You make **possible**

Unleash the Power of OpenConfig

Suprita
Sr. Network Test Engineer

DEVLIT-4014

Barcelona | January 27-31, 2020

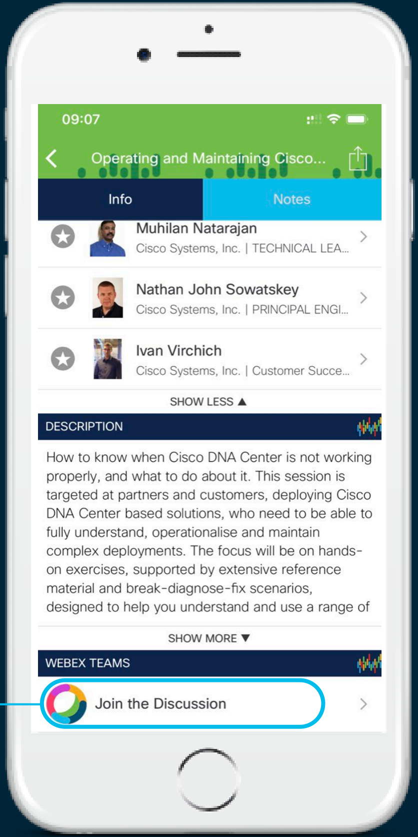
Cisco Webex Teams

Questions?

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

How

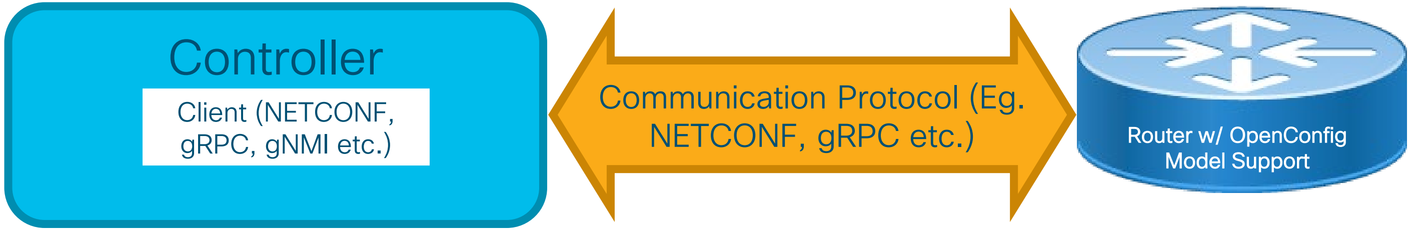
1. Find this session in the Cisco Events Mobile App
2. Click “Join the Discussion”
3. Install Webex Teams or go directly to the team space
4. Enter messages/questions in the team space


09:07
Operating and Maintaining Cisco...
Info
Notes
Muhilan Natarajan
Cisco Systems, Inc. | TECHNICAL LEA...
Nathan John Sowatskey
Cisco Systems, Inc. | PRINCIPAL ENGI...
Ivan Virchich
Cisco Systems, Inc. | Customer Succe...
SHOW LESS ▲
DESCRIPTION
SHOW MORE ▼
WEBEX TEAMS
Join the Discussion

How to know when Cisco DNA Center is not working properly, and what to do about it. This session is targeted at partners and customers, deploying Cisco DNA Center based solutions, who need to be able to fully understand, operationalise and maintain complex deployments. The focus will be on hands-on exercises, supported by extensive reference material and break-diagnose-fix scenarios, designed to help you understand and use a range of

Goals

- Understand Today’s Network Challenges
- Road to Model driven Programmability
- Introduction to OpenConfig for Network Provisioning & Operations.


Controller
Client (NETCONF, gRPC, gNMI etc.)
Communication Protocol (Eg. NETCONF, gRPC etc.)
Router w/ OpenConfig Model Support

Today’s Network Challenges

Todays Network is Open, and accessed by not just Network Engineer but by API developers.

- Overview of the 2002 IAB Network Management Workshop
  - Need for user-friendly Network Configuration protocol
  - Programmatic interface for device configuration.
  - Ability to Configure Services and not just devices.
  - SNMP is not optimized enough for Configuration & DevOps and is not transaction based.

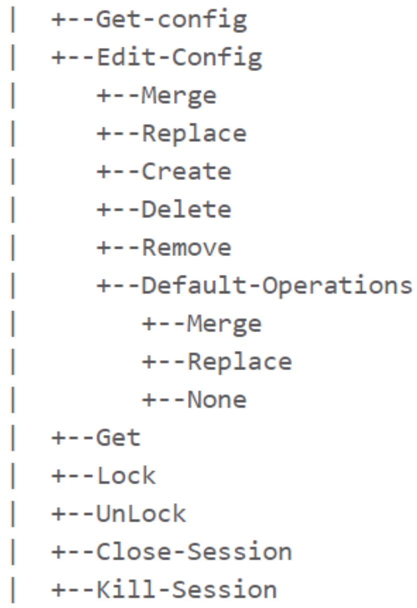
More details : https://tools.ietf.org/html/rfc3535

NETCONF & Yang

NETCONF

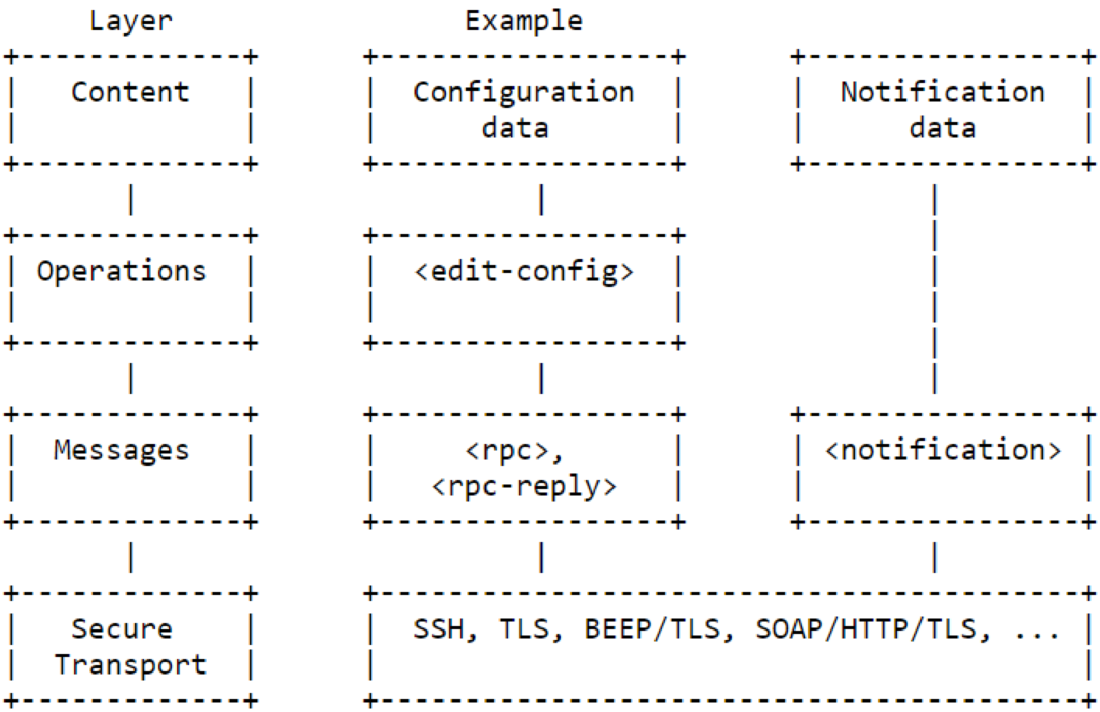
NETwork CONFiguration protocol

- IETF Standard [rfc6241, rfc6242]
  - Transaction based
  - Configure, manipulate & delete the config. [get-config, edit-config & delete-config]
  - Query Operational data [get]
- Operations are realized on top of RPC [remote procedure calls]
  - Uses XML data encoding



```
|   +--Get-config
|   +--Edit-Config
|      +--Merge
|      +--Replace
|      +--Create
|      +--Delete
|      +--Remove
|      +--Default-Operations
|         +--Merge
|         +--Replace
|         +--None
|   +--Get
|   +--Lock
|   +--UnLock
|   +--Close-Session
|   +--Kill-Session
```

NETCONF Protocol Layers



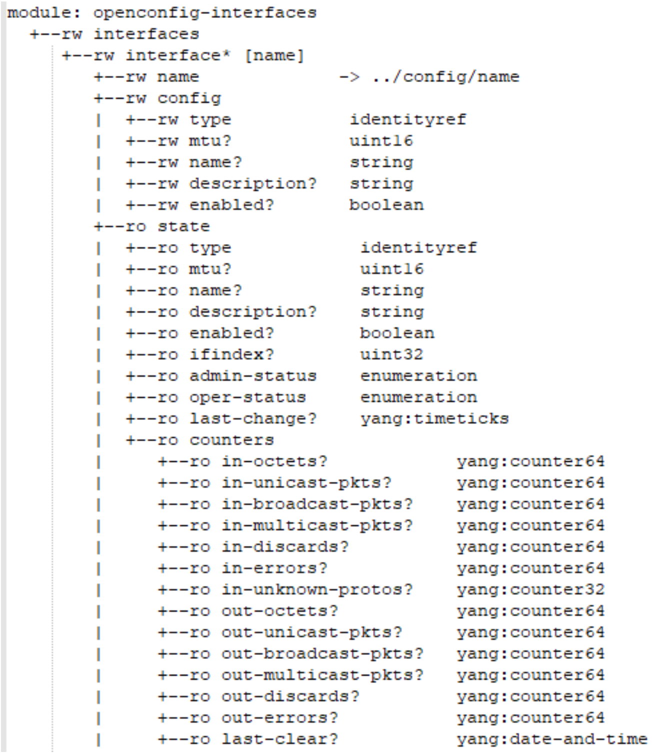
```
         Layer                 Example
    +-------------+      +-----------------+      +----------------+
    |   Content   |      |  Configuration  |      |  Notification  |
    |             |      |      data       |      |      data      |
    +-------------+      +-----------------+      +----------------+
           |                       |                      |
    +-------------+      +-----------------+              |
    | Operations  |      |  <edit-config>  |              |
    |             |      |                 |              |
    +-------------+      +-----------------+              |
           |                       |                      |
    +-------------+      +-----------------+      +----------------+
    |  Messages   |      |     <rpc>,      |      | <notification> |
    |             |      |   <rpc-reply>   |      |                |
    +-------------+      +-----------------+      +----------------+
           |                       |                      |
    +-------------+      +-----------------------------------------+
    |   Secure    |      |  SSH, TLS, BEEP/TLS, SOAP/HTTP/TLS, ... |
    |  Transport  |      |                                         |
    +-------------+      +-----------------------------------------+
```

Source : https://tools.ietf.org/html/rfc6241

YANG

Yet Another Next Generation

- YANG is a data modeling language used to model configuration and state data manipulated by the NETCONF protocol
- IETF Standard [rfc6020]
  - specifically targeted to the needs of configuration management
  - Easy to read, to promote adoption.
  - Provides mechanisms to validate models of configuration data for semantics and syntax.



```
module: openconfig-interfaces
  +--rw interfaces
     +--rw interface* [name]
        +--rw name             -> ../config/name
        +--rw config
        |  +--rw type           identityref
        |  +--rw mtu?           uint16
        |  +--rw name?          string
        |  +--rw description?   string
        |  +--rw enabled?       boolean
        +--ro state
        |  +--ro type            identityref
        |  +--ro mtu?            uint16
        |  +--ro name?           string
        |  +--ro description?    string
        |  +--ro enabled?        boolean
        |  +--ro ifindex?        uint32
        |  +--ro admin-status    enumeration
        |  +--ro oper-status     enumeration
        |  +--ro last-change?    yang:timeticks
        |  +--ro counters
        |     +--ro in-octets?            yang:counter64
        |     +--ro in-unicast-pkts?      yang:counter64
        |     +--ro in-broadcast-pkts?    yang:counter64
        |     +--ro in-multicast-pkts?    yang:counter64
        |     +--ro in-discards?          yang:counter64
        |     +--ro in-errors?            yang:counter64
        |     +--ro in-unknown-protos?    yang:counter32
        |     +--ro out-octets?           yang:counter64
        |     +--ro out-unicast-pkts?     yang:counter64
        |     +--ro out-broadcast-pkts?   yang:counter64
        |     +--ro out-multicast-pkts?   yang:counter64
        |     +--ro out-discards?         yang:counter64
        |     +--ro out-errors?           yang:counter64
        |     +--ro last-clear?           yang:date-and-time
```

NETCONF Client

- Satisfies the prerequisites for an SSH/TLS connection
- Opens/Ends a NETCONF session
- Sends NETCONF RPCs requesting/changing configuration / Operation data
- (Optional)Syntax and Semantics Verification.
- (Optional)locks/unlocks the candidate configuration
- Open source clients:
  - libnetconf: NETCONF library in C
  - ncclient: Python library for NETCONF clients

NETCONF Session

SSH & NETCONF configuration

- Connection-Oriented
  - Transport : SSH [default 830], TLS
- Netconf Client establishes session with server
- Session Establishment
  - Hello message
  - Exchanges capabilities, modules and features
- Session-termination : close/kill

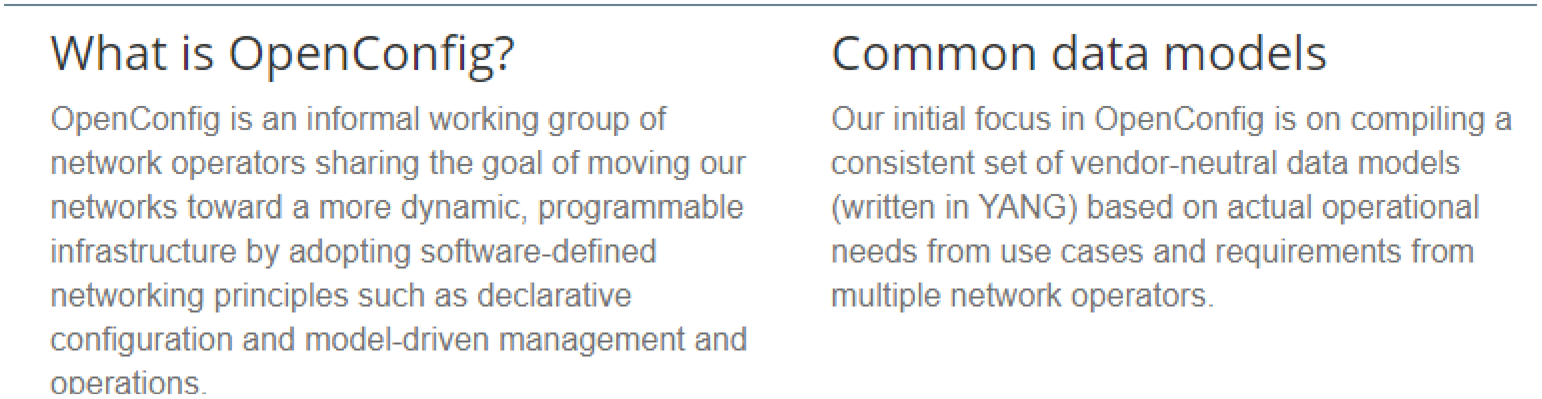
```
!
ssh server v2
ssh server vrf default
ssh server netconf vrf default
!
```

```
!
netconf agent tty
!
netconf-yang agent
 ssh
!
```

OpenConfig Models

What is it ?

http://www.openconfig.net/

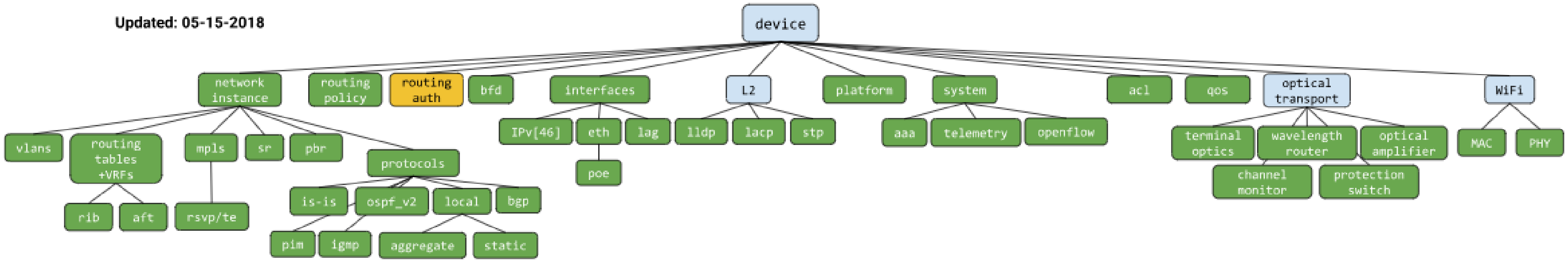


What is OpenConfig?

OpenConfig is an informal working group of network operators sharing the goal of moving our networks toward a more dynamic, programmable infrastructure by adopting software-defined networking principles such as declarative configuration and model-driven management and operations.

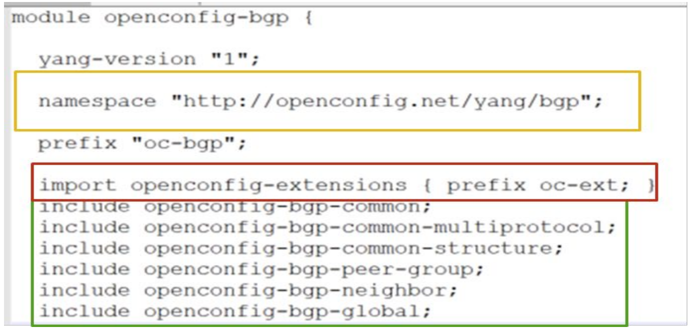
Common data models

Our initial focus in OpenConfig is on compiling a consistent set of vendor-neutral data models (written in YANG) based on actual operational needs from use cases and requirements from multiple network operators.



Updated: 05-15-2018

OpenConfig Models



```
module openconfig-bgp {

  yang-version "1";

  namespace "http://openconfig.net/yang/bgp";

  prefix "oc-bgp";

  import openconfig-extensions { prefix oc-ext; }
  include openconfig-bgp-common;
  include openconfig-bgp-common-multiprotocol;
  include openconfig-bgp-common-structure;
  include openconfig-bgp-peer-group;
  include openconfig-bgp-neighbor;
  include openconfig-bgp-global;
```

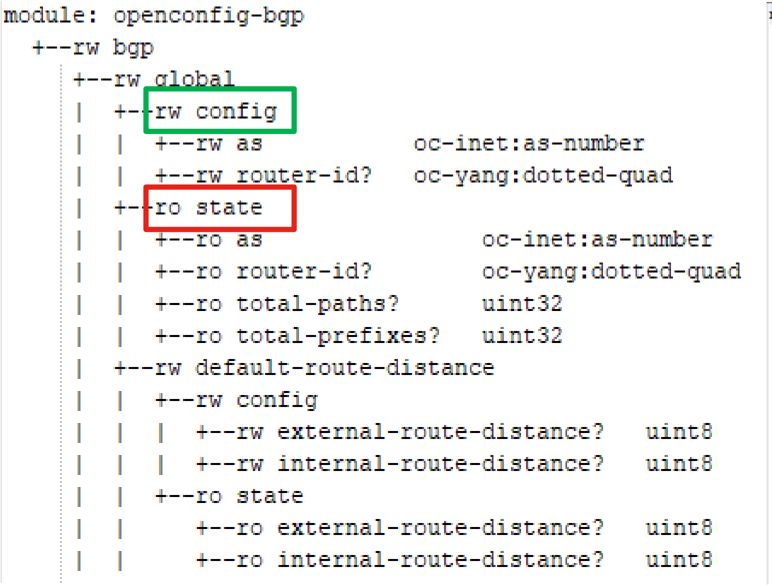
Unique name

External modules with a local prefix

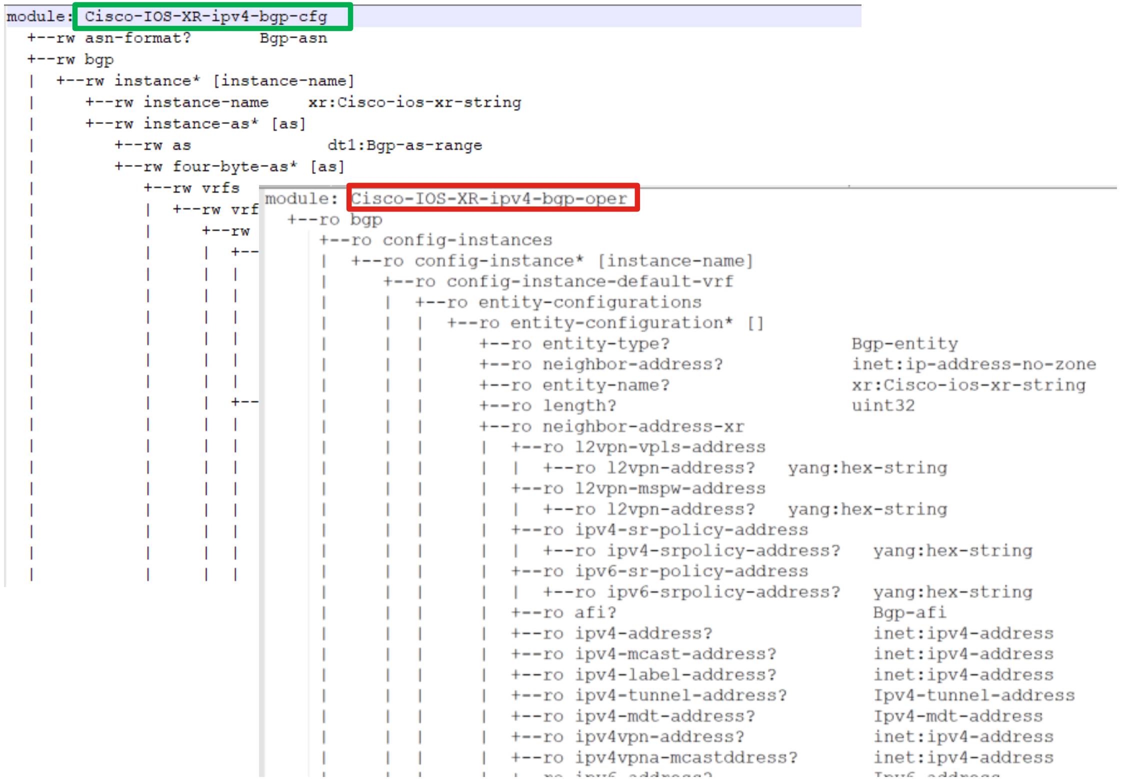
Sub-modules

- Module is the base unit of definition in YANG for eg. openconfig-bgp.yang
- YANG models can augment an existing data model with additional nodes
- "include" statement to include submodules
- "import" statement to reference external modules
- When a definition in an external module is referenced, a locally defined prefix MUST be used, followed by “:”

OpenConfig vs Cisco-defined Model : BGP



```
module: openconfig-bgp
  +--rw bgp
     +--rw global
     |  +--rw config
     |  |  +--rw as              oc-inet:as-number
     |  |  +--rw router-id?      oc-yang:dotted-quad
     |  +--ro state
     |  |  +--ro as                    oc-inet:as-number
     |  |  +--ro router-id?            oc-yang:dotted-quad
     |  |  +--ro total-paths?          uint32
     |  |  +--ro total-prefixes?       uint32
     |  +--rw default-route-distance
     |  |  +--rw config
     |  |  |  +--rw external-route-distance?   uint8
     |  |  |  +--rw internal-route-distance?   uint8
     |  |  +--ro state
     |  |     +--ro external-route-distance?   uint8
     |  |     +--ro internal-route-distance?   uint8
```

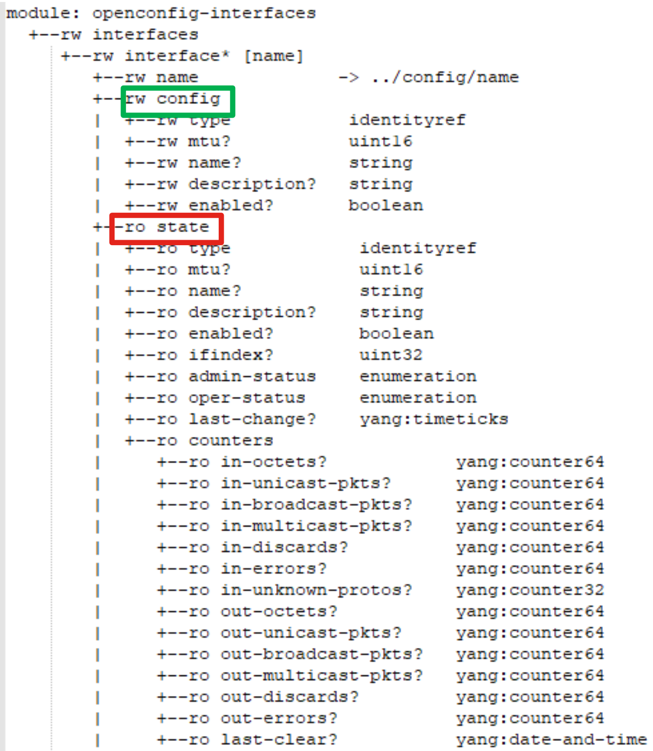


```
module: Cisco-IOS-XR-ipv4-bgp-cfg
  +--rw asn-format?          Bgp-asn
  +--rw bgp
  |  +--rw instance* [instance-name]
  |     +--rw instance-name     xr:Cisco-ios-xr-string
  |     +--rw instance-as* [as]
  |        +--rw as                dt1:Bgp-as-range
  |        +--rw four-byte-as* [as]
  |           +--rw vrfs
  |           |  +--rw vrf
  |           |     +--rw
  |           |     |  +--
  |           |     |  |
  |           |     |  |
  |           |     |  |
  |           |     |  |
  |           |     |  |
  |           |     |  |
  |           |     |  +--
  |           |     |  |
  |           |     |  |
  |           |     |  |
  |           |     |  |
  |           |     |  |
  |           |     |  |
  |           |     |  |
  |           |     |  |
```

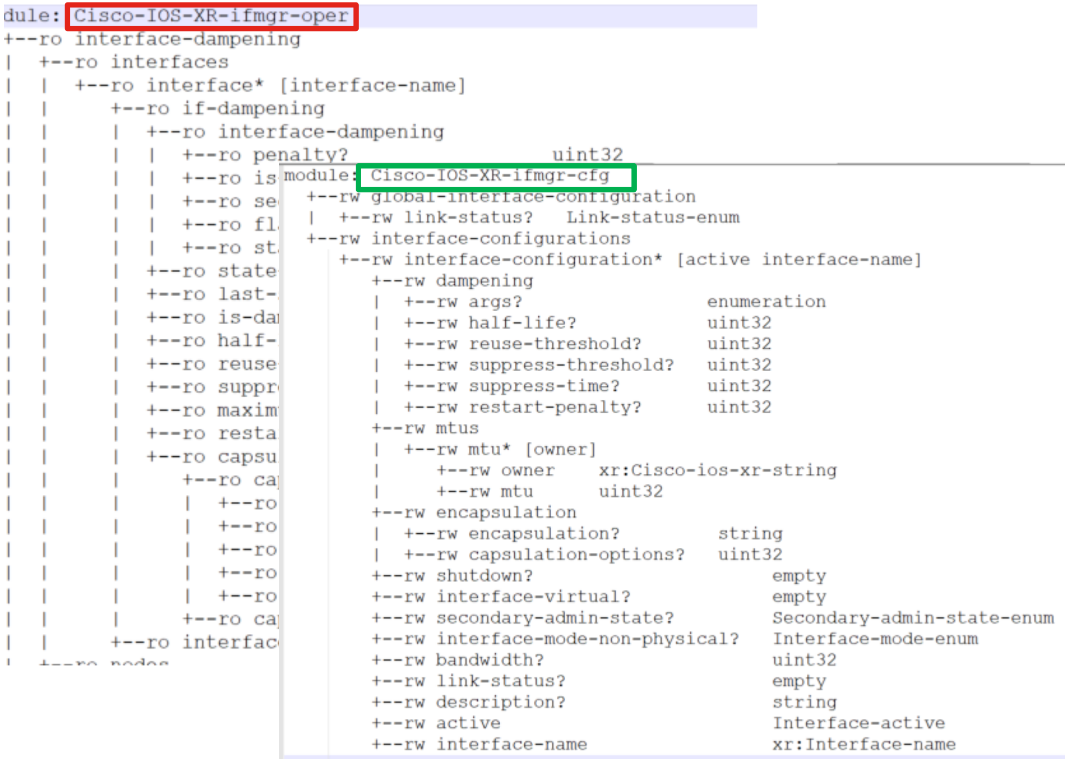
```
module: Cisco-IOS-XR-ipv4-bgp-oper
  +--ro bgp
     +--ro config-instances
     |  +--ro config-instance* [instance-name]
     |     +--ro config-instance-default-vrf
     |     |  +--ro entity-configurations
     |     |  |  +--ro entity-configuration* []
     |     |  |     +--ro entity-type?              Bgp-entity
     |     |  |     +--ro neighbor-address?         inet:ip-address-no-zone
     |     |  |     +--ro entity-name?              xr:Cisco-ios-xr-string
     |     |  |     +--ro length?                   uint32
     |     |  |     +--ro neighbor-address-xr
     |     |  |     |  +--ro l2vpn-vpls-address
     |     |  |     |  |  +--ro l2vpn-address?    yang:hex-string
     |     |  |     |  +--ro l2vpn-mspw-address
     |     |  |     |  |  +--ro l2vpn-address?    yang:hex-string
     |     |  |     |  +--ro ipv4-sr-policy-address
     |     |  |     |  |  +--ro ipv4-srpolicy-address?   yang:hex-string
     |     |  |     |  +--ro ipv6-sr-policy-address
     |     |  |     |  |  +--ro ipv6-srpolicy-address?   yang:hex-string
     |     |  |     |  +--ro afi?                        Bgp-afi
     |     |  |     |  +--ro ipv4-address?               inet:ipv4-address
     |     |  |     |  +--ro ipv4-mcast-address?         inet:ipv4-address
     |     |  |     |  +--ro ipv4-label-address?         inet:ipv4-address
     |     |  |     |  +--ro ipv4-tunnel-address?        Ipv4-tunnel-address
     |     |  |     |  +--ro ipv4-mdt-address?           Ipv4-mdt-address
     |     |  |     |  +--ro ipv4vpn-address?            inet:ipv4-address
     |     |  |     |  +--ro ipv4vpna-mcastddress?       inet:ipv4-address
```

Model Snipped for representation on slide

OpenConfig vs Cisco-defined Model : Interface



```
module: openconfig-interfaces
   +--rw interfaces
      +--rw interface* [name]
         +--rw name                  -> ../config/name
         +--rw config
         |  +--rw type              identityref
         |  +--rw mtu?              uint16
         |  +--rw name?             string
         |  +--rw description?      string
         |  +--rw enabled?          boolean
         +--ro state
         |  +--ro type               identityref
         |  +--ro mtu?               uint16
         |  +--ro name?              string
         |  +--ro description?       string
         |  +--ro enabled?           boolean
         |  +--ro ifindex?           uint32
         |  +--ro admin-status       enumeration
         |  +--ro oper-status        enumeration
         |  +--ro last-change?       yang:timeticks
         |  +--ro counters
         |     +--ro in-octets?             yang:counter64
         |     +--ro in-unicast-pkts?       yang:counter64
         |     +--ro in-broadcast-pkts?     yang:counter64
         |     +--ro in-multicast-pkts?     yang:counter64
         |     +--ro in-discards?           yang:counter64
         |     +--ro in-errors?             yang:counter64
         |     +--ro in-unknown-protos?     yang:counter32
         |     +--ro out-octets?            yang:counter64
         |     +--ro out-unicast-pkts?      yang:counter64
         |     +--ro out-broadcast-pkts?    yang:counter64
         |     +--ro out-multicast-pkts?    yang:counter64
         |     +--ro out-discards?          yang:counter64
         |     +--ro out-errors?            yang:counter64
         |     +--ro last-clear?            yang:date-and-time
```



```
dule: Cisco-IOS-XR-ifmgr-oper
+--ro interface-dampening
|  +--ro interfaces
|  |  +--ro interface* [interface-name]
|  |     +--ro if-dampening
|  |     |  +--ro interface-dampening
|  |     |  |  +--ro penalty?                 uint32
|  |     |  |  +--ro is
|  |     |  |  +--ro se
|  |     |  |  +--ro fl
|  |     |  |  +--ro st
|  |     |  +--ro state
|  |     |  +--ro last-
|  |     |  +--ro is-da
|  |     |  +--ro half-
|  |     |  +--ro reuse
|  |     |  +--ro suppr
|  |     |  +--ro maxim
|  |     |  +--ro resta
|  |     |  +--ro capsu
|  |     |     +--ro ca
|  |     |     |  +--ro
|  |     |     |  +--ro
|  |     |     |  +--ro
|  |     |     |  +--ro
|  |     |     |  +--ro
|  |     |     +--ro ca
|  |     +--ro interfac
|  +--ro nodes
```

```
module: Cisco-IOS-XR-ifmgr-cfg
  +--rw global-interface-configuration
  |  +--rw link-status?    Link-status-enum
  +--rw interface-configurations
     +--rw interface-configuration* [active interface-name]
        +--rw dampening
        |  +--rw args?                   enumeration
        |  +--rw half-life?              uint32
        |  +--rw reuse-threshold?        uint32
        |  +--rw suppress-threshold?     uint32
        |  +--rw suppress-time?          uint32
        |  +--rw restart-penalty?        uint32
        +--rw mtus
        |  +--rw mtu* [owner]
        |     +--rw owner      xr:Cisco-ios-xr-string
        |     +--rw mtu        uint32
        +--rw encapsulation
        |  +--rw encapsulation?          string
        |  +--rw capsulation-options?    uint32
        +--rw shutdown?                            empty
        +--rw interface-virtual?                   empty
        +--rw secondary-admin-state?               Secondary-admin-state-enum
        +--rw interface-mode-non-physical?         Interface-mode-enum
        +--rw bandwidth?                           uint32
        +--rw link-status?                         empty
        +--rw description?                         string
        +--rw active                               Interface-active
        +--rw interface-name                       xr:Interface-name
```

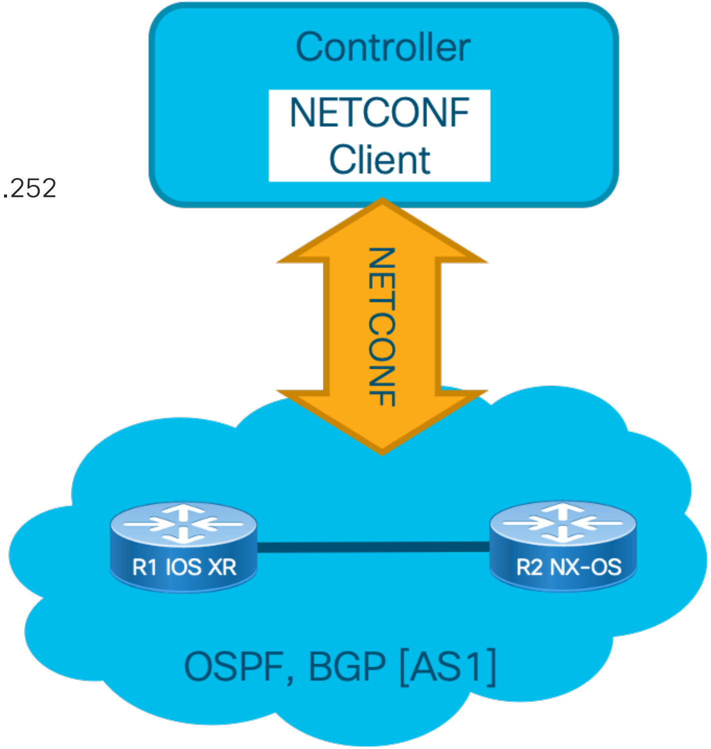
Model Snipped for representation on slide

pYANG Demo

OpenConfig Demo

R1 IOS XR Config

```
interface GigabitEthernet0/0/0/0
ipv4 address 100.1.2.1 255.255.255.252
!
router bgp 1
bgp router-id 1.1.1.1
address-family ipv4 unicast
!
neighbor 100.1.2.2
 remote-as 1
 address-family ipv4 unicast
```



R2 NX OS Config

```
interface Ethernet1/1
 ip address 100.1.2.2/30
!
router bgp 1
 router-id 2.2.2.2
 address-family ipv4 unicast
 neighbor 100.1.2.1
  remote-as 1
  address-family ipv4 unicast
```

Referred Sessions & Further Reading

- DEVWKS-1644 - Building IP Core Network with OpenConfig
- DEVWKS-1381 - Introduction to Using gRPC-based Protocol for Model-Driven Management of IOS-XR
- LTRSPG-2601 - Cisco IOS XR Programmability
- OpenConfig references
  - http://www.openconfig.net/
  - https://github.com/openconfig
- pYANG : https://github.com/mbj4668/pyang
- NETCONF & YANG RFCs
  - RFC 3535 : Overview of the 2002 IAB Network Management Workshop
  - RFC 6241 : Network Configuration Protocol
  - RFC 6242 : Using the NETCONF Protocol over Secure Shell
  - RFC 6020 : YANG - A Data Modeling Language for the Network Configuration Protocol
- Open Source Netconf Client :
  - libnetconf: NETCONF library in C intended for building NETCONF clients and servers.
  - ncclient: Python library for NETCONF clients

Learn more about the new DevNet Certifications and how you can prepare now!

	Associate Level	Specialist Level	Professional Level	Expert Level
Engineering	CISCO CERTIFIED CCNA	CISCO CERTIFIED SPECIALIST	CISCO CERTIFIED CCNP	CISCO CERTIFIED CCIE
Software	CISCO CERTIFIED DEVNET Associate	CISCO CERTIFIED DEVNET SPECIALIST	CISCO CERTIFIED DEVNET Professional	CISCO CERTIFIED DEVNET Expert Future Offering

Start Here | Upcoming Cisco DevNet Certifications

- Start at **Meet DevNet**

  DEVNET-2864: Getting ready for Cisco DevNet Certifications
  Offered daily at 9am, 1pm & 4pm at Meet DevNet

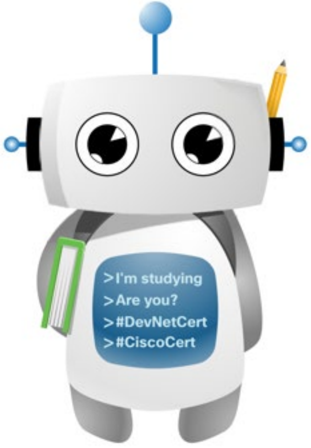
- Attend a **brownbag session**

  DEVNET-4099: DevNet Certifications: Bringing software practices & software skills to networking
  Offered daily 12:15-12:45 in the DevNet Zone Theater

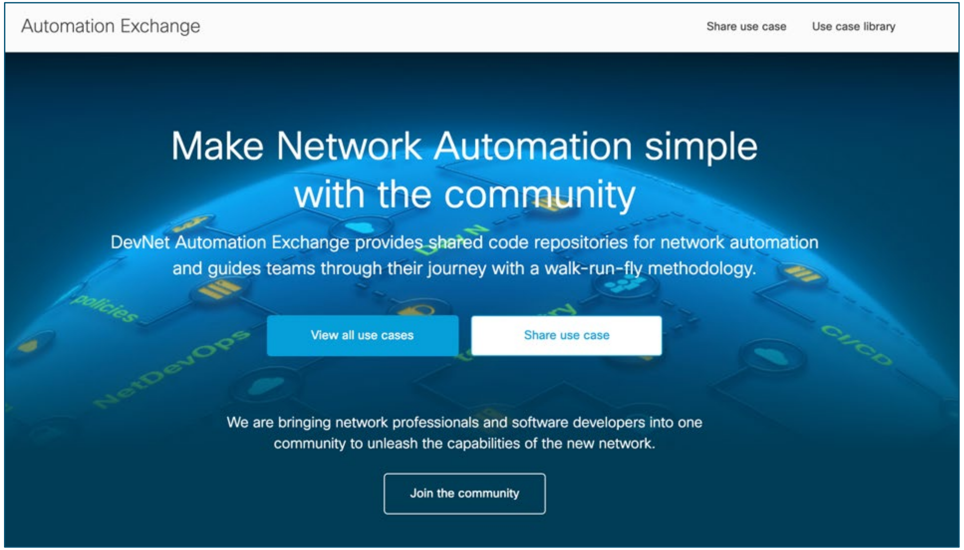
- Visit the **Learning@Cisco** booth

- Scan this code to **sign up** for the latest updates or go to http://cs.co/20eur02

DEVNET



Find shared code repositories of use cases for network automation & more



Automation Exchange

Make Network Automation simple with the community

DevNet Automation Exchange provides shared code repositories for network automation and guides teams through their journey with a walk-run-fly methodology.

We are bringing network professionals and software developers into one community to unleash the capabilities of the new network.

Don't miss our 5 **Automate Infrastructure demos** in the DevNet Zone!

Scan this code or go to the URL to **learn more**

DEVNET

http://cs.co/20eur01

Start at **Meet DevNet**

DEVNET-3010 [a-j] Learn how to make Network Automation Simple with the Community

Offered Monday 2pm & 5pm, Tuesday & Wednesday 10am, 2pm & 5pm, and Thursday 10am & 5pm at Meet DevNet

Complete your online session survey

- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (starting on Thursday) to receive your Cisco Live t-shirt.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Content Catalog on ciscolive.com/emea.

Cisco Live sessions will be available for viewing on demand after the event at ciscolive.com.

Continue your education

- Demos in the Cisco Showcase
- Walk-In Labs
- Meet the Engineer 1:1 meetings
- Related sessions

cisco Live!

Thank you

You make **possible**