Programmabilité Cisco Catalyst / IOS-XE

Community Live Webinar - http://reseauxblog.cisco.fr/

Jérôme Durand – Technical Solutions Architect
Antoine Orsoni – Systems Engineer

6 décembre 2022

Des nouveaux sujets tous les mois

La Communauté francophone Cisco grandit

Grâce à la participation de tous nos collaborateurs, notre communauté grandit pour vous offrir des nouveaux webinaires sur différentes technologies.

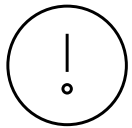
Inscrivez-vous et participez !

Prochains événements

Événements précédents

Webex et le RGPD (2022)
le mardi 13 décembre 2022

Redécouvrez Cisco SD-WAN
le mardi 17 janvier 2023



Connectez, Engagez, Collaborez !

Solutions

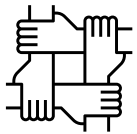
Acceptez les solutions qui sont correctes et complimentez ceux qui vous ont aidé ! Aidez autres utilisateurs à trouver les réponses correctes dans la fenêtre de recherche.

Accepter comme solution

Compliments

Mettez en évidence les autres membres. Les votes utiles motivent les membres enthousiastes en leur offrant un signe de reconnaissance !

0 Compliments



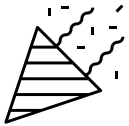
Community Helping Community

Aidez-nous à franchir un nouveau challenge !

Cette année la communauté a choisi deux associations au profit des enfants, mais aussi pour soutenir l'approvisionnement des hôpitaux et des réfugiés en Ukraine.

Nous avons besoin de votre aide pour parvenir à 13'000 interactions et être en mesure d'atteindre notre objectif.

Si vous souhaitez collaborer vous n'avez pas besoin de verser de l'argent, votre activité lorsque vous vous connectez à la communauté sera comptabilisée pour nous aider. Cliquez ici



Jérôme Durand

Présentateur

Jérôme Durand a rejoint Gip Renater en 2002. Il a commencé par travailler sur des projets de R&D, mais est rapidement devenu responsable d'exploitation en 2006. En 2009, il prend la responsabilité de l'équipe services. Jérôme a rejoint Cisco en 2011 en tant qu'expert des technologies de routage et de commutation. Actuellement, il est très impliqué dans la programmation et l'automatisation des réseaux, notamment les solutions SD-WAN et SD-Access. Il est également l'auteur de la RFC 7454 - BGP Operations and Security.

Antoine Orsoni

Présentateur

Antoine Orsoni a rejoint Cisco en 2017. Après un rôle de spécialiste Enterprise Networking, il accompagne depuis 2022 des clients opérateurs télécoms français. Il travaille avec ses clients sur des projets de SP Routing, Data Center et Automatisation.

Téléchargez la présentation !

https://bit.ly/WEB1sld-dec22

Agenda

- Introduction à la programmabilité
- YANG
- Les modèles YANG en pratique

- NETCONF
- RESTCONF
- GNXI

- Télémétrie
- Démo YANG Suite
- Pour poursuivre l’aventure...

Introduction à la programmabilité

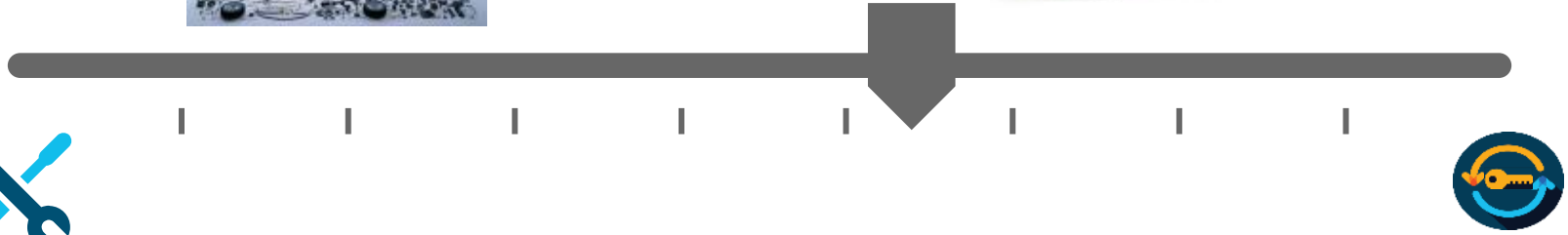
Do it Yourself or Integrated strategy

NETCONF
RESTCONF
gRPC
gNMI
gNOI

DNA Automation
DNA Assurance
SD-Access

DIY

Integrated



Do it Yourself or Integrated strategy

NETCONF
RESTCONF
gRPC
gNMI

API

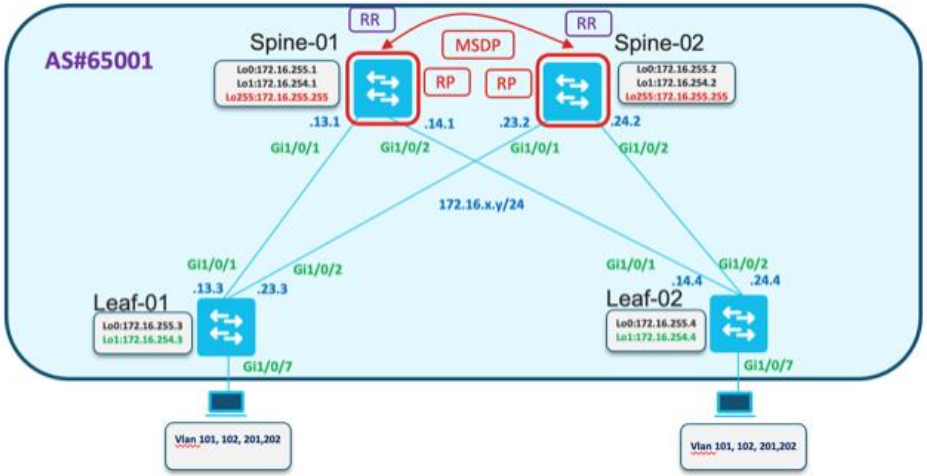
DNA Automation
DNA Assurance
SD-Access

DIY

Integrated

BGP EVPN VXLAN solution management with Ansible

The Ansible BGP EVPN solution is publicly available now on GitHub with step-by-step examples of getting started with DAG, L2VNI and L3VNI.



Try it out yourself and engage directly on GitHub:
https://github.com/Cat9kEVPN/cat9k-evpn-ansible

Quick start

For the quick start with DAG provisioning next steps have to be executed:

Step 1

On this step Underaly is provisioned.

Step 1a

Edit `inventory.yml` and set proper name and management ip address.

```
all:
  children:
    leaf:
      hosts:
        Leaf-01:
          ansible_host: 10.1.1.1

<...snip...>
```

Detailed information could be found here

Step 1b

Edit `group_vars/all.yml` and set proper login and password

```
ansible_connection: ansible.netcommon.network_cli
ansible_network_os: cisco.ios.ios
ansible_python_interpreter: "python"
ansible_user: cisco
ansible_ssh_pass: cisco123
```

`ansible_user` must have privildge level 15. Example of the configuration is below

```
username cisco privilege 15 password 0 cisco123
```

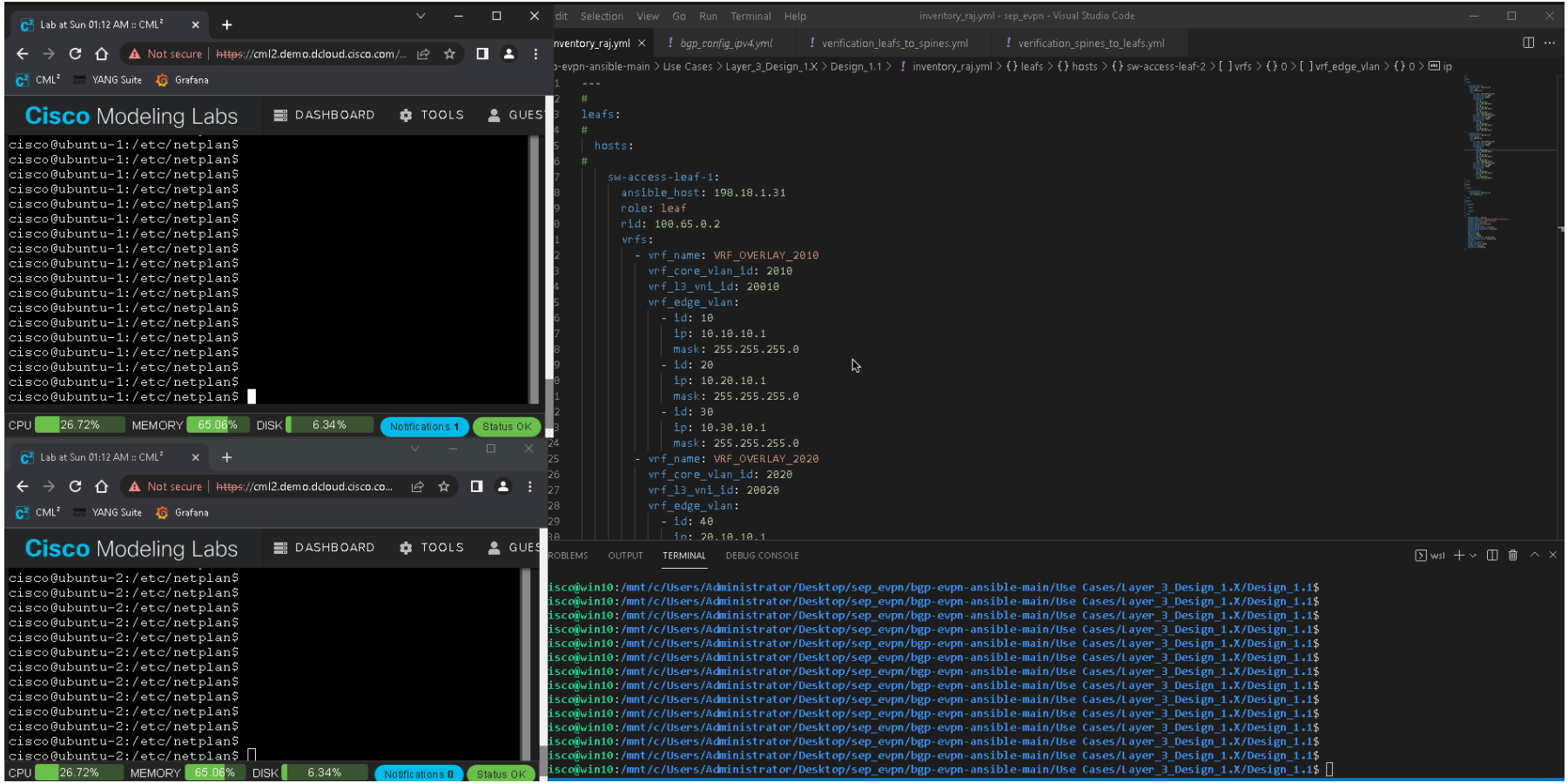
If enable password should be used, check the Enable Mode documentation.

Detailed information could be found here

Step 1c

Edit `host_vars/<hostname>.yml` and set required parameters for underlay

Deployment of BGP EVPN VXLAN with Ansible



IOS XE Programmable Device Lifecycle

Secure ZTP

Provisioning Automation

python™

Device Onboarding

Model Driven Programmability

Network Configuration Protocol (NETCONF), RESTCONF, gNMI

YANG Data Models, OpenConfig,

gNMI IPv6 support

YANG Suite, Terraform, Ansible

Device Configuration

Day 0

Day 1

Day 2

Day N

INTENT

CONTEXT

Intent-based Network Infrastructure

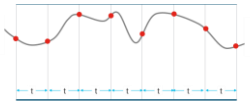
Device Optimization

Software Image Management

Guest Shell + DNS

Device Monitoring

Model Driven Telemetry



gNMI Dial-In

Proto encoding for gNMI

gRPC Dial-Out + DNS + mTLS

IOS XE Programmable Device Lifecycle

Secure ZTP

Provisioning Automation

python™

Device Onboarding

Day 0

Model Driven Programmability

Network Configuration Protocol (NETCONF), RESTCONF, gNMI

YANG Data Models, OpenConfig,

gNMI IPv6 support

YANG Suite, Terraform, Ansible

Device Configuration

Day 1

INTENT

CONTEXT

Intent-based Network Infrastructure

Day N

Device Optimization

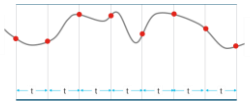
Software Image Management

Guest Shell + DNS

Day 2

Device Monitoring

Model Driven Telemetry



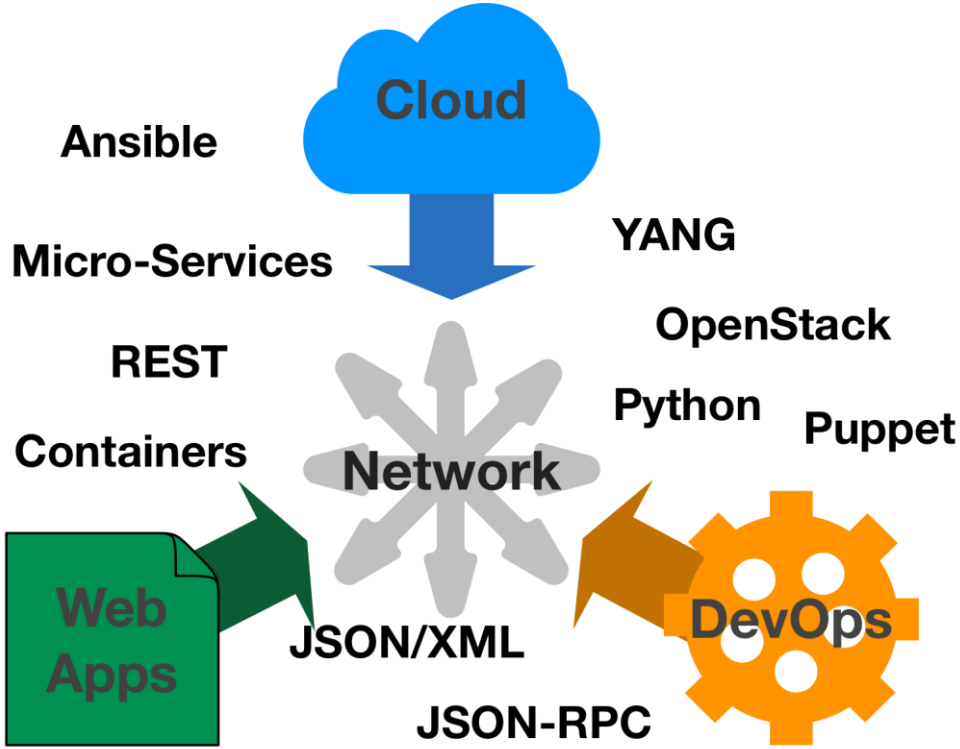
gNMI Dial-In

Proto encoding for gNMI

gRPC Dial-Out + DNS + mTLS

Why Model Driven Programmability ?

The Network is No Longer Isolated

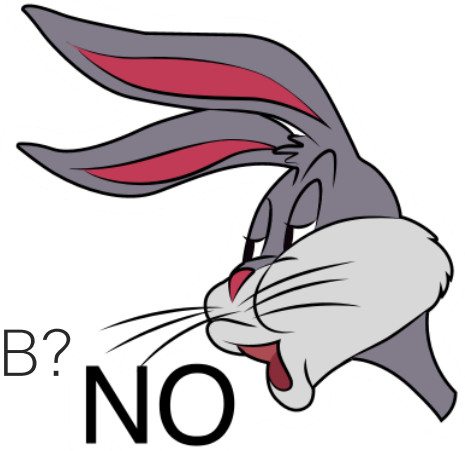


Congrats !
Your configuration has been successfuly deployed on 217 / 276 devices

What about SNMP?

SNMP works "reasonably well for device monitoring"

RFC 3535: Overview of the 2002 IAB Network Management Workshop – 2003
https://tools.ietf.org/html/rfc3535



Standard MIB?
Scalability?
Retrieve full device config?
Filtered output?
Configuration?

RFC 3535: What is Needed?

- A programmatic interface for device configuration
- Separation of Configuration and State Data
- Ability to configure "services" NOT "devices"
- Integrated error checking and recovery

Programmable Interfaces

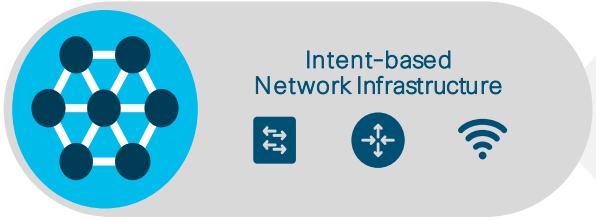
CLI

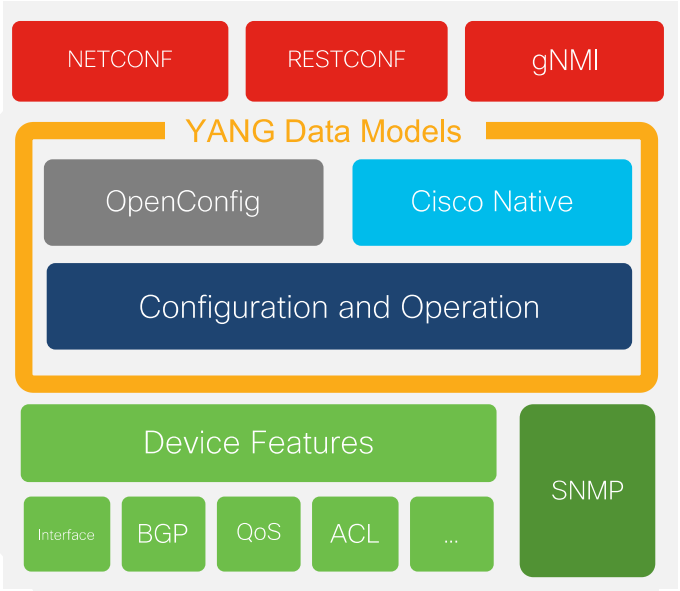
SNMP

WebUI

The NETCONF, RETCONF and gNMI 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 and gNMI

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





IETF

OPENCONFIG

IEEE

CISCO

YANG

Why models ?

```
SA-Edge-1#sh int gi 1/0/17
GigabitEthernet1/0/17 is up, line protocol is up (connected)
  Hardware is Gigabit Ethernet, address is 70d3.7984.0f11 (bia 70d3.7984.0f11)
  MTU 9100 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 1000Mb/s, media type is 10/100/1000BaseTX
  input flow-control is on, output flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/2000/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     23530484 packets input, 3412544575 bytes, 0 no buffer
     Received 3871554 broadcasts (1816130 multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 1816130 multicast, 0 pause input
     0 input packets with dribble condition detected
     24346526 packets output, 5625088240 bytes, 0 underruns
     Output 281128 broadcasts (0 multicasts)
```

Why models ?

```
SA-Edge-1#sh int gi 1/0/17
GigabitEthernet1/0/17 is up, line protocol is up (connected)
  Hardware is Gigabit Ethernet, address is 70d3.7984.0f11 (bia 70d3.7984.0f11)
  MTU 9100 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 1000Mb/s, media type is 10/100/1000BaseTX
  input flow-control is on, output flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/2000/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     23530484 packets input, 3412544575 bytes, 0 no buffer
     Rec      d 3871554 broadcasts (1816130 multicasts)
     0        0 giants, 0 throttles
              errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     watchdog, 1816130 multicast, 0 pause input
     0 input packets with dribble condition detected
     24346526 packets output, 5625088240 bytes, 0 underruns
     Output 281128 broadcasts (0 multicasts)
```

What is a Data Model?

A data model is simply a well understood and agreed upon method to describe "something". As an example, consider this simple "data model" for a person.

- *Person*
  - **Gender** – male, female, other
  - **Height** – Feet/Inches or Meters
  - **Weight** – Pounds or Kilos
  - **Hair Color** – Brown, Blond, Black, Red, other
  - **Eye Color** – Brown, Blue, Green, Hazel, other

Where do YANG models come from ?

- **Standard definition**
  (IETF, ITU, OpenConfig, etc.)
- Compliant with standard
  `ietf-diffserv-policy.yang`
  `ietf-diffserv-classifer.yang`
  `ietf-diffserv-target.yang`

- **Vendor definition**
  (Cisco, Juniper, etc.)
- Unique to Vendor Platforms
  `cisco-memory-stats.yang`
  `cisco-flow-monitor`
  `cisco-qos-action-qlimit-cfg`

https://github.com/YangModels/yang
https://github.com/openconfig/public

What is OpenConfig?

Models Designed by Operators for Operators

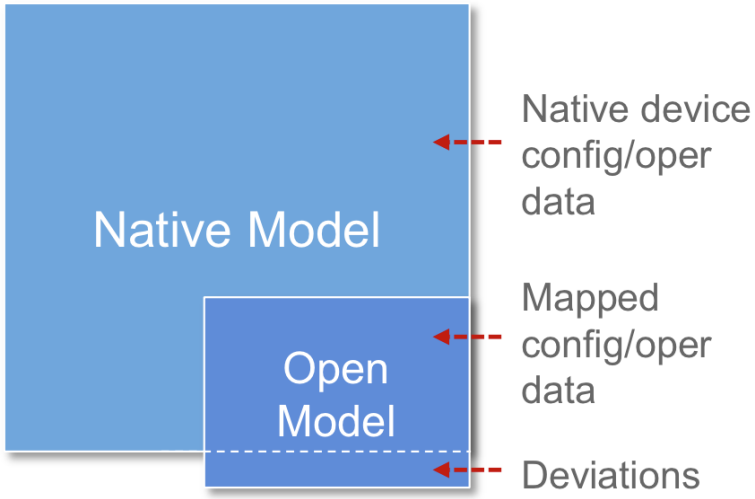
"OpenConfig's initial focus is on compiling a consistent set of vendor-neutral data models based on actual operational needs from use cases and requirements from multiple network operators."

OpenConfig FAQ:
www.openconfig.com

What is OpenConfig?

Models Designed by Operators for Operators

- Focused on creating vendor-neutral data models written in YANG
- Models combine both configuration and operational data
- Model coverage still limited with an active development community
- Support from multiple routing vendors (e.g. Cisco, Juniper, Arista)



A look at a Standard YANG Model for Interfaces

```
DevNet$ pyang -f tree ietf-interfaces.yang

module: ietf-interfaces
  +--rw interfaces
  |  +--rw interface* [name]
  |     +--rw name                        string
  |     +--rw description?                string
  |     +--rw type                        identityref
  |     +--rw enabled?                    boolean
  |     +--rw link-up-down-trap-enable?   enumeration {if-mib}?
  |     +--ro admin-status                enumeration {if-mib}?
  |     +--ro oper-status                 enumeration
  |     +--ro last-change?                yang:date-and-time
  |     +--ro if-index                    int32 {if-mib}?
  |     +--ro phys-address?               yang:phys-address
```

Example output edited for simplicity and brevity

dna-dne-code/intro-mdp/yang/models/ietf-interfaces.yang

Programmable Interfaces

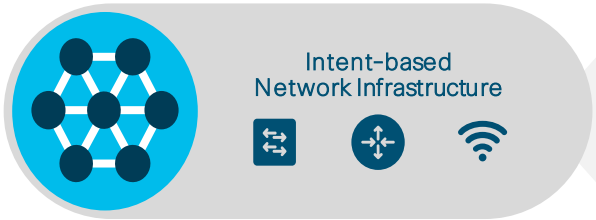
CLI

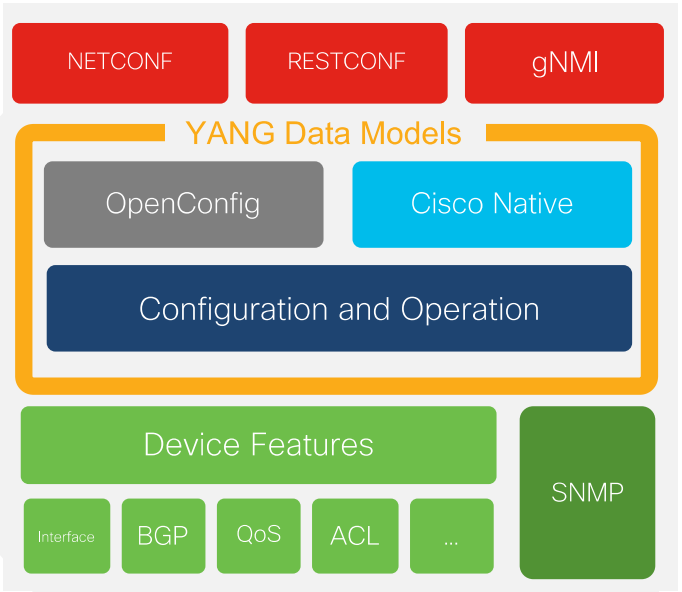
SNMP

WebUI

The NETCONF, RETCONF and gNMI 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 and gNMI

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





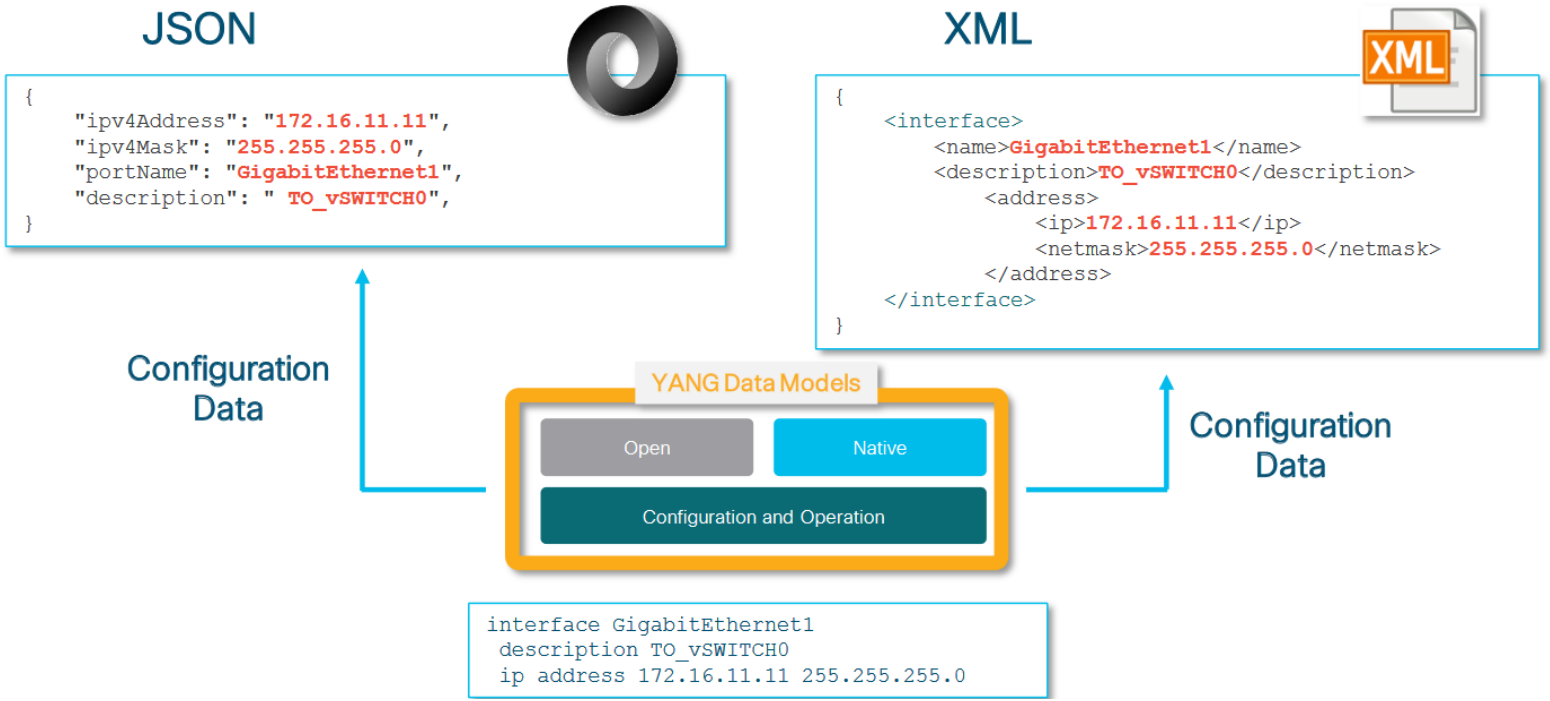
IETF

OPENCONFIG

IEEE

CISCO

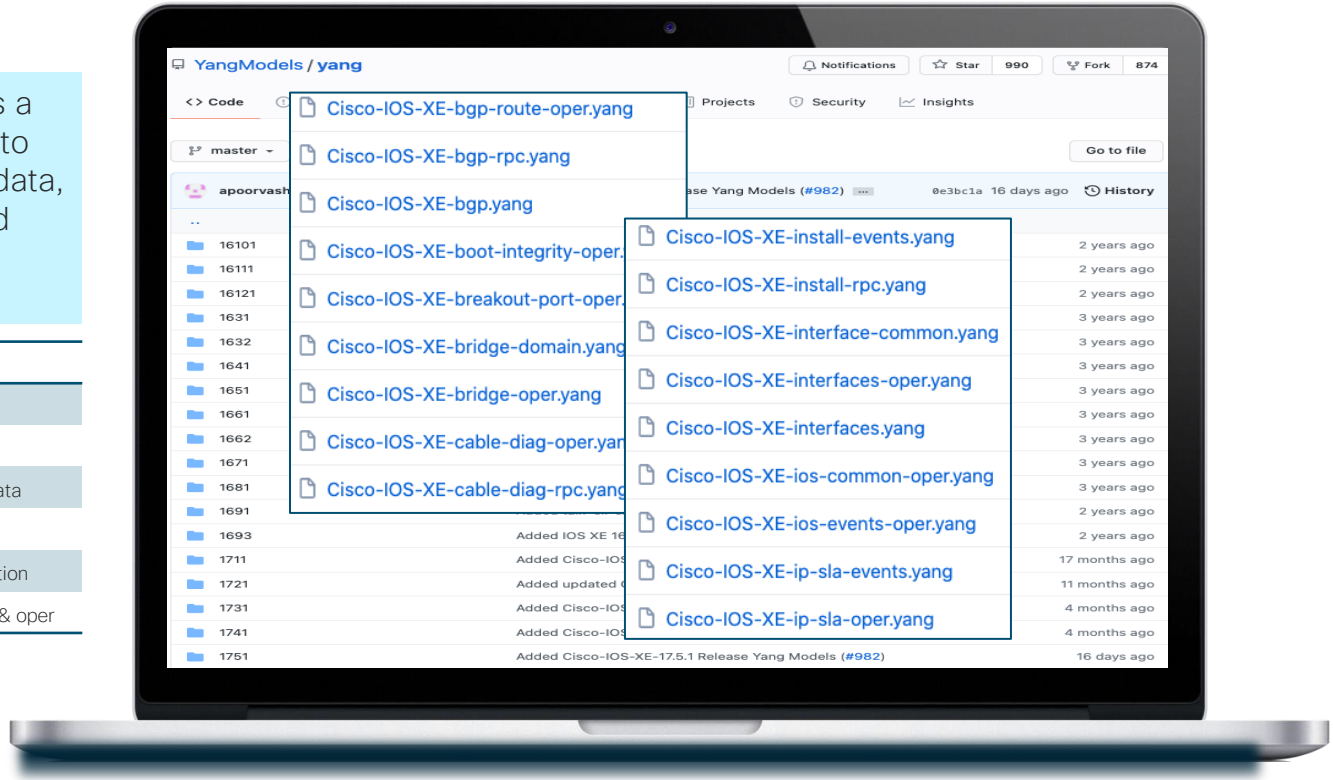
YANG Relationship to JSON and XML



IOS XE – YANG Model Coverage on GitHub

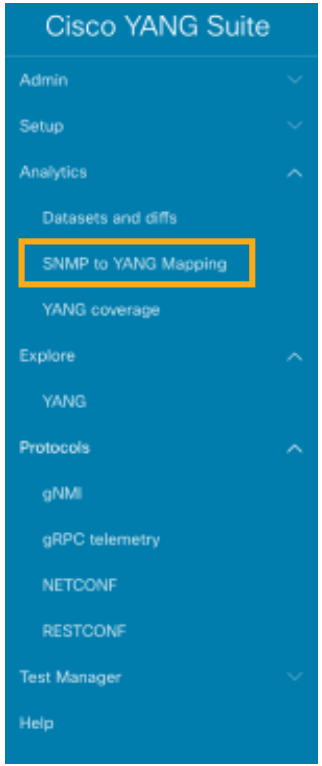
RFC7950 states that “YANG is a data modeling language used to model configuration data, state data, Remote Procedure Calls, and notifications for network management protocols”

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

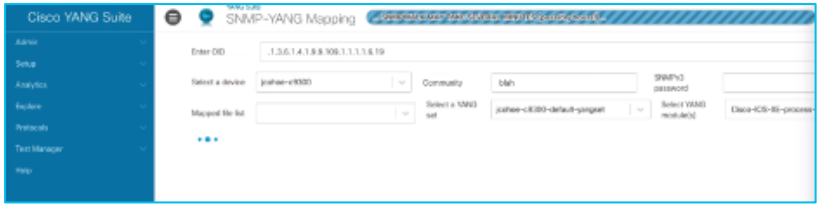


https://github.com/YangModels/yang/tree/master/vendor/cisco/xe

SNMP to YANG migration mapping



Ease the transition from SNMP OID to YANG Xpath and easily verify the responses from both.



OID: .1.3.6.1.4.1.9.9.109.1.1.1.1.6.19

MIB Path .iso.org.dod.internet.private.enterprises.9.9.276.1.6.1.1.2.7.84.101.49.47.48.47.52

YANG Path /cpu-usage/cpu-utilization/five-seconds

YANG Suite

```
SNMPGET:
.iso.org.dod.internet.private.enterprises.9.9.276.1.6.1.1.2.7.84.101.49.47.48.47.52 = INTEGER: 12


YANG GET:
/cpu-usage/cpu-utilization/five-seconds
VALUES:
['1']
```

Right click > Run to retrieve from SNMP and NETCONF simultaneously.

This solution utilizes the Python library for "fuzzy matching" of OID and XPATH values to identify most accurate match.

Modèles YANG en pratique

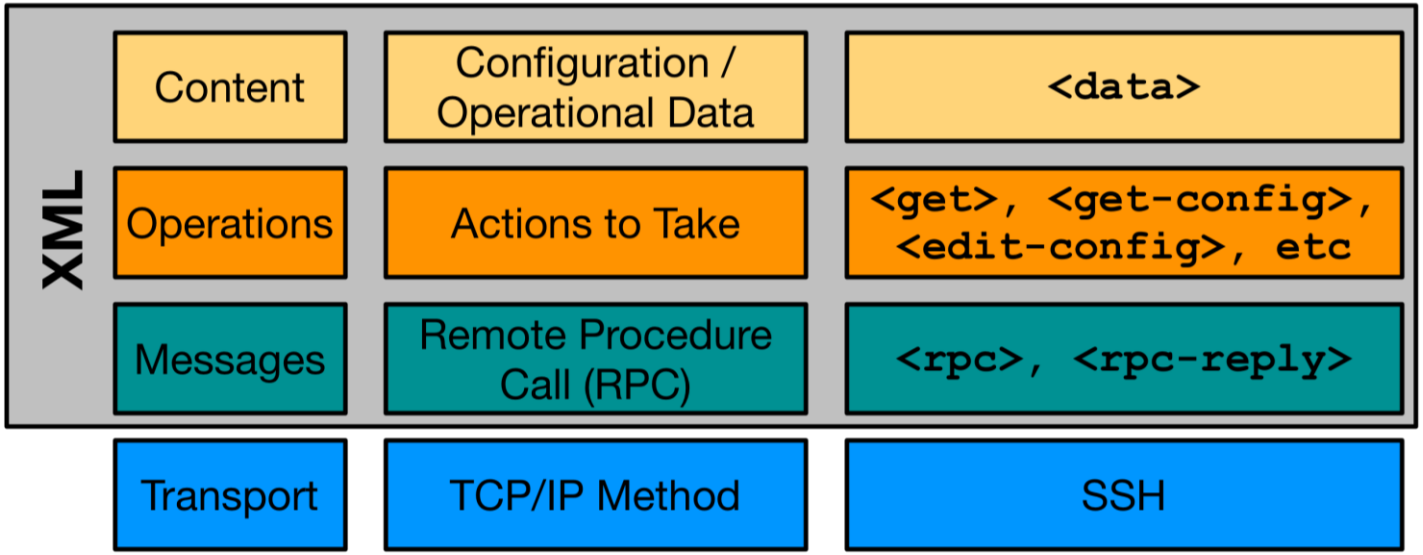
Polling Question 1

Où puis-je trouver les modèles YANG pour IOS-XE ?

1) Cisco.com
2) GitHub
3) Contact TAC

NETCONF

NETCONF protocol



Understanding the Capabilities List

```
urn:ietf:params:netconf:base:1.0
urn:ietf:params:netconf:base:1.1

urn:ietf:params:xml:ns:yang:ietf-interfaces?
 module=ietf-interfaces&revision=2014-05-08&
 features=pre-provisioning,if-mib,arbitrary-names&
 deviations=ietf-ip-devs
```

Example edited for simplicity and brevity

Two General Types

- Base NETCONF capabilities (= NETCONF version)
- Data Models Supported

YANG 1.0 to 1.1 transition – YANG advertisement

Legacy YANG 1.0 capabilities exchange and NETCONF "hello" message will soon be unsupported

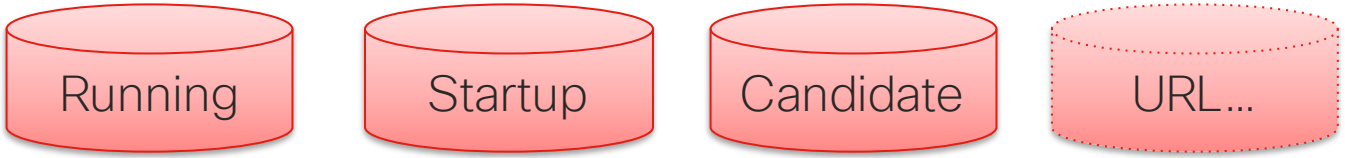
```
(v) auto@pod24-xelab:~/ncc$
(v) auto@pod24-xelab:~/ncc$ netconf-console --host c9300 --port 830 --user admin --password Cisco123 --hello
/home/auto/ncc/v/lib/python3.8/site-packages/paramiko/kex_ecdh_nist.py:39: CryptographyDeprecationWarning: encode_point has been deprecated on EllipticCurvePublicNumbers and will be
 compressed and uncompressed point encoding.
  m.add_string(self.Q_C.public_numbers().encode_point())
/home/auto/ncc/v/lib/python3.8/site-packages/paramiko/kex_ecdh_nist.py:91: CryptographyDeprecationWarning: Support for unsafe construction of public numbers from encoded data will be
  self.Q_S = ec.EllipticCurvePublicNumbers.from_encoded_point(
/home/auto/ncc/v/lib/python3.8/site-packages/paramiko/kex_ecdh_nist.py:103: CryptographyDeprecationWarning: encode_point has been deprecated on EllipticCurvePublicNumbers and will be
n compressed and uncompressed point encoding.
  hm.add_string(self.Q_C.public_numbers().encode_point())
<?xml version='1.0' encoding='UTF-8'?>
<nc:hello xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <nc:capabilities>
    <nc:capability>urn:ietf:params:netconf:base:1.0</nc:capability>
    <nc:capability>urn:ietf:params:netconf:base:1.1</nc:capability>
    <nc:capability>urn:ietf:params:netconf:capability:writable-running:1.0</nc:capability>
    <nc:capability>urn:ietf:params:netconf:capability:rollback-on-error:1.0</nc:capability>
    <nc:capability>urn:ietf:params:netconf:capability:validate:1.0</nc:capability>
    <nc:capability>urn:ietf:params:netconf:capability:validate:1.1</nc:capability>
    <nc:capability>urn:ietf:params:netconf:capability:xpath:1.0</nc:capability>
    <nc:capability>urn:ietf:params:netconf:capability:notification:1.0</nc:capability>
    <nc:capability>urn:ietf:params:netconf:capability:interleave:1.0</nc:capability>
    <nc:capability>urn:ietf:params:netconf:capability:with-defaults:1.0?basic-mode=explicit&amp;also-supported=report-all-tagged,report-all</nc:capability>
    <nc:capability>urn:ietf:params:netconf:capability:yang-library:1.0?revision=2016-06-21&amp;module-set-id=4702f66e9d4acb0c47ccf523dcafba47</nc:capability>
    <nc:capability>http://tail-f.com/ns/netconf/actions/1.0</nc:capability>
    <nc:capability>http://cisco.com/ns/cisco-xe-ietf-ip-deviation?module=cisco-xe-ietf-ip-deviation&amp;revision=2016-08-10</nc:capability>
    <nc:capability>http://cisco.com/ns/cisco-xe-ietf-ipv4-unicast-routing-deviation?module=cisco-xe-ietf-ipv4-unicast-routing-deviation&amp;revision=2015-09-11</nc:capability>
    <nc:capability>http://cisco.com/ns/cisco-xe-ietf-ipv6-unicast-routing-deviation?module=cisco-xe-ietf-ipv6-unicast-routing-deviation&amp;revision=2015-09-11</nc:capability>
    <nc:capability>http://cisco.com/ns/cisco-xe-ietf-ospf-deviation?module=cisco-xe-ietf-ospf-deviation&amp;revision=2018-02-09</nc:capability>
    <nc:capability>http://cisco.com/ns/cisco-xe-ietf-routing-deviation?module=cisco-xe-ietf-routing-deviation&amp;revision=2016-07-09</nc:capability>
    <nc:capability>http://cisco.com/ns/cisco-xe-openconfig-acl-deviation?module=cisco-xe-openconfig-acl-deviation&amp;revision=2017-08-25</nc:capability>
    <nc:capability>http://cisco.com/ns/cisco-xe-openconfig-aft-deviation?module=cisco-xe-openconfig-aft-deviation&amp;revision=2018-12-05</nc:capability>
    <nc:capability>http://cisco.com/ns/cisco-xe-openconfig-isis-deviation?module=cisco-xe-openconfig-isis-deviation&amp;revision=2018-12-05</nc:capability>
    <nc:capability>http://cisco.com/ns/cisco-xe-openconfig-lldp-deviation?module=cisco-xe-openconfig-lldp-deviation&amp;revision=2018-07-25</nc:capability>
    <nc:capability>http://cisco.com/ns/cisco-xe-openconfig-mpls-deviation?module=cisco-xe-openconfig-mpls-deviation&amp;revision=2019-06-27</nc:capability>
    <nc:capability>http://cisco.com/ns/cisco-xe-openconfig-segment-routing-deviation?module=cisco-xe-openconfig-segment-routing-deviation&amp;revision=2018-12-05</nc:capability>
    <nc:capability>http://cisco.com/ns/cisco-xe-openconfig-system-management-deviation?module=cisco-xe-openconfig-system-management-deviation&amp;revision=2019-07-01</nc:capability>
```

YANG 1.1 example: "ietf-yang-library" to retrieve supported YANG modules

```
Sending:

#275
<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:b966c2ff-a59b-46a3-aa31-cb1cc5e97e44">
  <nc:get>
    <nc:filter>
      <modules-state xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library"/>
    </nc:filter>
  </nc:get>
</nc:rpc>

##

Received message from host

<?xml version="1.0" ?>
<rpc-reply message-id="urn:uuid:b966c2ff-a59b-46a3-aa31-cb1cc5e97e44" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:par
  <data>
    <modules-state xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library">
      <module-set-id>4702f66e9d4acb0c47ccf523dcafba47</module-set-id>
      <module>
        <name>BGP4-MIB</name>
        <revision>1994-05-05</revision>
        <schema>http://localhost:9938/restconf/tailf/modules/BGP4-MIB/1994-05-05</schema>
        <namespace>urn:ietf:params:xml:ns:yang:smiv2:BGP4-MIB</namespace>
        <conformance-type>implement</conformance-type>
      </module>
      <module>
        <name>BRIDGE-MIB</name>
        <revision>2005-09-19</revision>
        <schema>http://localhost:9938/restconf/tailf/modules/BRIDGE-MIB/2005-09-19</schema>
        <namespace>urn:ietf:params:xml:ns:yang:smiv2:BRIDGE-MIB</namespace>
        <conformance-type>implement</conformance-type>
      </module>
```

If the desired application previously parsed the NETCONF "hello" message to retrieve the supported YANG models, the parsing must be modified to reflect how version 1.1 advertises via "ietf-yang-library" instead of the NETCONF "hello" message.

NETCONF Data Stores



- Data stores are named contains that *may* hold an entire copy of the configuration
- Not all data stores are supported by all devices
- Running is the only mandatory data store
- Not all data stores are writable
  - Check the device's capabilities
  - To make changes to a non-writeable data store, copy from a writable one

Operations - NETCONF Actions

Operation	Description
<get>	Retrieve running configuration and device state information
<get-config>	Retrieve all or part of specified configuration data store
<edit-config>	Loads all or part of a configuration to the specified configuration data store
<copy-config>	Replace an entire configuration data store with another
<delete-config>	Delete a configuration data store
<commit>	Copy candidate data store to running data store
<lock> / <unlock>	Lock or unlock the entire configuration data store system
<close-session>	Graceful termination of NETCONF session
<kill-session>	Forced termination of NETCONF session

Challenges of most of tools out there

Create

Convert service request into valid config

Easy: need to check resource availability

Update

Config change from change in service

Medium: need to add/release resources

Delete

Remove service instance, release resources

Hard: need reference counting of resources

Repair

Recover from arbitrary resource changes

Very Hard: identify/fix or migrate to new infra

NETCONF
example with Ansible

Playbook Ansible (module cli_command)

```
- ios_config:
  lines:
    - ntp server 2.2.2.2
```

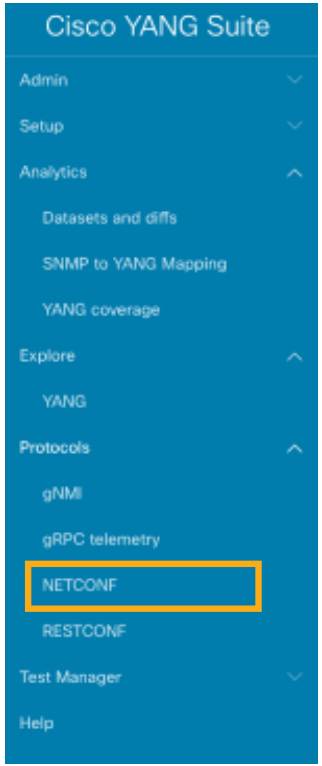
What if there is already an NTP server ?

You’re lucky there is a module that handles bettrer NTP on Cisco IOS: ios_ntp_global_module

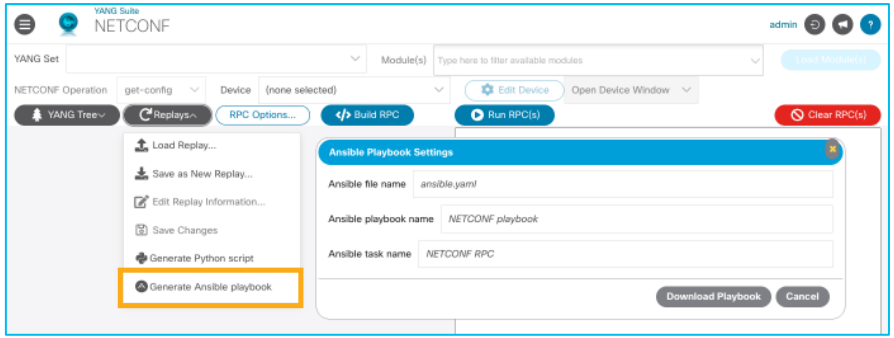
Playbook Ansible (module netconf_rpc)

```
- netconf_rpc:
  rpc: edit-config
  content: |
    <target>
      <running/>
    </target>
    <config>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <ntp>
          <server xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-ntp"
operation='replace'>
            <server-list>
              <ip-address>2.2.2.2</ip-address>
            </server-list>
          </server>
        </ntp>
      </native>
    </config>
```

YANG Suite + Ansible integrations
using NETCONF, RESTCONF & gNMI OpenConfig



Quickly and easily generate Ansible playbook for deployments to be used with the inventory, similar to the “Generate Python script” button.



RESTCONF

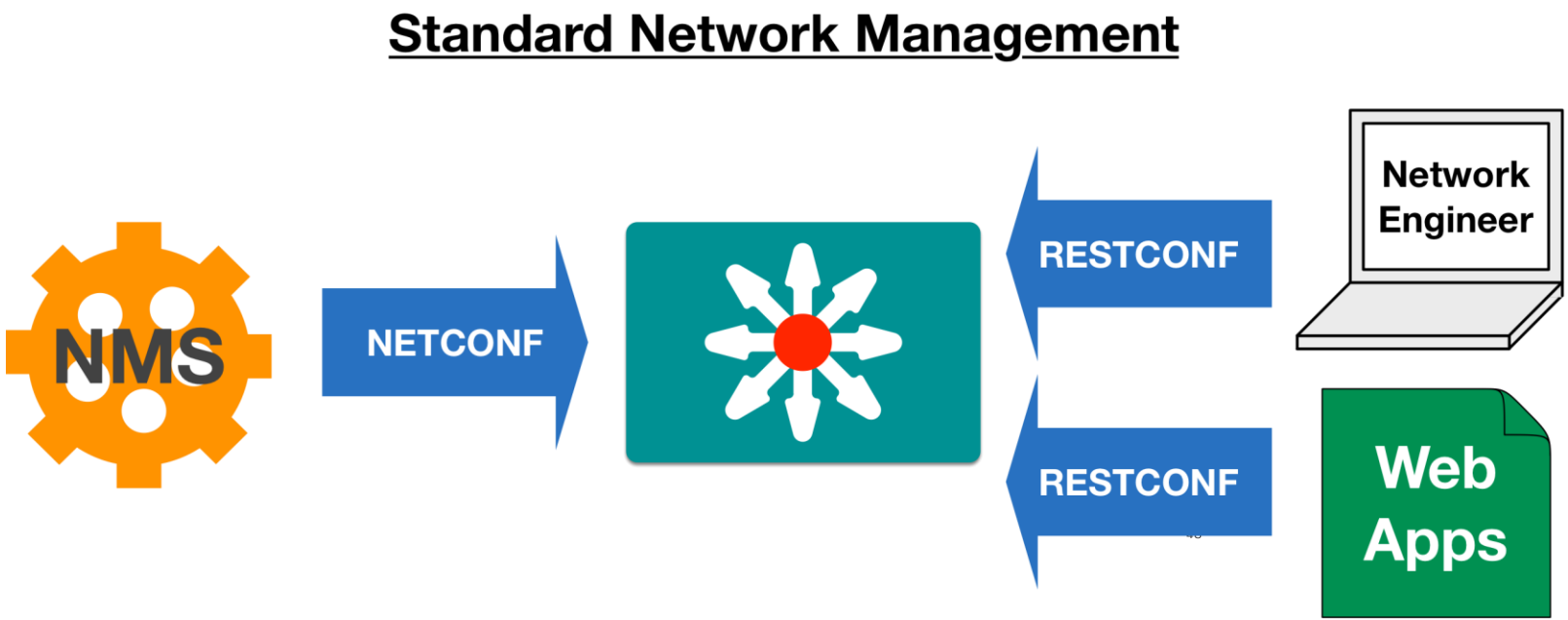
RESTCONF Details

"an HTTP-based protocol that provides a programmatic interface for accessing data defined in YANG..."

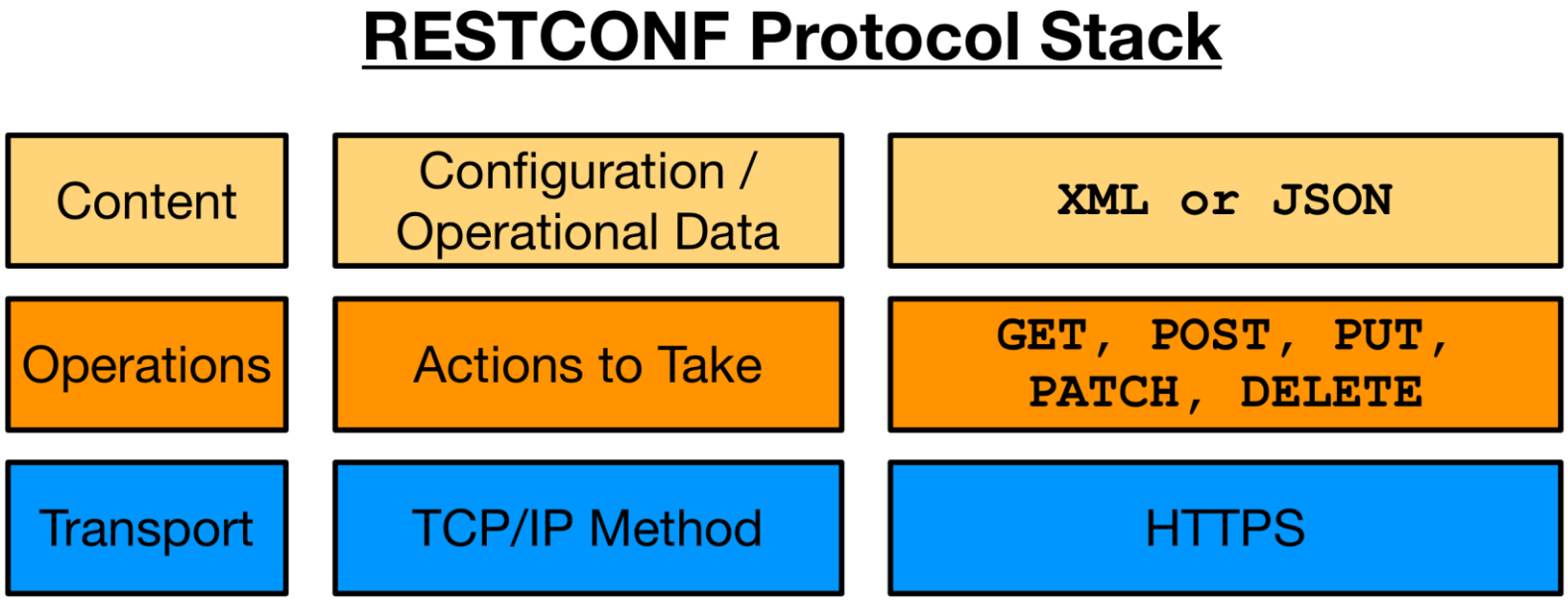
https://tools.ietf.org/html/rfc8040

- RFC 8040 - January 2017
- Uses HTTPS for transport
- Tightly coupled to the YANG data model definitions
- Provides JSON or XML data formats

What about NETCONF?



RESTCONF Protocol Stack & Transport



Content - XML or JSON

HTTP Headers

- **Content-Type**: Specify the type of data being sent from the client
- **Accept**: Specify the type of data being requested by the client

RESTCONF MIME Types

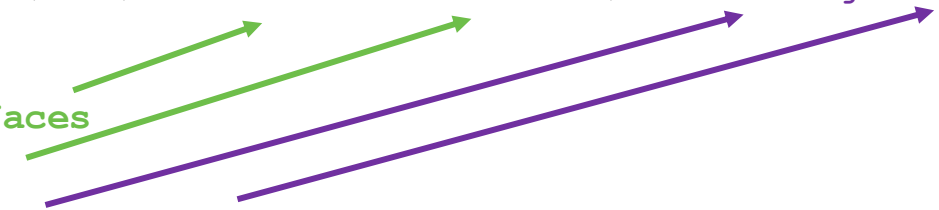
- application/yang-data+json
- application/yang-data+xml

Operations - HTTP CRUD

RESTCONF	NETCONF
GET	<get> , <get-config>
POST	<edit-config> (operation="create")
PUT	<edit-config> (operation="create/replace")
PATCH	<edit-config> (operation="merge")
DELETE	<edit-config> (operation="delete")

URL Creation Review

https://<ADDRESS>/restconf/data/ietf-interfaces:interfaces/interface=GigabitEthernet1?depth=unbounded



```
module: ietf-interfaces
   +--rw interfaces
   |  +--rw interface* [name]
   |     +--rw name                          string
   |     +--rw description?                  string
   |     +--rw type                          identityref
   |     +--rw enabled?                      boolean
   |     +--rw link-up-down-trap-enable?     enumeration
```

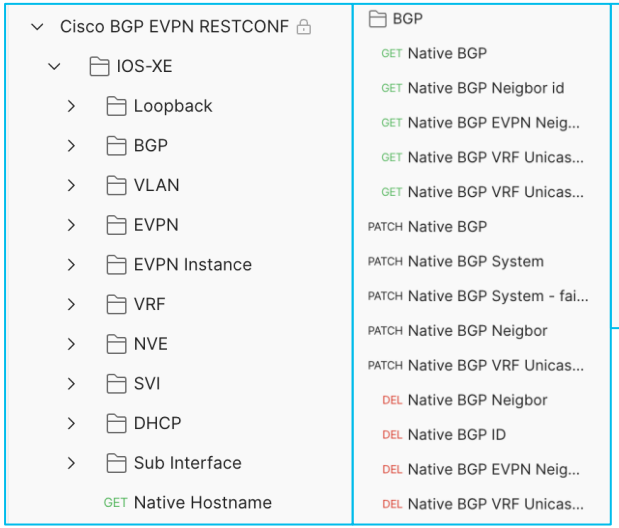
Options Examples:

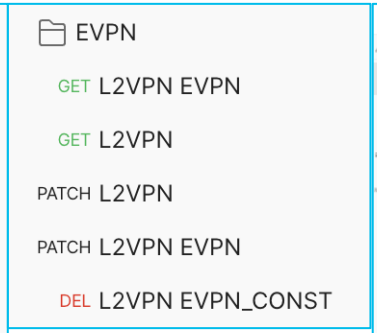
- `depth=unbounded`
  Follow nested models to end. Integer also supported
- `content=[all, config, nonconfig]`
  Query option controls type of data returned.
- `fields=expr`
  Limit what leafs are returned

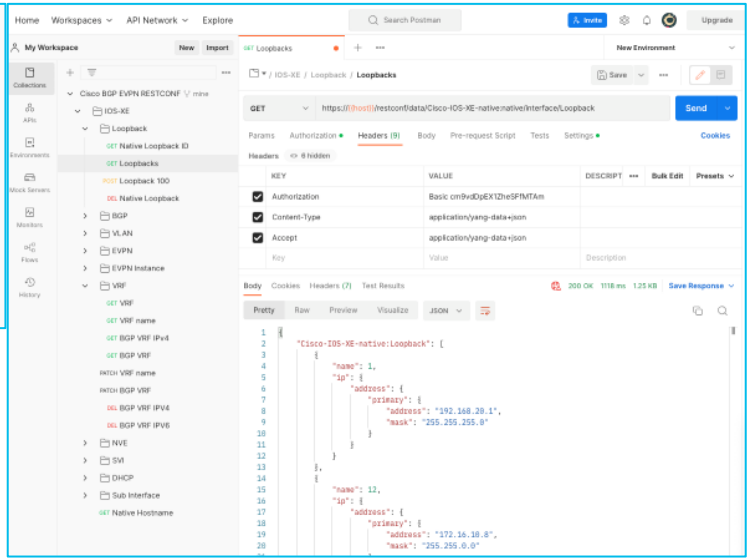
Key:
https://<ADDRESS>/<ROOT>/data>/<[YANG MODULE:]CONTAINER>/<LEAF>[?<OPTIONS>]

Postman collections (BGP EVPN VXLAN)

A Postman collection contains examples for managing EVPN configurations
https://www.postman.com/ciscodevnet/workspace/cisco-devnet-s-public-workspace







RESTCONF
Example with Terraform

Terraform is…

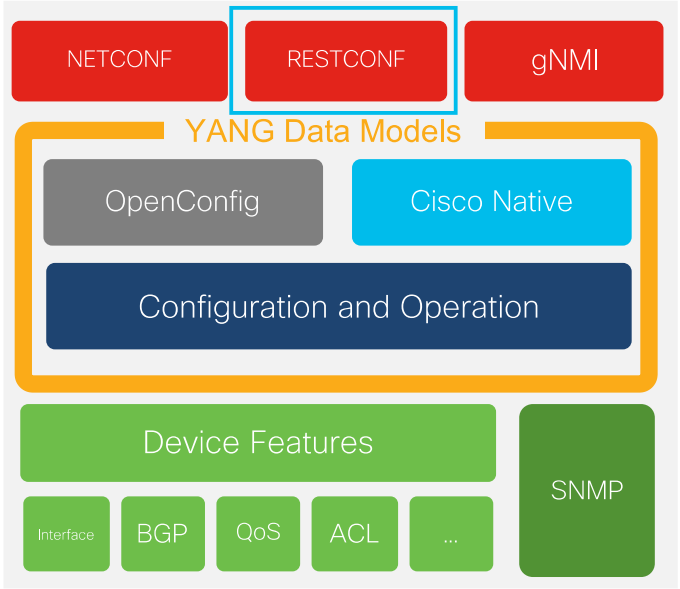
Open-source Infrastructure as Code (IaC) Software Tool providing a consistent CLI workflow to manage hundreds of cloud services. Terraform codifies cloud APIs into declarative configuration files.

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

Resources:

Ask IOS XE Terraform Provider Webex space: https://eurl.io/#PtsT8eJFI
GitHub Provider Examples: https://github.com/CiscoDevNet/terraform-provider-iosxe/
Provider Binary: https://registry.terraform.io/search/providers?namespace=CiscoDevNet
Go Client: https://github.com/CiscoDevNet/iosxe-go-client
Blogs at https://blogs.cisco.com/tag/terraform

Terraform uses the RESTCONF API



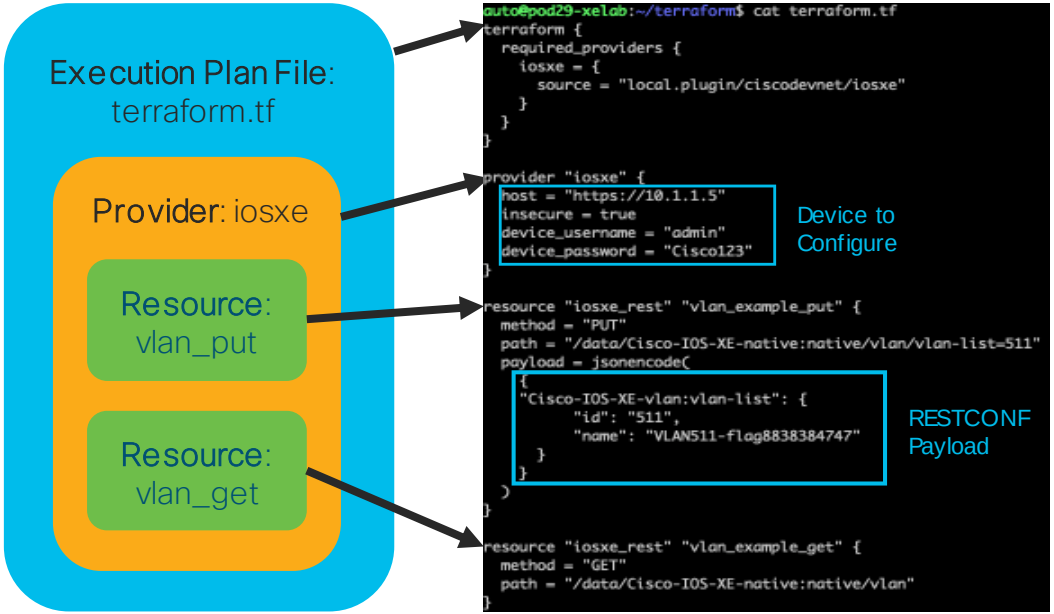
Terraform Terminology

Terraform uses an execution plan file with a provider and resource definitions

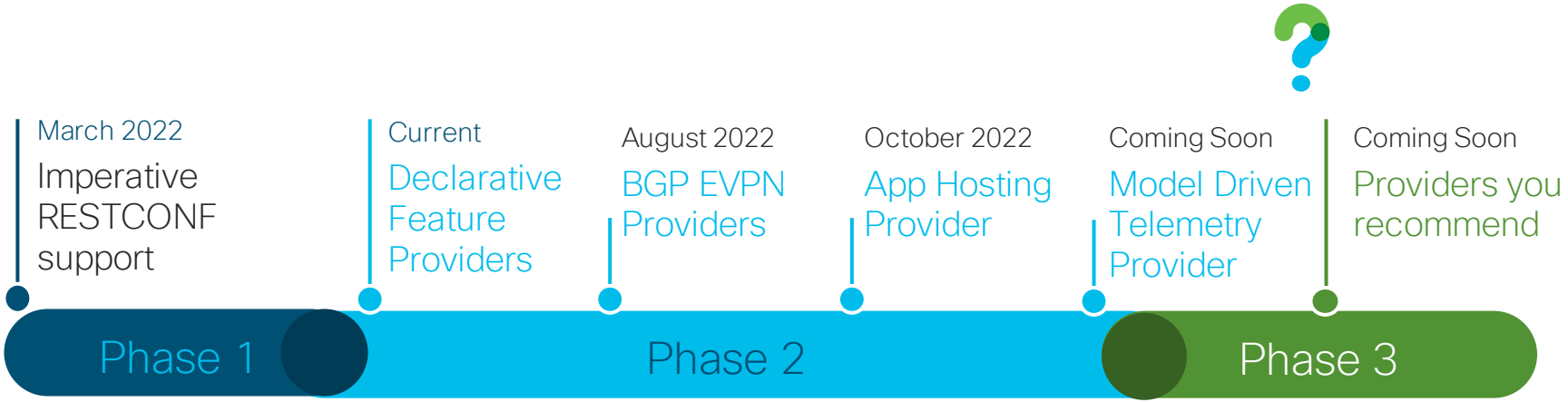
An **execution plan file** defines the provider and resources. It is written in HashiCorp Configuration Language (HCL), similar to JSON, and stored with a .tf extension

A **provider** is a plugin to make a collection of resources accessible

A **resource** (or infrastructure resource) describes one or more infrastructure objects managed by Terraform. With the IOS XE Terraform provider, resources can be considered the same as a configurable feature



Evolution of Terraform Provider

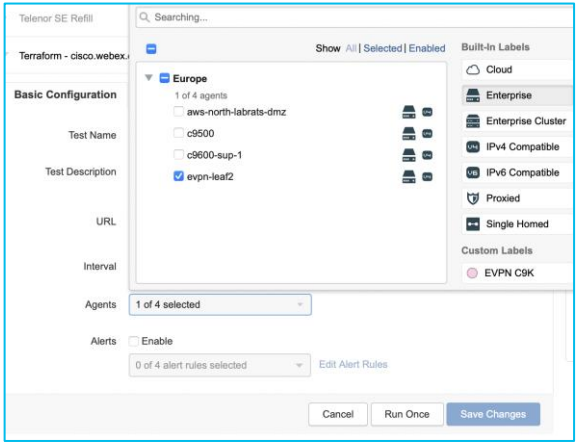


Declarative providers leverage the SDK from the Phase 1 imperative provider

Terraform ThousandEyes lifecycle management

1. Deploy TE agent on switch catalyst 9000
2. Pass variables including the the Agent ID to the ThousandEyes API
3. Create test and attach the C9K TE Agent ID to the test
4. Trigger test to run

```
terraform {
  required_providers {
    ciscoapphosting = {
      source  = "robertcsapo/ciscoapphosting"
      version = "1.0.0"
    }
  }
}

provider "ciscoapphosting" {
  username = var.username
  password = var.password
  insecure = var.insecure
  timeout  = var.timeout
}

resource "ciscoapphosting_app" "app" {
  host = "127.0.0.1"
  image = "https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-4.2.2.cisco.tar"
  app_gigabit_ethernet = "1/0/1"
  vlan_trunk = false
  vlan = 1
  env = {
    TEAGENT_ACCOUNT_TOKEN = "token"
  }
}
```



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

GNMI

gRPC and gNMI

gRPC

Google
Remote Procedure Call

- gRPC's are used when interfacing with the gNMI interface
- Implemented in IOS XE for Model Driven Telemetry (MDT)
- Dial-Out / Configured Telemetry
- Low-latency and scalable
- HTTP/2 transport

gNMI

gRPC
Network Management Interface

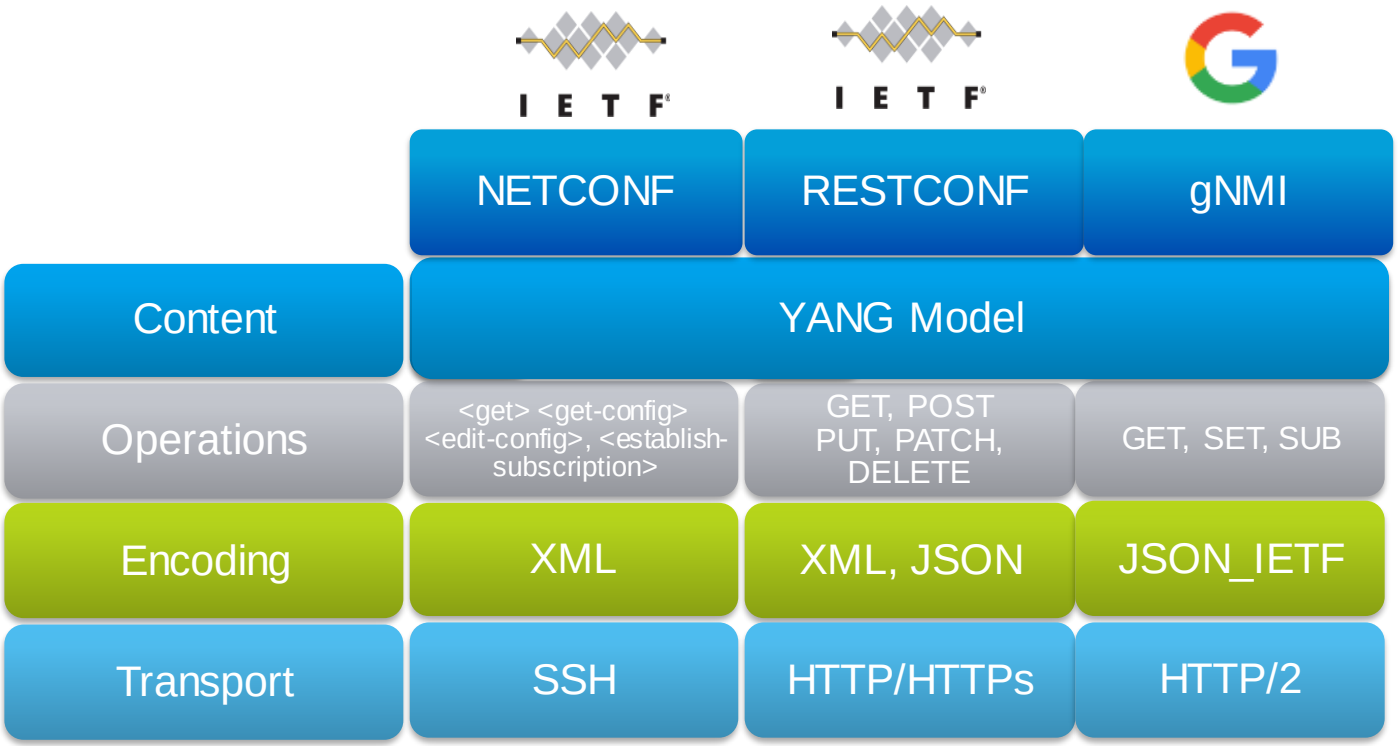
- Network management protocol
- Manage configuration and view operational data of network devices
- Developed by Google
- Modeled using YANG
- Programmability and Dial-In / Dynamic Telemetry (Dial-out soon)

gNMI

gNMI Operations

CAP	Capabilities exchange – list all supported YANG modules	
GET	Prefix	Shortcut if Paths share root paths
	Paths	Defines what info is gathered
	Data Type	OPER, CONFIG, ALL
SET	Like GET	Prefix, Paths
	Type	Update, Replace or Delete
SUBSCRIBE	Subscribe	Prefix, Path

API Interfaces

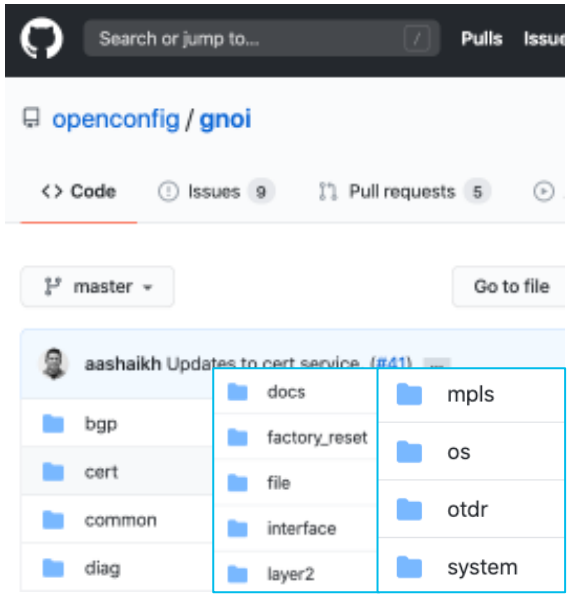


gNOI gRPC Network Operations Interface

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 wipe	`factory-reset`	17.7
File.proto	Not implemented	`copy, delete`	N/A
System.proto	Not implemented	`reload, set boot`	N/A

https://github.com/openconfig/gnoi



Polling Question 2

Quel API de programmation est à privilégier pour upgrader un équipement ?

1) CLI
2) RESTCONF
3) GNOI

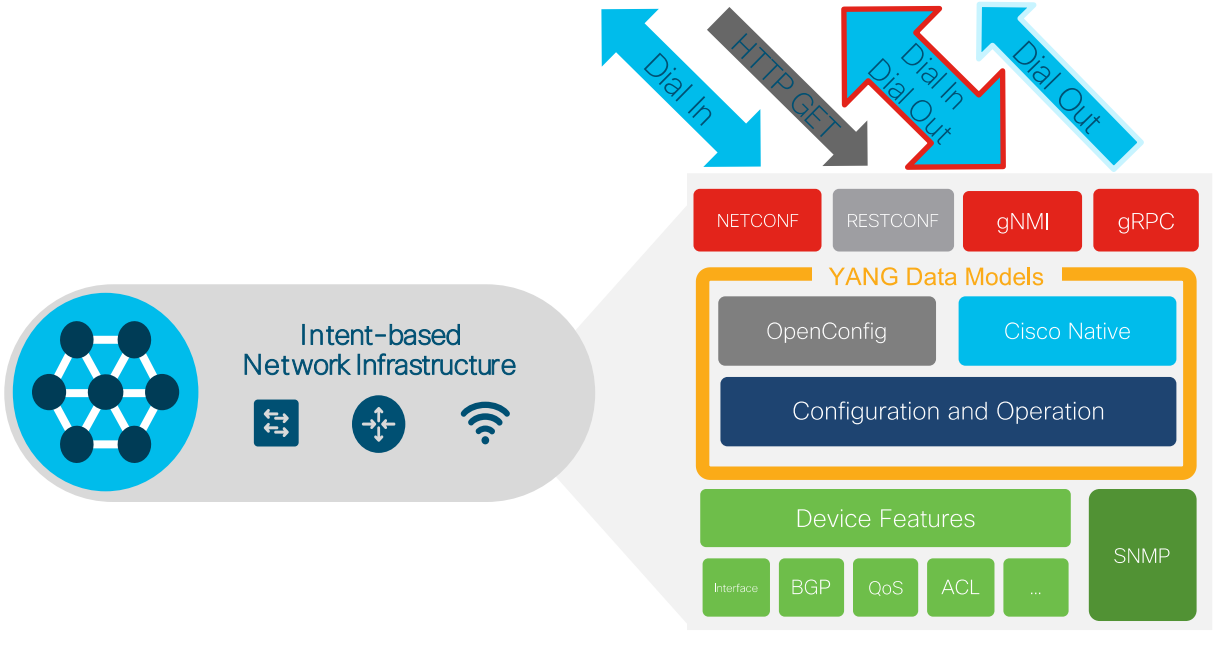
Télémétrie

Model Driven Telemetry Interfaces

Dial In: Collector establishes a connection to the device then subscribes to telemetry (pub/sub)

Out: Telemetry is pushed from the device to the collector based off configuration (push)

Publication / Subscription

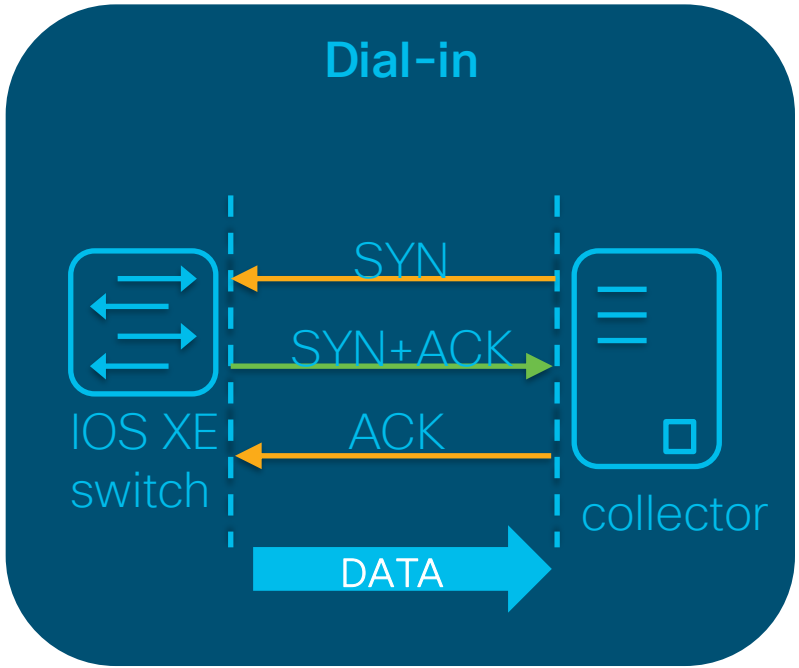


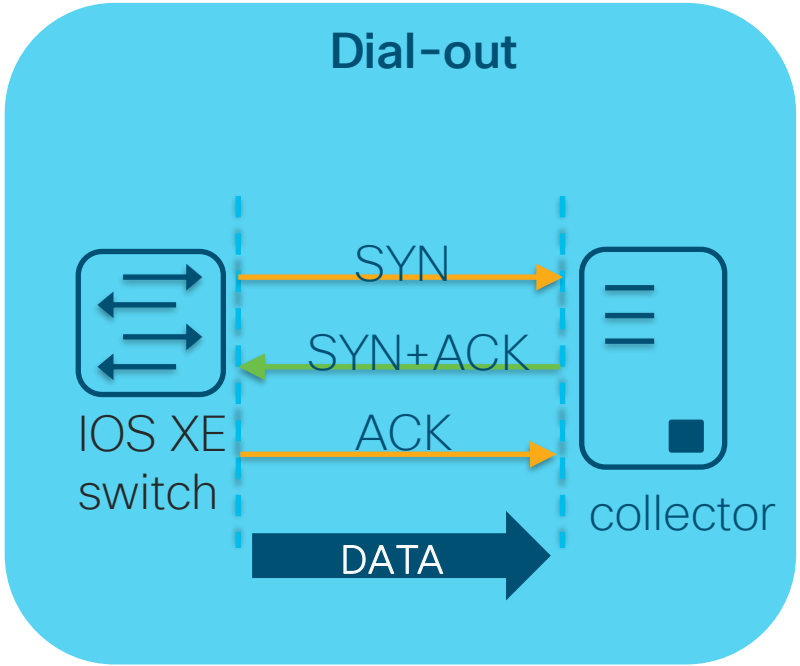
XML, JSON, proto and kvGPB encoding

Consistent YANG data models between interfaces

On-change event and time-based publication options

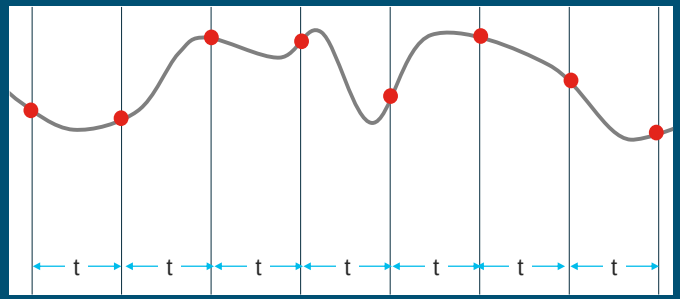
Telemetry Dial-in vs Dial-out



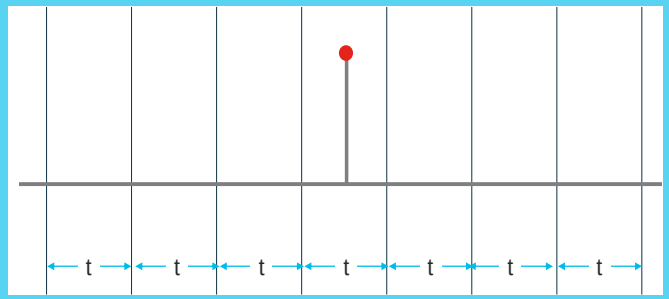


Telemetry Periodic vs On-Change

Periodic



On-Change



Model Driven Telemetry Interface Comparison

	NETCONF	gRPC Dial-Out	gNMI Dial-In	gNMI Dial-Out
Minimum IOS XE Version	16.6	16.10	16.12	17.11
Recommended Version	17.6	17.6	17.6	17.11
Telemetry Direction	Dial-In, IOS XE is server	Dial-Out IOS XE is client	Dial-In IOS XE is server	Dial-Out
Configuration	Dynamic per session	Static per configuration	Dynamic per session	Static
Telemetry Collector	Client	Server	Client	Server
Encoding	XML	KV GPB	JSON_IETF	PROTO + JSON_IETF
Security	SSH + PKI certificate or password	TLS or plain-text	TLS certificate with user authentication	TLS certificate with user authentication
Transport Protocol	SSH	HTTP2	HTTP2	HTTP2
Data Models	YANG	YANG	YANG	YANG

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

IOS XE Model Driven Telemetry

Cisco IOS XE

CLI

...or with...

YANG

pyATS ANSIBLE NSO Terraform

gNMI Dial-In/Dynamic
NETCONF Dial-In

gRPC Dial-Out/Configured

Collector/Receiver
Decodes to text

telegraf

splunk>

elastic

Storage
Time Series Database

InfluxDB

Monitoring
and Visualizations

Grafana

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

Migrating from SNMP to gRPC Dial-Out Telemetry

What is the expected increase in CPU/Memory when using the gRPC Dial-Out telemetry interface, compared to SNMP ?

gRPC adds 2% for each telemetry collector
SNMP adds 6% and an additional 4% for each collector

Testbed:

Ubuntu Linux VM
Telegraf Tooling
SNMP + gRPC

C9300-48

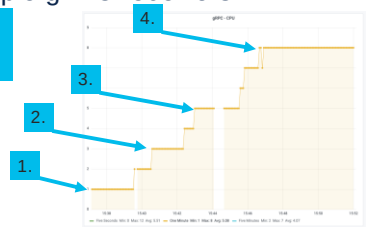
Spirent 48 port traffic generator

CPU impact with multiple gRPC receivers

1. No Interface collection, CPU @ 1%
2. Add gRPC 1 receiver, CPU @ 3% +2
3. Add 2nd gRPC receiver, CPU @ 5% +2
4. Add 3rd gRPC receiver, CPU @ 8% +3

gRPC configuration

```
telemetry ietf subscription 1000
 encoding encode-kvgpb
 filter xpath /interfaces-ios-xe-oper:interfaces/interface
 source-address 128.107.223.252
 stream yang-push
 update-policy periodic 500
 receiver ip address 128.107.223.248 57500 protocol grpc-tcp

telemetry ietf subscription 1001
 encoding encode-kvgpb
 filter xpath /interfaces-ios-xe-oper:interfaces/interface
 source-address 128.107.223.252
 stream yang-push
 update-policy periodic 500
 receiver ip address 128.107.223.210 57500 protocol grpc-tcp

 telemetry ietf subscription 1002
 encoding encode-kvgpb
 filter xpath /interfaces-ios-xe-oper:interfaces/interface
 source-address 128.107.223.252
 stream yang-push
 update-policy periodic 500
 receiver ip address 128.107.223.230 57500 protocol grpc-tcp
```

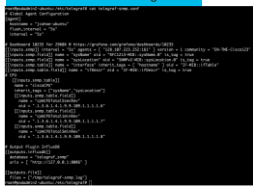


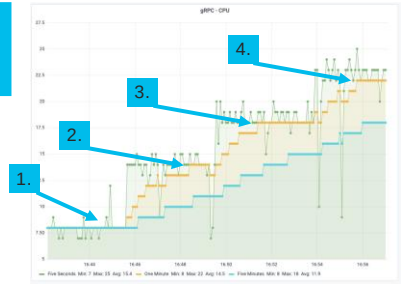
CPU impact multiple SNMP pollers

Telegraf process, or:
$ snmpwalk -v 1 -c EN-TME-Cisco123 128.107.223.252 1.3.6.1.2.1.2.2

1. Baseline CPU @ 8 %
2. Add 1 SNMP CPU @ 14% +6
3. Add 2nd SNMP CPU @ 18% +4
4. Add 3rd SNMP CPU @ 22% +4

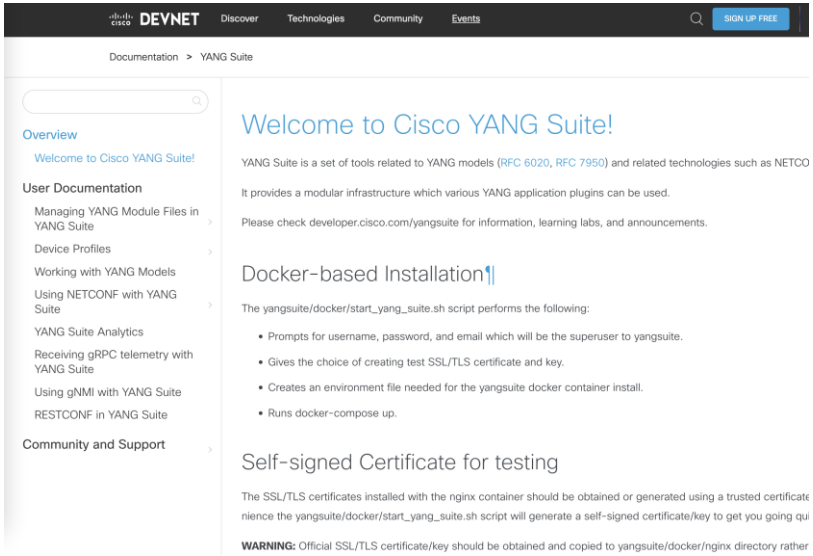
SNMP collection configuration





Démo Yang Suite

Install YANG Suite




```
git clone
https://github.com/CiscoDevNet/yangsuite

yangsuite/docker/start_yang_suite.sh
```

https://developer.cisco.com/docs/yangsuite/

Polling Question 3

Pour quelle raison n'installeriez-vous pas Yang Suite après ce Webinar ?

1) Cela prend trop de temps à installer
2) C'est trop cher
3) Aucune de ces 2 réponses

Pour poursuivre vos recherches…

IOS XE Operational Consistency

	Switching									Wireless				Routing								IOT				SPAG			Cable
	CAT 3650/ 3850	CAT 9200CX	CAT 9200L	CAT 9200	CAT 9300L	CAT 9300/ 9400	CAT 9500	CAT 9500H	CAT 9600	CAT 9100 -EWC	CAT 9800-CL	CAT 9800-L	CAT 9800 -40/80	ISR 1000	ISR 4000	C8300 C8200	CSR 1000v	C8KV	ASR 1000 Fixed	ASR 1000 Modular	C8500 / C8500L	IR 1100	ESR 6300	IE 3x00	ESS 3300	ASR 900 / 920	NCS 520	NCS 4200	cBR-8
Provisioning Automation																													
ZTP	16.5+	17.8		16.12+	16.12+	16.6+		16.8+	16.12+			17.7+	16.12+	16.9+*	16.5+	17.3+	17.2+	17.4+	16.7+	16.8+	17.3+			17.1+	17.1+				
PXE	16.5+	17.8		16.9+	16.9+	16.6+		16.8+	16.11+																				
Model Driven Configuration Management																													
NETCONF	16.5+	17.8	16.9+	16.9+	16.9+	16.6+		16.8+	16.11+	16.12+	16.10+	16.12+	16.10+	16.8+	16.3+	17.3+	16.3+	17.4+	16.3+	16.3+	17.3+	16.10+	17.1+	16.11+	16.11+	16.8+	16.10+	16.8+	16.8+
RESTCONF	16.7+	17.8	16.9+	16.9+	16.9+	16.7+		16.8+	16.11+	16.12+	16.11+	16.12+	16.11+	16.8+	16.6+	17.3+	16.6+	17.4+	16.6+	16.6+	17.3+	16.10+	17.1+	16.11+	16.11+	16.8+	16.10+	16.8+	16.8+
gNMI		17.8	16.12+	16.12+	16.12+	16.8+		16.10+	16.11+		17.8+	17.8+	17.8+	17.8+	17.8+	17.8+		17.12+	17.2	17.2	17.3+	17.2+	17.2+	16.12+	16.12+	17.1	17.1	17.1	16.12+
Model Driven Telemetry																													
NETCONF Dial-In	16.6+	17.8	16.9+	16.9+	16.9+	16.6+		16.8+	16.11+	16.12+ ****				16.8+	16.7+	17.3+	16.10+	17.4+	16.7+	16.8+	17.3+	16.10+	17.1+	16.12+	16.12+	16.9+	16.10+	16.9+	16.9+
gRPC Dial-Out		17.8	16.10+	16.10+	16.10+	16.10+		16.10+	16.11+	N/A	17.6 ***	17.6 ***	17.6+ ***	16.10+	16.10+	17.3+	16.10+	17.4+	16.10+	16.10+	17.3+	16.10+	17.1+	16.12+	16.12+	16.10+	16.10+	16.10+	16.10+
gNMI Dial-In		17.8	16.12+	16.12+	16.12+	16.12+		16.12+	16.12+	N/A	17.6 ***	17.6 ***	17.6+ ***	17.8+	17.8+	17.8+		17.12+	17.2+	17.2+	17.3+	17.2+	17.2+	17.1+	17.1+	17.1+	17.1+	17.1+	16.12+
gNMI Dial-out		17.11+	17.11+	17.11+	17.11+	17.11+		17.11+	17.11+																			17.11+	
gNMI explicit wildcard		17.8	17.5+	17.5+	17.5+	17.5+		17.5+	17.5+																				
Software Image Management																													
GuestShell (On Box Python)	16.5+ *	17.8		16.12+	16.12+	16.6+		16.8+	16.12+			17.8+	16.11+	16.9+ **	16.5+	17.3+	16.7+	17.4+	16.7+	16.8+	17.3+			17.1+	17.1+				
gRPC Network Operations Interface (gNOI)																													
cert.proto		17.8	17.3+	17.3+	17.3+	17.3+		17.3+	17.3+		17.8+	17.8+	17.8+	17.8+	17.8+	17.8+		17.12+	17.8+	17.8+	17.8	17.8+	17.8+	17.8+	17.8+	17.8+	17.8+	17.8+	17.8+
os.proto		?		17.5+	17.5+	17.5+	17.5+	17.5+																					
reset.proto		?	17.7+	17.7+	17.7+	17.7+		17.7+	17.7+				17.7+									17.8+	17.8+	17.8+	17.8+				

Not Planned | Planned | Shipping

Source of truth: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/179/b_179_programmability_cg.html

CISCO DEVNET

developer.cisco.com

Start Now

Videos and Tutorials

Sandbox

Learning Lab

Automation and Code Exchange

Learning and Certifications

Community and Study Groups

developer.cisco.com

Cisco IOS XE Programmability – Booksprint Book

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

Table of Contents

Programmability Configuration Guide

Book Table of Contents

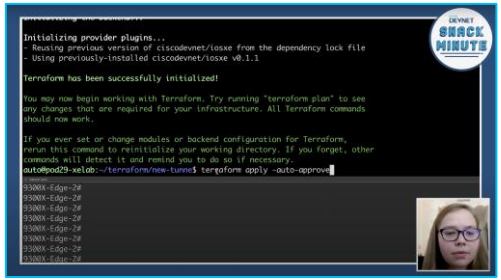
Programmability Configuration Guide, Cisco IOS XE Cupertino 17.9.x

First Published: 2022-08-01

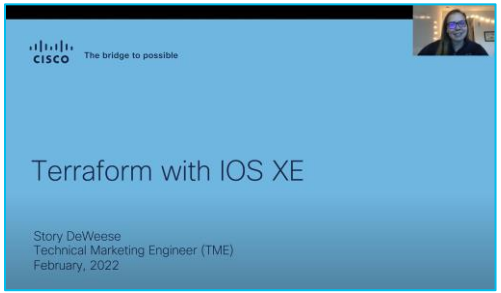
https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/179/b_179_programmability_cg.html

Blog and Resources: Terraform

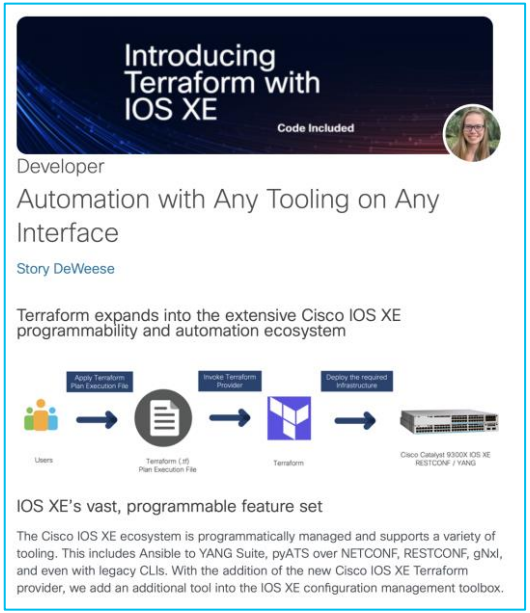
https://github.com/CiscoDevNet/terraform-provider-iosxe/
https://registry.terraform.io/search/providers?namespace=CiscoDevNet



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



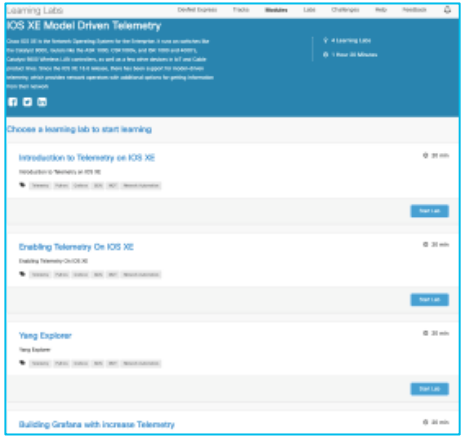
Intro to IOS XE Terraform Provider Video:
https://www.youtube.com/watch?v=GEY_hyXimbA

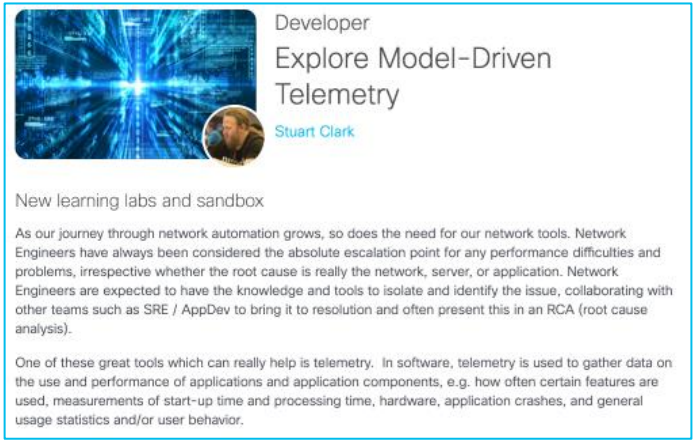


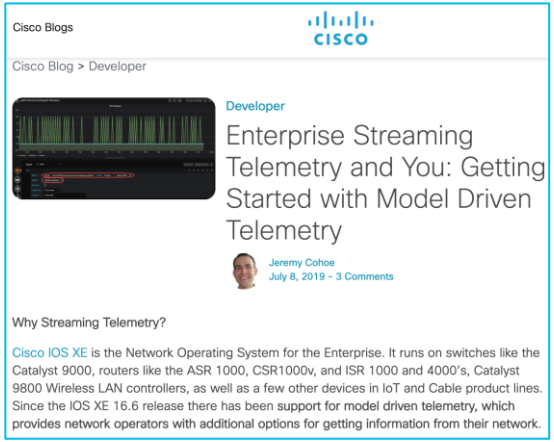
https://blogs.cisco.com/developer/terraformiosxe01

Learning Lab and Blog: Telemetry

https://developer.cisco.com/learning/modules/iosxe_telemetry
https://blogs.cisco.com/developer/model-driven-telemetry-sandbox
https://blogs.cisco.com/developer/getting-started-with-model-driven-telemetry
https://youtu.be/QwwZakkWBng







reseauxblog.cisco.fr

Avez-vous encore des questions ?

Nos experts vous répondent

Si vous avez posé une question sur le panneau de Q&R (Q&A en anglais) ou que vous revenez sur la communauté dans les jours qui suivent notre webinaire, nos experts peuvent encore vous aider !

Participez dans le forum de Ask Me Anything (AMA) avant le 16 décembre 2022.

https://bit.ly/AMA-dec22

Nos réseaux sociaux

LinkedIn

Cisco Community

Twitter

@cisco_support

YouTube

CiscoSupportChannel

Facebook

CiscoSupportCommunity

Faites valoir votre opinion en répondant à notre enquête !

Cliquez sur le lien

https://bit.ly/WEBenq-dec22

The bridge to possible