository
 - Uploading YANG files from the local filesystem to a YANG repository
 - Downloading YANG files via NETCONF from a device to a YANG repository
 - Copying YANG modules via SCP from a server to a YANG repository
 - Importing YANG modules from a Git repository to a YANG repository
 - Defining a YANG module set
 - Managing YANG module files in YANG Suite
- Working with YANG Models
 - Exploring YANG Models
- YANG Suite Analytics
 - Datasets
- Using gNMI with YANG Suite
- YANG Suite gRPC telemetry receivers
 - Telemetry over gRPC Clear Channel
 - Configuring gRPC Telemetry Receivers
 - Telemetry over gRPC Secure Channel
- Using NETCONF with YANG Suite
 - Using NETCONF RPCs
 - Working with NETCONF Notification Streams
 - Using Replays for repeated workflows
 - Locking and unlocking datastores
- RESTCONF in YANG Suite
 - YANG Suite RESTCONF
- Test management with YANG Suite
 - Using Test Manager to define and execute tests
 - Convert Ytool Test Suites to YANG Suite Formats to activate Windows.
 - Generating YANG model Tests
 - Importing and exporting YANG Suite tests

Demo: Generate Python from NETCONF

Access YANG Suite >
Protocols > NETCONF

Select YANG model then
Build RPC payload

Select Replays

Select Generate Python
script

The screenshot displays the Cisco YANG Suite interface. On the left, a sidebar shows the navigation menu with 'Admin', 'Setup', 'Explore', and 'Protocols'. The 'Protocols' section is expanded, showing 'NETCONF'. The main area is titled 'YANG Suite / NETCONF / YANG set "c9300-default-yangset" / Modules'. It shows the 'c9300-default-yangset' YANG set selected, with the 'Module(s)' dropdown set to 'Cisco-IOS-XE-native'. The 'NETCONF Operation' dropdown is set to 'edit-config'. The 'Device' dropdown is set to 'C9300'. The 'RPC Operations...' button is highlighted, and a blue box with arrows points to the 'Load Replay...' and 'Generate Python script' options in the 'Replays' dropdown menu. The 'Nodes' tree on the left shows the hierarchy of the YANG model, with 'ios-vlan' selected. The 'Value' field shows '28'. The 'Operation' field shows 'create'. The 'Run RPC(s)' button is visible. The right pane shows the generated XML payload for the 'edit-config' operation, and the bottom pane shows the generated Python script.

```
#!/usr/bin/env python
2
import lxml.etree as et
3
from argparse import ArgumentParser
4
from ncclient import manager
5
from ncclient.operations import RPCError
6
7
payload = [
8
9
    <get xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
10
    <filter>
11
    <lag-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-lacp-oper"/>
12
    </filter>
13
    </get>
14
    ''
15
    ]
16
17
if __name__ == '__main__':
18
    parser = ArgumentParser(description='Usage:')
19
20
    # script arguments
21
    parser.add_argument('-a', '--host', type=str, required=True,
22
                        help="Device IP address or Hostname")
23
    parser.add_argument('-u', '--username', type=str, required=True,
24
                        help="Device Username (netconf agent username)")
25
    parser.add_argument('-p', '--password', type=str, required=True,
26
                        help="Device Password (netconf agent password)")
27
    parser.add_argument('--port', type=int, default=830,
28
                        help="Netconf agent port")
29
    args = parser.parse_args()
```

NETCONF + Ansible

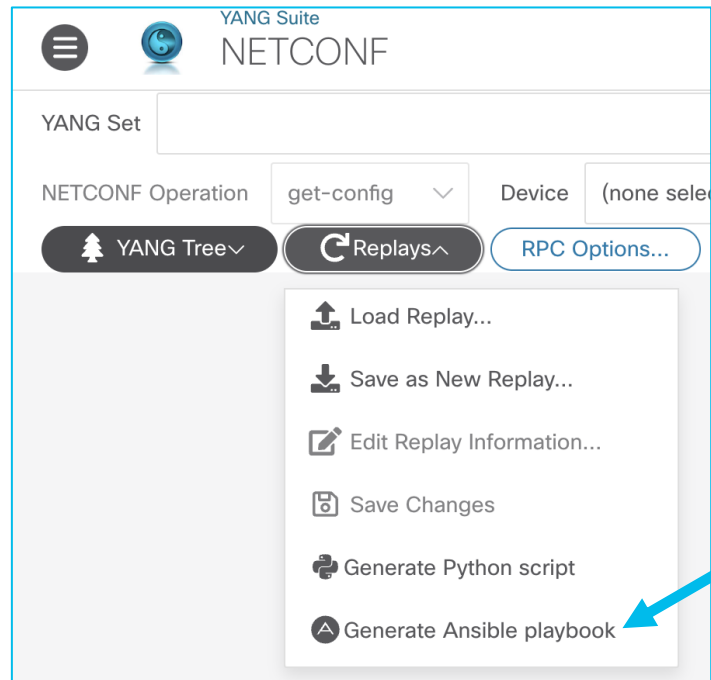
Similar to the “Generate Python” functionality, a new functionality generates YAML formatted for Ansible.

Requirements

1. Install Ansible
2. Install NETCONF collection:
`ansible-galaxy collection install NETCONF`

```
- name : conf-host
  hosts: c9300
  connection: netconf
  gather_facts: no

  tasks:
    - name: hostname-conf
      netconf_config:
        xml: |
          <config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
            <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
              <hostname>c9300-pod29</hostname>
            </native>
          </config>
```



NETCONF + Ansible Update Interface Description Demo

Cisco YANG Suite

Admin Setup Analytics Explore Protocols Test Manager Help

YANG Suite / Help
Help: Welcome to Cisco YANG Suite!

admin

Welcome to Cisco YANG Suite!

YANG Suite is a set of tools related to YANG models ([RFC 6020](#), [RFC 7950](#)) and related technologies such as NETCONF ([RFC 6241](#)).

It provides a modular infrastructure which various YANG application plugins can be used.

Please check [developer.cisco.com/yangsuite](#) for information, learning labs, and announcements.

Docker-based Installation

The `yangsuite/docker/start_yang_suite.sh` script performs the following:

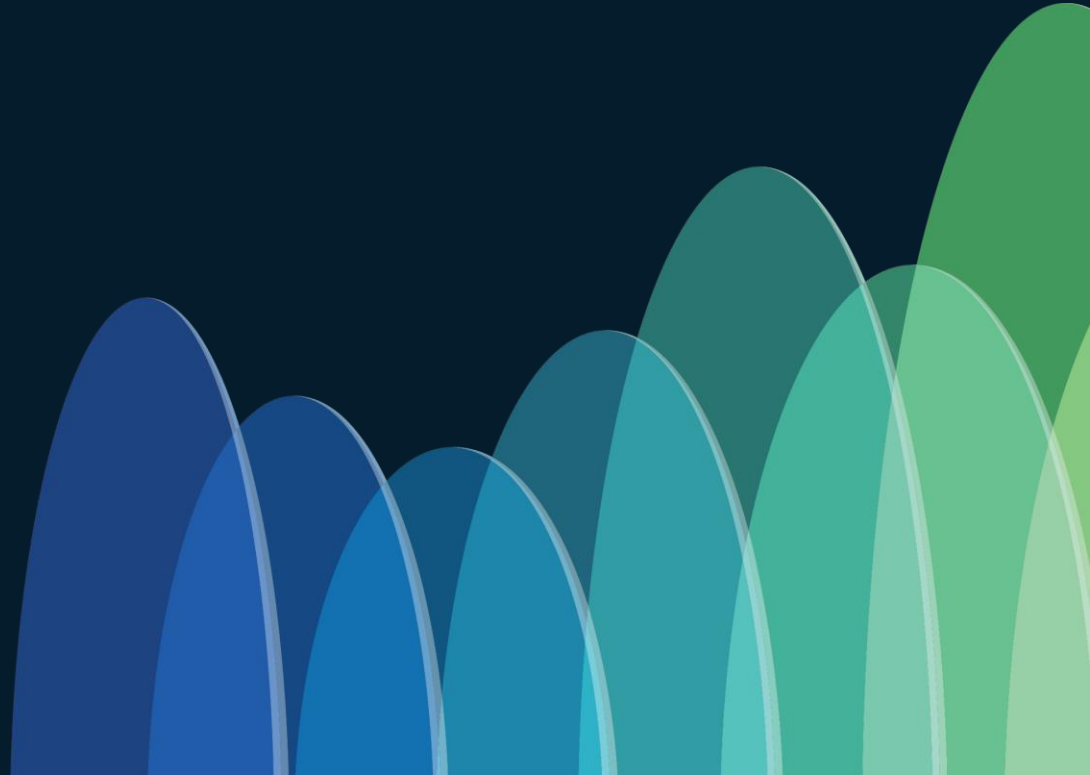
- Prompts for username, password, and email which will be the superuser to yangsuite.
- Gives the choice of creating test SSL/TLS certificate and key.
- Creates an environment file needed for the yangsuite docker container install.
- Runs `docker-compose up`.

YANG Suite Documentation

Search docs

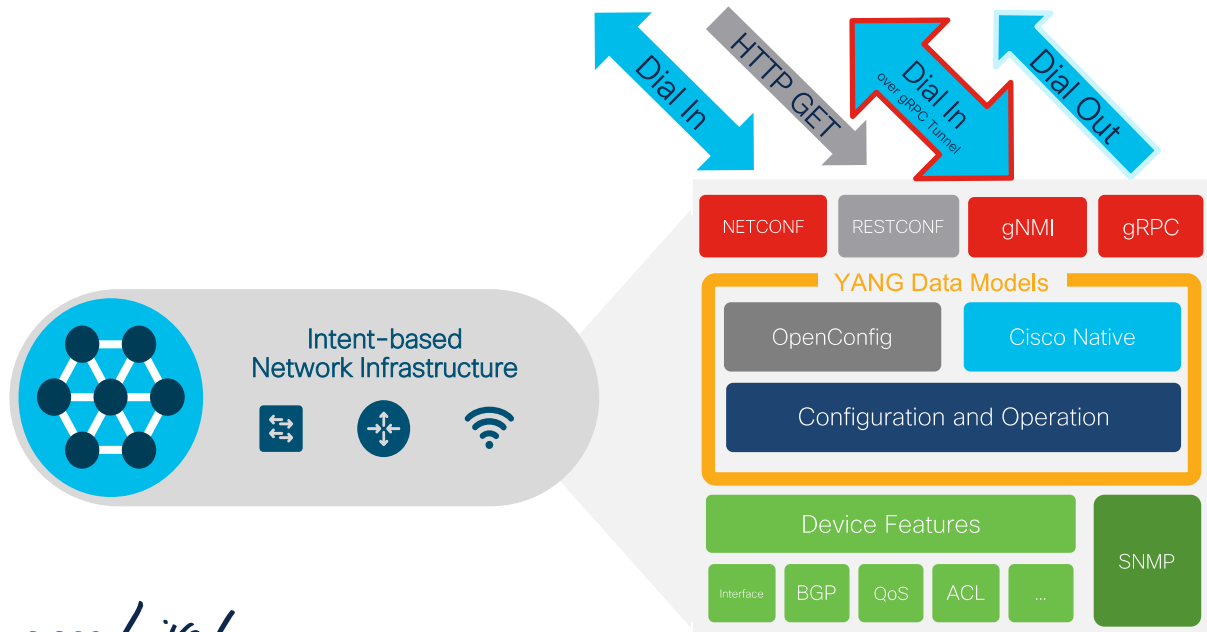
- Welcome to Cisco YANG Suite!
- Device Profiles
 - Managing device profiles
 - Setting up YANG Suite TLS/SSL
- File Manager
 - Constructing and populating a YANG module repository
 - Uploading YANG files from the local filesystem to a YANG repository
 - Downloading YANG files via NETCONF from a device to a YANG repository
 - Copying YANG modules via SCP from a server to a YANG repository
 - Importing YANG modules from a Git repository to a YANG repository
 - Defining a YANG module set
 - Managing YANG module files in YANG Suite
- Working with YANG Models
 - Exploring YANG Models
- YANG Suite Analytics
 - Datasets
 - Mapping SNMP OIDs to YANG XPath
- Using gNMI with YANG Suite
- YANG Suite gRPC telemetry receivers
 - Telemetry over gRPC Clear Channel
 - Configuring gRPC Telemetry Receivers
 - Telemetry over gRPC Secure Channel
- Using NETCONF with YANG Suite
 - Using NETCONF RPCs
 - Working with NETCONF Notification Streams
 - Using Replays for repeated workflows
 - Locking and unlocking datastores
- RESTCONF in YANG Suite
 - YANG Suite RESTCONF
- Test management with YANG Suite
 - Using Test Manager to define and execute tests
 - Generating YANG model Tests
 - Managing and editing replay files

Day 2



Model Driven Telemetry Interfaces

- ↔ Dial In: Collector establishes a connection to the device then subscribes to telemetry (pub/sub)
 - ↔ Dial Out: Telemetry is pushed from the device to the collector based off configuration (push)
- Publication / Subscription**



XML, JSON, proto and kvGPB encoding

Consistent YANG data models between interfaces

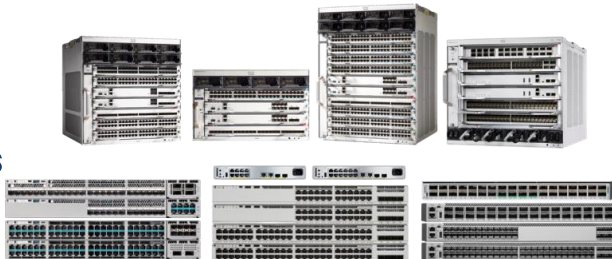
On-change event and time-based publication options

Updated TIG_MDT container now available!

Recently updated!

```
docker pull jeremycohoe/tig_mdt
docker run -ti -p 3000:3000 -p 57500:57500 jeremycohoe/tig_mdt
```

Cisco IOS XE Devices



Collector/Receiver
Decodes to text



Storage
Time Series Database



Monitoring
and Visualizations



https://hub.docker.com/r/jeremycohoe/tig_mdt

<https://github.com/jeremycohoe/cisco-ios-xe-mdt>
https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/179/b_179_programmability_cg/m_179_prog_ieff_telemetry.html

cisco *Live!*

Model Driven Telemetry Interface Comparison

	NETCONF	gRPC (Dial-Out)	gNMI
Minimum IOS XE Version	16.6	16.10	Dial-In: 16.12 over gRPC tunnel: 17.11
Recommended Version	17.9	17.9	Dial-In: 17.9 over gRPC tunnel 17.11
Telemetry Direction	Dial-In, IOS XE is server	Dial-Out IOS XE is client	Dial-In IOS XE is server Dial-Out gRPC Tunnel
Configuration	Dynamic per session	Static per configuration	Dynamic per session
Telemetry Collector	Client	Server	Client
Encoding	XML	KV GPB	JSON_IETF + PROTO
Security	SSH + PKI certificate or password	mTLS or plain-text	mTLS certificates <small>mTLS cert only or mTLS cert + user/pass authentication</small>
Transport Protocol	SSH	HTTP2	HTTP2
Data Models	YANG	YANG	YANG

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

Cisco Telemetry Data Broker (Telegraf)

Cisco Telemetry Broker provides many benefits include brokering, filtering, and transforming data. It provides the ability to replicate telemetry data.

- Cisco Secure Network Analytics (Stealthwatch) UDP Director (UDPD) replicates UDP traffic to multiple destinations.
- Cisco Telemetry Broker
 - Builds upon UDPD
 - Optimizes telemetry pipelines for the hybrid cloud
 - Simplifies the consumption of telemetry data for customers' business-critical tools by brokering hybrid cloud data, filtering unneeded data, and transforming data to a usable format



► Brokering Data:

The ability to route and replicate telemetry data from a source location to multiple destination consumers.

Quickly onboard new telemetry-based tools!

► Filtering Data:

The ability to filter data that is being replicated to consumers for fine grain control over what consumers are able to see and analyze.

Save money sending data to expensive tools!

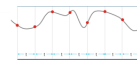
► Transforming Data:

The ability to transform data protocols from the exporter to the consumer's protocol of choice.

Enable tools to consume multiple data formats!

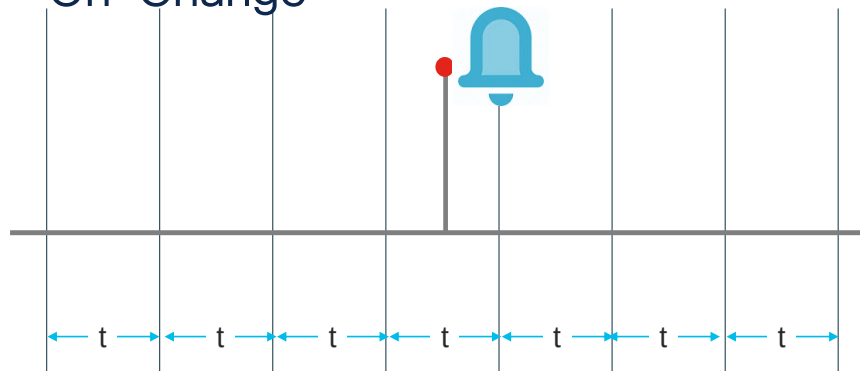
PRODUCTS	UNIT LIST PRICE	STATUS	QUANTITY	ACTION
Virtual License Cisco Telemetry Broker Essential License - 100GB/Day 10-100-100GB/Day	\$10.00 Per 100GB/Day/12 Month	Active	20	Delete
Support - Virtual License Enhanced Support for Cisco Telemetry Broker SVC-10-SUP-E	200.00 Per Each/Month	Active	1	Delete
Premium Support for Cisco Telemetry Broker SVC-10-SUP-P	Take an action to see the List Price Per Each/Month	No action	1	Delete

<https://cs.co/telemetrybroker> aka <https://www.cisco.com/c/en/us/products/security/telemetry-broker/index.html>
<https://blogs.cisco.com/security/taking-full-control-of-your-telemetry-with-the-intelligent-telemetry-plane>



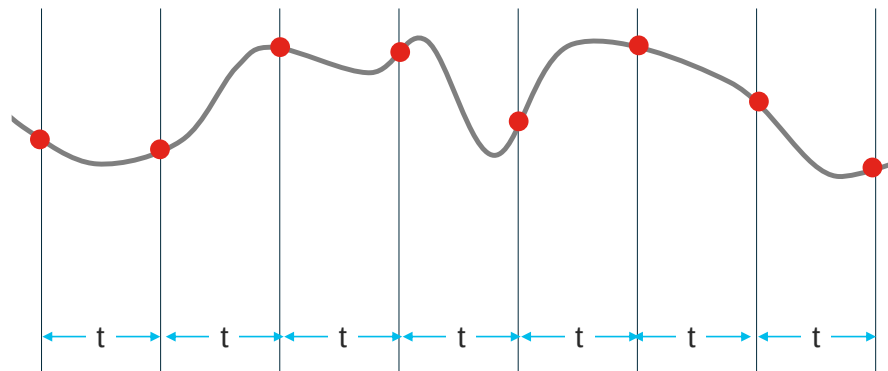
Publication options

On-Change



Feature Model "On-Change" Notifications
Event Notifications (failed login, optic fault, etc)
State and Configuration

Periodic



Feature Model "Periodic" Notifications
Time based publication
Minimum interval 100 centiseconds (1s)

OpenConfig YANG with GNXI, not NETCONF

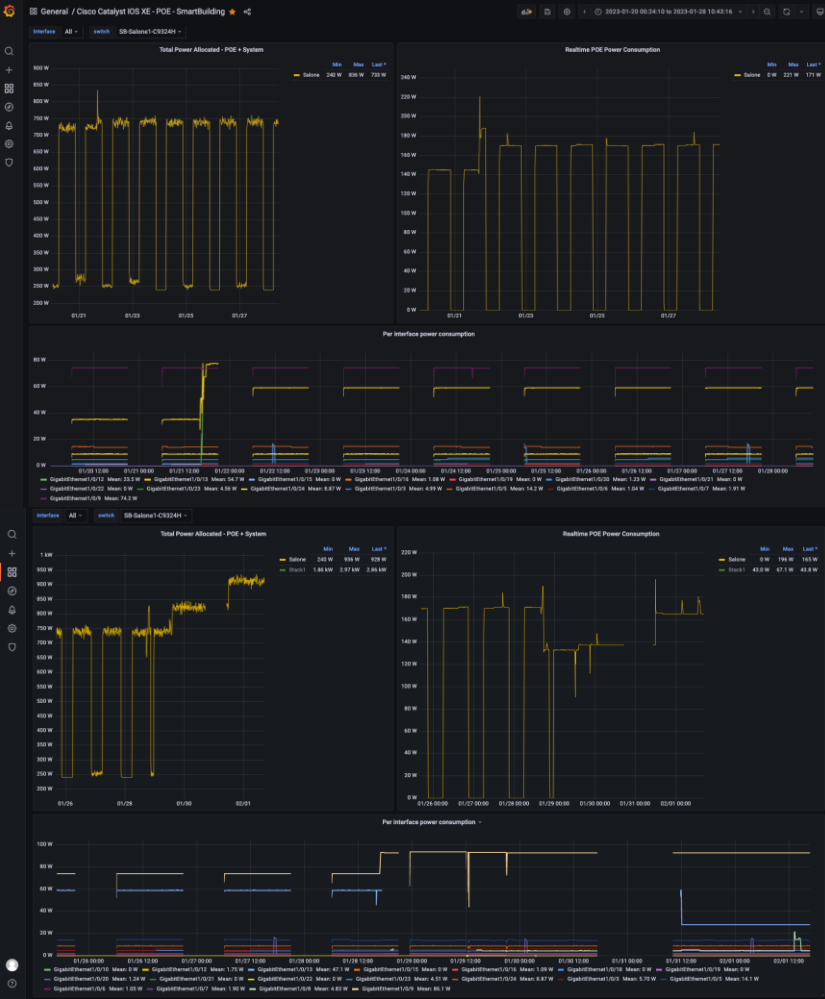
Configure a gRPC Telemetry Subscription

Configuring telemetry subscriptions like the following to collect CPU data over time

On Cisco IOS XE Device:

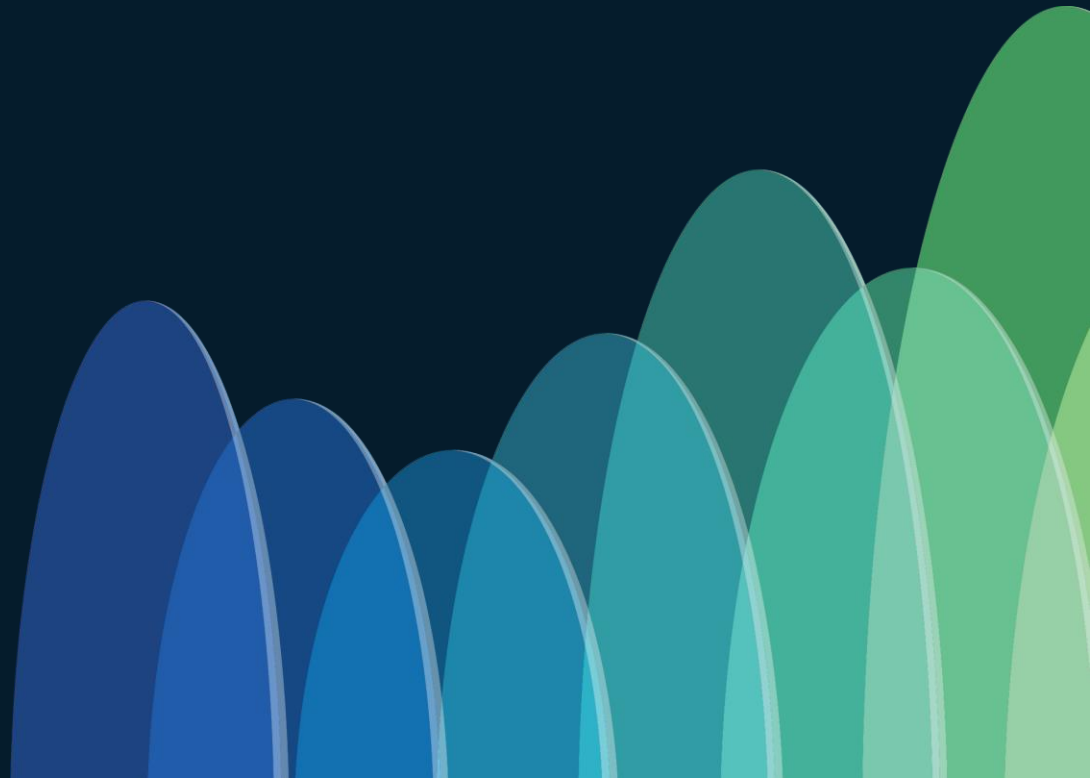
```
configure terminal
telemetry ietf subscription 1
  encoding encode-kvgpb
  filter xpath /process-cpu-ios-xe-oper:cpu-usage/cpu-utilization
  stream yang-push
  update-policy periodic 60000
  receiver ip address 10.1.1.3 57500 protocol grpc-tcp
```

See more examples at: <https://github.com/jeremycohoe/cisco-ios-xe-mdt>



Interested in learning more?
Attend tomorrow's sustainability session
BRKAPP-2245

Day N



gNOI – gRPC Network Operations Interface

1. gRPC Network Operations Interface, or gNOI, is a set of gRPC-based microservices, used for executing operational commands on network devices
2. gNOI operations are executed against the gNMI API interface
3. gNOI is defined and implemented on a per proto basis
4. There are many protos defined – some are more mature and evolve and different pace

Protobuf RPC	Use	Related CLI	Release
Cert.proto	TLS Certificate management	crypto pki ...	17.3
Os.proto	Network Operating System management	install add file ...	17.5
Reset.proto	Factory Reset and secure wipe	write erase ...	17.7
File.proto	Not implemented on IOS XE, use YANG	copy, delete	N/A
System.proto	Not implemented on IOS XE, use YANG	reload, set boot	N/A

<https://github.com/openconfig/gnoi>

<https://github.com/jeremycohoe/cisco-catalyst-gnoi-reset.proto> Factory Reset API Lab Guide

<https://github.com/jeremycohoe/cisco-catalyst-gnoi-cert.proto> Certificate Management API Lab Guide

<https://www.youtube.com/watch?v=cGxgBAlefOQ> GNOI OS.proto – Operating System API demo

GNOI OS.proto demo

```
jcohoe@jcohoe-ubuntu18-lab:~$ cd ~/certs-jcohoe-c9300-2/ ; gnoi_os -insecure -target_addr 10.85.134.92:9339 -op install -target_name c9300 -alsologtostderr -cert ./client.crt -ca ./rootCA.pem -key ./rootCA.key -version 17.06.01.0.135639.1618187331 -time_out 999s -os /tftpboot/cat9k_iosxe.17.06.01-20210411.bin
jcohoe@jcohoe-ubuntu18-lab:~/certs-jcohoe-c9300-2$ cd
jcohoe@jcohoe-ubuntu18-lab:~$
jcohoe@jcohoe-ubuntu18-lab:~$ cd ~/certs-jcohoe-c9300-2/ ; gnoi_os -insecure -target_addr 10.85.134.92:9339 -op activate -target_name c9300 -alsologtostderr -cert ./client.crt -ca ./rootCA.pem -key ./rootCA.key -version 17.06.01.0.135639.1618187331 -time_out 999s -os /tftpboot/cat9k_iosxe.17.06.01-20210411.bin
jcohoe@jcohoe-ubuntu18-lab:~/certs-jcohoe-c9300-2$
jcohoe@jcohoe-ubuntu18-lab:~/certs-jcohoe-c9300-2$ cd ~/certs-jcohoe-c9300-2/ ; gnoi_os -insecure -target_addr 10.85.134.92:9339 -op verify -target_name c9300 -alsologtostderr -cert ./client.crt -ca ./rootCA.pem -key ./rootCA.key
I0418 11:57:10.555504 29388 gnoi_os.go:98] Running OS version: 17.06.01.0.135639.1618187331
jcohoe@jcohoe-ubuntu18-lab:~/certs-jcohoe-c9300-2$
```

```
telnet
C9300#
*Apr 18 18:29:46.613: %INSTALL-5-INSTALL_START_INFO: Switch 1 R0/0: install_engine: Started install remove
*Apr 18 18:32:09.394: %PLATFORM_FEP-1-FRU_PS_ACCESS: Switch 1: power supply A is not responding
*Apr 18 18:32:11.246: %INSTALL-5-INSTALL_COMPLETED_INFO: Switch 1 R0/0: install_engine: Completed install remove
*Apr 18 18:35:24.621: %INSTALL-5-INSTALL_START_INFO: Switch 1 R0/0: install_engine: Started install add flash:gNOI_iosxe.17.06.01.0.135639.1618187331.bin
*Apr 18 18:38:56.711: %INSTALL-5-INSTALL_COMPLETED_INFO: Switch 1 R0/0: install_engine: Completed install add PACKAGE flash:gNOI_iosxe.17.06.01.0.135639.1618187331.bin
*Apr 18 18:38:56.711: %ISSU-3-ISSU_COMP_CHECK_FAILED: Switch 1 R0/0: install_engine: ISSU compatibility check failed for 17.06.01.0.135639
C9300#
C9300#
*Apr 18 18:44:26.608: %INSTALL-5-INSTALL_START_INFO: Switch 1 R0/0: install_engine: Started install activate
*Apr 18 18:44:26.608: %INSTALL-5-INSTALL_AUTO_ABORT_TIMER_PROGRESS: Switch 1 R0/0: install_mgr: Install auto abort timer will expire in 7200 seconds
```

```
C9300#show gnxi os summary
Mode: Install Mode
Current Operation: No operation in progress
No image has been added as Inactive
Current Activated Version: 17.05.01.0.144.1617180620
C9300#show gnxi os summary
Mode: Install Mode
Current Operation: No operation in progress
No image has been added as Inactive
Current Activated Version: 17.06.01.0.135639.1618187331
Last Executed RPC: Verify, Success
```

[Demo video on YouTube](#)

Verify:

```
cd ~/certs-jcohoe-c9300-2/ ; gnoi_os -insecure -target_addr 10.85.134.92:9339 -op verify -target_name c9300 -alsologtostderr -cert ./client.crt -ca ./rootCA.pem -key ./rootCA.key
Running OS version: 17.05.01.0.144.1617180620
```

Install:

```
cd ~/certs-jcohoe-c9300-2/ ; gnoi_os -insecure -target_addr 10.85.134.92:9339 -op install -target_name c9300 -alsologtostderr -cert ./client.crt -ca ./rootCA.pem -key ./rootCA.key -version 17.06.01.0.135639.1618187331 -time_out 999s -os /tftpboot/cat9k_iosxe.17.06.01-20210411.bin
```

Activate:

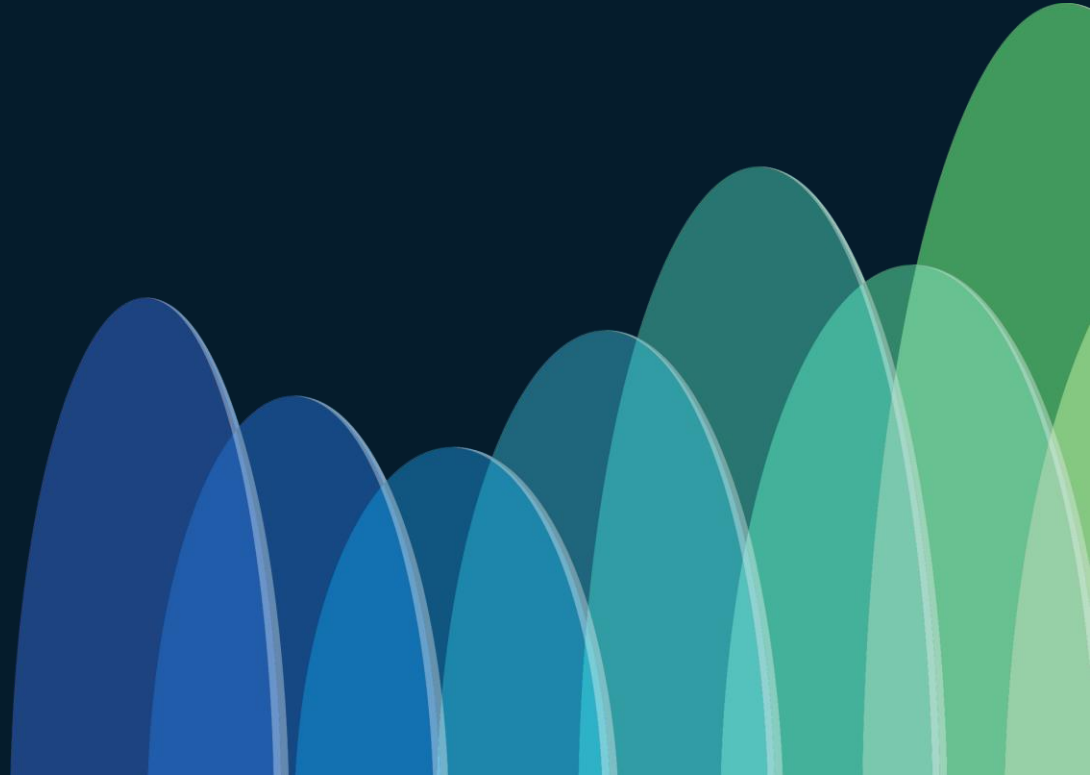
```
cd ~/certs-jcohoe-c9300-2/ ; gnoi_os -insecure -target_addr 10.85.134.92:9339 -op activate -target_name c9300 -alsologtostderr -cert ./client.crt -ca ./rootCA.pem -key ./rootCA.key -version 17.06.01.0.135639.1618187331 -time_out 999s -os /tftpboot/cat9k_iosxe.17.06.01-20210411.bin
```

Verify:

```
Running OS version: 17.06.01.0.135639.1618187331
```

show CLI: show gnxi os summary

Tooling



YANG Suite Capabilities

- Initial Release:
 - Plugin and YANG File Manager, Datasets and diffs
 - Device Manager
 - NETCONF (Python), gRPC Telemetry
 - Docker install support with HTTPS
- Second Release:
 - RESTCONF
 - gNMI
 - Python Integrations
- Third Release:
 - gRPC Telemetry with TLS Support
 - SNMP OID to YANG Xpath Mapping
 - Ansible Integrations
 - Pip install support

Core plugins

Additional plugins

Cisco YANG Suite

Admin

- Manage users
- Manage plugins
- View logs

Setup

- YANG files and repositories
- YANG module sets
- Device profiles

Analytics

- Datasets and diffs
- SNMP to YANG Mapping

Protocols

- gNMI
- gRPC telemetry
- NETCONF
- RESTCONF

Explore

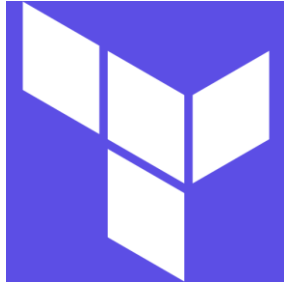
- YANG

What is Ansible?



- Open-source configuration management tool
- Commercially supported by Red Hat
- Declarative and idempotent
- Ansible can be Imperative when needed
- Can manage a wide range of systems: VMs, network devices, cloud instances, etc.
- Agentless: no requirement for installation of application software to run
- Has Python server-side dependencies

They can coexist... it's not an either/or story



- Terraform keeps state locally
- Terraform knows what is configured vs desired end-state
- Terraform can automatically destroy/recreate resources



- Ansible mutates the infrastructure
- Ansible does not keep state (mostly)
- Terraform can call Ansible to perform tasks post resource deployment (e.g., on VMs deployed)

Create a MDT Subscription using Ansible

CLI

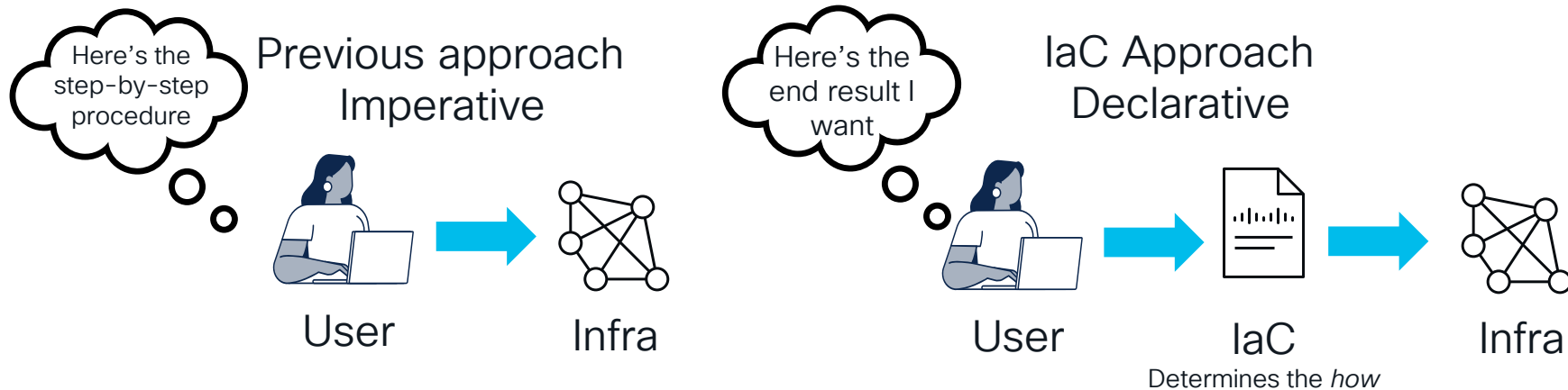
```
configure terminal
telemetry ietf subscription 1
  encoding encode-kvgpb
  filter xpath /process-cpu-ios-xe-oper:cpu-usage/cpu-utilization
  stream yang-push
  update-policy periodic 60000
  receiver ip address 10.1.1.3 57500 protocol grpc-tcp
```

```
1  - name: add telemetry subscriptions
2    hosts: c9300
3    connection: netconf
4    gather_facts: no
5
6    tasks:
7      - name: establish subscription
8        netconf_config:
9          xml: |
10             <config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
11               <mdt-config-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-cfg">
12                 <mdt-subscription>
13                   <subscription-id>1</subscription-id>
14                   <base>
15                     <stream>yang-push</stream>
16                     <encoding>encode-kvgpb</encoding>
17                     <source-address>10.1.1.5</source-address>
18                     <source-vrf>Mgmt-vrf</source-vrf>
19                     <period>2000</period>
20                     <xpath>/process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds</xpath>
21                   </base>
22                   <mdt-receivers>
23                     <address>10.1.1.3</address>
24                     <port>57500</port>
25                     <protocol>grpc-tcp</protocol>
26                   </mdt-receivers>
27                 </mdt-subscription>
28               </mdt-config-data>
29             </config>
```

https://github.com/jeremycohoe/ansible-config-samples/blob/master/add_sub_17.7.yaml

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>

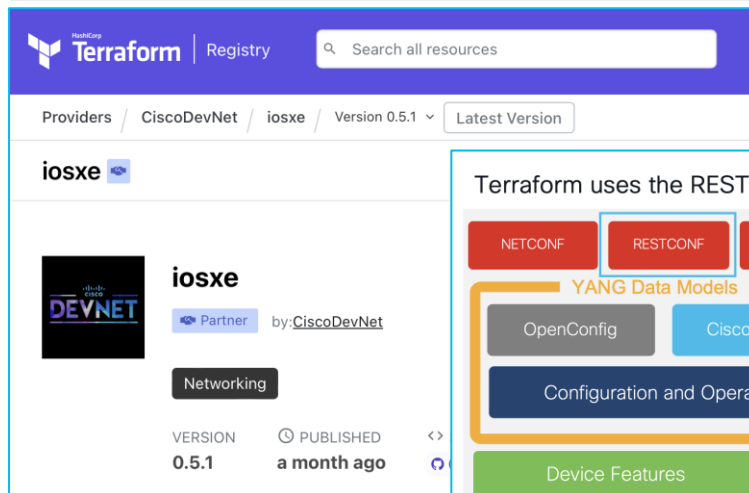
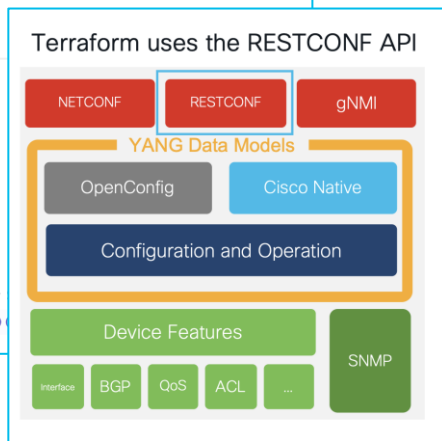
Cisco IOS XE Terraform Provider

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.

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

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

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



iosxe provider

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

General

Resources

iosxe_cli
iosxe_restconf
iosxe_save_config

Data Sources

iosxe_restconf

Terraform for Model Driven Telemetry

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

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

Terraform will perform the following actions:

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

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

Lab Guide Steps:

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

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: ~ (ssh)
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:~$ █

auto@pod27-xelab: ~ (ssh)
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 █

```

Resources

Cisco Catalyst Programmability Sessions at Cisco Live APJC 2024

Sunday Nov 10

Monday Nov 11

Tuesday Nov 12

Wednesday Nov 13

Thursday Nov 14

Session Levels:

Beginner

Intermediate

4:15 PM

DEVNET-2385

Atomic Config
Replace with Cisco
Catalyst 9000

2:00 PM

DEVNET-1249

Secure Zero Touch
Provisioning (SZTP)
with Cisco Catalyst
Switches

11:30 AM

DEVNET-1283

Programmability,
Automation Model
Driven Telemetry on
Cisco IOS XE with a
dash of YANG Suite?

8:30 AM

BRKAPP-2245

Programmability on
Catalyst 9000
enabling
Sustainability
Telemetry
Dashboards

Secure ZTP Blog



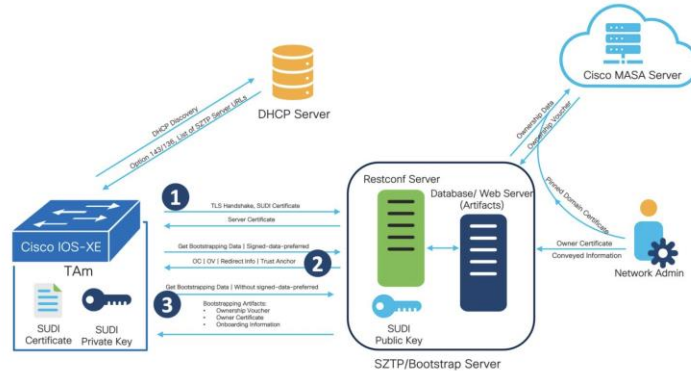
Developer

Building Trust from the Ground Up: The Role of Secure ZTP in Zero Trust Networks

6 min read

Ashil Parekh

Secure ZTP employs three-step validation, including device validation, server validation, and artifact validation, to securely onboard the device. The diagram provided below illustrates the various steps involved in the device onboarding and provisioning process within a secure ZTP framework. Let's take a closer look at each of these steps:



<https://blogs.cisco.com/developer/secureztp01>

ZTP in Action

Start the hands-on lab for Day 0



Module: Zero Touch Provisioning

In this module, you will verify and confirm the prerequisites for Zero Touch Provisioning (ZTP), the feature of IOS XE on the Catalyst 9300 switch. At the end of this module, you will issue the 'write erase' command, reload the switch, and watch as the ZTP process completes and the switch is configured programmatically and automatically.

What is ZTP? When a device that supports Zero-Touch Provisioning boots up, and does not find the startup configuration (during initial installation), the device enters the Zero-Touch Provisioning mode. The device searches for an IP from a DHCP server and bootstraps itself by enabling the Guest Shell. The device then obtains the IP address or URL of an HTTP/TFTP server, and downloads a Python script from an server to configure the device.

```
write erase  
reload
```

ping

<https://github.com/sdeweese/CLUS22-LTROPs-1836-programmability-and-automation/blob/main/ZTP2.md>

Open Programmable Infrastructure (OPI) SZTP

Ready to create your own bootstrapping server, consider the OPI one to get started

Device Getting Onboarding Information

Read the configuration back and validate it is correct:

```
docker run --rm --network sztp_opi docker.io/curlimages/curl:8.5.0 --silent --fail-with-body -
```

Get onboarding info (from device perspective)

```
$ docker run --rm --user 0 --network sztp_opi -v sztp_cli
{
  "ietf-sztp-bootstrap-server:output": {
    "conveyed-information": "MIIDfwYLKoZIHvcNAQkQASugggNul
  }
}
```

Decode payload

```
$ jq -r -s "\nietf-sztp-bootstrap-server:output\n" /tmp/post_rpc_input.js
{
  "ietf-sztp-conveyed-info:onboarding-information": {
    "boot-image": {
      "download-uri": [
        "https://web:443/my-boot-image.img",
        "https://web:990/my-boot-image.img"
      ],
      "image-verification": [
        {
          "hash-algorithm": "ietf-sztp-conveyed-info:sha-256",
          "hash-value": "7b:ca:e6:ac:23:06:d8:79:06:8c:ac:03:80:e2:16:44:7e:40:6a:65:fa:d4:69:"
        }
      ]
    },
    "pre-configuration-script": "IyEvYm1uL2Jhc2gKZWNoYiAiaW5zaWR1IHRoZSBwcmluY29uZm1ndXJhdGlvb
    "configuration-handling": "merge",
    "configuration": "PHRvcC84bWxucz8iaHR8cHM6L2V4YW1wbGUuY29tL2NvbmZpZyI+CjAgPGFueS14bWwY29u
    "post-configuration-script": "IyEvYm1uL2Jhc2gKZWNoYiAiaW5zaWR1IHRoZSBub3N0bWVbmZpZ3V5YXRp
  }
}
```

<https://github.com/opiproject/sztp>

THE LINUX FOUNDATION PROJECTS

OPEN PROGRAMMABLE INFRASTRUCTURE PROJECT

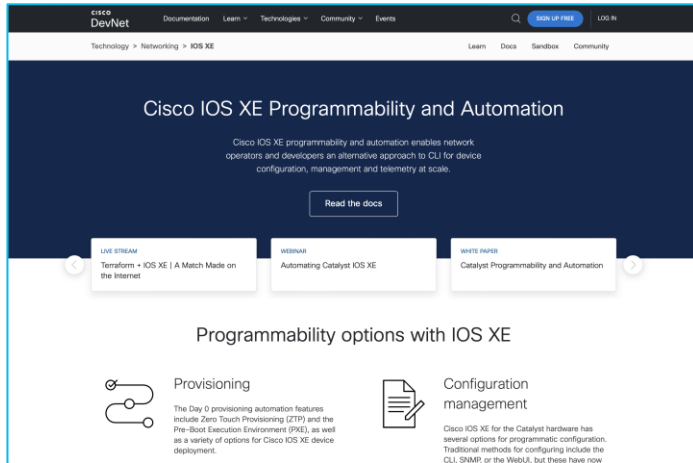
Open ecosystems for next generation architectures and frameworks

The objective of the Open Programmable Infrastructure Project is to foster a community-driven standards-based open ecosystem for next generation architectures and frameworks based on DPU/IPU-like technologies.

JOIN US

Programmability Website

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



- Community Forum
- IOS XE FAQ
- White Papers
- Code Exchange
- IOS XE Docs & Guide
- Learning Tracks and Labs
- Sandboxes
- ... and more !



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

Cisco YANG Suite



YANG API Testing and Validation Environment

Construct and test YANG based APIs over
NETCONF, RESTCONF, gRPC and gNMI

IOS XE / IOS XR / NX OS platforms

The top screenshot shows the 'Explore YANG Models' view. It features a sidebar with navigation links (Admin, Setup, Explore, Protocols, Help) and a main area with a tree of YANG models. The selected model is 'Cisco-IOS-XE-interfaces-oper'. A 'Node Properties' table is visible on the right.

Node Properties	
Name	statistics
Node type	container
Description	A collection of interface-related statistics objects
Module	Cisco-IOS-XE-interfaces-oper
Revision	2020-07-01
Xpath	/interfaces/interface/statistics
Prefix	interfaces-ios-xe-oper
Namespace	http://cisco.com/ns/yang/Cisco-IOS-XE-interfaces-oper
Access	read-only

The bottom screenshot shows the 'NETCONF' view. It features a sidebar with navigation links (Admin, Setup, Explore, Protocols, Help) and a main area with a tree of nodes. The selected node is 'Cisco-IOS-XE-interfaces-oper'. A 'Value' field is visible next to the node. An XML output is shown on the right.

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <get>
    <filter>
      <interfaces xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-interfaces-oper">
        <interface>
          <name/>
          <statistics/>
        </interface>
      </interfaces>
    </filter>
  </get>
</rpc>
```

developer.cisco.com/yangsuite

github.com/CiscoDevNet/yangsuite

Get hands-on using the new learning lab!

<https://developer.cisco.com/learning/labs/intro-yangsuite/>

Docker container innovation 1 container



API White Paper

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 ^

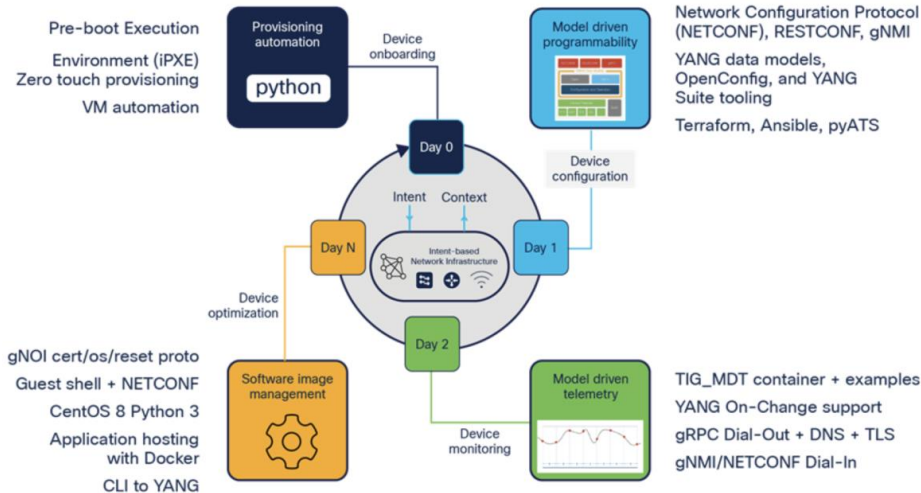
Conclusion

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>

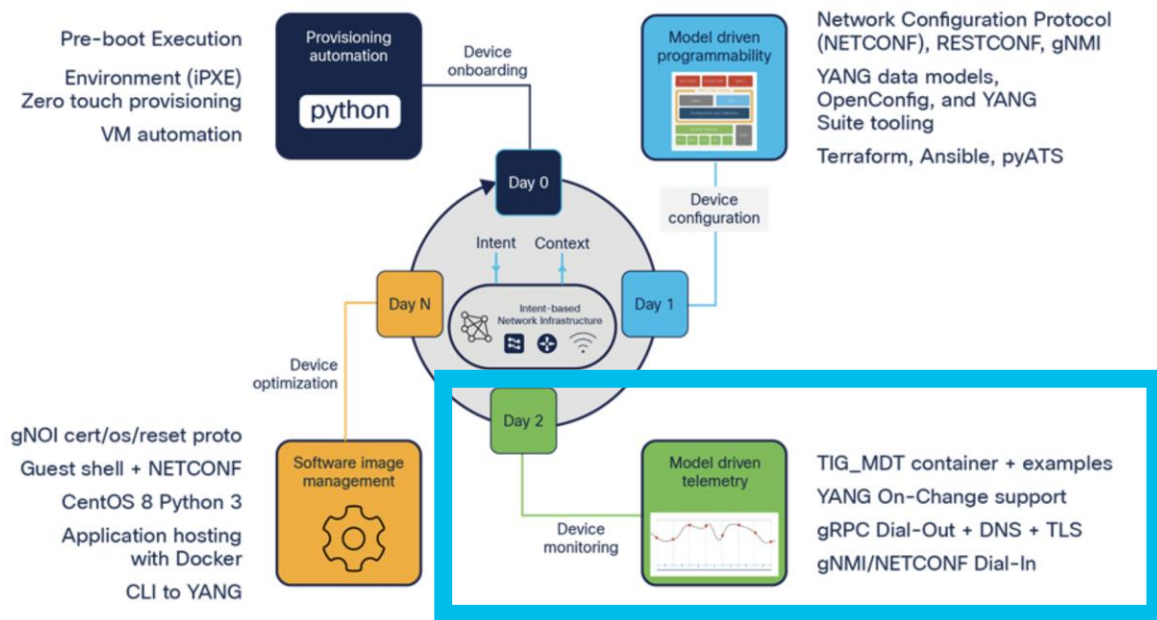
PDF: <https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9300-series-switches/nb-06-catalyst-programmability-automation-wp.pdf>

Webinar with live demos & examples: <https://www.youtube.com/watch?v=Ldck5PnPu2I>

cisco Live!

Model Drive Telemetry (MDT) White Paper

The Model Driven Telemetry White Paper includes examples, use cases and tooling related to telemetry. This paper is now available online and in PDF form!



Website



PDF



View online: <https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9300-series-switches/model-driven-telemetry-wp.html>
View as PDF: <https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9300-series-switches/model-driven-telemetry-wp.pdf>

cisco Live!

Introduction to Cisco IOS ... ^ Q

Table of Contents

Introduction to Cisco IOS XE

Introduction to telemetry

Benefits of model driven tele...

Network monitoring challeng...

Architecture and databases

Dial-in and dial-out MDT +

Publication notification optio...

YANG data modeling language +

Benchmarking and comparis... +

Cisco controller solutions +

Cloud solutions +

Tooling +

Dashboarding and validation +

Configuration examples +

Telemetry configuration man... +

Troubleshooting and validati... +

Best practices and lessons l... +

Conclusion

Resources

Cisco IOS XE – Reservable Virtual Sandbox

This DevNet reservable sandbox has IOS XRv + N9Kv + IOS XE
The Catalyst 8000V within the DevBox is enabled for Day 0/ZTP usecases

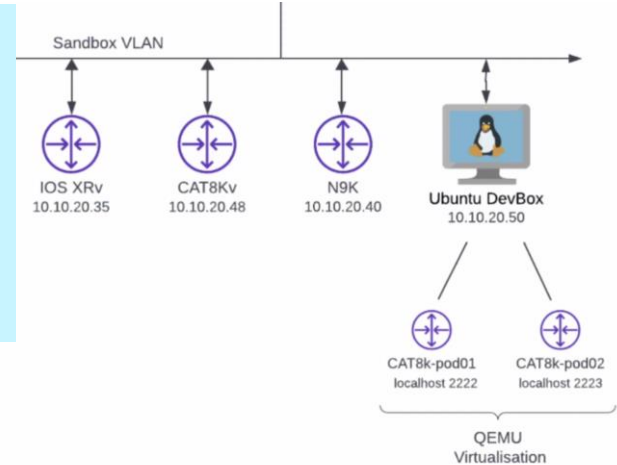
<https://devnetsandbox.cisco.com>

“IOS XE on Cat8kv”

<https://devnetsandbox.cisco.com/DevNet/catalog/IOS%20XE%20on%20Cat8kv>

The “Enterprise Networking” Learning Labs contains guides for the supported usecases

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



Explore YANG data models and NETCONF to query and configure IOS XE network devices. Then use Ansible to manage configurations on your IOS XE devices, covering the fundamentals of Ansible. Lastly, enable...

IOS XE on Cat8kv

The IOS XE on Catalyst 17.9.2 Sandbox offers developers access to an IOS XE device running the latest IOS XE code release available on cisco.com. Here you can test out the newest programmability...

Networking

reservable



Launch

6 hrs - 4 modules - 12 labs

Sandbox Capabilities:

TIG_MDT Telemetry
YANG Suite API
Terraform + Ansible
ZTP & Guest Shell
... and more

Table of Contents

- Overview
- Network Topology
- Grafana Dashboard
- Access Details
- Additional Resources
- VPN Access

Catalyst 8000v Credentials

host	port	username	password
10.10.20.48	22	developer	C1sco12345
10.10.20.48	830	developer	C1sco12345
10.10.20.48	443	developer	C1sco12345

Developer Box Credentials

host	port	username	password
10.10.20.50	22	developer	C1sco12345
telnet localhost (CAT8k-1)	2222	developer	C1sco12345
telnet localhost (CAT8k-2)	2223	developer	C1sco12345

Cisco IOS XE – Always On Virtual Sandbox

The “[IOS XE on Cat8kv Always On](#)” virtual sandbox requires no reservation

The c8kv can be accessed with SSH, NETCONF, and RESTCONF

Hostname: sandbox-iosxe-latest-1.cisco.com

Username: admin Password: C1sco12345

SSH port 22, NETCONF port 830, RESTCONF HTTPS

<https://devnetsandbox.cisco.com>

devnetsandboxiosxe.cisco.com

<https://devnetsandbox.cisco.com/DevNet/catalog/ios-xe-always-on>

The “Enterprise Networking” Learning Labs contains guides for the supported usecases

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

IOS XE on Cat8kv AlwaysOn



This AlwaysOn sandbox offers developers access to an IOS XE device running the latest IOS XE code release available on cisco.com (currently 17.11.x). Here you can test out the newest programmability feature...

Always-On



Launch

Access Details:

Developers and network engineers access the IOS XE on Catalyst 17.11.x sandbox directly using the following information:

- Cat8000v Host
 - Address: `sandbox-iosxe-latest-1.cisco.com`
 - Username: admin
 - Password: C1sco12345
 - NETCONF port: 830
 - gRPC telemetry port: 57500
 - ssh port: 22

Learning Track

Enterprise Networking

Explore YANG data models and NETCONF to query and configure IOS XE network devices. Then use Ansible to manage configurations on your IOS XE devices, covering the fundamentals of Ansible. Lastly, enable...

6 hrs - 4 modules - 12 labs

Cisco University (Cisco U) part of L&D

<https://u.cisco.com>

<https://u.cisco.com/search/tutorial?query=Story%20DeWeese,%20Jeremy%20Coho,%20not%20berry>

Direct link to Tutorial, requires login to u.cisco.com first:

1. <https://ondemandlearning.cisco.com/apollo-alpha/tc-iosxe-ztp/pages/1>
2. <https://ondemandlearning.cisco.com/apollo-alpha/tc-terraform-ios-xe/pages/1>
3. <https://ondemandlearning.cisco.com/apollo-alpha/tc-yangsuite-netconf/pages/1>
4. <https://ondemandlearning.cisco.com/apollo-alpha/tc-yangsuite-restconf/pages/1>

Cisco U.

Welcome to Cisco U.

You're in the right place, whether you're looking to earn a certification or gain new skills. In Cisco U., you'll find courses, community, and learning content to help you reach your goals.

[Learn more about Cisco U.](#), or come on in and get started!

Log in with your Cisco Account ID

Create a free account

cisco *Live!*

TUTORIAL

 Jeremy Coho

Cisco IOS XE Zero-Touch Provisioning

In this tutorial, you will learn the basics of the IOS XE ZTP feature.

Beginner 45m

Created October 23, 2023

TUTORIAL

 Story DeWeese

Using YANG Suite with NETCONF

In this tutorial, you'll learn how to use YANG Suite to visualize the NETCONF protocol with network devices.

Intermediate 30m

Created October 24, 2023

TUTORIAL

 Story DeWeese

Using YANG Suite with RESTCONF and gRPC

In this tutorial, you'll learn how to use YANG Suite to visualize the RESTCONF and gRPC protocols with network devices.

Intermediate 10m

Created October 24, 2023

TUTORIAL

 Story DeWeese

Cisco IOS XE Terraform Provider

Get started leveraging the power of Terraform. In this tutorial, we'll use this cloud-native provisioning tool to create a new VLAN.

Beginner 15m

Created February 21, 2023

Keynote Deep Dives

Wednesday
10:30am – 11:30am



Experiences Amplified:
How AI Can Fuel Better Employee and Customer Experiences

Level 1
Room 106



Smart, Secure, Seamless:
Transforming Experiences with Next-Generation Networking

Level 2
Room 204



Harness a Bold New Era:
Transform Data Centre and Service Provider Connectivity

Level 2
Room 203



Securing User to Application and Everything in Between

Level 2
Melbourne Room 2



Unlocking Digital Resilience through Unified Observability

The HUB
Centre Stage

Complete Your Session Evaluations



Complete a minimum of 4 session surveys and the Overall Event Survey to claim a **Cisco Live T-Shirt**.



Complete your surveys in the **Cisco Live mobile app**.



Continue your education



- Visit the Cisco Stand for related demos
- Book your one-on-one Meet the Expert meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand



Thank you

CISCO *Live!*

#CiscoLiveAPJC

CISCO *Live!*

GO BEYOND

#CiscoLiveAPJC