



Energy Management for Catalyst

IOS XE Device Telemetry Workshop

Jeremy Cohoe & Story DeWeese
Technical Marketing

April 22, 2024 (Earth Day)

Blog, Sign-Up, and Overview



<https://blogs.cisco.com/developer/developer-sustainability-week-events>

Energy Management and Sustainability Workshop April 25

On Wednesday, April 25, a hands-on coding workshop, led by [Jeremy Cohoe](#) and [Story DeWeese](#), will guide participants through for coding for energy management on Cisco Catalyst products. As a bring-your-own-hardware (BYOH) event, you can apply this learning on your own devices. In this free workshop, you will get experience in coding for energy management on Cisco Catalyst products and learn coding practices that can be immediately applied to your own hardware for instant sustainability gains.

Register for the free workshop: [Power Up Savings: Cisco Catalyst Energy Management with Jeremy Cohoe & Story DeWeese](#)

About Us

This hands-on coding workshop is designed for the eco-conscious developer. As part of Cisco DevNet's Developer Sustainability Week, this session, led by Jeremy Cohoe (Technical Marketing Engineer) and Story DeWeese (Technical Marketing Engineer), will guide participants through the intricacies of writing code that elevates energy management to new heights within Cisco Catalyst products. As a bring-your-own-hardware (BYOH) event, attendees will have the unique opportunity to apply their learning, transforming theory into practice on their very own devices.

In this workshop, you will:

- Acquire hands-on experience in coding for energy management on Cisco Catalyst products.
- Develop a toolkit of coding practices that can be immediately applied to your own hardware for instant sustainability gains.
- Have the confidence to customize and innovate upon your newly learned coding skills to further optimize energy management within their networks.
- Become part of an elite group of tech professionals who are at the forefront of driving sustainable practices in the networking industry.

Secure your spot in this workshop today!

cisco DevNet

Developer Sustainability Week

Developer Sustainability Week: Energy Management for Catalyst Devices Workshop

Power Up Savings: Cisco Catalyst Energy Management with Jeremy Cohoe & Story DeWeese

Virtual

Developer Sustainability Week: Energy Management for Catalyst Devices Workshop

[Get Ticket](#)

Thu, Apr 25, 2024, 9:00 AM -
Thu, Apr 25, 2024, 11:00 AM
America/Los_Angeles (UTC -07:00)

[Add to Calendar](#)



Workshop Agenda

1. Getting into the Lab(s)
 - Demo in dCloud
2. Overview of IOS XE Telemetry
3. Using Telemetry with YANG Suite
 - YANG Suite demo in POD8
4. Collecting with TIG_MDT
 - Docker demo in POD8
5. Terraform for configuration management
 - Demo 8 subscriptions
6. Sustainability APIs and data
 - Grafana Demo
7. Resources

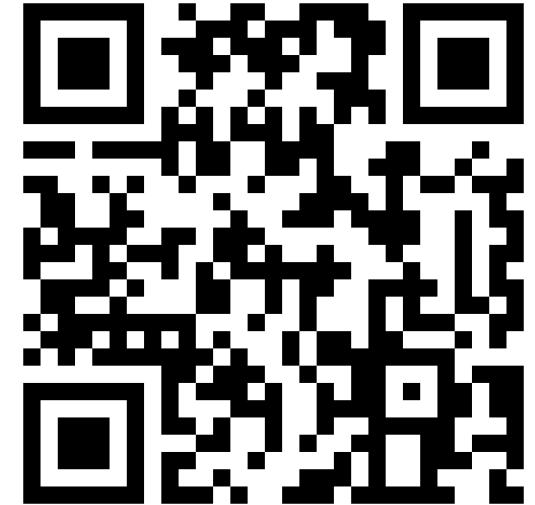
Lab Environments

IOS XE Device Programmability Website

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

The screenshot shows the Cisco DevNet website for IOS XE. The header includes links for Documentation, Learn, Technologies, Community, and Events, along with a search bar, sign-up button, and log-in link. The main content area is titled "Cisco IOS XE Programmability and Automation" and discusses the enabling of network operators and developers for device configuration, management, and telemetry at scale. It features a "Read the docs" button and three cards: "LIVE STREAM Terraform + IOS XE | A Match Made on the Internet", "WEBINAR Automating Catalyst IOS XE", and "WHITE PAPER Catalyst Programmability and Automation". Below this, there's a section titled "Programmability options with IOS XE" with two sections: "Provisioning" (using ZTP and PXE) and "Configuration management" (using Cisco IOS XE for Catalyst hardware). Each section has a brief description and a small icon.

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



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

Learning Lab and Blogs: Telemetry

<https://developer.cisco.com/learning/modules/ansible-mdt/>

<https://blogs.cisco.com/developer/model-driven-telemetry-sandbox>

<https://blogs.cisco.com/developer/getting-started-with-model-driven-telemetry>

<https://youtu.be/QwwZakkWBng>



Developer
Explore Model-Driven
Telemetry
Stuart Clark

New learning labs and sandbox

As our journey through network automation grows, so does the need for our network tools. Network Engineers have always been considered the absolute escalation point for any performance difficulties and problems, irrespective whether the root cause is really the network, server, or application. Network Engineers are expected to have the knowledge and tools to isolate and identify the issue, collaborating with other teams such as SRE / AppDev to bring it to resolution and often present this in an RCA (root cause analysis).

One of these great tools which can really help is telemetry. In software, telemetry is used to gather data on the use and performance of applications and application components, e.g. how often certain features are used, measurements of start-up time and processing time, hardware, application crashes, and general usage statistics and/or user behavior.

Why Streaming Telemetry?

Cisco IOS XE is the Network Operating System for the Enterprise. It runs on switches like the Catalyst 9000, routers like the ASR 1000, CSR1000v, and ISR 1000 and 4000's, Catalyst 9800 Wireless LAN controllers, as well as a few other devices in IoT and Cable product lines. Since the IOS XE 16.6 release there has been support for model driven telemetry, which provides network operators with additional options for getting information from their network.

Learning Lab



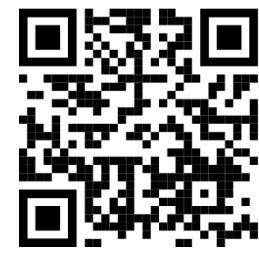
Blog



YouTube



DevNet Sandboxes – IOS XE Overview



Easily access IOS XE as part of the DevNet Sandbox ecosystem

This DevNet reservable sandbox has IOS XRv + N9Kv + IOS XE
The c8k within the DevBox is enabled for Day 0/ZTP usecases

<https://devnetsandbox.cisco.com>
“IOS XE on Cat8kv”

The “IOS XE on Cat8kv Always On” virtual sandbox requires no reservation

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

Hostname: devnetsandboxiosxe.cisco.com

Username: admin Password: C1sco12345

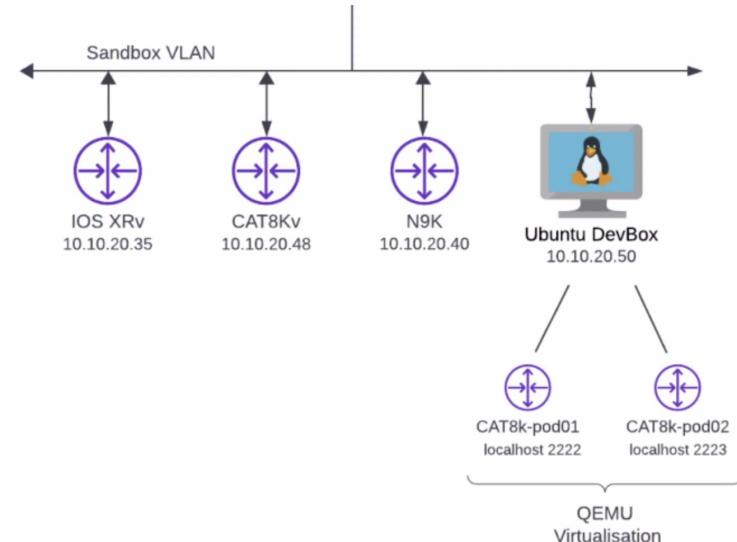
SSH port 22, NETCONF port 830

devnetsandboxiosxe.cisco.com
“IOS XE on Cat8kv AlwaysOn”

Reservable – Physical 9300, 9300X, 9200 + stacks

The physical labs are being rebuilt and are not quite ready yet

The screenshot shows the Cisco DevNet Sandbox interface. At the top, there are tabs for 'INSTRUCTIONS', 'COMMANDS', 'NAVIGATOR', 'STYLE', and 'BEHAVIOR'. Below the tabs, there's a section titled 'OVERVIEW' which contains a brief description of the IOS XE on Catalyst sandbox. It mentions that it offers access to an IOS XE device running the latest IOS XE code release available on cisco.com (currently 17.9.2). It also notes that the sandbox is updated approximately every 4 months. A note states that the code provided is for IOS XE (approximately every 4 months). Below this, there's a section titled 'PROGRAMMABILITY FEATURES' which lists several features such as Model Driven Programmability with YANG Data Models and NETCONF/YANG support, Linux Guest Shell on-box for running Linux Applications and run Python Scripts directly at the edge, and a KVM-based DevBox for developers and network engineers to build and deploy applications (custom or off the shelf) on their network devices. It also mentions Day 0 Provisioning with Network Plug and Play (NPnP) and Zero Touch Provisioning (ZTP), a DevBox (running KVM) providing a Linux environment for running applications and testing ZTP. In addition, a stand-alone Catalyst 8000v with SSH access has been provided to validate other features. A note states that IOS XE is the software running on many platforms from Cisco, but not all programmability features are available on every platform. Check specific platform documentation for details. At the bottom, there's a section titled 'ACCESS DETAILS' which provides instructions for reserving the sandbox and receiving software VPN access information and credentials via email at the start of the reservation or by viewing details in the 'Output' view in the reservation details.





DevNet Sandbox: 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/>



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

① 6 hrs - 4 modules - 12 labs

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

Table of Contents

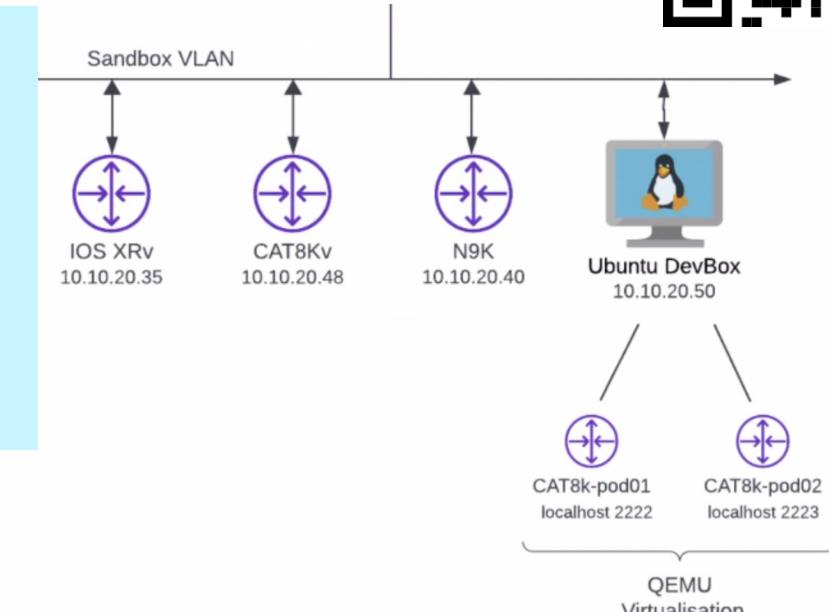
- Overview
- Network Topology
- Grafana Dashboard
- Access Details
- Additional Resources
- VPN Access

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





DevNet Sandbox: 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

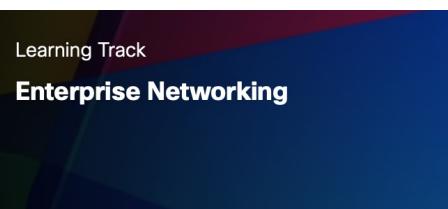


Launch

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



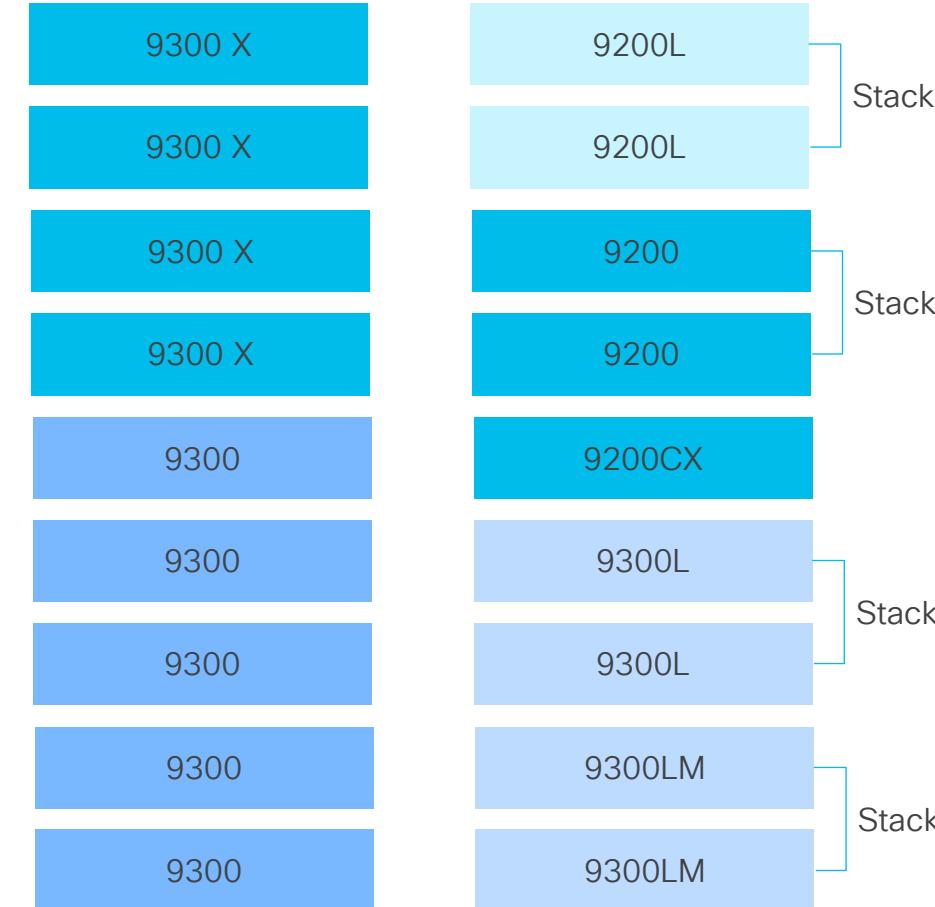
Explore YANG data models and NETCONF to query and configure IOS XE network devices. Then use Ansible to manage configurations on your IOS XE devices, covering the fundamentals of Ansible. Lastly, enable...

⌚ 6 hrs - 4 modules - 12 labs



DevNet Sandbox: Physical Reservable

- 20+ new physical reservable instances will soon be available for reservation in the DevNet Sandbox platform
- These physical devices are enabled for the API and Telemetry use cases, including the PSU, Fan, and Temperature KPI's that are otherwise not available on the virtualized platforms
- Reservations can be made for up to 1 week



dCloud Programmability



<https://dcloud.cisco.com>

"Cisco Catalyst 9000 IOS XE Programmability & Automation Lab v1"

<https://dcloud2.cisco.com/demo/catalyst-9000-ios-xe-programmability-automation-lab-v1>

Use Cases:

EVPN:

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

Model Driven Telemetry

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

YANG Programmability

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

Tooling and Integrations

YANG Suite

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

Telemetry

- TIG stack in Docker
- Grafana dashboard for device health

Postman / RESTCONF

- EVPN fabric API calls

Terraform/RESTCONF

- Declarative EVPN fabric management

Ansible

- EVPN solution enablement using CLI

Ubuntu VM Details:

Syslog receiver from all switches
TFTP config backup
See slide

Windows VM Details

VS Code

Terraform @ folder
Ansible @ folder

Chrome browser

YANG Suite, Grafana

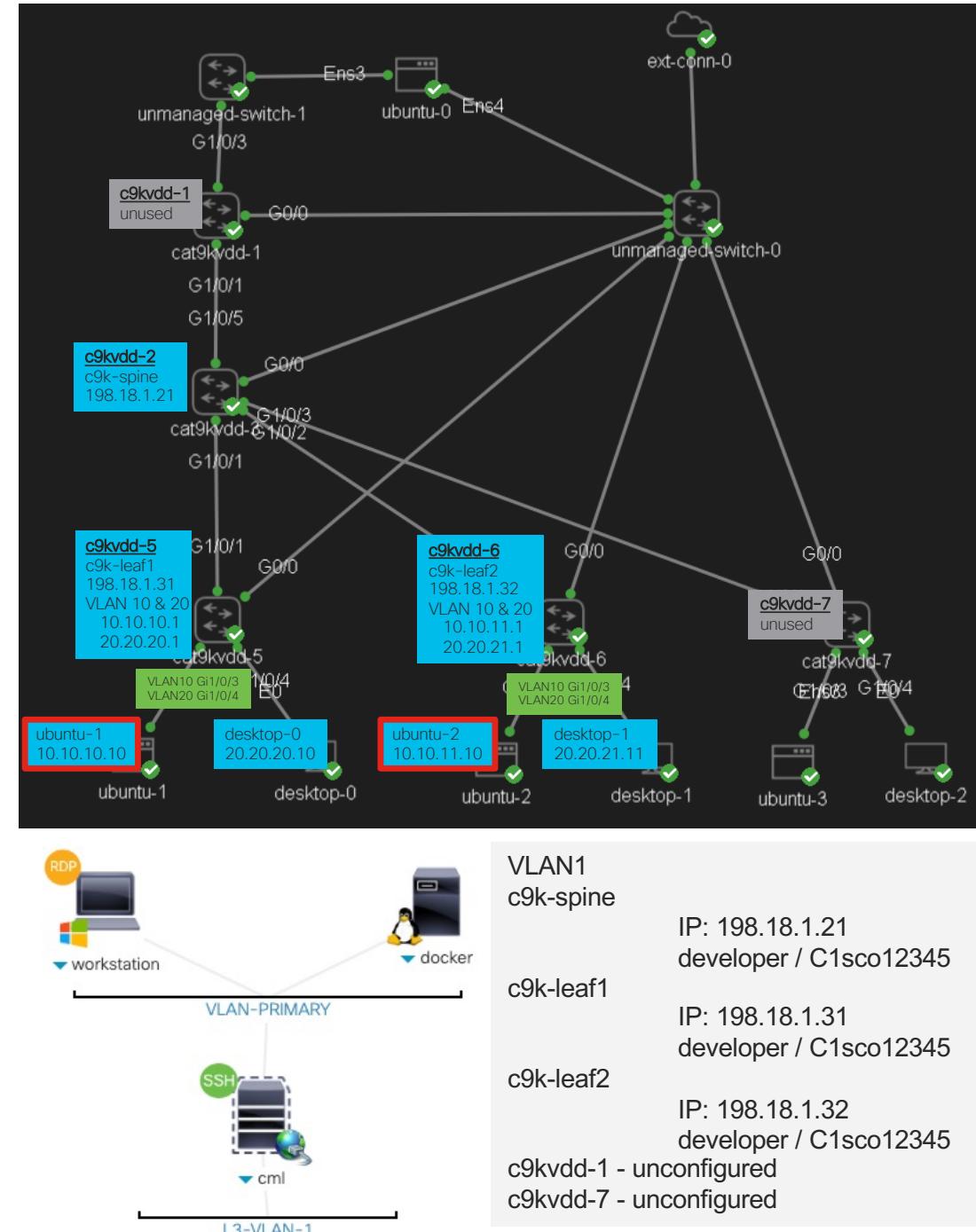
Bash/PS/Cmd shells

SSH into C9K or Ubuntu

Postman

Workspace for EVPN

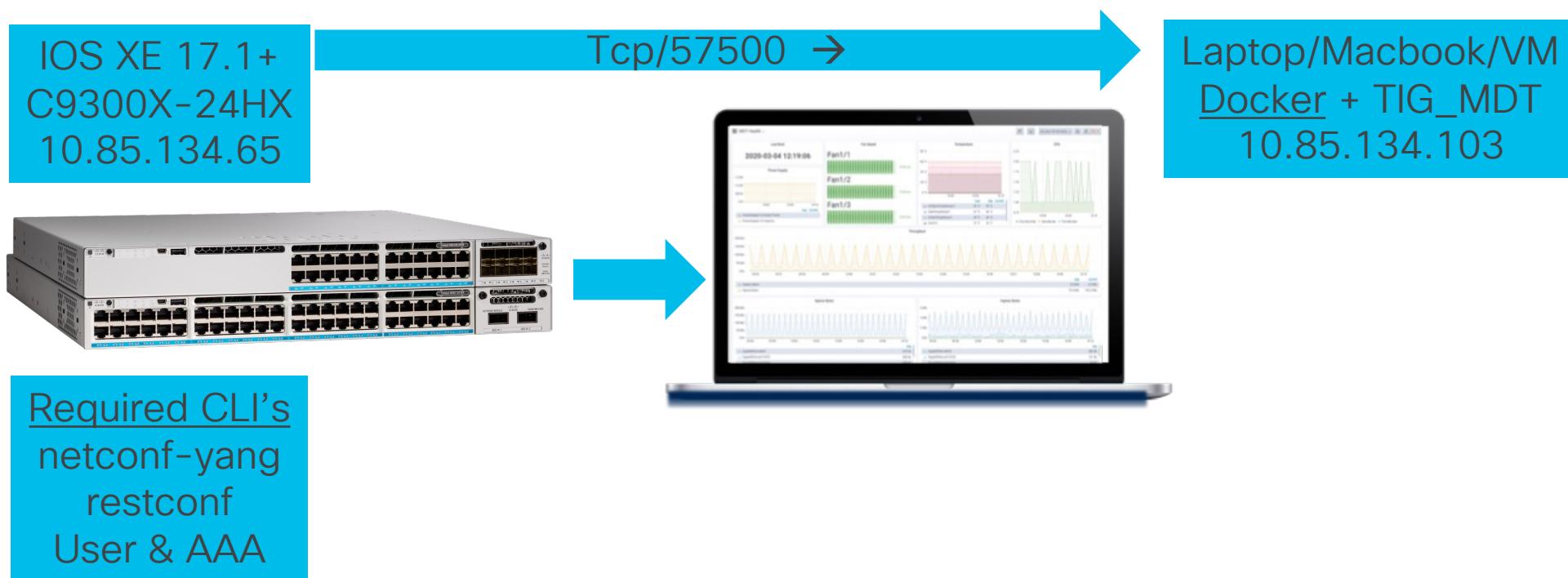
3x C9K Virtual Switch



BYOD Lab Architecture: gRPC Dial-Out

The IOS XE Switch will send telemetry data out on port 57500 to the collector across the LAN using the gRPC “Dial-Out” telemetry feature

Detailed lab guide for telemetry is available from
<https://github.com/jeremycoho/cisco-ios-xe-mdt>





Lab Demo – Reserve the dCloud environment

The screenshot shows the Cisco dCloud Catalog interface. On the left, there's a sidebar with categories like Content Producers, Content Categories, Solutions, Languages, User Devices, and Access Level. The main area shows a search result for "programmability". A specific item, "Cisco Catalyst 9000 IOS XE Programmability & Automation Lab v1", is highlighted with an orange border. This item has details such as ID: 341777, Published Date: 25-Jul-2023 06:37, and tags like Lab, Enterprise Networks, Switches, English. Below the item, there's a note about it being community-developed and a link to learn more about Community Development Services. At the bottom of the card, there are "Unfavorite" and "Related Documents" buttons, and a green "Schedule" button.

Cisco Catalyst 9000 IOS XE Programmability & Automation Lab v1

ID: 341777 Published Date: 25-Jul-2023 06:37 Lab Enterprise Networks Switches English

IMPORTANT: This content is Community Developed and therefore is not subject to standard dCloud validation or support.

Learn more about Community Development Services and how you can build your own demo by visiting the Community Development Collections pages and joining our User and Support Forum.
https://dcloud-docs.cisco.com/c/r/dcloud-docs/sites/en_us/explore/comm-dev.html

Keywords: restconf, programmability, automation, c9k, catalyst, mdt, mdp, postman, api, ztp, yang, yang suite, evpn, tig

Schedule

Schedule a Single Session

Schedule an Event (Multiple Sessions)

Reserve a dCloud instance:

[Cisco Catalyst 9000 IOS XE Programmability & Automation Lab v1](https://dcloud2-sjc.cisco.com/content/demo/341777?returnPathTitleKey=content-view)

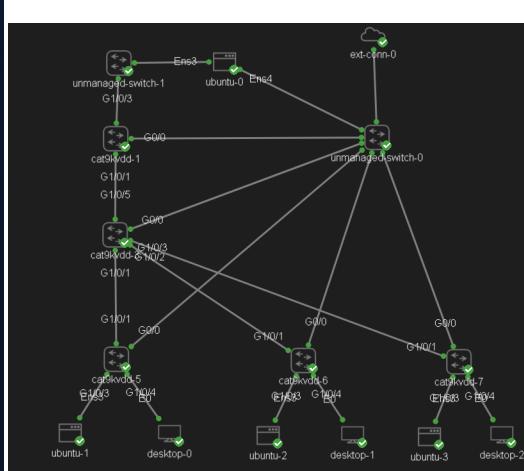
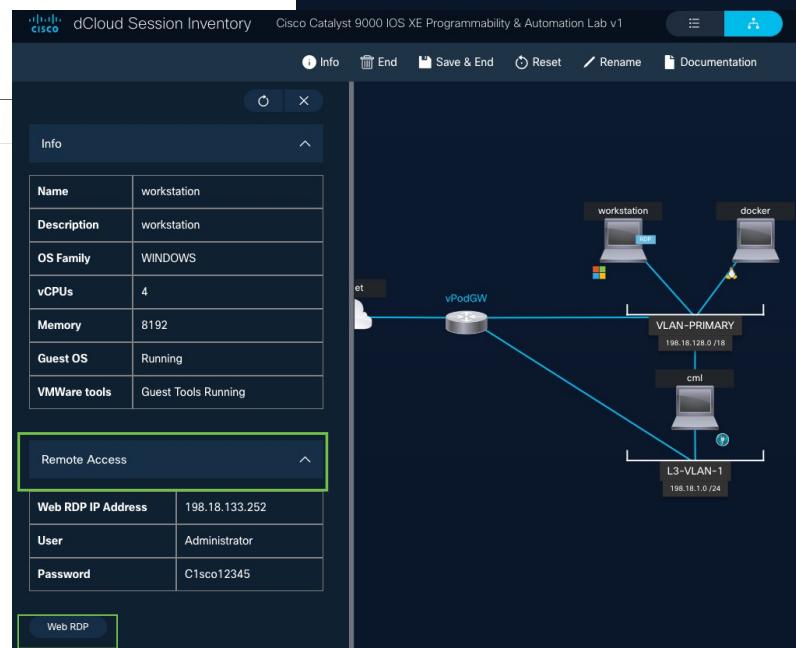
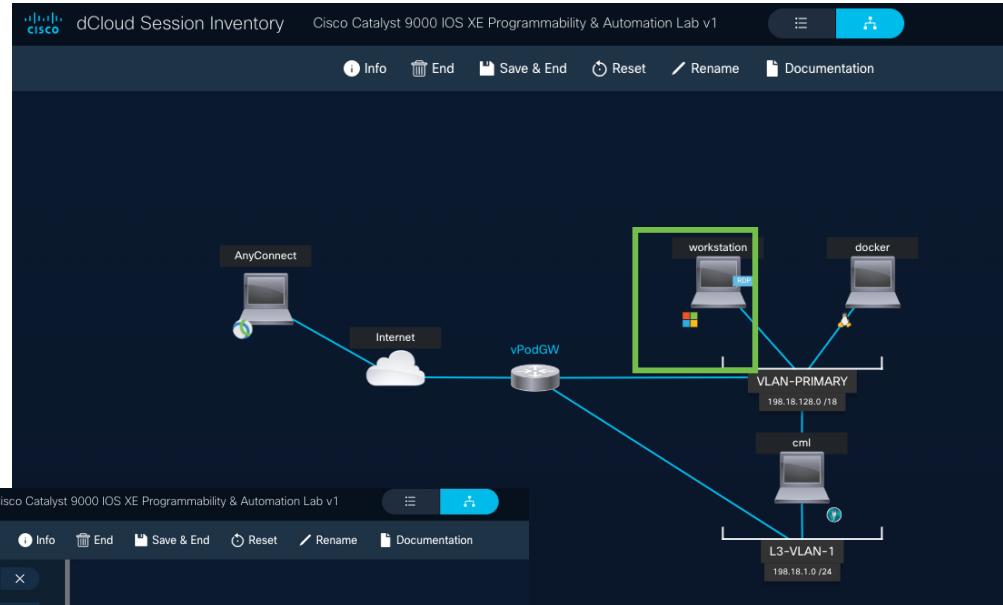
1. From dCloud, search for “programmability” in the Catalog or simply use this link:
<https://dcloud2-sjc.cisco.com/content/demo/341777?returnPathTitleKey=content-view>
2. Select the green “Schedule” button
3. Select “Schedule a Single Session”
4. In the schedule page select Primary Use as “Develop & Test” then add required details
5. Next, click the green “Schedule” button



Lab Demo – Access the dCloud environment

Once your session is up and available

1. you can access it through the dCloud > MyHub tab
 2. Select the green View button to open your session
 3. Select the workstation to get a sidebar
 4. Select “Remote Access” in the sidebar
 5. Select the “Web RDP” button
 6. Login to RDP to access YANG Suite, Grafana & more

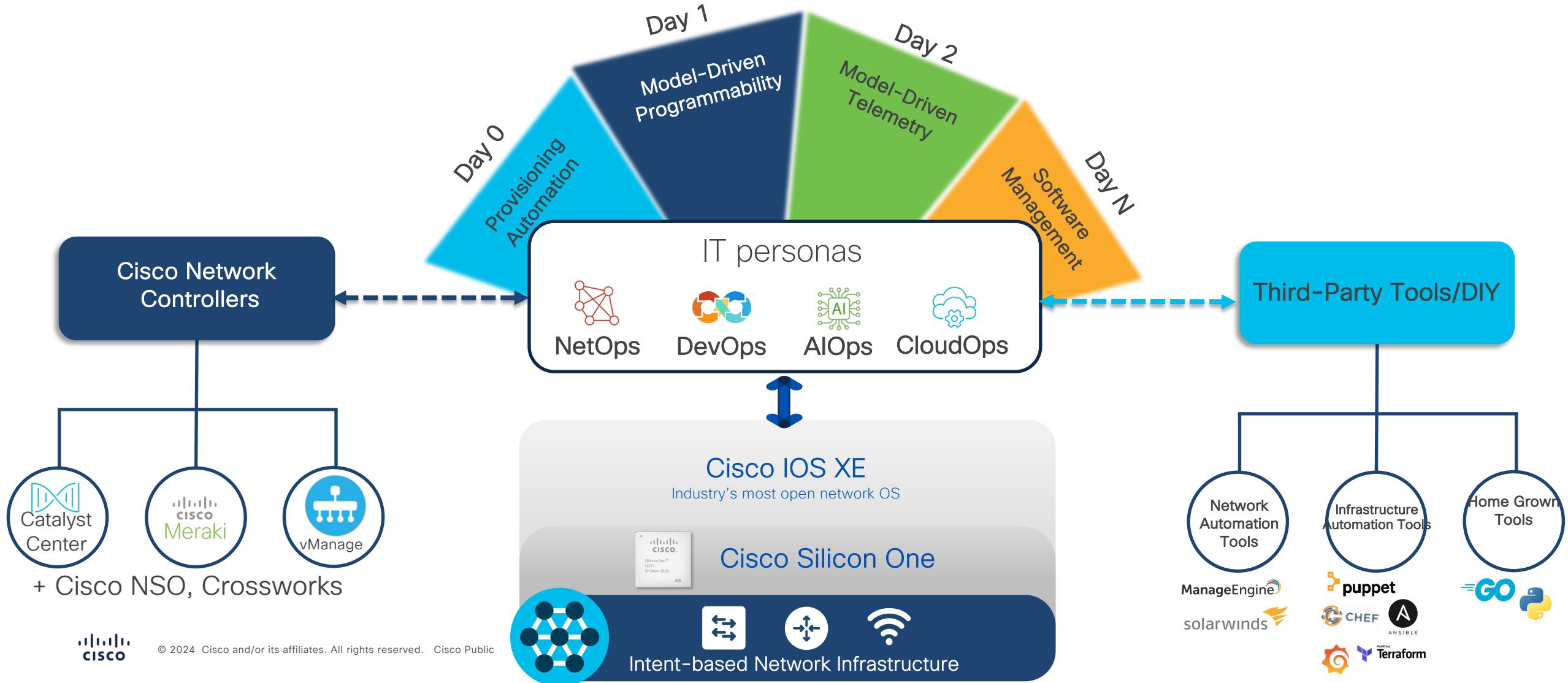


Intro to Cisco IOS XE Programmability and Telemetry



Industry's most open Network Operating System

Receptive to a wide range of management and orchestration strategies



Enterprise Network Automation

with Cisco IOS XE Programmability

Deploy. Configure. Monitor. Optimize.

Day
0

Provisioning Automation

Bring up network devices into a functional state with minimal to no-touch

Network Plug-N-Play
Secure Zero Touch Provisioning (ZTP/SZTP)
Preboot eXecution Environment (PXE)

Software Image management

Manage OS, certificates and third-party Linux applications

gNOI
Guest Shell and application hosting

Day
N

Day
1

Model-Driven Programmability

Best in class open Programmability to empower your NetOps and DevOps

NETCONF, RESTCONF and gNMI
YANG Data models – Cisco Native, IETF, OpenConfig, IEEE

Day
2

Model-Driven Telemetry

Real-time access to operational statistics

NETCONF Dial-In gNMI Dial-In & Dial-out
gRPC Dial-out
On-Change and periodic telemetry



Empower your NetOps with Cisco YANG Suite

Best in class tooling for open programmability

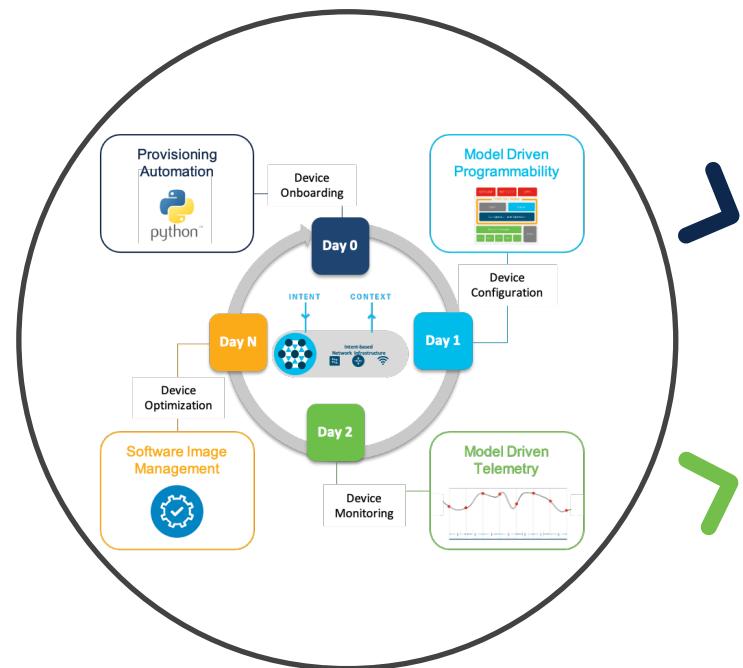
<https://developer.cisco.com/yangsuite>

Complex Network Operations

Cisco YANG Suite

Customer Benefits

Use Cases



YANG SUITE
Cisco YANG Suite



Simplified migration from
CLI and SNMP



Toolkit to integrate with
existing workflows



Guided workflow for all
things YANG



API Testing and
Validation Environment

**Legacy Protocol
Migration**

**Device
Automation**

**Network
Monitoring**

**Compliance and
Coverage**

Simplify your device programmability journey with Cisco YANG Suite

IOS XE Programmability and Telemetry API's

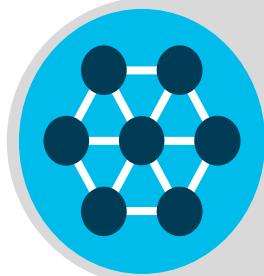
CLI

SNMP

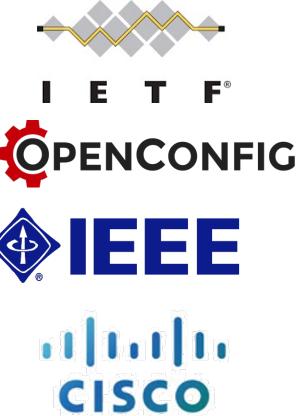
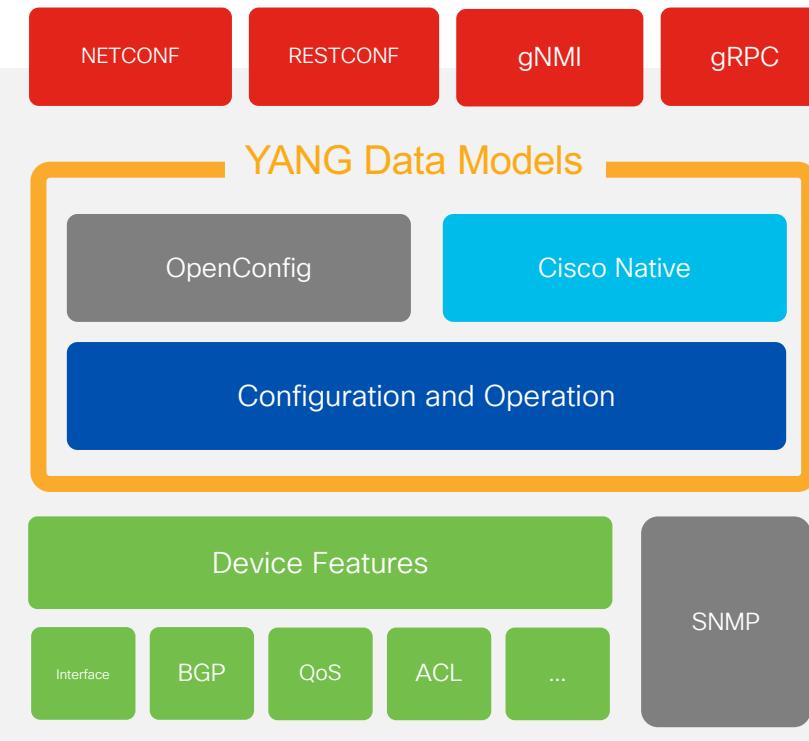
WebUI

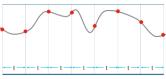
The NETCONF, RESTCONF, gNMI and gRPC are programmatic interfaces that provide additional methods for interfacing with the IOS XE device – Just like the CLI, SNMP, and WebUI is used for configuration changes and operational metrics so can the programmatic interfaces of NETCONF, RESTCONF, gNMI, and gRPC.

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



Intent-based
Network Infrastructure

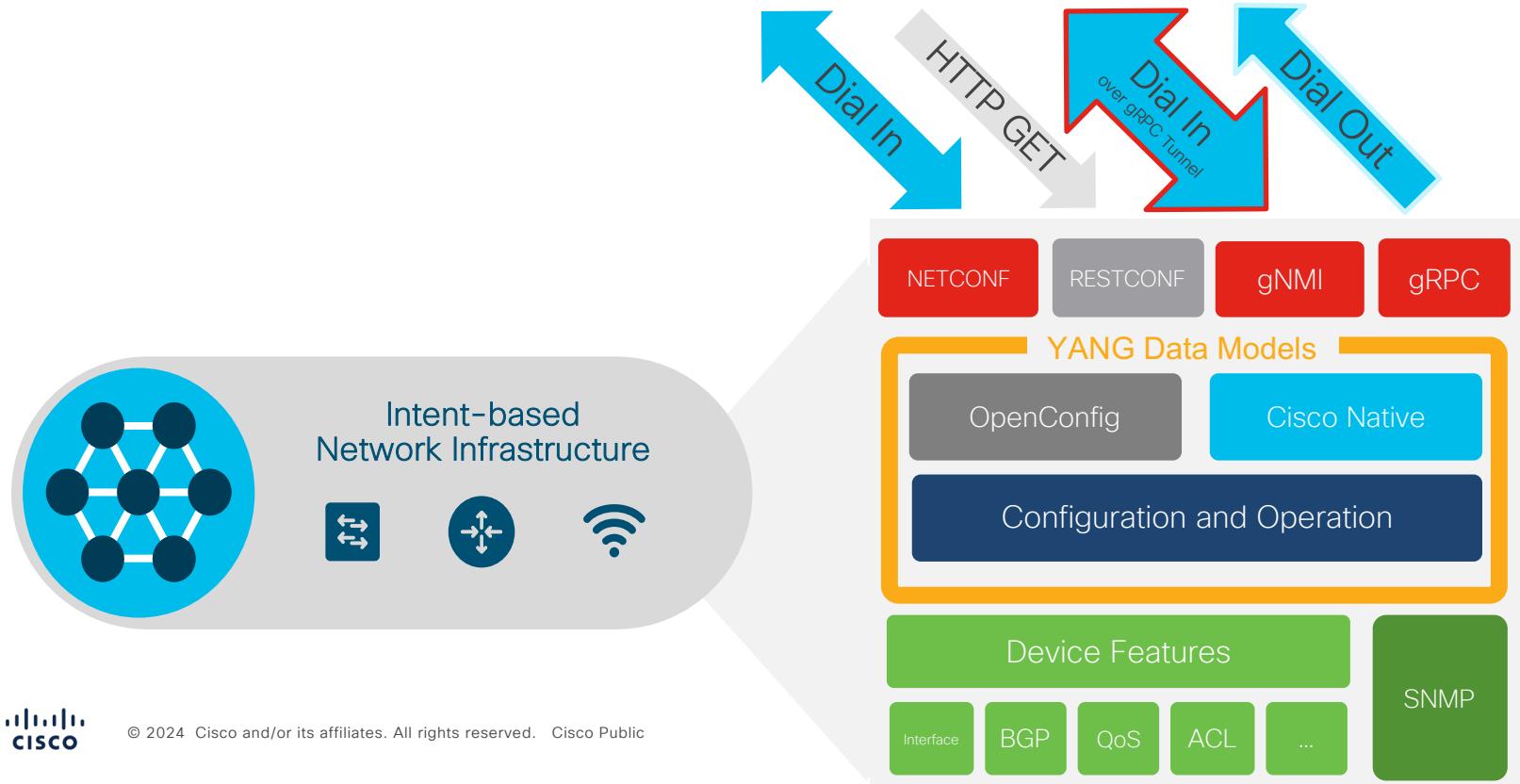


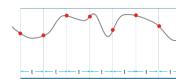


Model Driven Telemetry API's

- 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)

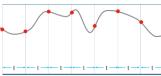
Publication / Subscription





Model Driven Telemetry Interface Comparison

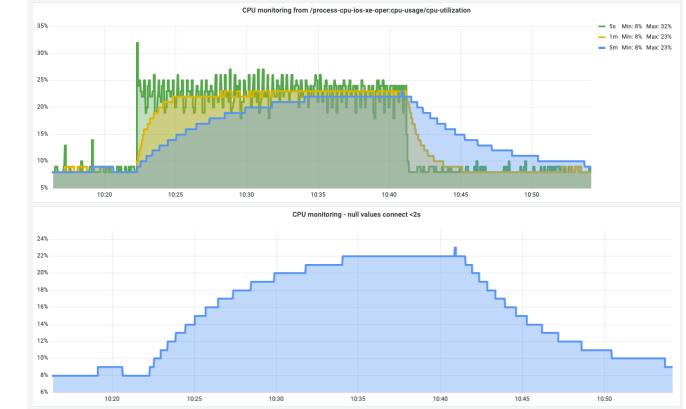
	NETCONF	gRPC (Dial-Out)	gNMI	
Minimum IOS XE Version	16.6	16.10	Dial-In: 16.12 over gRPC tunnel: 17.11	
Recommended Version	17.12	17.12	Dial-In: 17.12	
Telemetry Direction	Dial-In, IOS XE is server	Dial-Out IOS XE is client	Dial-In IOS XE is server Dial-Out gRPC Tunnel	Network architecture, security posture and policy, YANG data modules, tools and language preferences, and <u>standards</u> , and software version, are some considerations when leveraging the various MDT interfaces
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 mTLS cert only or mTLS cert + user/pass authentication	
Transport Protocol	SSH	HTTP2	HTTP2	
Data Models	YANG	YANG	YANG	



Model Driven Telemetry Usage Comparison

60-minute collection sample with 60-second update interval

Interface	CPU Impact	PCAP file size/data size (MB)	Data byte Rate	Data bit rate	Average Packet Rate (sec)	Average Packet Size (bytes)
gNMI	+3%	23 MB	6 kBps	53 kbps	5	1180
gRPC	+3%	69 MB	19 kBps	155 kbps	58	333
NETCONF	+2%	83 MB	23 kBps	185 kbps	29	780
RESTCONF	+4%	200 MB	35 kBps	281 kbps	37	945
SNMP *	+6%	120 / 87	24 kBps	197 kbps	90	273

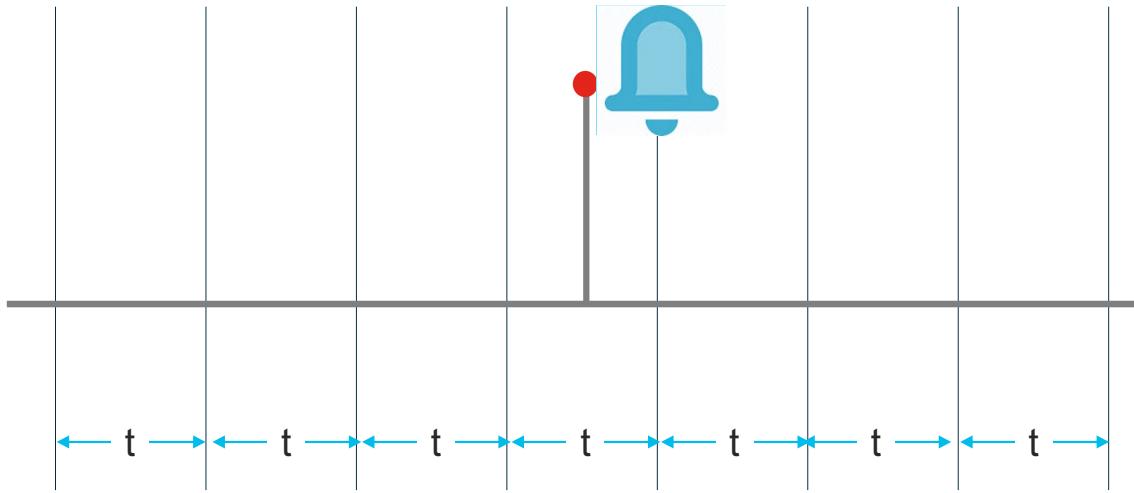


17 xpas collected at 60 second update interval
 /arp-ios-xe-oper:arp-data
 /cdp-ios-xe-oper:cdp-neighbor-details
 /environment-ios-xe-oper:environment-sensors
 /if:interfaces-state
 /interfaces-ios-xe-oper:interfaces/interface
 /ios:native
 /lldp-ios-xe-oper:lldp-entries
 /matm-ios-xe-oper:matm-oper-data
 /mdt-oper:mdt-oper-data/mdt-subscriptions
 /memory-ios-xe-oper:memory-statistics/memory-statistic
 /oc-if:interfaces/interface/state/counters
 /oc-platform:components
 /oc-sys:system
 /platform-ios-xe-oper:components
 /poe-ios-xe-oper:poe-oper-data/poe-switch
 /process-cpu-ios-xe-oper:cpu-usage/cpu-utilization
 /process-memory-ios-xe-oper:memory-usage-processes
 + Device-hardware-oper + Switch-stack-oper + more ?

This demonstrates that even when SNMP is only measuring Interfaces the load is still significantly higher than YANG which is measuring significantly more YANG data

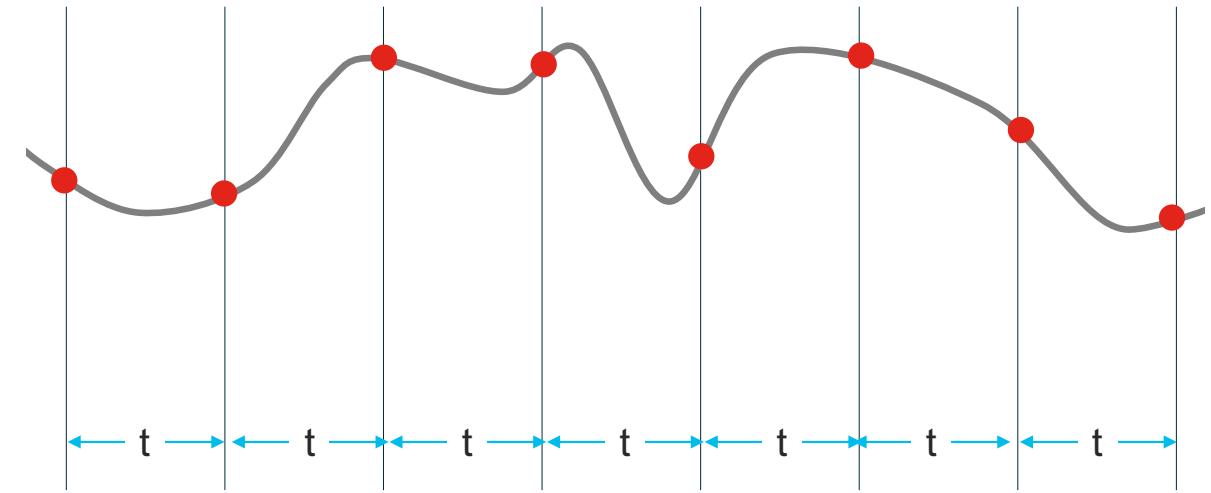
Publication options

On-Change

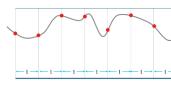


Feature Model “On-Change” Notifications
Event Notifications (failed login, optic fault, etc)
State and Configuration

Periodic



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



IOS XE Model Driven Telemetry overview

Cisco IOS XE



gNMI Dial-In/Dynamic
NETCONF Dial-In

↑ ↓ gRPC Dial-Out/Configured

Collector/Receiver
Decodes to text



splunk®

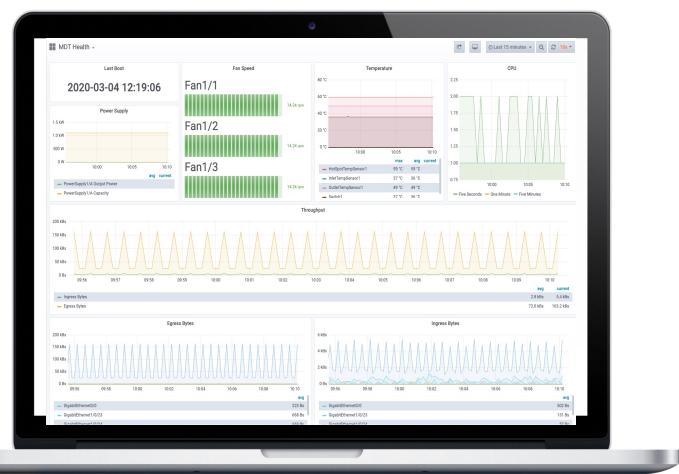


elastic

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_ietf_telemetry.html



Grafana Example Dashboards

Example dashboards for Switch Health, C9800 Client Stats, and POE

<https://grafana.com/grafana/dashboards/13462>

<https://grafana.com/grafana/dashboards/12468>

<https://grafana.com/grafana/dashboards/17238-catalyst-poe-dashboard/>

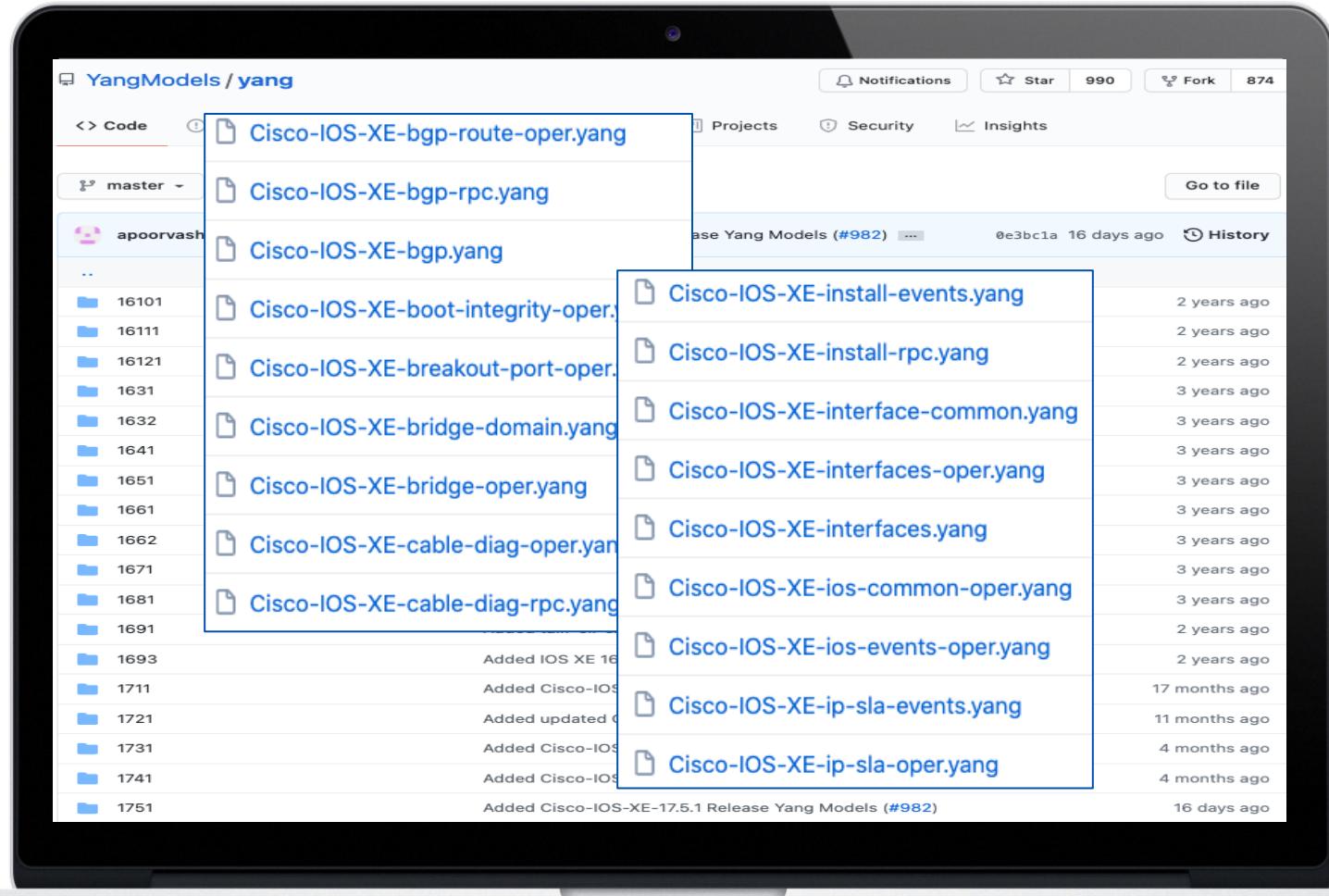


Cisco IOS XE - YANG models on GitHub



- 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>



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 screenshot displays two main windows of the Cisco YANG Suite:

- YANG Suite - Explore YANG Models:** Shows the YANG model "Cisco-IOS-XE-interfaces-oper". The left sidebar lists "Admin", "Setup", "Explore", "Protocols", and "Help". The right pane shows the schema structure for "interface" nodes, including "name", "interface-type", "admin-status", "oper-status", "last-change", and "if-index". A "Node Properties" panel on the right provides details about the "statistics" node.
- YANG Suite - NETCONF:** Shows the NETCONF interface for the "Cisco-IOS-XE-interfaces-oper" module. It includes tabs for "NETCONF Operation" (set to "get"), "Device" (set to "JCOHOE-DMZ-C9300"), and "RPC Options...". The "Nodes" pane shows the structure of the "Cisco-IOS-XE-interfaces-oper" module, specifically the "interface" node with its attributes. To the right, the XML representation of the RPC message is shown:

```
<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>
```

Now Available !

developer.cisco.com/yangsuite

github.com/CiscoDevNet/yangsuite

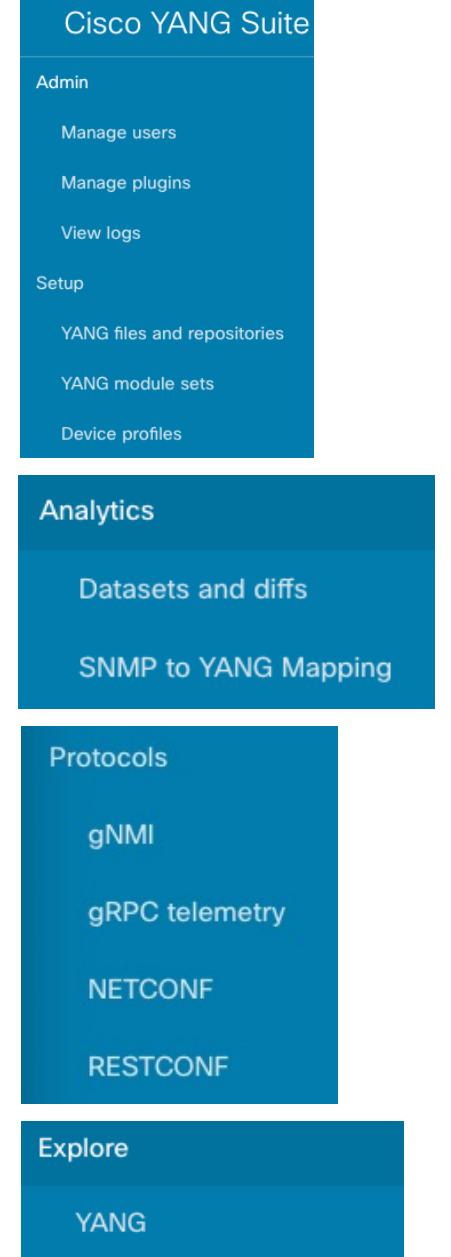


YANG Suite: Whats Included

- 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



YANG Suite: Docker quick-start

1. git clone https://github.com/CiscoDevNet/yangsuite
2. Build certificates: cd yangsuite/docker/ ; ./gen_test_certs.sh
3. docker-compose up

```
jcohoe@JCOHOE-M-C6NA docker % docker-compose up --build
Building yangsuite
Step 1/19 : FROM ubuntu:18.04
--> 2c047404e52d
Step 2/19 : ENV DOCKER_RUN true
--> Using cache
--> 3b8e0efd0cfcd
Step 3/19 : ARG PY=python3.6
```

2. Access the tool at https://localhost

Cisco YANG Suite User Agreement

YANG Suite users must agree to the "Cisco End User License Agreement" and "Privacy Statement".

Choose to accept or decline "Cisco End User License Agreement".

Decline Accept

Choose to accept or decline "Cisco Online Privacy Statement".

Decline Accept

https://www.cisco.com/c/en/us/about/legal/cloud-and-software/end_user_license_agreement.html
<https://www.cisco.com/c/en/us/about/legal/privacy-full.html>



3. Review and accept license and privacy policy

The screenshot shows the Cisco End User License Agreement (EULA) page. At the top, there's a navigation bar with links for 'MENU', 'cisco' logo, 'Software Contracting Basics /', and search/filter options. The main content area is titled 'End User License Agreement'. It discusses the simplification of licensing documents and the availability of the Cisco EULA in various languages. Below this, there's a 'Table of Contents' section listing 13 sections from 'Scope and Applicability' to 'Definitions'. To the right, there's a 'Related Links' sidebar with links to 'Product Specific Terms', 'Software License Portability Policy', 'Software License Transfer and Re-Use Policy', 'Online Privacy Statement', and 'Master Data Protection Agreement'.

4. Login to the tool at http://localhost

Default credentials in the Dockerfile
admin : superuser

Log in to YANG Suite

Please login to access this page.

Username: [6]

Password: [6]

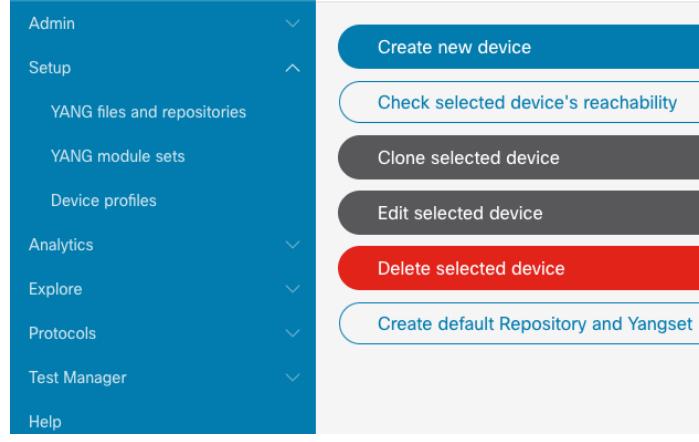
[Lost your password?](#)

YANG Suite + Telemetry

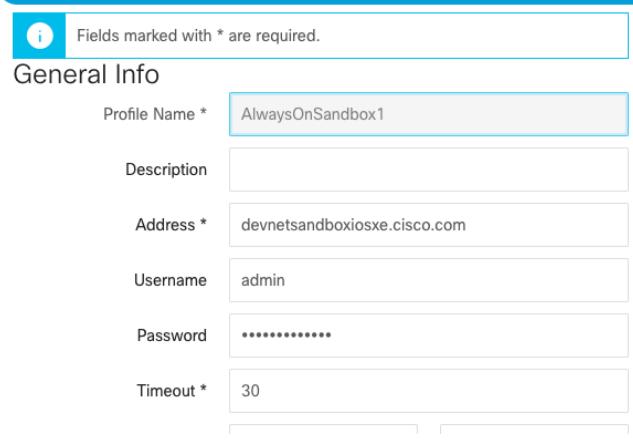
Use the AlwaysOn Sandbox with YANG Suite

Within YANG Suite

1. Add a new device by navigating to Setup > Device Profiles
2. Click the blue “Create” new device button and add the following for the always on sandbox
 1. Profile Name: AlwaysOn
 2. Address: devnetsandbox.cisco.com
 3. Username: admin
 4. Password: C1sco12345
3. Ensure the checkboxes for NETCONF and gNIM are selected. Also click “Skip SSH key validation for this device when applicable



The screenshot shows the YANG Suite interface with a sidebar containing 'Admin', 'Setup', 'YANG files and repositories', 'YANG module sets', 'Device profiles', 'Analytics', 'Explore', 'Protocols', 'Test Manager', and 'Help'. On the right, there is a list of actions: 'Create new device' (blue button), 'Check selected device's reachability', 'Clone selected device', 'Edit selected device', 'Delete selected device' (red button), and 'Create default Repository and Yangset'. Below this is a 'New Device Profile' dialog box for 'gNMI'. It includes fields for 'Platform *' (selected as 'IOS XE'), 'gNMI insecure port *' (50052), 'gNMI secure port *' (9339), and a checkbox for 'Use TLS Certificate'. A note at the top of the dialog says 'Device supports gNMI'.



The screenshot shows the 'General Info' configuration page. It includes fields for 'Profile Name *' (AlwaysOnSandbox), 'Description', 'Address *' (devnetsandboxiosxe.cisco.com), 'Username' (admin), 'Password' (redacted), and 'Timeout *' (30). A note at the top states 'Fields marked with * are required.'



The screenshot shows the 'Connectivity check results' page for a device named 'c930...'. It displays a summary: 'Connectivity check results:' followed by a list of checked items: ping, gNMI, NETCONF, and RESTCONF.

NETCONF Demo in dCloud YANG Suite

NETCONF Get for Cisco-IOS-XE-platform-oper

The screenshot shows the Cisco YANG Suite interface. On the left, a sidebar menu includes Admin, Setup, Analytics, Explore, Protocols (gNMI, gRPC telemetry), NETCONF, RESTCONF, Test Manager, and Help. The NETCONF section is selected. The main area is titled 'NETCONF' and shows 'YANG Set' as '198-18-1-21-default-yangset'. A dropdown for 'Module(s)' has 'Cisco-IOS-XE-platform-oper.x' selected. Below this, 'NETCONF Operation' is set to 'get', and 'Device' is set to '198.18.1.21'. Buttons for 'Edit Device', 'Open Device Window', 'Run RPC(s)', and 'Clear RPC(s)' are present. A 'Build RPC' button is also visible. The central part of the interface displays a YANG tree for 'Cisco-IOS-XE-platform-oper'. The 'components' node is expanded, showing 'component' under it. The 'component' node has 'cname' and 'state' children. Under 'platform-properties', there is a 'platform-property' node with 'name', 'value', and 'configurable' children. To the right of the tree, there is a 'Value' input field with three checkboxes checked. Below the tree, an XML representation of the get RPC message is shown:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
<get>
<filter>
<components xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-platform-oper">
<component>
< cname >Switch1</ cname >
</ component >
< component >
< cname >c3xxx Stack</ cname >
</ component >
< component >
< cname >FixedModule1/0</ cname >
</ component >
< component >
< cname >GigabitEthernetContainer1/0/1</ cname >
</ component >
< component >
< cname >GigabitEthernetContainer1/0/2</ cname >
</ component >
</ components >
</ filter >
</ get >
</ rpc >
```

Results from the API call:

```
<data>
<components xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-platform-oper">
<component>
< cname >Switch1</ cname >
</ component >
< component >
< cname >c3xxx Stack</ cname >
</ component >
< component >
< cname >FixedModule1/0</ cname >
</ component >
< component >
< cname >GigabitEthernetContainer1/0/1</ cname >
</ component >
< component >
< cname >GigabitEthernetContainer1/0/2</ cname >
</ component >
</ components >
</ data >
```

Equivalent gRPC Dial-Out telemetry Xpath:
filter xpath /platform-ios-xe-oper:components/component/platform-properties/platform-property

gNMI Demo in dCloud YANG Suite

GET the platform properties using gNMI

Cisco YANG Suite

YANG Suite / gNMI / YANG set " / Modules

gRPC Network Management Interface (gNMI)

admin

YANG Set: c9300x-default-yangset | Module(s): openconfig-platform x Cisco-IOS-XE-platform-oper x | Load Module(s)

Device: c9300x | Edit Device | Capabilities | gNMI Operation | Get | Set | Subscribe | Prefix path

GET type: All | Config | State | Operational | Openconfig | RFC 7951 | Module name | Other | Encoding type: JSON_IETF | JSON | PROTO | ASCII

Search XPaths | Legend | Replays | Build RPC | Clear Values | Run RPC(s) | Clear RPC(s)

Nodes

Value	Operation
openconfig-platform	
Cisco-IOS-XE-platform-oper	
components	✓
component	✓
cname	*
state	
platform-properties	✓
platform-property	✓
name	*
value	
configurable	
platform-subcomponents	

Results from the API call:

JSON Decoded

```
=====
{
  "name": "Switch1 Hotspot Yellow Threshold",
  "value": {
    "string": "105"
  },
  "configurable": false
}
```

Model Driven Telemetry – gRPC explained

- YANG Suite's gRPC Telemetry plugin can be used to receive and validate the telemetry data
- Instead of using YANG Suite the Telegraf/Influx/Grafana solution can also be used to better visualize and store the data

The screenshot shows the YANG Suite / gRPC Telemetry interface. At the top, it says "gRPC Telemetry". Below that, there are two input fields: "Listen at IP address" (127.0.0.1) and "Listen at port" (57344). To the right of these fields are two buttons: "Stop telemetry receiver" (highlighted in red) and "Clear output".

The main area displays configuration details for a telemetry subscription:
Subscription ID: 202
Type: Configured
State: Valid
Stream: yang-push
Filter: Filter type: xpath
XPath: /process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds
Update policy:
 Update Trigger: periodic
 Period: 1000
Encoding: encode-kvpgb
Source VRF:
Source Address:
Notes:

Legacy Receivers:

Address	Port	Protocol	Protocol Profile
10.1.1.3	57344	grpc-tcp	

The telemetry is configured and validated on the C9300 with sample configuration below

```
C9300#conf t
Enter configuration commands, one per line. End with CNTL/Z.
C9300(config)#telemetry ietf subscription 202
C9300(config-mdt-subs)# encoding encode-kvpgb
C9300(config-mdt-subs)# filter xpath /process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds
C9300(config-mdt-subs)# stream yang-push
C9300(config-mdt-subs)# update-policy periodic 1000
C9300(config-mdt-subs)# receiver ip address 10.1.1.3 57344 protocol grpc-tcp
C9300#show telemetry ietf subscription 202 detail
Telemetry subscription detail:

Subscription ID: 202
Type: Configured
State: Valid
Stream: yang-push
Filter:
  Filter type: xpath
  XPath: /process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds
Update policy:
  Update Trigger: periodic
  Period: 1000
Encoding: encode-kvpgb
Source VRF:
Source Address:
Notes:

Legacy Receivers:
  Address          Port    Protocol   Protocol Profile
  -----           -----
  10.1.1.3        57344   grpc-tcp
```

```
telemetry ietf subscription 202
encoding encode-kvpgb
filter xpath /process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds
stream yang-push
update-policy periodic 1000
receiver ip address 10.1.1.3 57344 protocol grpc-tcp
```

Telegraf
InfluxDB
Grafana

Docker Container Workshop



Pre-built TIG_MDT Docker container

Upgrade coming to Telegraf, Influx, and Grafana Model Driven Telemetry (TIG_MDT) Docker container

Making it easier to consume telemetry in production

Upgraded Telegraf, InfluxDB, and Grafana tools

Additional dashboards for

Device Health, Wireless Client, Wireless AP, RF etc

Examples for device CLI configuration for telemetry

Details of scale and data storage requirements

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

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_ietf_telemetry.html

docker pull jeremycohoe/tig_mdt

Download the TIG_MDT container from DockerHub that has the pre-configured tools available

```
auto@pod8-xelab:~$  
auto@pod8-xelab:~$ docker pull jeremycohoe/tig_mdt  
Using default tag: latest  
latest: Pulling from jeremycohoe/tig_mdt  
74ac377868f8: Pull complete  
65f7b0cb16ee: Pull complete  
f03ca7828856: Pull complete  
19de70153a07: Pull complete  
6c96311a8b51: Pull complete  
5e91a562b510: Pull complete  
d46ee566c83f: Pull complete  
0b775b94c0bf: Pull complete  
65ef77d4be31: Pull complete  
Digest: sha256:16136b521fc97cf417d87479a4d47c1b87fbdac44616e543c3ddbd817c65b49  
Status: Downloaded newer image for jeremycohoe/tig_mdt:latest  
docker.io/jeremycohoe/tig_mdt:latest  
auto@pod8-xelab:~$ docker run -ti -p 3000:3000 -p 57500:57500 jeremycohoe/tig_mdt  
Startin TIG_MDT docker container for Cisco IOS XE elemetry at Fri Apr 19 16:59:18 UTC 2024
```

```
docker pull jeremycohoe/tig_mdt  
docker run -ti -p 3000:3000 -p 57500:57500 jeremycohoe/tig_mdt  
or start detatched (-d)  
docker run -dti -p 3000:3000 -p 57500:57500 jeremycohoe/tig_mdt
```

Validate docker container

The telegraf, influxdb, and Grafana processes are running inside the Docker container

```
$ docker ps  
$ docker exec -it <containerID> bash
```

Check running processes: ps xa

```
auto@pod8-xelab:~$ docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS  
a5d40e8f9c55 jeremycohoe/tig_mdt "/start -d" 38 minutes ago Up 38 minutes 0.0.0.0:3000->3000/tcp, :::3000->3000/tcp, 0.0.0.0:57500->57500/tcp, ::  
auto@pod8-xelab:~$  
auto@pod8-xelab:~$ docker exec -it a5d40e8f9c55 bash  
root@a5d40e8f9c55:/#  
root@a5d40e8f9c55:/# ps xa  
 PID TTY      STAT   TIME COMMAND  
  1 pts/0    Ss+   0:00 /bin/bash /start -d  
  9 pts/0    S+    0:00 /usr/bin/sleep infinity  
 19 pts/0    Sl+   0:05 /usr/bin/influxd -pidfile /var/run/influxdb/influxd.pid -config /etc/influxdb/influxdb.conf  
 52 ?        Sl    0:03 /usr/share/grafana/bin/grafana server --pidfile=/var/run/grafana-server.pid --config=/etc/grafana/grafana.ini --packaging=deb  
 69 pts/0    SLl+  0:02 /usr/bin/telegraf --config /etc/telegraf/telegraf.conf  
109 pts/1    Ss    0:00 bash  
121 pts/1    R+    0:00 ps xa  
root@a5d40e8f9c55:/#
```

Send telemetry from switch to collector

```
cat9300x-pod08a#sh telemetry ietf subscription 2024001 receiver  
Telemetry subscription receivers detail:
```

```
Subscription ID: 2024001  
Name: grpc-tcp://10.1.1.3:57500  
Connection: 25  
State: Connected  
Explanation:  
Last Error: Transport lost  
Target State: Connected
```

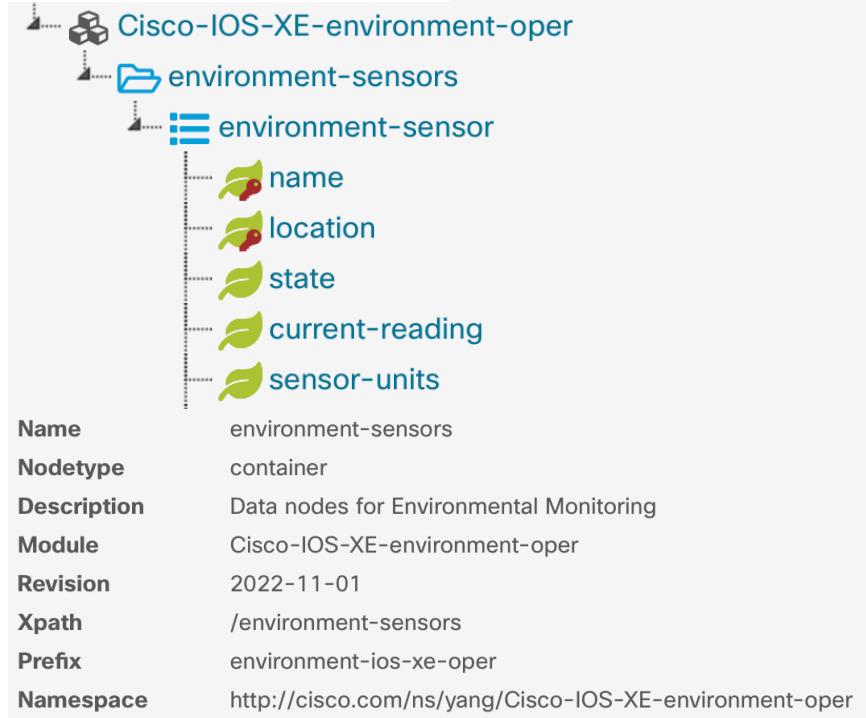
```
cat9300x-pod08a#sh telemetry ietf subscription 2024001 de  
cat9300x-pod08a#sh telemetry ietf subscription 2024001 detail  
Telemetry subscription detail:
```

```
Subscription ID: 2024001  
Type: Configured  
State: Valid  
Stream: yang-push  
Filter:  
  Filter type: xpath  
  XPath: /environment-sensors  
Update policy:  
  Update Trigger: periodic  
  Period: 60000  
Encoding: encode-kvpgpb  
Source VRF:  
Source Address:  
Notes: Subscription validated
```

Named Receivers:	Name	Last State Change	State	Explanation
	grpc-tcp://10.1.1.3:57500	04/19/24 09:59:34	Connected	

```
telemetry ietf subscription 2024001  
encoding encode-kvpgpb  
filter xpath /environment-sensors  
stream yang-push  
update-policy periodic 60000  
receiver ip address 10.1.1.3 57500 protocol grpc-tcp
```

```
sh telemetry ietf subscription 2024001 receiver  
sh telemetry ietf subscription 2024001 detail
```



Validate data is received in the container

Review the raw telemetry data that has been received in the container by the telegraf tool

```
# tail /tmp/telegraf-grpc.log
```

```
root@a5d40e8f9c55:~# tail /tmp/telegraf-grpc.log
Compiled Thu 21-Mar-24 14:20 by IOSXE-Group",state/serial_no="NULL",state/part_no="NULL",state/removable=false,state/oper
te/allocated_power=0i,state/used_power=0i 1713548389810000000
openconfig-platform:components/component,host=jcohoe-tigmdt,name=TenGigabitEthernet1/0/47,path=openconfig-platform:compon
nfig/name="TenGigabitEthernet1/0/47",state/name="TenGigabitEthernet1/0/47",state/type="PORT",state/id="1108",state/locati
on="NULL",state/firmware_version="NULL",state/software_version="Cisco IOS Software [IOSXE]", Catalyst L3 Switch Software
IOSXE-Group
Copyright (c) 1986-2024 by Cisco Systems, Inc.
Compiled Thu 21-Mar-24 14:20 by IOSXE-Group",state/serial_no="NULL",state/part_no="NULL",state/removable=false,state/oper
te/allocated_power=0i,state/used_power=0i 1713548389810000000
openconfig-platform:components/component,host=jcohoe-tigmdt,name=TenGigabitEthernet1/0/48,path=openconfig-platform:compon
nfig/name="TenGigabitEthernet1/0/48",state/name="TenGigabitEthernet1/0/48",state/type="PORT",state/id="1109",state/locati
on="NULL",state/firmware_version="NULL",state/software_version="Cisco IOS Software [IOSXE]", Catalyst L3 Switch Software
IOSXE-Group
Copyright (c) 1986-2024 by Cisco Systems, Inc.
Compiled Thu 21-Mar-24 14:20 by IOSXE-Group",state/serial_no="NULL",state/part_no="NULL",state/removable=false,state/oper
te/allocated_power=0i,state/used_power=0i 1713548389810000000
Cisco-IOS-XE-process-cpu-oper:cpu-usage/cpu-utilization,host=jcohoe-tigmdt,path=Cisco-IOS-XE-process-cpu-oper:cpu-usage/c
pu_usage/average_seconds=0i 1713548674528000000
```

If data is not received ?

Q. Is the gRPC packet from the IOS XE switch reaching to the TIG_MDT collector ?

A. Collect a packet capture on the host or within the Docker container on the TCP port 57500 to confirm

```
# apt install tcpdump  
# tcpdump -i any port 57500
```

```
root@02a59b6c5b43:~# tcpdump -i any port 57500  
tcpdump: data link type LINUX_SLL2  
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode  
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes  
09:09:30.697628 eth0  Out IP 02a59b6c5b43.57500 > 10.1.1.5.58983: Flags [.], ack 1796663049,  
09:09:30.698053 eth0  In  IP 10.1.1.5.58983 > 02a59b6c5b43.57500: Flags [.], ack 1, win 502,  
^C  
2 packets captured  
2 packets received by filter  
0 packets dropped by kernel  
root@02a59b6c5b43:~# █
```

If packet is not seen, check firewall, ACL, routing.

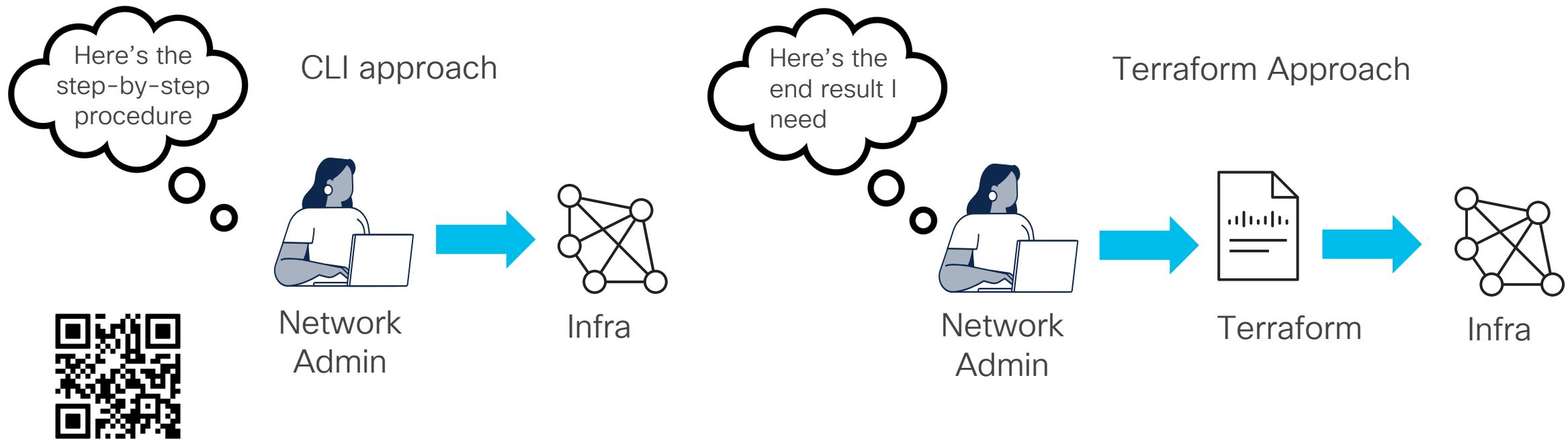
Ensure IOS XE device is configured correctly to send from the correct source-address and source-vrf

Telemetry Subscription Management

with Terraform

What's Terraform?

Terraform is a declarative Infrastructure as Code (IaC) cloud-native tooling. It is the process of managing changes through code, rather than a manual process. This means, we can provide the end result rather than describing step-by-step which changes to make as we would with CLI



Learn more about IaC here:

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



Terraform resource: iosxe_mdt_subscription

Use Terraform to manage the gRPC Dial-Out Telemetry subscriptions

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

Example files: <https://github.com/jeremycohoe/cisco-ios-xe-mdt/tree/master/sustainability>

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-kvvpb"
  source_vrf           = "Mgmt-vrf"
  source_address        = "1.2.3.4"
  update_policy_on_change = true
  filter_xpath          = "/ios-events-ios-xe-oper:ospf-neighbor-state-ch
receivers = [
  {
    address  = "5.6.7.8"
    port     = 57600
    protocol = "grpc-tcp"
  }
]
```

Copy

```
terraform {
  required_providers {
    iosxe = {
      source = "CiscoDevNet/iosxe"
      version = "0.5.5"
    }
  }
}

provider "iosxe" {
  username = var.host_username
  password = var.host_password
  url      = var.host_url
}

variable host_username {
  type = string
  default = "admin"
}
```

```
iosxe_mdt_subscription.example["2024005"]: Modifying... [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024005]
iosxe_mdt_subscription.example["2024001": Modifying... [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024001]
iosxe_mdt_subscription.example["2024002": Modifying... [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024002]
iosxe_mdt_subscription.example["2024003": Modifying... [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024003]
iosxe_mdt_subscription.example["2024007": Modifying... [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024007]
iosxe_mdt_subscription.example["2024008": Modifying... [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024008]
iosxe_mdt_subscription.example["2024004": Modifying... [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024004]
iosxe_mdt_subscription.example["2024006": Modifying... [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024006]
iosxe_mdt_subscription.example["2024001": Modifications complete after 0s [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024001]
iosxe_mdt_subscription.example["2024005": Modifications complete after 0s [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024005]
iosxe_mdt_subscription.example["2024006": Modifications complete after 0s [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024006]
iosxe_mdt_subscription.example["2024003": Modifications complete after 0s [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024003]
iosxe_mdt_subscription.example["2024007": Modifications complete after 0s [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024007]
iosxe_mdt_subscription.example["2024002": Modifications complete after 0s [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024002]
iosxe_mdt_subscription.example["2024004": Modifications complete after 1s [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024004]
iosxe_mdt_subscription.example["2024008": Modifications complete after 1s [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024008]
```

```
Apply complete! Resources: 0 added, 8 changed, 0 destroyed.
auto@pod29-xelab:~/terraform/sustainability$ terraform apply
```



Terraform for gRPC Dial-Out telemetry management

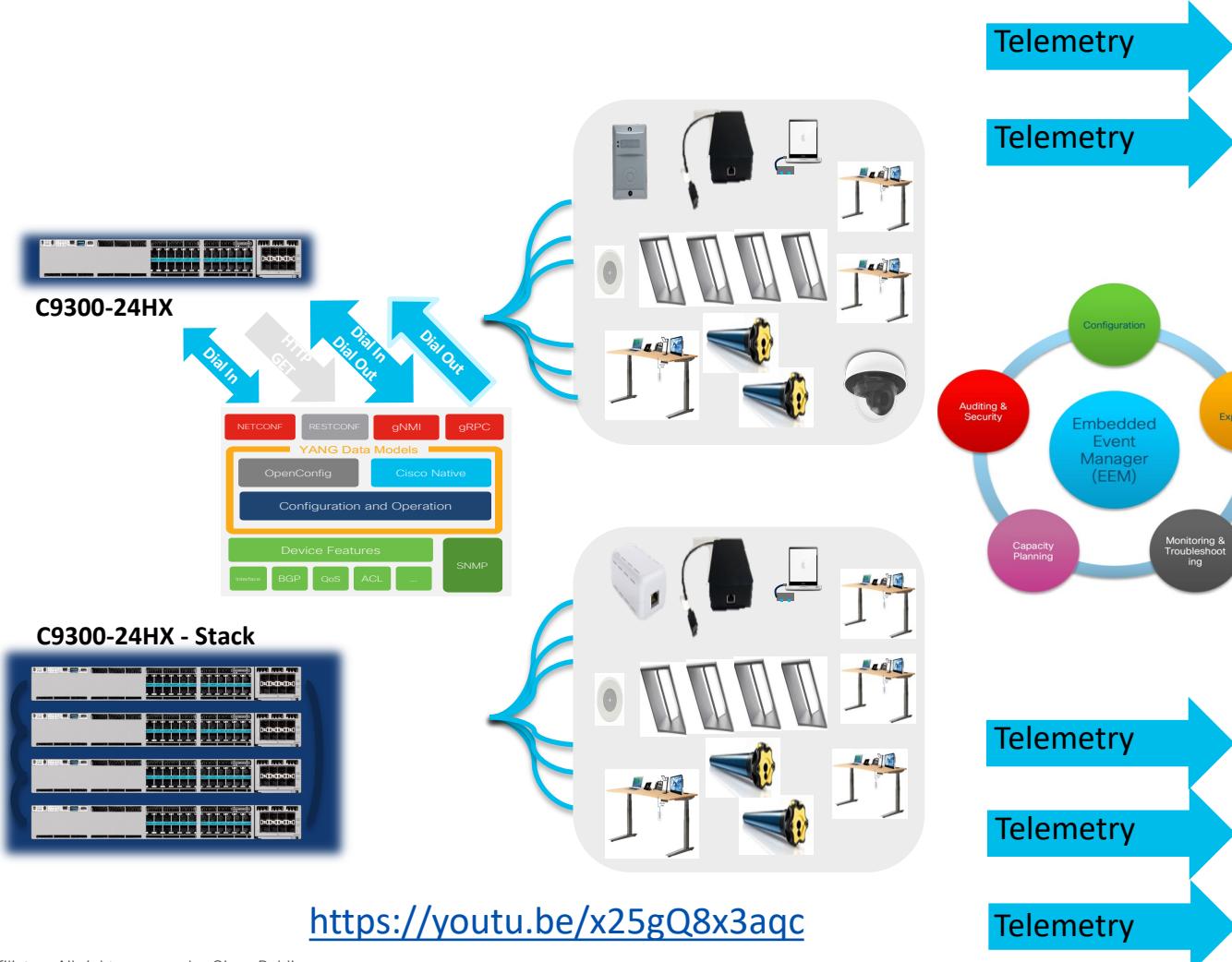
1. Modify variables to match your source and destination
 2. Run “terraform init” to ensure your terraform is updated and will use the proper providers
 3. Run “terraform plan” to confirm the config to add is correct
 4. Run “terraform apply” to modify the device config
 5. Validate that the correct config was added to your device

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** On the left, it shows two files: "sustainability.tf" and "terraform.tfvars".
- Code Editor:** The main area displays the "sustainability.tf" file content. The code defines a provider "iosxe" with source "CiscoDevNet/iosxe" and version "0.5.5". It also defines a variable "host_username" of type string with default value "admin" and description "The username for the Cisco IOS XE device to configure".
- Terminal:** Below the editor, the terminal window shows repeated entries from a session on a Cisco device named "tme-yangsuite". Each entry starts with "tme@tme-yangsuite:~/terraform/sustainability\$".
- Output:** A sidebar on the right lists several log entries, all starting with "VNC2-BORDER1-X#".

Sustainability Use Cases

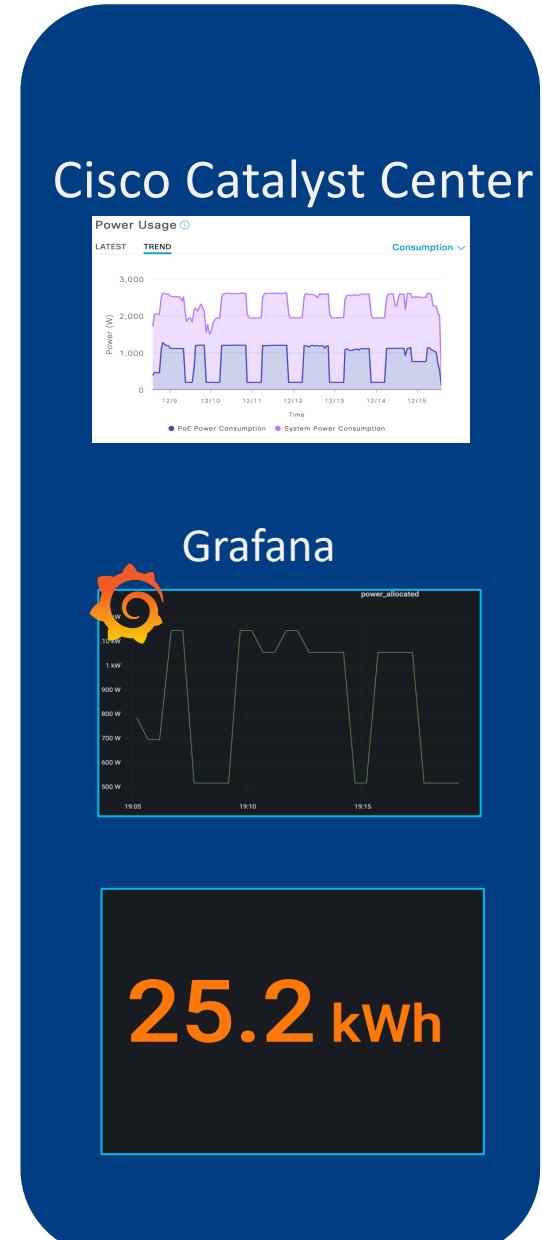
Catalyst Power Telemetry for Smart Buildings



<https://youtu.be/x25gQ8x3aqc>



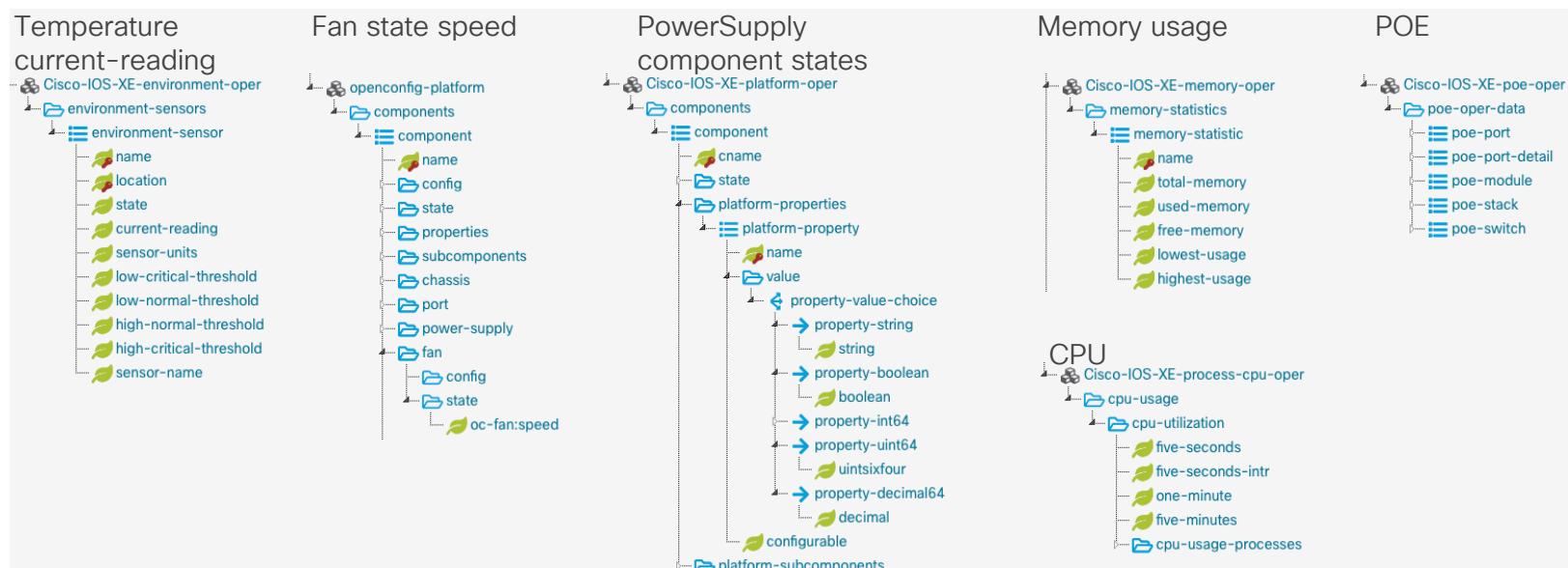
© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public



Which YANG API's are used ?

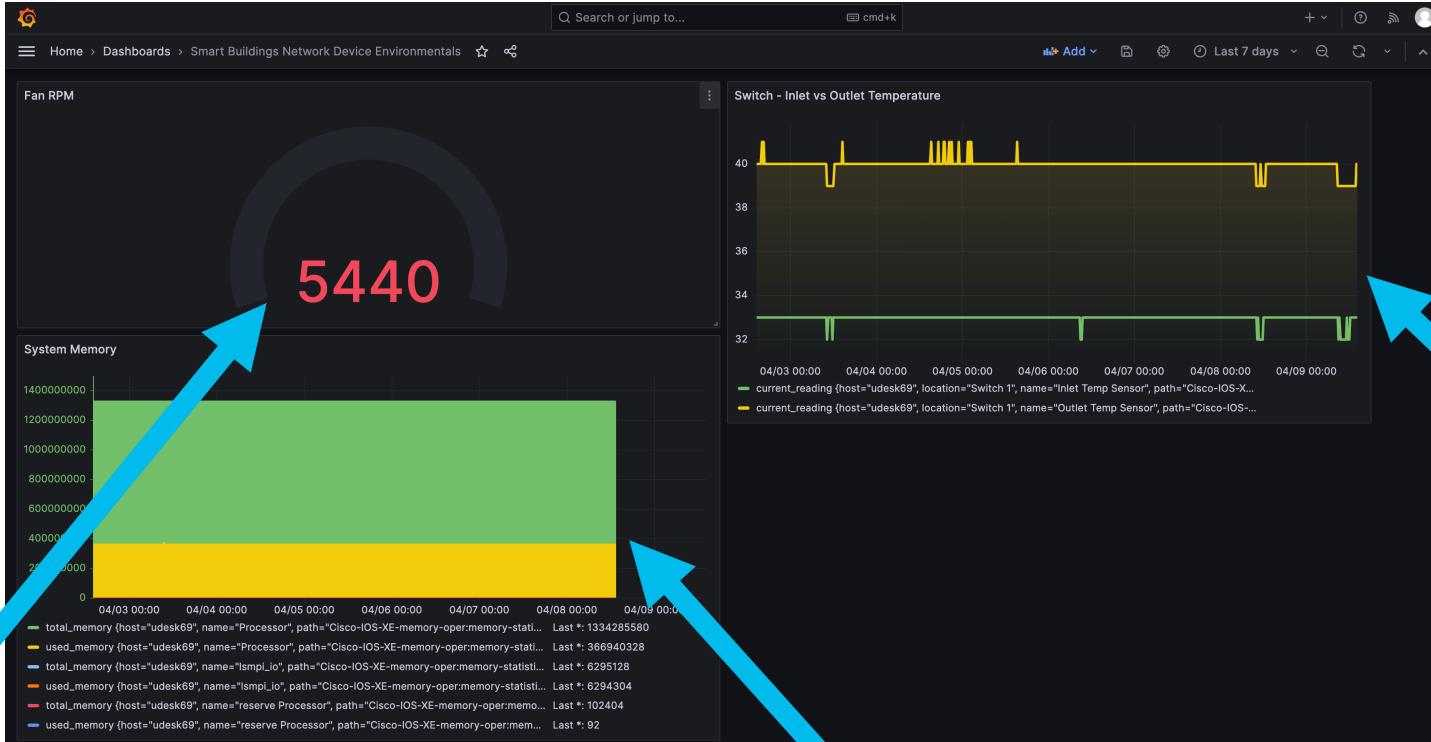
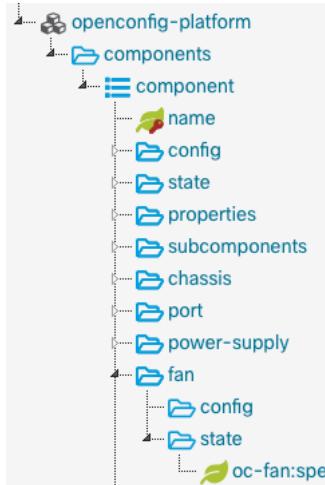
#	Feature	YANG Data Model	Subscription Xpath Configuration
1	Temperature	Cisco-IOS-XE-environment-oper	filter xpath /environment-ios-xe-oper:environment-sensors
2	Fan Speed	openconfig-platform.yang	filter xpath /oc-platform:components
3	PowerSupply	Cisco-IOS-XE-platform-oper.yang	filter xpath /platform-ios-xe-oper:components/component/platform-properties/platform-property
4	POE Module	Cisco-IOS-XE-poe-oper.yang	filter xpath /poe-ios-xe-oper:poe-oper-data/poe-module
5	POE Port	Cisco-IOS-XE-poe-oper.yang	filter xpath /poe-ios-xe-oper:poe-oper-data/poe-port-detail
6	POE Stack	Cisco-IOS-XE-poe-oper.yang	filter xpath /poe-ios-xe-oper:poe-oper-data/poe-stack
7	POE Switch	Cisco-IOS-XE-poe-oper.yang	filter xpath /poe-ios-xe-oper:poe-oper-data/poe-switch
8	Memory	Cisco-IOS-XE-memory-oper.yang	filter xpath /process-cpu-ios-xe-oper:cpu-usage/cpu-utilization
9	CPU	Cisco-IOS-XE-process-cpu-oper.yang	filter xpath /process-memory-ios-xe-oper:memory-usage-processes

- Every feature has a YANG data model
- Data models contain containers and lists of information about the feature
- The data can be subscribed to using the Xpath from



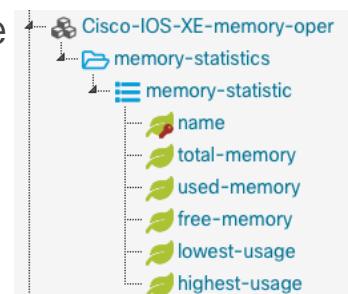
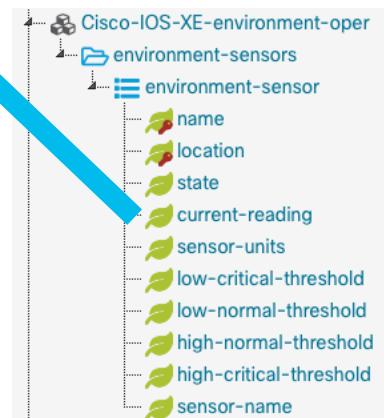
Which YANG API's are used ?

Fan state speed
= 5440 RPM



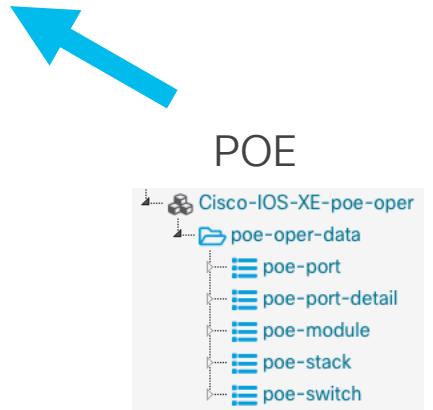
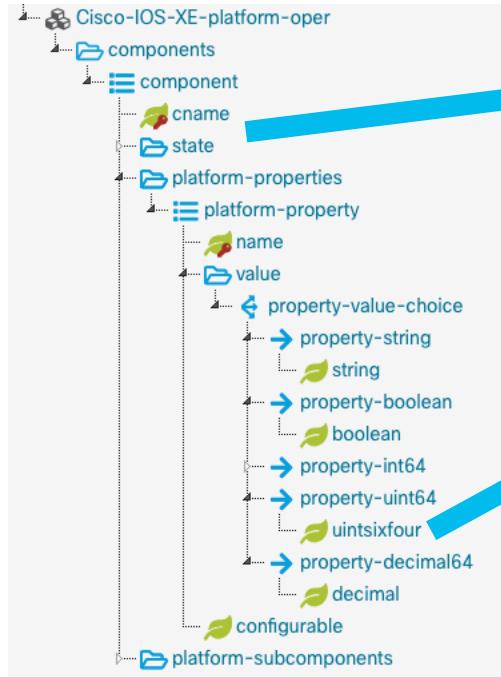
Memory usage
16 GB Total
4 GB Used

Temperature current-reading
Inlet = 32 F
Outlet = 40 F



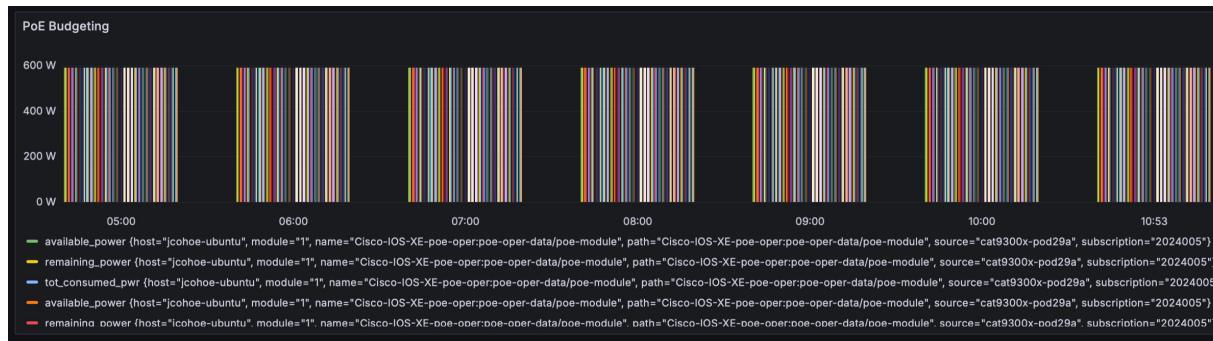
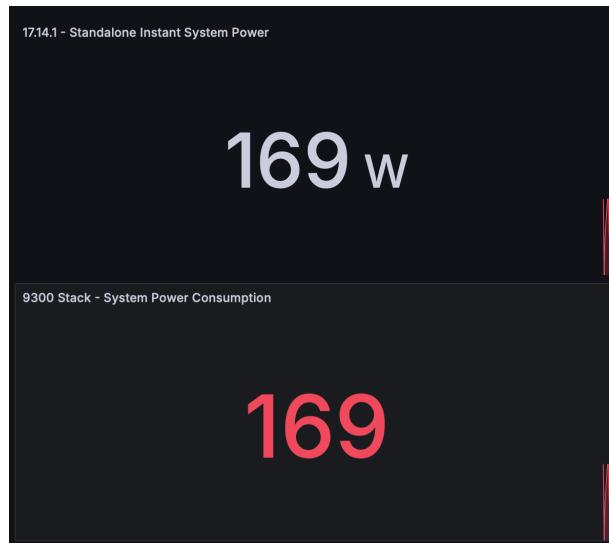
Which YANG API's are used ?

PowerSupply
component states



Visualize the data

Access the Grafana on port 3000 with admin/Cisco123 to see the dashlets
Fan Speed, Temperatures, Power Supply Utilization, PoE Budget



POE Telemetry

telemetry configuration

```
no telemetry ietf subscription 69001
```

```
telemetry ietf subscription 69001
```

```
filter xpath /poe-ios-xe-oper:poe-oper-data/poe-port-detail
```

```
receiver ip address 10.85.134.66 57508 protocol grpc-tcp
```

```
source-address 10.85.134.70
```

```
source-vrf Mgmt-vrf
```

```
stream yang-push
```

```
update-policy periodic 3000
```

```
encoding encode-kvvpb
```

```
no telemetry ietf subscription 69002
```

```
telemetry ietf subscription 69002
```

```
filter xpath /poe-ios-xe-oper:poe-oper-data/poe-switch
```

```
receiver ip address 10.85.134.66 57508 protocol grpc-tcp
```

```
source-address 10.85.134.70
```

```
source-vrf Mgmt-vrf
```

```
stream yang-push
```

```
update-policy periodic 3000
```

```
encoding encode-kvvpb
```

```
no telemetry ietf subscription 69003
```

```
telemetry ietf subscription 69003
```

```
filter xpath /poe-ios-xe-oper:poe-oper-data/poe-stack
```

```
receiver ip address 10.85.134.66 57508 protocol grpc-tcp
```

```
source-address 10.85.134.70
```

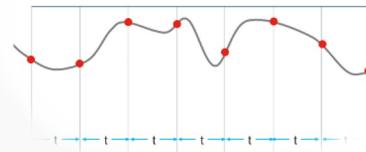
```
source-vrf Mgmt-vrf
```

```
stream yang-push
```

```
update-policy periodic 3000
```

```
encoding encode-kvvpb
```

Model Driven Telemetry



YANG data model

poe-port-detail	Name	poe-port-detail
intf-name	Nodetype	list
power-used	Description	List of PoE interfaces, keyed by interface name
pd-class	Module	Cisco-IOS-XE-poe-oper
device-detected	Revision	2022-07-01
device-name	Xpath	/poe-oper-data/poe-port-detail
police	Prefix	poe-ios-xe-oper
power-admin-max	Namespace	http://cisco.com/ns.yang/Cisco-IOS-XE-poe-oper
power-from-pse	Schema Node Id	/poe-oper-data/poe-port-detail
power-to-pd	Keys	• "intf-name"
poe-switch	Access	read-only
switch-num	Operations	• "get"
power-budget	Name	poe-switch
power-allocated	Nodetype	list
low-port-priority	Description	List of PoE switches, keyed by switch number
high-port-priority	Module	Cisco-IOS-XE-poe-oper
switch-priority	Revision	2022-07-01
port-one-status	Xpath	/poe-oper-data/poe-switch
port-two-status	Prefix	poe-ios-xe-oper
poe-stack	Namespace	http://cisco.com/ns.yang/Cisco-IOS-XE-poe-oper
power-stack-name	Schema Node Id	/poe-oper-data/poe-switch
mode	Keys	• "switch-num"
topolgy	Access	read-only
total-power	Operations	• "get"
rsvd-power	Name	poe-stack
alloc-power	Nodetype	list
unused-power	Description	List of PoE stacks, keyed by stack name
num-sw	Module	Cisco-IOS-XE-poe-oper
num-ps	Revision	2022-11-01
	Xpath	/poe-oper-data/poe-stack
	Prefix	poe-ios-xe-oper
	Namespace	http://cisco.com/ns.yang/Cisco-IOS-XE-poe-oper
	Schema Node Id	/poe-oper-data/poe-stack
	Keys	• "power-stack-name"
	Access	read-only
	Operations	• "get"

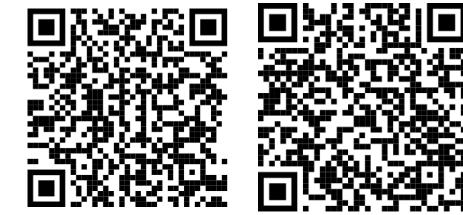
POE Dashboard on next slide



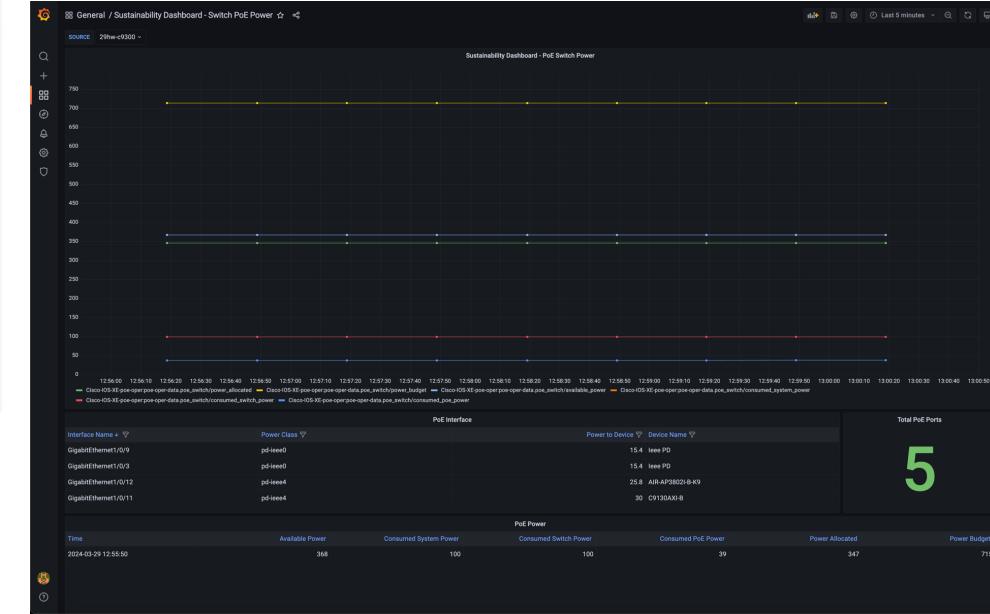
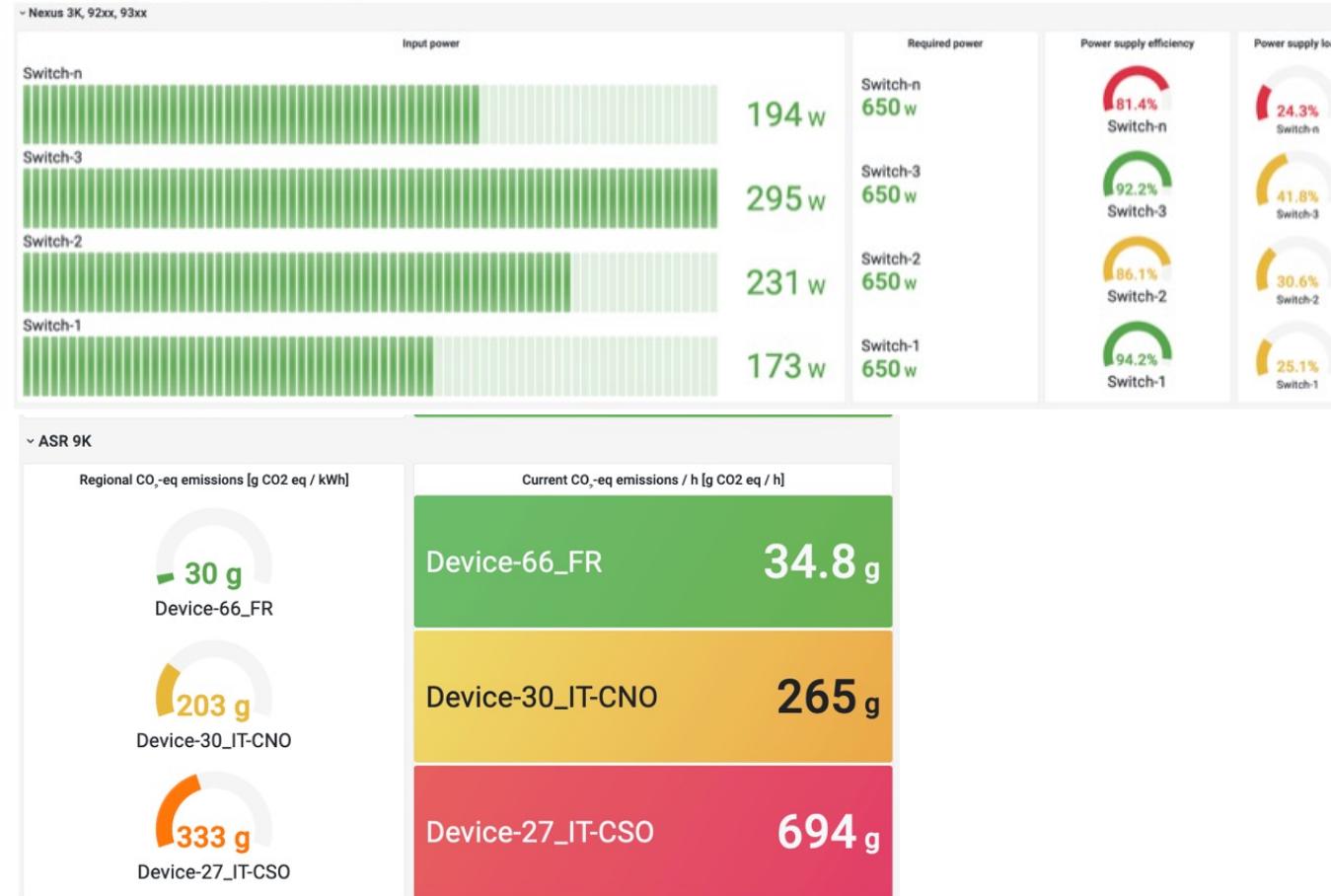
Resources

Additional sustainability telemetry use cases

<https://developer.cisco.com/codeexchange/github/repo/cisco-open/green-monitoring/>
<https://github.com/rickbauer9482/IOS-XE-Sustainability-Dashboard>



Overview power - DC - Nexus



GitHub.com and Grafana.com documentation

Details lab guide, Grafana Dashboard and YANG data model details available at
<https://github.com/jeremycoho/cisco-mdt-poe/>



README.md

Cisco Catalyst POE

Details about monitoring the POE and power consumption and utilization on Cisco Catalyst IOS XE.

This use case relies on the TIG_MDT Docker container which has the Telegraf, InfluxDB, and Grafana toolset available from [The cisco-ios-xe-mdt repository on Github](#)

Prerequisites

You need Cisco Catalyst 9300 POE

Some POE devices connected too

Minimum IOS XE Software release: _____ ?

POE YANG

The YANG module that has details about Power Over Etherner operational data is the [Cisco-IOS-XE-poe-oper:YANG](#) which has contains a lot of information about the feature.

There are two specific containers that have data that is relevant to POE and power monitoring: **poe-switch** which models port specific details, and the **poe-port-detail** container which models device level power information.

The **poe-switch** container has ...

poe-switch	Name	poe-switch
switch-num	Nodetype	list
power-budget	Description	List of PoE switches, keyed by switch number

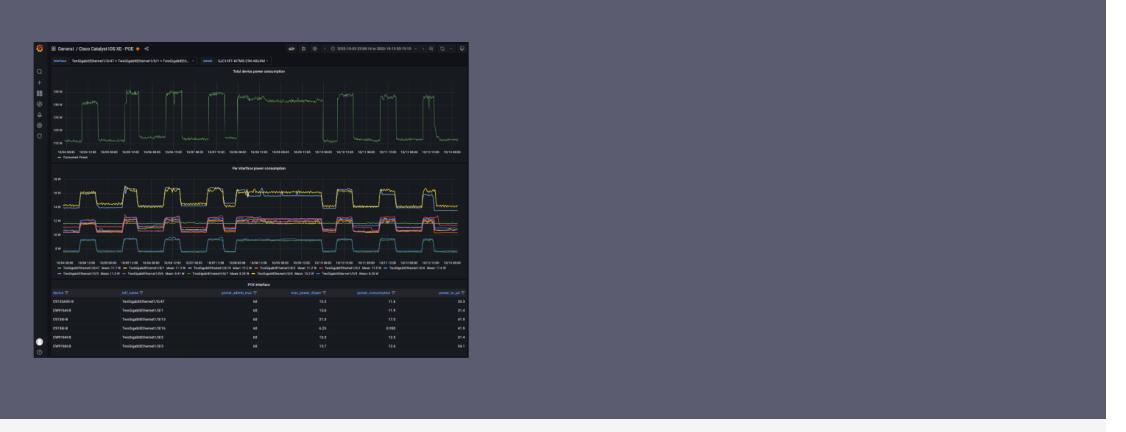
GrafanaLabs Products Open source Solutions Learn Company

← All dashboards

Cisco Catalyst POE

Cisco Catalyst Model Driven Telemetry (MDT) Power Over Ethernet (POE) dashboard

Overview Revisions Reviews Edit →



The screenshot shows a complex Grafana dashboard titled "Cisco Catalyst POE". It features several panels: a top panel with four line graphs showing power consumption over time; a middle panel with a larger graph showing power fluctuations across multiple ports; and a bottom panel displaying a table of power statistics for various ports, including "Total PoE power consumption", "Port 1 power consumption", "Port 2 power consumption", "Port 3 power consumption", and "Port 4 power consumption". The table includes columns for "Port", "Power consumption (W)", and "Status".

Programmability Configuration Guide

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

gNMI Dial-Out Using the gRPC Tunnel Service

Model Based AAA

Model-Driven Telemetry

In-Service Model Update

✓ Application Hosting

Application Hosting

ThousandEyes Enterprise Agent



https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/1714/b_1714_programmability_cg.html

API White Paper

Programmability and auto... ▾ 

Table of Contents

Programmability and automation... -

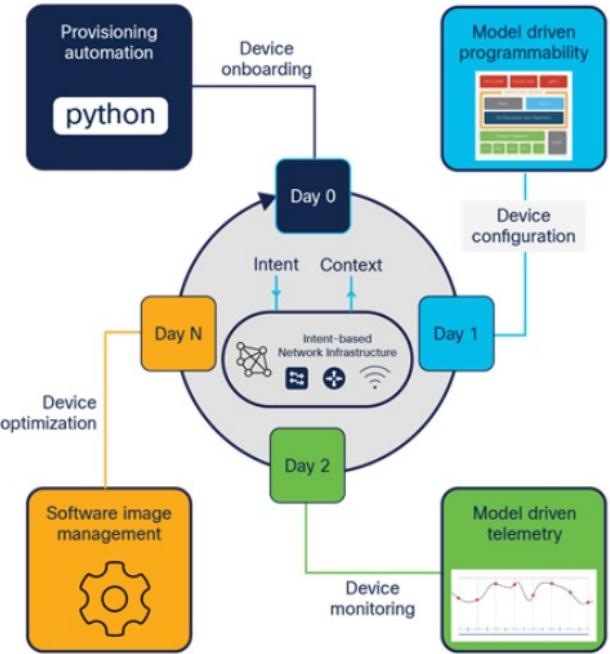
- Day 0: Provisioning automation
- Day 1: Model-driven programmability
- Day 2: Model-driven telemetry
- Day N: Device optimization
- Cisco IOS XE operational consistency

Yet Another Next Generation (YANG) ... +

- Day 1: Model-driven program... +
- Tooling: Cisco YANG Suite +
- Day 2: Model-driven telemetry +
- Day N: Device optimization +
- Conclusion
- Additional resources +
- Blogs

Catalyst Programmability and Automation

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



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

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

The Model Driven Telemetry White Paper includes examples, use cases and tooling related to telemetry

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 opti...

YANG data modeling language +

Benchmarking and comparis... +

Cisco controller solutions +

Cloud solutions +

Tooling +

Dashboarding and validation +

Configuration examples +

Telemetry configuration man... +

Troubleshooting and validati... +

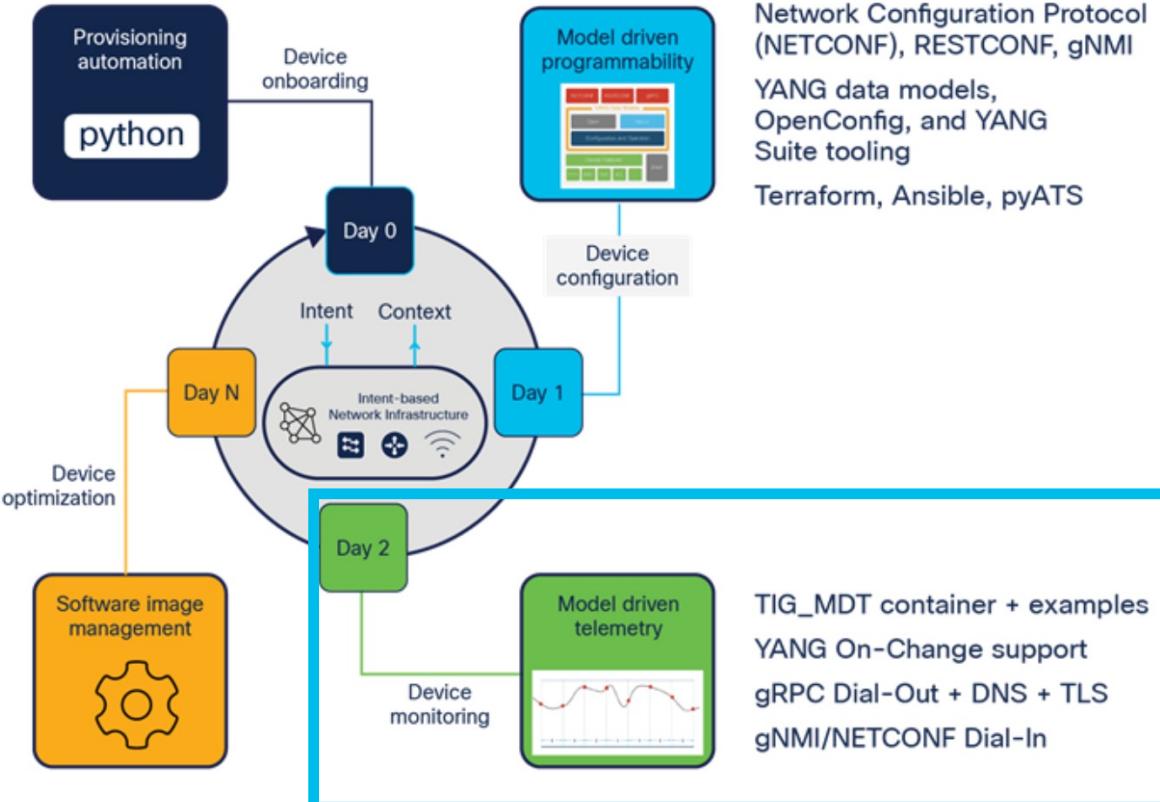
Best practices and lessons l... +

Conclusion

Resources

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

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



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



<http://cs.co/mdtwp>



<http://cs.co/mtpwppdf>





Thank you for watching and attending this interactive session

Energy Management for Catalyst

IOS XE Device Telemetry Workshop

Jeremy Cohoe & Story DeWeese
Technical Marketing

April 22, 2024 (Earth Day)