Atomic Configuration Replace (ACR)

with Cisco IOS XE

Story DeWeese
Technical Marketing Engineer
@storydeweese

Ashil Parekh
Product Management
@Ashil34571977

Agenda

01 Misconfiguration Consequences

02 Intro to Atomic Config Replace (ACR)

03 ACR Demos

04 NETCONF CLI RPC

05 Hands-On Opportunities

06 Resources

Cisco Webex App

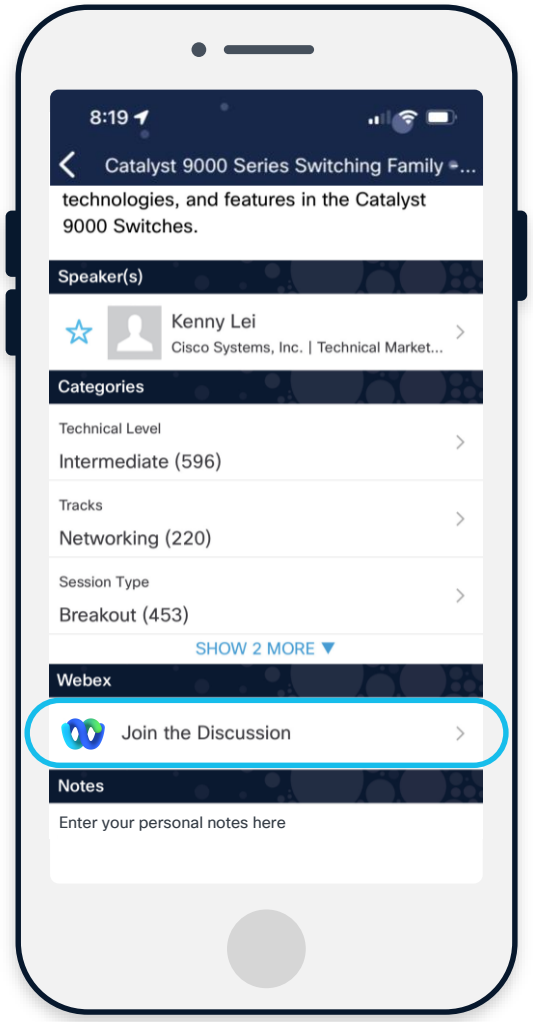
Questions?

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

How

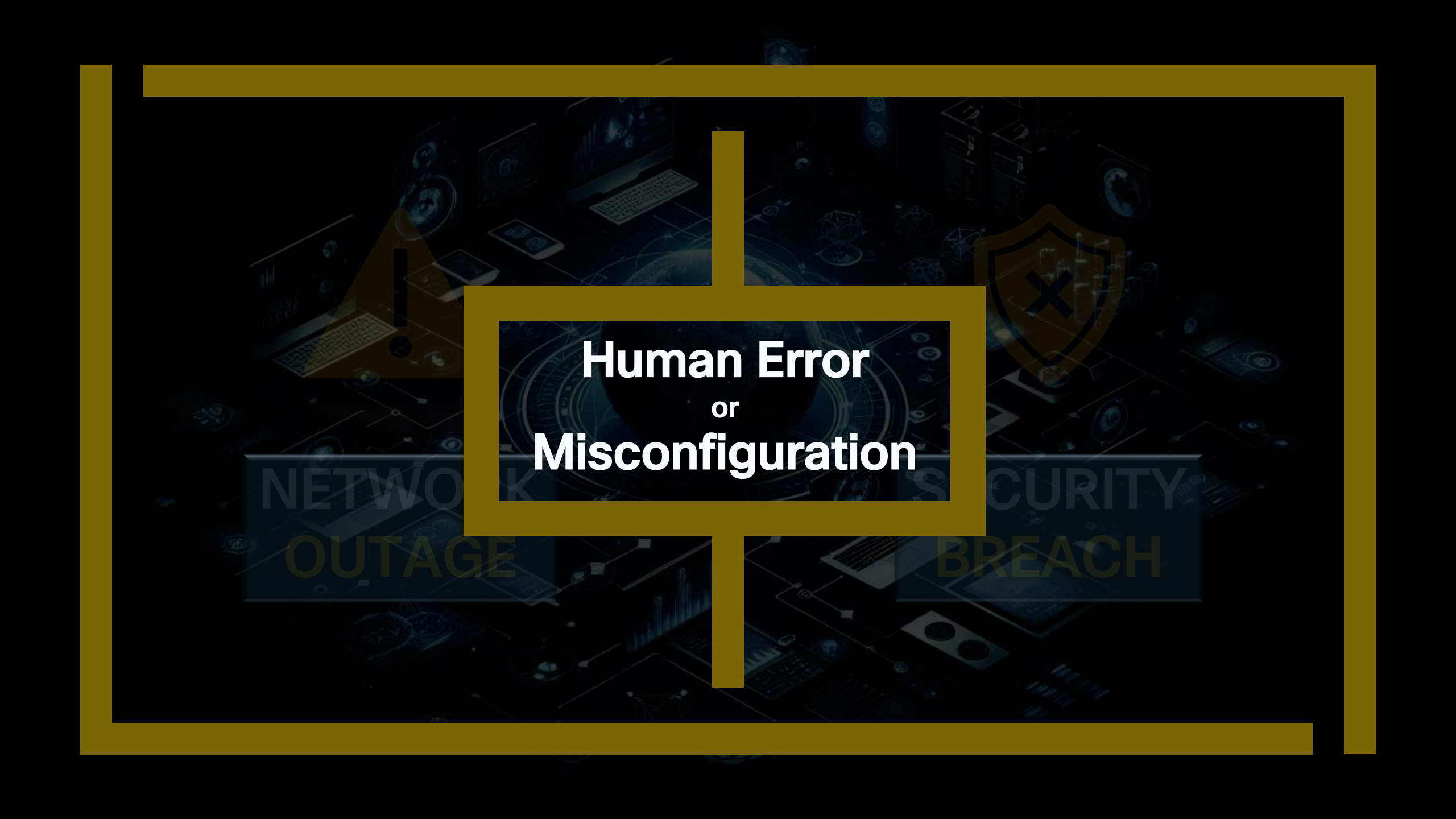
1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until June 13, 2025.



Misconfiguration Consequences


NETWORK
OUTAGE
SECURITY
BREACH


Human Error
or
Misconfiguration
NETWORK
OUTAGE
SECURITY
BREACH

State of Network Misconfigurations

45%

Network-related outages are caused by configuration failure. [1]

22%

Data breaches are caused by human errors. [2]

$1 Million

25% respondents said their most recent outage cost more than $1 million. [1]

$4.8 Million

Is the global average cost of a data breach, increased by 10% compared to the previous year. [2]

[1] - Annual outages analysis 2023 – Uptime

[2] - Cost of a Data Breach Report 2024 – IBM

Business Impact of Misconfiguration

$60 Million

A fast-food chain lost over $60 million in revenue due to a global IT outage caused by misconfiguration.

$350 Million

Multinational telecom company lost over $350 million in revenue due to network-wide outage caused by a config error.

What problems do you face during device config?

Why It’s Time for a Change?

The Complexity and Struggles of CLI-Based Automation

Creating Configurations That Just Work

Error Handling & Rollback

Scalability & Operational Efficiency

Why It's Time for a Change?

Creating Configurations That Just Work

1. Declarative Config Management

Terraform

2. Two-Phase Commit

Verify before apply

Error Handling & Rollback

1. Atomic Config Replace

Transcactional Config Change

2. Multiple Rollback Options

Instant Rollback
Post Deployment Rollback

Scalability & Operational Efficiency

1. Programmable Interfaces

NETCONF RESTCONF gNMI gRPC OPENCONFIG

APIs & Yang Models

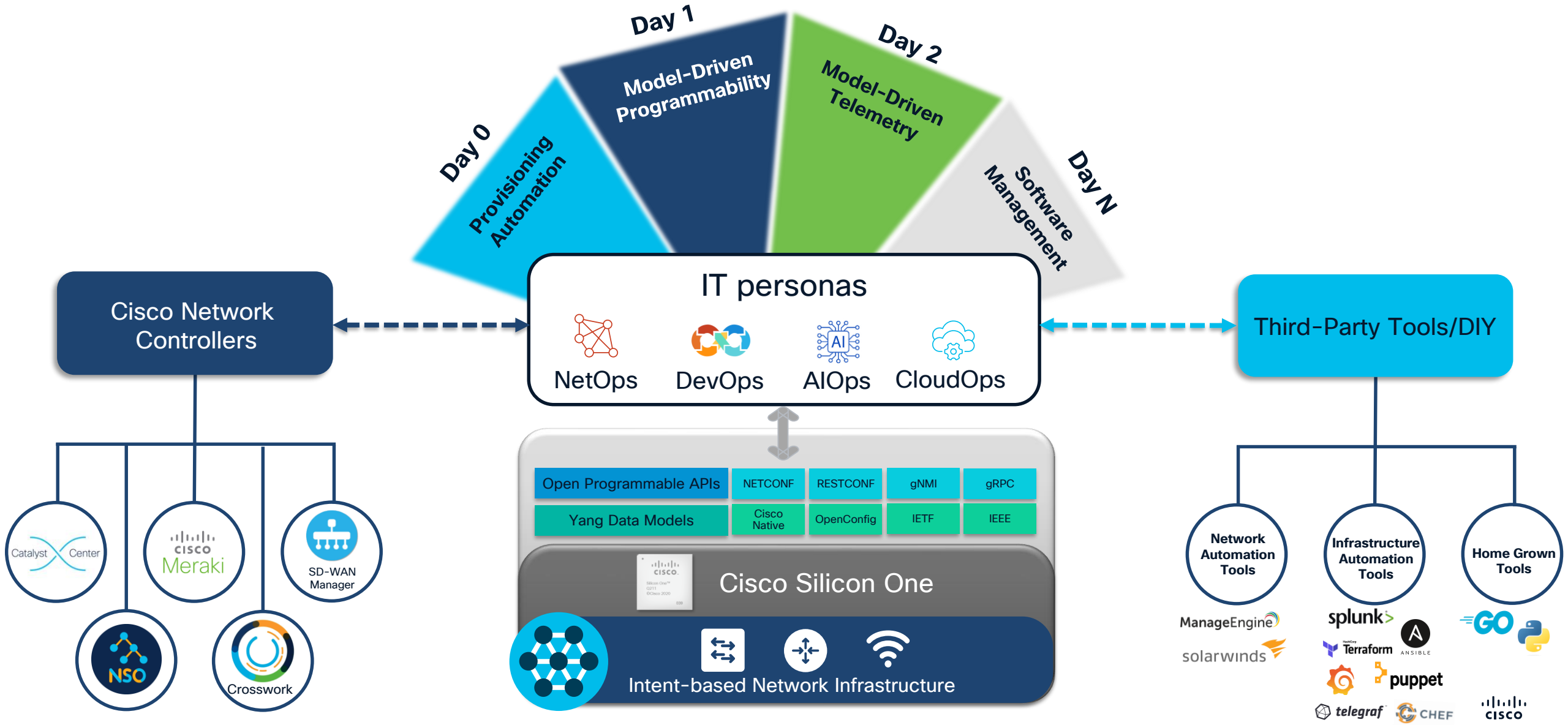
2. Infrastructure as Code

Structured Data: The Fuel for Reliable Automation

Unstructured Data (CLI)		Structured Data (YANG, XML/JSON)
Human-Oriented, Unstructured Text	Data Format & Parsing	Machine-Oriented, Structured Data
Low-Level, Procedural Logic, No intent abstraction, just commands	Abstraction & Intent	Intent-Based and Declarative, Describe what you want (intent)
Siloed and Manual, Poor integration with ITSM, CI/CD, observability tools	IT Ecosystems Integration	API-Driven and Event-Aware, Easily integrates with ITSM, GitOps, telemetry pipelines
Syntax differs by vendor, OS, and version, Scripts are bloated with conditionals	Multi-Vendor Complexity	Standard YANG models across vendors, Unified logic for multi-vendor networks

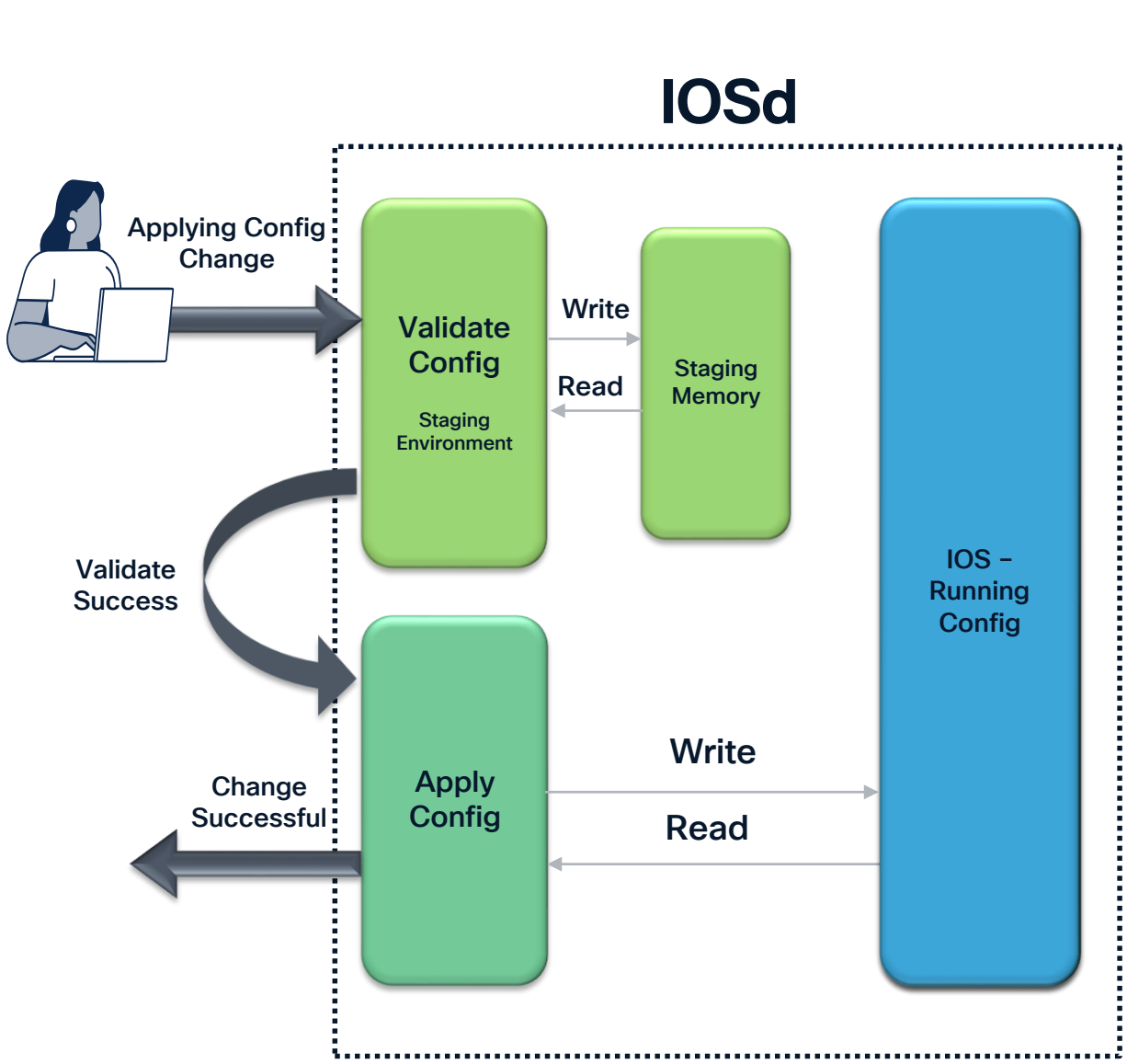
Industry's most open network OS

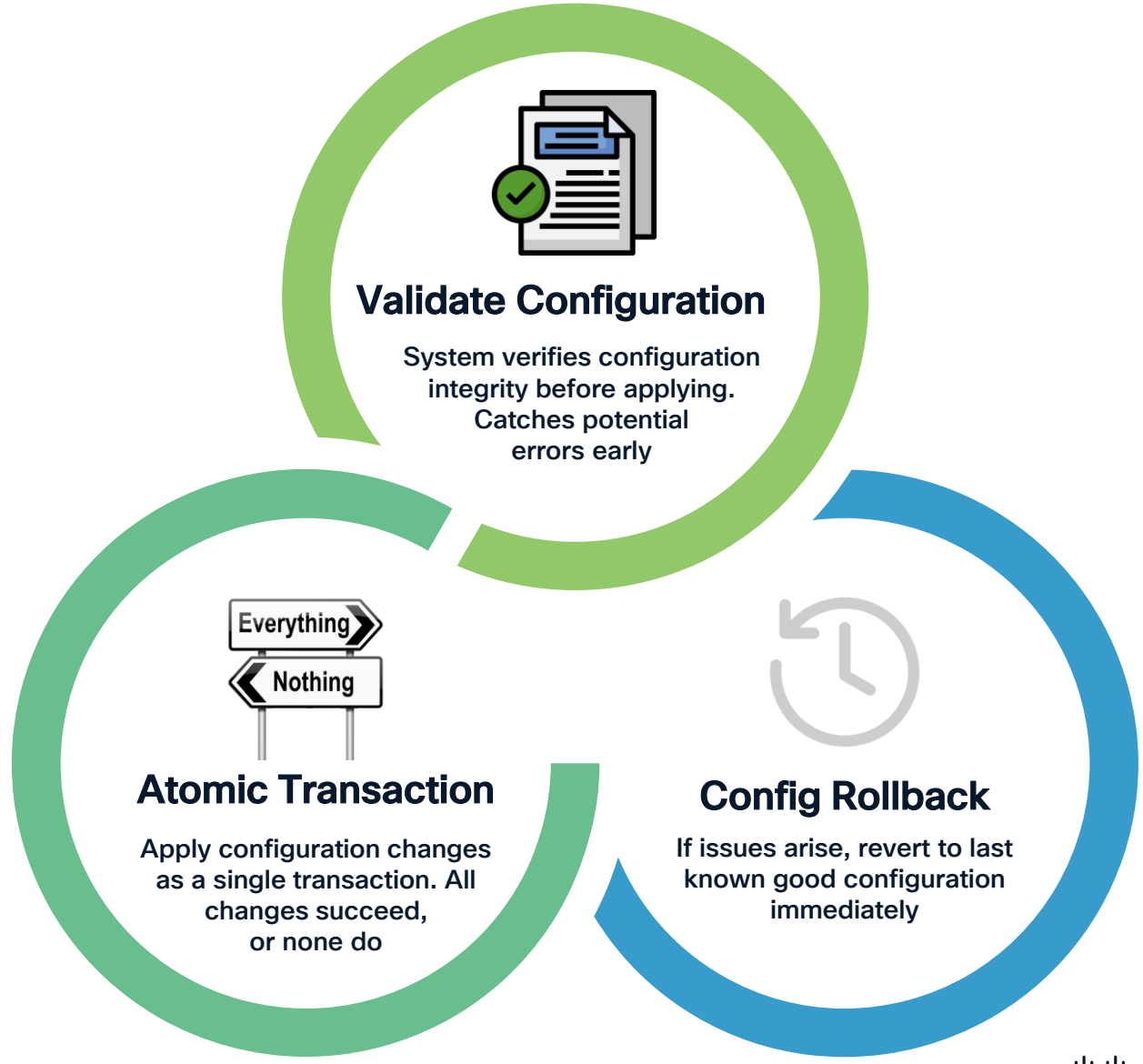
Receptive to a wide range of management and orchestration strategies



Intro to Atomic Config Replace (ACR)

Introducing Atomic Configuration Replace





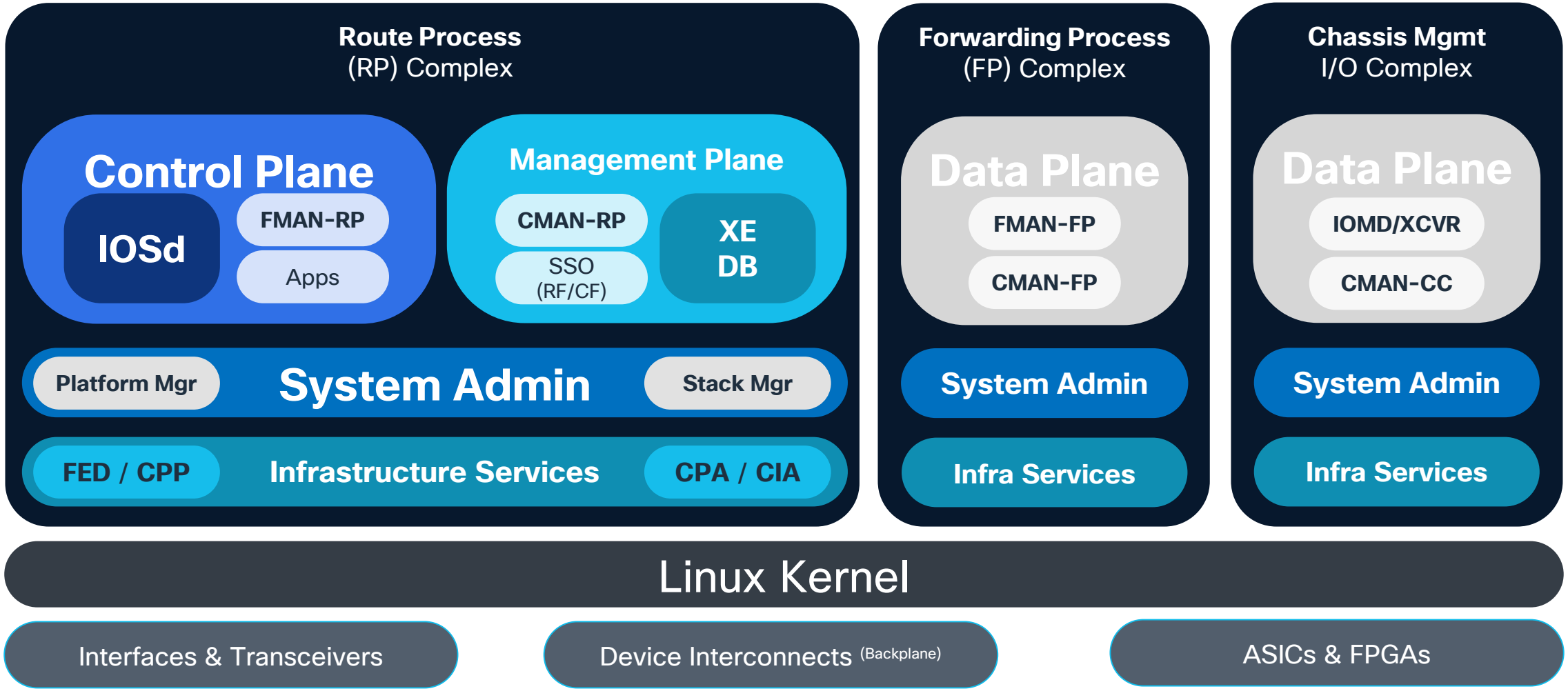
Evaluating ACR Benefits

Without Atomic Replace		With Atomic Replace
Manual Verification	Config Verification	Syntax, Semantic and Dependency Verification
Immediate Command Execution - Incremental Changes	Config Changes	Pre-Validated Configuration Deployment - Transactional Integrity
Costly configuration errors, Reactive troubleshooting with high risk of outage	Troubleshooting & Outage	Proactive error prevention and reduced risk of outage to enable seamless network management
Errors can leave the network exposed with higher risk of non-compliant configurations	Security & Compliance	Ensures adherence to security policies and compliance requirements

VS

Cisco IOS XE Architecture

Modularized Components for Software Abstraction



ACR demos

Python Script Using ACR

```
#! /usr/bin/env python
from ncclient import manager
from xml.dom import minidom
from netmiko import ConnectHandler
import difflib
import lxml.etree as et
from xml.etree import ElementTree as ET
import xmltodict
from lxml import etree
from ncclient.operations import RPCError
from datetime import datetime
import time,os

get_modelled_config_running = '''<get-modelled-config-clis xmlns="http://cisco.com/n
                          <datastore>running</datastore>
                        </get-modelled-config-clis> '''
get_modelled_config_candidate = '''<get-modelled-config-clis xmlns="http://cisco.com
                          <datastore>candidate</datastore>
                        </get-modelled-config-clis> '''

global candidate_flag,pre_filename_shrun,file1_lines,post_filename_shrun,file2_lines

class Device:
    def __init__(self, host, username, password):
        self.host = host
        self.username = username
        self.password = password
        self.__ssh_session = None
        self.__netconf_session = None
```

Detailed Python Script workflow:

1. Start
2. Initialize Device
3. Netconf Connect
4. Discard Changes
5. Get Pre-check Config
6. Apply Config (edit_config)
7. Get Post-check Config
8. Compare Pre & Post Configs
9. Confirmed Commit
10. Get Post-confirmed Commit Config
11. Commit Changes
12. Compare Pre & Final Configs
13. End

ACR Demo 1: Syntax & Dependency Error Isolation

See demo of these steps in the next slide!

1. Send a “full-replace” operation to a C9300 switch to fully replace all the current config on the device with the new config provided in the “target_C9K_config.xml” file
2. Notice syntax error is found in the “target_C9K_config.xml” file by ACR and the exact line of the error is provided
3. Fix the syntax error in the “target_C9K_config.xml” file
4. Send the “full-replace” operation once again to the C9300 switch
5. Verify that the ACR result is “Configuration applied successfully”

```
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
```

ACR Demo 2: rollback config because no confirm commit

See demo of these steps in the next slide!

1. Send a “full-replace” operation to a C9300 switch to fully replace all the current config on the device with the new config provided in the “target_C9K_config.xml” file
2. Notice that although the configuration is valid and applied to the device, the device returns to its previous known state because no “confirm commit” was issued

auto@pod22-xelab: ~/acr/test1 (ssh)

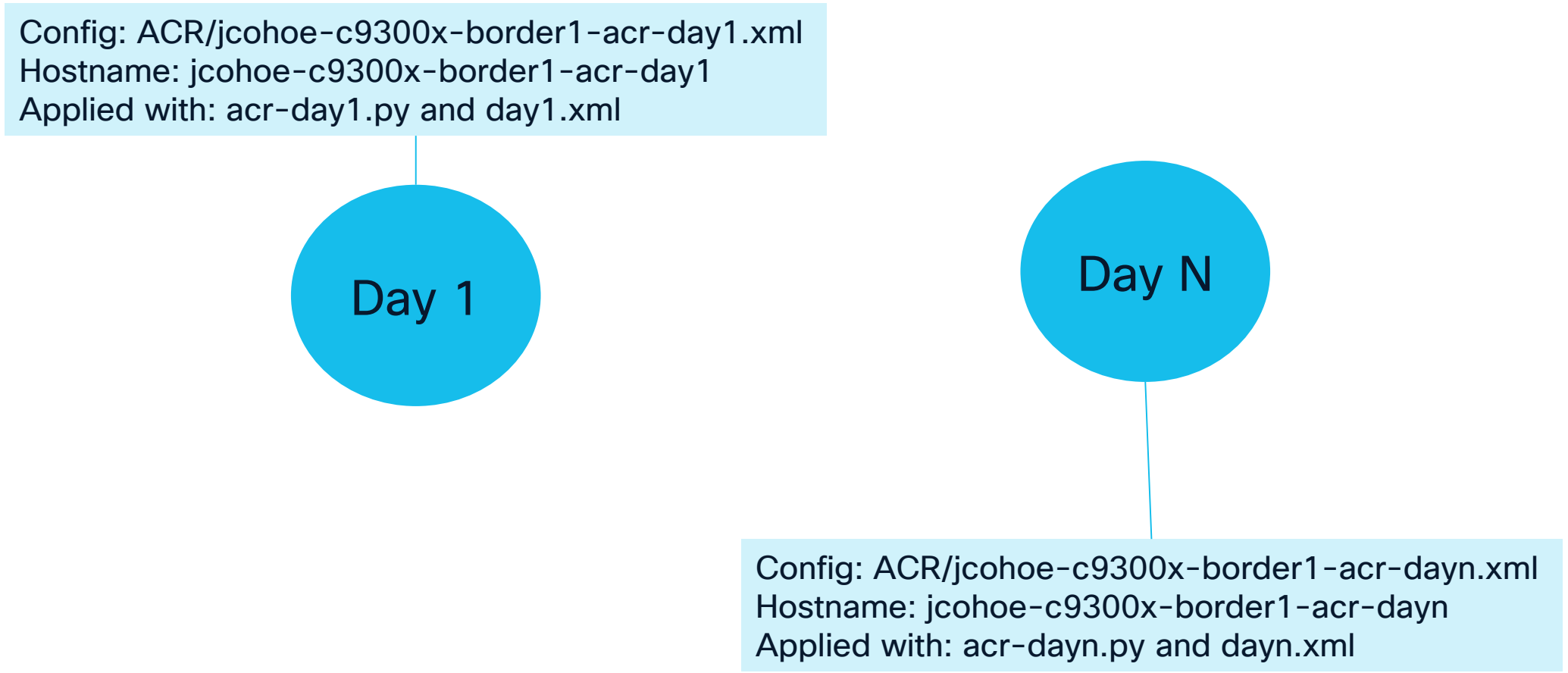
```
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
auto@pod22-xelab:~/acr/test1$
```

auto@pod22-xelab: ~ (ssh)

```
cat9300-pod22b#
cat9300-pod22b#
cat9300-pod22b#
cat9300-pod22b#
cat9300-pod22b#
cat9300-pod22b#
cat9300-pod22b#
cat9300-pod22b#
cat9300-pod22b#
cat9300-pod22b#
cat9300-pod22b#
```

ACR Demo Scenario: Day1 to Day N

Atomic Config Replace with NETCONF/YANG



Jcohoe-c9300x-border1.config

tme@tme-yangsuite:~/acr/jcohoe-c9300x-border1-acr$ cp jcohoe-c9300x-border1-acr.xml jcohoe-c9300x-border1-acr.xml.good1

cat ~/acr/jcohoe-c9300x-border1-acr/jcohoe-c9300x-border1-acr.xml

```
tme@tme-yangsuite:~/acr/jcohoe-c9300x-border1-acr$ wc -l jcohoe-c9300x-border1-acr.
481 jcohoe-c9300x-border1-acr.xml
tme@tme-yangsuite:~/acr/jcohoe-c9300x-border1-acr$ cat jcohoe-c9300x-border1-acr.xm
<config-ios-cli-trans xmlns ="http://cisco.com/ns/yang/Cisco-IOS-XE-cli-rpc">
 <clis>
  version 17.15
memory free low-watermark processor 130582
service internal
service timestamps debug datetime msec
service timestamps log datetime msec
service call-home
no service dhcp
no service tcp-small-servers
no service udp-small-servers
hostname jcohoe-c9300x-border1
control-plane
 service-policy input system-cpp-policy
```

```
line vty 0 4
 exec-timeout 0
 length 0
 transport input all
!
line vty 5 15
 exec-timeout 0
 transport input all
!
line vty 16 31
 transport input ssh
!
ntp server 10.81.254.202
event manager applet catchall
 event cli pattern .* sync no skip no
 action 1 syslog msg $_cli_msg
 exit
!
no process cpu extended history
diagnostic bootup level minimal
!switch 1 provision c9300x-48hx
 </clis>
 <operation>full-replace</operation>
  <do-commit>false</do-commit>
</config-ios-cli-trans>tme@tme-yangsuite:~/acr/jcohoe-c9300x-border1-acr$
tme@tme-yangsuite:~/acr/jcohoe-c9300x-border1-acr$
```

ACR.py: device configuration & config file

Specify Device (line 244)

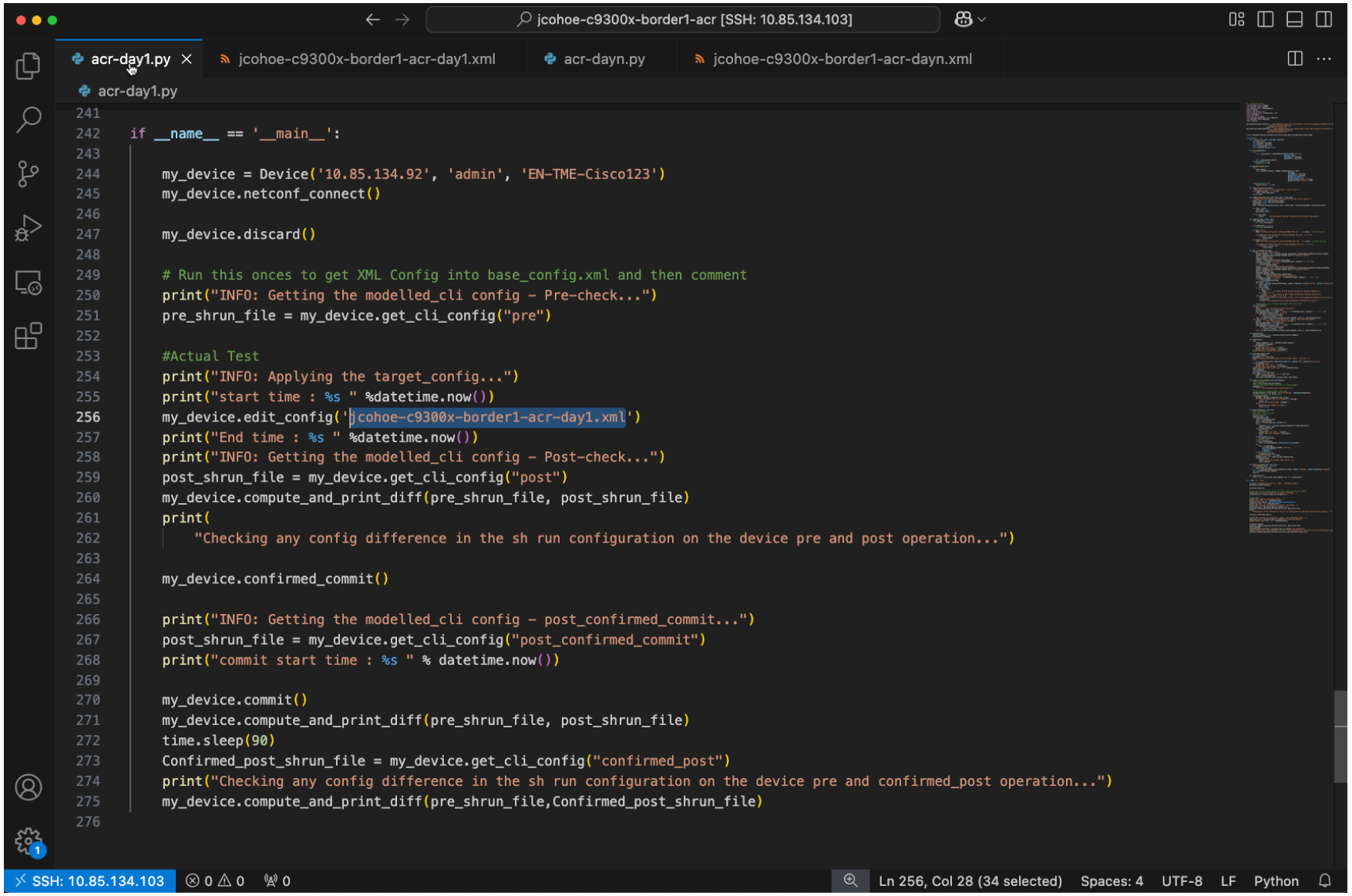
Authentication User/Pass

XML configuration file (line 256)

```
242  if __name__ == '__main__':
243
244      my_device = Device('10.85.134.92', 'admin', 'EN-TME-Cisco123')
245      my_device.netconf_connect()
246
247      my_device.discard()
248
249      # Run this onces to get XML Config into base_config.xml and then comment
250      print("INFO: Getting the modelled_cli config - Pre-check...")
251      pre_shrun_file = my_device.get_cli_config("pre")
252
253      #Actual Test
254      print("INFO: Applying the target_config...")
255      print("start time : %s " %datetime.now())
256      my_device.edit_config('jcohoe-c9300x-border1-acr-dayn.xml')
257      print("End time : %s " %datetime.now())
258      print("INFO: Getting the modelled_cli config - Post-check...")
259      post_shrun_file = my_device.get_cli_config("post")
260      my_device.compute_and_print_diff(pre_shrun_file, post_shrun_file)
```

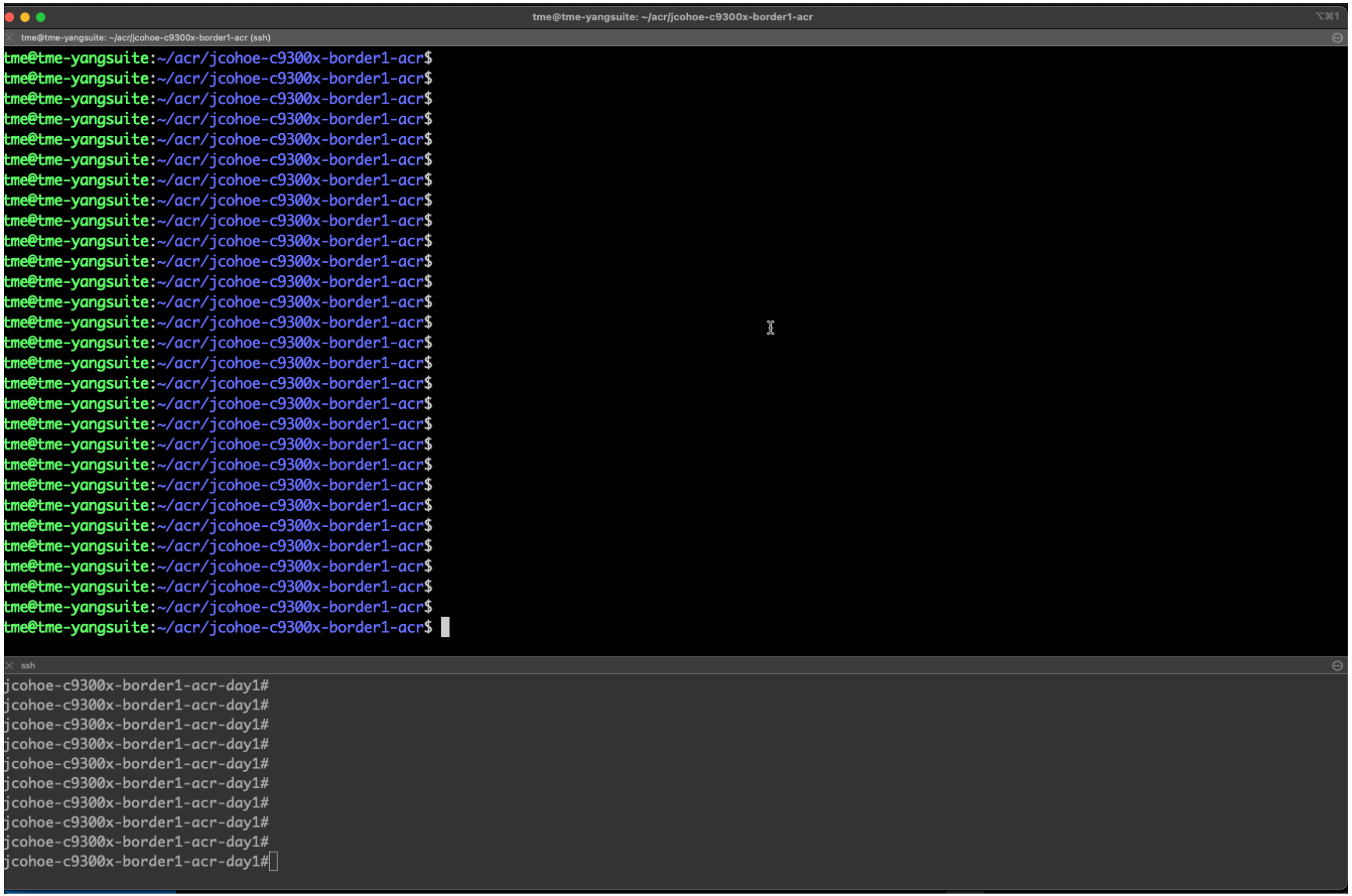
Any current configuration on box to Day - 1

Hostname before:
VNC2-BORDER1-X
Hostname after:
jcohoe-c9300x-border1-acr-day1



Day-1 to Day-N

Hostname before:
jcohoe-c9300x-border1-acr-day1
Hostname after:
jcohoe-c9300x-border1-acr-dayn



NETCONF CLI RPC

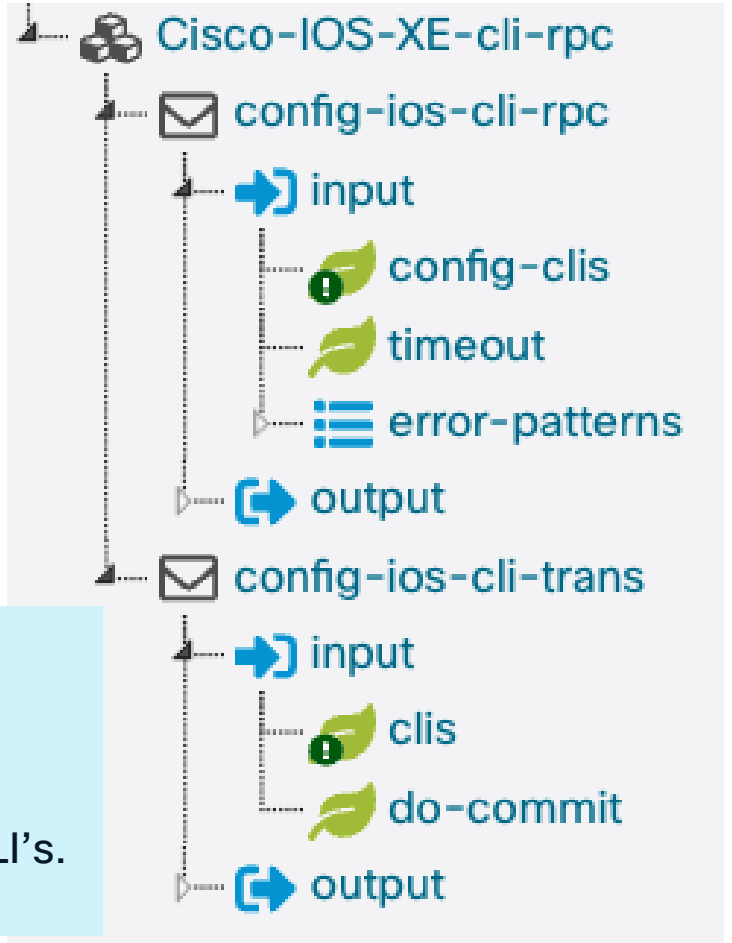
YANG model for CLI execution

Original 17.9

Any configure CLI can now be sent within the YANG payload

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
message-id="101">
<config-ios-cli-rpc
xmlns=http://cisco.com/ns/yang/Cisco-IOS-XE-cli-rpc>
<config-clis>
interface Loopback111
description configured-via-CLI-YANG
no shutdown
</config-clis>
</config-ios-cli-rpc>
</rpc>]]>]]>
```

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
message-id="101">
<config-ios-cli-trans
xmlns=http://cisco.com/ns/yang/Cisco-IOS-XE-cli-rpc>
<clis>
interface Loopback111
description configured-via-CONFD-YANG
no shutdown
</clis>
</config-ios-cli-trans>
</rpc>]]>]]>
```



clis

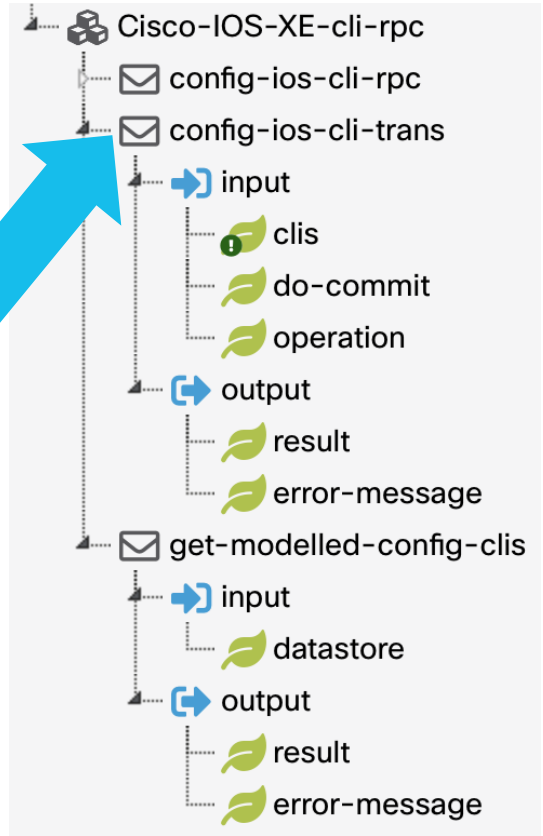
config-clis

"cli rpc" sends CLI to the IOS parser
This is similar to configuring CLI on the VTY
Directly into running-config, then **synchronized** to ConfD
"transactional cli rpc" sends a list of CLI to ConfD
This is similar to sending edit-config RPCs corresponding to the CLI's.
Synchronized from ConfD into the CLI running-config

https://github.com/YangModels/yang/blob/main/vendor/cisco/xe/1791/Cisco-IOS-XE-cli-rpc.yang

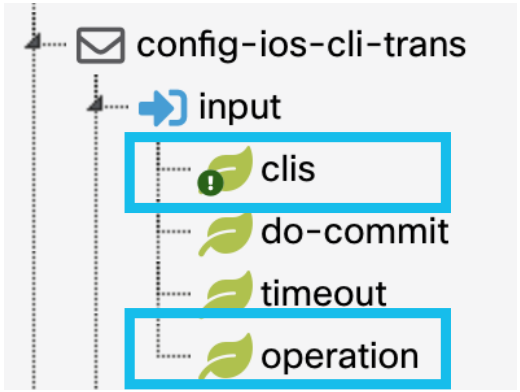
Cisco-IOS-XE-CLI-RPC.YANG

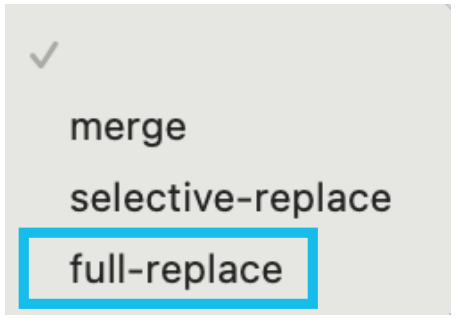
This YANG data model allows sending CLI through the YANG API interfaces
Previously only YANG modelled data was supported



Name	config-ios-cli-trans
Nodetype	rpc
Description	Configure using modelled config CLI. Wireless, app-hosting, telemetry features are not supported through this RPC.
Module	Cisco-IOS-XE-cli-rpc
Revision	2023-11-01
Xpath	/config-ios-cli-trans
Prefix	cli-ios-xe-rpc
Namespace	http://cisco.com/ns/yang/Cisco-IOS-XE-cli-rpc
Schema Node Id	/config-ios-cli-trans
Access	write
Operations	• "rpc"

Name	get-modelled-config-clis
Nodetype	rpc
Description	Retrieve configuration in CLI format from Running or Candidate datastore. Wireless, app-hosting, telemetry features are not supported through this RPC.
Module	Cisco-IOS-XE-cli-rpc
Revision	2023-11-01
Xpath	/get-modelled-config-clis
Prefix	cli-ios-xe-rpc
Namespace	http://cisco.com/ns/yang/Cisco-IOS-XE-cli-rpc
Schema Node Id	/get-modelled-config-clis
Access	write
Operations	• "rpc"

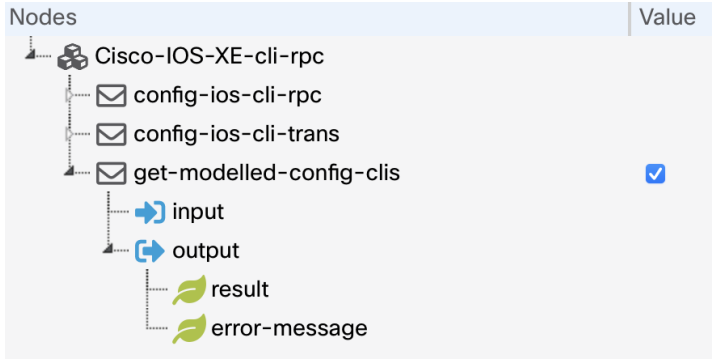




https://github.com/YangModels/yang/blob/main/vendor/cisco/xe/1791/Cisco-IOS-XE-cli-rpc.yang

Get Modelled Config CLI RPC

- Sending the “get-modelled-config-clis” RPC returns the modelled running-config in CLI format
- Anything not modelled will not be returned (AppH)
- Unsupported model config will be ignored (AppH)
- This is used as the template to update the device with after being modified as needed




```
Sending:
#246
<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:afff
  <get-modelled-config-clis xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-cli-rpc"/>
</nc:rpc>


##
```

```
Received message from host
```

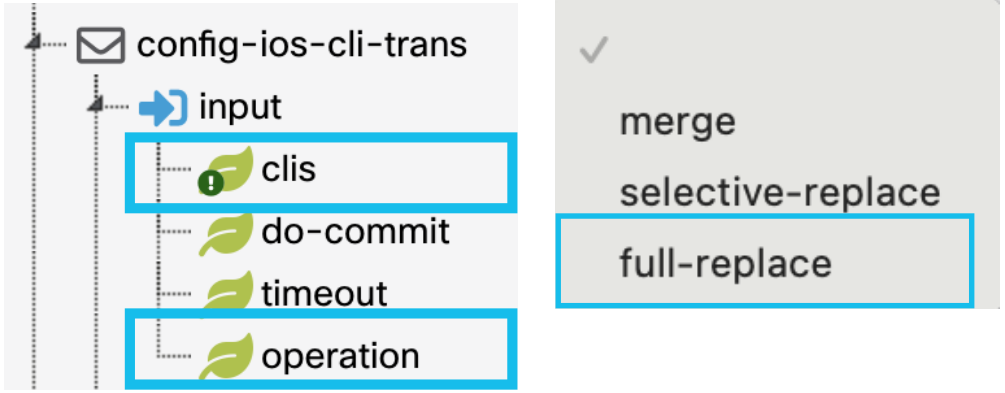
```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params
<result xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-cli-rpc">version 17.14
memory free low-watermark processor 130582
service timestamps debug datetime msec
service timestamps log datetime msec
service call-home
no service tcp-small-servers
no service udp-small-servers
hostname JCOHOE-C9300-2
control-plane
 service-policy input system-cpp-policy
!
clock summer-time PDST recurring
clock timezone pacific -8 0
login on-success log
license boot level network-advantage addon dna-advantage
transceiver type all
 monitoring
!
iox
call-home
 contact-email-addr sch-smart-licensing@cisco.com
 profile CiscoTAC-1
  active
  destination transport-method http
```

```
RPC:
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <get-modelled-config-clis xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-cli-rpc"/>
</rpc>
```

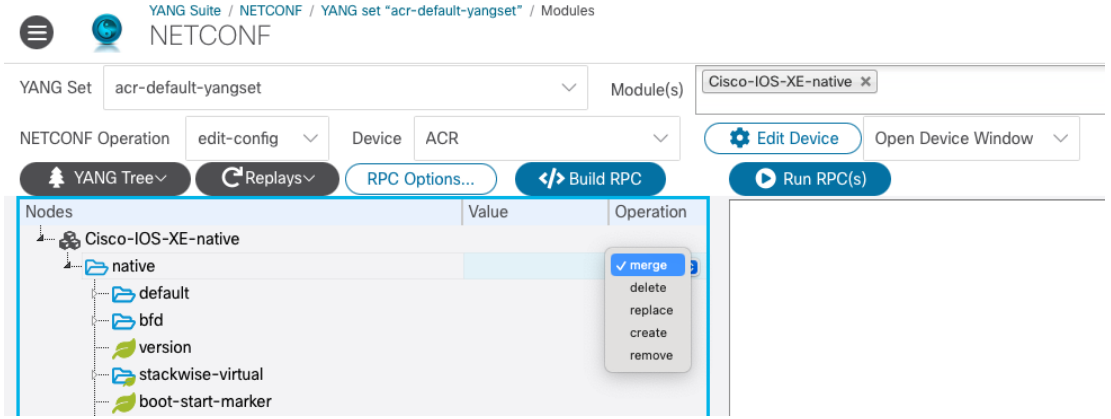
Atomic Config Replace – ACR

Atomic Config Replace enables full or partial config replace
Ability to send an entire configuration to the device in an XML/JSON payload
Support for traditionally documented CLI's over the CLI-RPC.YANG

Full and selective replace supported as part of CLI RPC over NETCONF/YANG



Merge, Replace operations supported as part of NETCONF/YANG

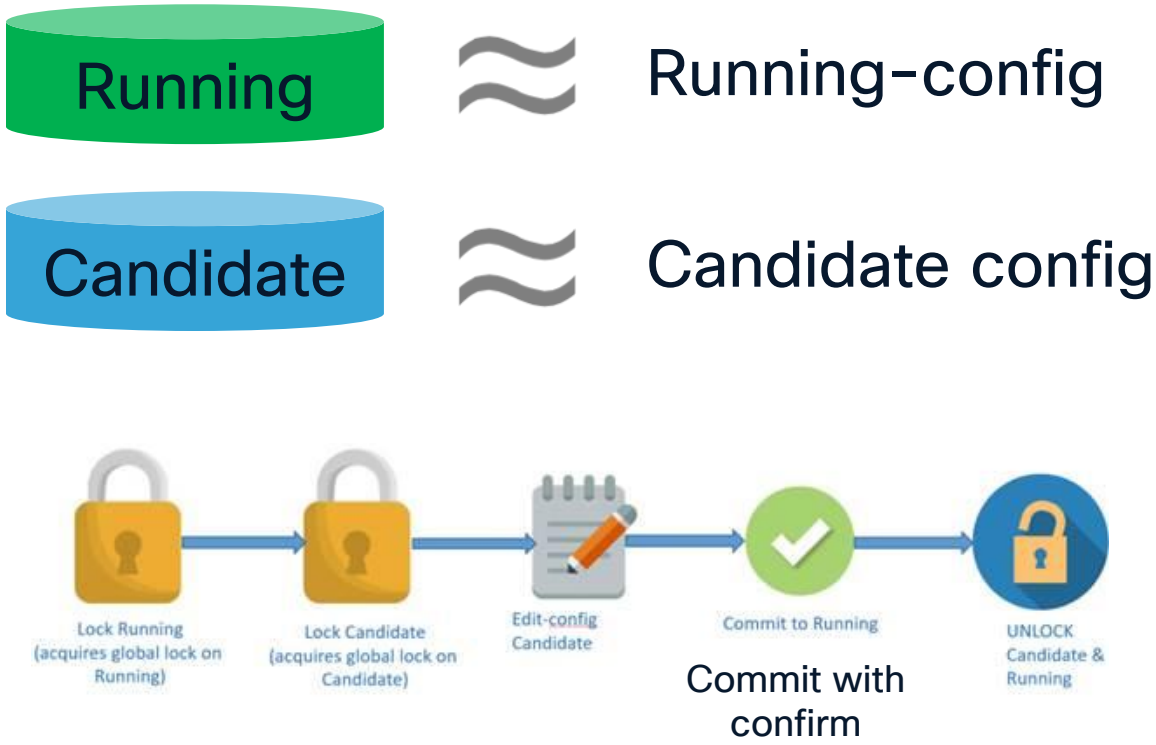


IOS XE NETCONF Datastores

"A Datastore holds a copy of the configuration data that is required to get a device from its initial default state into a desired operational state"

Running is the default and only mandatory Datastore

The Candidate Configuration feature enables support for candidate capability by implementing RFC 6241 with a simple commit option.
The candidate datastore provides a temporary workspace in which a copy of the device's running configuration is stored.
The candidate configuration supports the confirmed commit capability



https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/1717/b_1717_programmability_cg/m_1717_prog_yang_netconf.html#id_78218

2-Stage Commit

- The 2-Stage commit process includes error and syntax checking
- It enabled a multi-stage commit process with verify before apply
- It is a non-disruptive application processes for the changes – no impact to packet processing
- 2-Stage Commit is only seen when config is rejected as there is no disruption to service

```
VNC2-BORDER1-X#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
VNC2-BORDER1-X(config)#yang-interfaces feature ios-two-stage
VNC2-BORDER1-X(config)#end
VNC2-BORDER1-X#sh run | i two-stage
yang-interfaces feature ios-two-stage
VNC2-BORDER1-X#
```

Enable 2-stage commit with	# yang-interfaces feature ios-two-stage
CLI will be simplified to	# yang-interfaces features atomic-config

Pre-requisites for ACR

On the Catalyst device, the following CLI's must be configured for ACR:

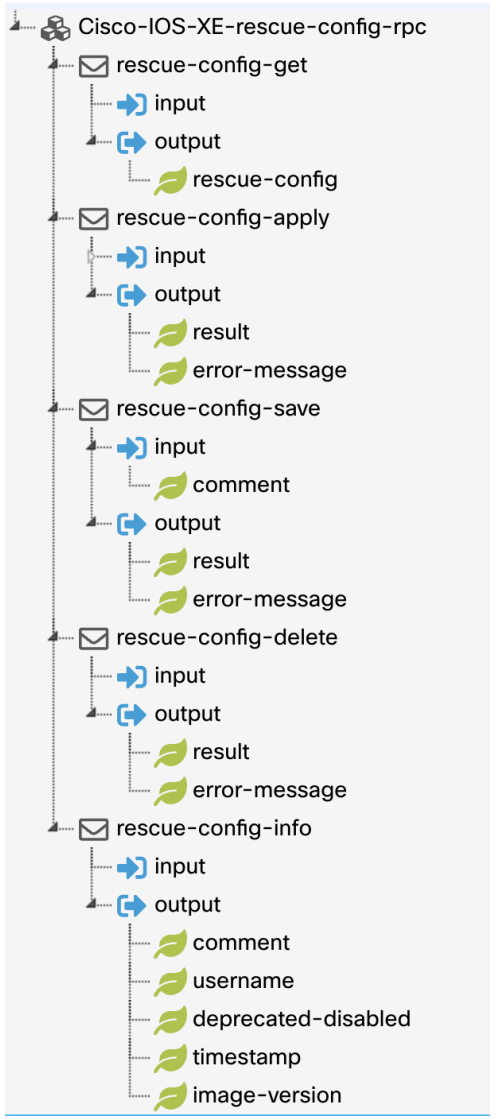
1. netconf-yang
2. netconf-yang feature candidate-datastore
3. yang-interfaces feature atomic-config
4. yang-interfaces feature deprecated disable

```
VNC2-BORDER1-X(config)#netconf-yang fe
VNC2-BORDER1-X(config)#netconf-yang feature ?
  candidate-datastore  Enable the Candidate Datastore
  side-effect-sync     Use the side-effect sync instead of full sync, whenever possible

VNC2-BORDER1-X(config)#netconf-yang feature ca
VNC2-BORDER1-X(config)#netconf-yang feature candidate-datastore ?
  <cr>  <cr>

VNC2-BORDER1-X(config)#netconf-yang feature candidate-datastore
```

```
VNC2-BORDER1-X#sh run | i yang
netconf-yang
netconf-yang feature candidate-datastore
yang-interfaces feature atomic-config
yang-interfaces feature deprecated disable
VNC2-BORDER1-X#
```

```
VNC2-BORDER1-X(config)#yang-interfaces feature de
VNC2-BORDER1-X(config)#yang-interfaces feature deprecated ?
  disable  Disable deprecated Yang model elements
  enable   Enable deprecated Yang model elements

VNC2-BORDER1-X(config)#yang-interfaces feature deprecated di
VNC2-BORDER1-X(config)#yang-interfaces feature deprecated disable
VNC2-BORDER1-X(config)#
```

ACR Rescue Config



The rescue config is a golden configuration that can be used to roll back to if an undesired state is reached.

The rescue config RPC YANG model provides options to get, apply, save, delete or access rescue config info.

Rescue Config Get

Request

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">

  <rescue-config-get xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-rescue-config-rpc"/>

</rpc>
```

CLI

```
jcohoe-c9300-2-acm#sh run | format netconf-xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <cloud-services-cfg-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-cloud-services-cfg">
    <cloudm-config>
    </cloudm-config>
  </cloud-services-cfg-data>
  <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
    <version>17.19</version>
    <memory>
      <free>
        <low-watermark>
          <processor>105089</processor>
        </low-watermark>
      </free>
    </memory>
    <call-home>
      <alert-group xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-call-home">
```

Reply

```
<?xml version="1.0" ?>

<rpc-reply message-id="urn:uuid:58dc2e7f-1393-4628-b87b-87a063de683c" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc

  <rescue-config xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-rescue-config-rpc"><config xmlns="http://tail-f.com/ns/config/1.

  <cloud-services-cfg-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-cloud-services-cfg">

    <cloudm-config>

    </cloudm-config>

  </cloud-services-cfg-data>

  <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">

    <version>17.19</version>

    <memory>

      <free>

        <low-watermark>

          <processor>105089</processor>

        </low-watermark>

      </free>

    </memory>

…
```

Rescue Config Apply

Request

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">

  <rescue-config-save xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-rescue-config-rpc"/>

</rpc>
```

Reply

```
<?xml version="1.0" ?>

<rpc-reply message-id="urn:uuid:f0df1990-88bd-4e16-9b80-1d48a35a38e9" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">

  <result xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-rescue-config-rpc">Successfully applied rescue config</result>

</rpc-reply>
```

Rescue Config Save (with comment)

Request

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">

  <rescue-config-save xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-rescue-config-rpc">

    <comment>CL-rescue</comment>

  </rescue-config-save>

</rpc>
```

Reply

```
<?xml version="1.0" ?>

<rpc-reply message-id="urn:uuid:d270af87-3868-4bea-a8e5-517afc8acfa4" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">

  <result xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-rescue-config-rpc">Successfully saved rescue config</result>

</rpc-reply>
```

Rescue Config Info

Request

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">

  <rescue-config-info xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-rescue-config-rpc"/>

</rpc>
```

Reply

```
<?xml version="1.0" ?>

<rpc-reply message-id="urn:uuid:d270af87-3868-4bea-a8e5-517afc8acfa4" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">

  <result xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-rescue-config-rpc">Successfully saved rescue config</result>

</rpc-reply>

<rpc-reply message-id="urn:uuid:e900c0e9-8673-473d-8a96-621f99992b00" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">

  <comment xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-rescue-config-rpc">CL-rescue</comment>

  <username xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-rescue-config-rpc">admin</username>

  <deprecated-disabled xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-rescue-config-rpc">false</deprecated-disabled>

  <image-version xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-rescue-config-rpc">17.19.01</image-version>

  <timestamp xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-rescue-config-rpc">2025-06-08 17:27:25</timestamp>

</rpc-reply>
```

Rescue Config Delete

Request

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">

  <rescue-config-delete xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-rescue-config-rpc"/>

</rpc>
```

Reply

```
<?xml version="1.0" ?>

<rpc-reply message-id="urn:uuid:5e581ba9-ed6d-49a5-bce3-fd4440c75cc6" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">

  <result xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-rescue-config-rpc">Successfully deleted rescue config</result>

</rpc-reply>
```

Ready to get hands-on?

ACR: Scope

- Validates only device-level configurations
  - Network-level configurations are not validated
- Pre-Release Feature: Available for Early Field Trials
- Supported on Catalyst 9300, 9500 & 9800 (WLC)
  - 9200, 9400 & 9600 support coming soon!
- Supported on programmable interfaces
  - Exclusively with NETCONF & NETCONF CLI RPC
- Works with specific set of features

Category	Features
Basic L2	Ethernet interfaces, Port channel interfaces, Port channel, Spanning tree, LACP, Logging, Err-disable
L3 and SVL	VRF, VLAN interfaces, Loopback interfaces, IP, IP DHCP, IP Route, MPLS, ARP, Track
Policy (Security and Others)	Class-map, Policy-map, Route-map, AAA, Ssh, IP ACL, TACACS, Crypto, certs etc, Username
Management, Device configuration and access etc.	HTTP, SNMP, Banner, Line, NTP, Monitor, Call home, Hostname, Service, Archive, PnP, Event Manager
Advanced features	Post-deployment rollback, Config Diff, Commit Comment, ISSU, Manage static imperative device objects

Resources

Cisco Live US Programmability Learning Map

Sunday–8th

TECOPS-2314 9:00A

Automating All Things YANG, All the time - Programmability and Automation 101 with Cisco IOS XE

Monday–9th

BRKOPS-2256 10-11:30

Exploring Practical AIOps Use Cases for Enterprise Networks with Splunk 1-2PM

BRKENS-2604

Atomic Config Replace with Cisco Catalyst 9000 3:30-5PM

BRKOPS-1401

Cisco IOS XE: Telemetry, Automation, and YANG–Oh My!

Tuesday–10th

DEVNET-1110 11:30-1:15

Modern approaches for IOS XE network device management on Cat9k

BRKDEV-2017 1:30-2:30

gRPC, gNMI, gNOI... Oh My! An Enterprise Network Automation Journey 2-3:30

BRKOPS-2223

Network of the Future is Here - Let's Automate your IPv6 deployment with Python!

Wednesday–11th

CISCOU-1059 4-4:30PM

Observability at TorIX: custom telemetry solutions on next-gen campus switching

Thursday–12th

https://blogs.cisco.com/developer/cisco-ios-xe-automation-clus25

Walk-in labs (open Monday-Thursday)

- ✓ Efficiently monitoring device statistics in real-time using gRPC Dial-out with IOS XE [LABPRG-2004]
- ✓ Explore and test YANG models for model driven telemetry on IOS XE with Cisco's YANG Suite [LABOPS-2000]
- ✓ Hands-On Lab: Monitoring Cisco IOS-XE Devices with RESTCONF [LABDEV-2001]

Make sure to attend below for detailed information.

EXPLORER FULL CONFERENCE IT LEADERSHIP

Opening Keynote: Vision for the Future - KEYGEN-1001

Chuck Robbins, Chair and Chief Executive Officer, Cisco
Jeetu Patel, President and Chief Product Officer, Cisco
Carrie Palin, SVP and Chief Marketing Officer, Cisco
DJ Sampath, SVP, AI Software and Platform, Cisco
Kevin Weil, Chief Product Officer, OpenAI
Alan Rosa, Chief Information Security Officer and SVP of Infrastructure and Operations, CVS Health

Schedule **Tuesday, Jun 10 | 8:30 AM - 10:30 AM PDT | SDCC - Hall G/H, Keynote Theater**

FULL CONFERENCE IT LEADERSHIP

Campus Switching Innovations for Future Proofed Workspaces - BRKENS-2609

Minhaj Uddin, Leader, Technical Marketing, Cisco - **Distinguished Speaker**

Schedule **Tuesday, Jun 10 | 11:00 AM - 12:30 PM PDT | SDCC - Upper Level, Room 33A**

FULL CONFERENCE IT LEADERSHIP

Cisco Catalyst 9000 Series, Cisco Silicon One, and IOS XE Architecture and Innovation - BRKARC-2092

Shawn Wargo, Principal Technical Marketing Engineer, Cisco - **Distinguished Speaker, Hall of Fame Speaker**

Schedule **Tuesday, Jun 10 | 11:00 AM - 12:30 PM PDT | SDCC - Upper Level, Room 28CD**

FULL CONFERENCE IT LEADERSHIP

Campus Switching Architecture for Future Proofed Workspaces - BRKARC-2668

Kenny Lei, TME, Cisco - **Distinguished Speaker**

Schedule **Wednesday, Jun 11 | 3:30 PM - 5:00 PM PDT | SDCC - Upper Level, Room 28AB**

Cisco Live US Catalyst 9000 Learning Map

Sunday–8th

TECENS-2620 9:00AM
Catalyst 9000 Switching Architecture and Software Innovations

TECENS-2680 2:00PM
Cisco Catalyst 9000 Switching Family Architecture

TECARC-2446 2:00PM
BGP EPVN in Enterprise Campus with Catalyst 9000 Switching Platforms

Monday–9th

BRKENS-1500 8:00AM
Introduction to Campus Network Design and Multilayer Architectures

BRKENS-2092 8:00AM
BGP EVPN in Enterprise Campuses with Catalyst 9000 Series Switches

BRKENS-2095 9:30AM
Designing Highly Available Networks using Catalyst 9000 Series Switches

BRKENS-2652 11:00AM
Connecting Beyond Fabric: Catalyst 9000 BGP EVPN Handoff Scenarios

BRKENS-2604 1:00PM
Atomic Config Replace with Cisco Catalyst 9000

BRKENS-2608 2:30PM
Future-proofing Campus Switching for WiFi7

BRKARC-1012 2:30PM
Investment Protection with Catalyst 9000 Series Switching & Wireless: A Competitive Edge

BRKENS-2099 4:00PM
Innovations on Cisco Campus Switching for Sustainability and Energy Management

Tuesday–10th

LTRARC-3001 8:00AM
Mastering Catalyst 9000 Switches: Architectural Insights and Troubleshooting Strategies

LTRENS-2429 8:00AM
AI/ML in Cisco Catalyst Center: Transforming Network Operations!

BRKARC-2092 11:00AM
Unlocking the Automation Power in Catalyst Center for Wired and Wireless Networks

BRKENS-2609 11:00AM
Deploy Cisco Catalyst Center with Rest-API's

BRKENS-2655 1:30PM
Catalyst Center Network Operations Essentials using UI and APIs

BRKENS-1402 3:00PM
Deploying Cisco Catalyst Center with CICD

BRKARC-2039 4:00PM
Cisco Catalyst 9000 Switching QoS with Silicon One ASICs Deep Dive

BRKENS-2603 4:00PM
Catalyst Switching enabled Smart Buildings : Beyond PoE Connectivity

Wednesday–11th

BRKENS-2610 10:30AM
Catalyst Center Network Operations Essentials using UI and APIs

CIUG-1109 10:30AM
Catalyst 9000 Switching Innovations & Roadmap

LTRENS-2256 1:00PM
Cisco Catalyst Switching Innovations Lab

BRKENS-2500 1:30PM
Advanced Campus Network Design: Multilayer Architectures and Next-Gen Protocols

BRKARC-2668 3:30PM
Campus Switching Architecture for Future Proofed Workspaces

Thursday–12th

BRKENS-2094 9:30AM
Media & Time Sensitive Networking with C9K Switches: Converging Time Sensitive Applications & Devices onto Ethernet

BRKARC-2099 10:30AM
Catalyst 9000 Series Switching Family: Core and Distribution

BRKARC-2098 10:30AM
Open-Source GenAI Bot for Catalyst Center

BU-led sessions

Cisco Live US Catalyst Center Learning Map

Sunday–8th

TECOPS-2001 9:00AM
The Ultimate Guide to Install, Onboard, and Operate Your Campus Network with Cisco Catalyst Center

LTRSEC-2005 9:00AM
Building Cisco SD-Access with Cisco Catalyst Center & ISE

TECENS-2680 2:00PM
BGP EPVN in Enterprise Campus with Catalyst 9000 Switching Platforms

Monday–9th

BRKOPS-2698 8:00AM
Choosing the Right Cisco Catalyst Center Deployment Model for Your Network

CIUG-1100 10:00AM
Cisco Catalyst Center: AI-Driven Switching: Revolutionizing Automation and Assurance

LTRXAR-3783 1:00PM
Cross-Architecture Integration Experience Lab

BRKENS-1601 1:30PM
Catalyst Center and Meraki Cloud: The Right Choice for your Catalyst 9000 Switch Management!

BRKOPS-2609 1:30PM
Cisco Catalyst Center: Built-In Integrations for Streamlined Network Operations

LTROPS-2341 2:00PM
Build a Flexible Network Automation Workflow with GitLab CI/CD, Catalyst Center, NetBox, and Ansible

IBOENS-1100 2:30PM
Cisco Catalyst Center and SD-Access Design Fundamentals

BRKEWN-2029 4:00PM
Separating hype from reality, real world use cases of AIOps and Assurance for wireless within Catalyst Center

BRKCOC-2483 4:00PM
Cisco IT: Streamlining Network Management and Decisions with Catalyst Center Automation and Splunk

CISCOU-3004 5:00PM
Configuring and Troubleshooting Catalyst Center Templates

Tuesday–10th

BRKEWN-2306 1:30PM
Wireless Network Automation and Assurance with Cisco Catalyst Center

IBOOPS-2391 1:30PM
AI/ML in Cisco Catalyst Center: Transforming Network Operations!

BRKOPS-2697 2:00PM
Unlocking the Automation Power in Catalyst Center for Wired and Wireless Networks

DEVWKS-1004 2:30PM
Deploy Cisco Catalyst Center with Rest-API's

Wednesday–11th

DEVNET-2660 10:00AM
Catalyst Center Network Operations Essentials using UI and APIs

DEVNET-2176 10:30AM
Deploying Cisco Catalyst Center with CICD

BRKTRS-2821 2:30PM
Troubleshooting Strategies for Cisco Catalyst Center & SD-Access

BRKXAR-1013 2:30PM
4 Ways to Streamline Your Licensing with Cisco's Networking Subscription Across Your Portfolio

BRKOPS-2379 3:30PM
Automate Catalyst Center with Cisco Workflows

BRKOPS-2835 4:00PM
5 new things you need to know about Catalyst Center licensing

Thursday–12th

BRKIOT-2016 8:30AM
Streamline Your Success: Automating OT Services with Cisco Catalyst Center Best Practices

BRKOPS-2442 8:30AM
Leveraging Digital Twin for Advanced Network Management with Cisco Catalyst Center

DEVNET-3000 9:30AM
Open-Source GenAI Bot for Catalyst Center

BRKOPS-2570 10:30AM
AI-Powered Automation: Building Smarter Apps for Cisco Catalyst Center Operations

BRKOPS-2492 10:30AM
Let's Deploy Catalyst Center Global Manager (CCGM): Single Pane of Glass for Multiple Catalyst Centers

BRKOPS-2343 10:30AM
Decoding Site Reliability Engineering Through Catalyst Center

BU-led sessions

Catalyst Center

Cisco Live US SD-Access Fabric Learning Map

Sunday–8th

TECENS-2820 9:00AM
Cisco Software-Defined Access LISP: Architecture Overview

LTRENS-2509 9:00AM
Mastering Cisco SD-Access: LISP Pub/Sub and its Benefits Made Simple

TECENS-2850 2:00PM
Security in Enterprise - A cross domain security primer across LAN, wLAN and WAN

Monday–9th

BRKENS-2810 10:00AM
Cisco Software-Defined Access LISP Solution Fundamentals

LTRENS-3751 1:00PM
SD-Access as Code with Cisco Catalyst Center and ISE Automation

IBOENS-1100 2:30PM
Cisco Catalyst Center and SD-Access Design Fundamentals

BRKENS-1804 3:30PM
The Power of Cisco SD-Access LISP Fabric: Simplified Deployment to Advanced Use Cases - Part 1

BRKENS-1851 4:00PM
Zero Trust: Secure the Workplace with Cisco Software-Defined Access

Tuesday–10th

BRKENS-1805 11:00AM
SD-Access in Action: Trusted Outcomes Across Education and Finance-Featuring UC Riverside & CIBC Bank

BRKENS-2824 2:00PM
Deploying Your First Cisco SD-Access Project

BRKENS-2804 4:00PM
The Power of Cisco SD-Access LISP Fabric: Simplified Deployment to Advanced Use Cases - Part 2

IBOENS-2828 4:30PM
Network Quest: Exploring Campus Fabrics and Secure Segmentation

Wednesday–11th

BRKENS-2816 10:30AM
Cisco SD-Access Transit: Advanced Design Principles

IBOENS-2826 10:30AM
Cisco SD-Access Design and Deployment Best Practices

BRKENS-2836 10:30AM
Endpoint profiling and segmentation using AI endpoint Analytics and Cyber Vision for next generation SD Access manufacturing plants

BRKENS-1806 1:00PM
Transforming Enterprise Networks with Cisco SD-Access: Real-World Strategies from CDW

BRKENS-3826 3:30PM
Advanced LISP SD-Access Forwarding Architecture

Thursday–12th

BRKENS-2650 8:30AM
Designing and Deploying Cisco SD-Access with BGP EVPN

BRKENS-2700 8:30AM
Fabric Networking in the Campus: What's the fuss and what are the choices?

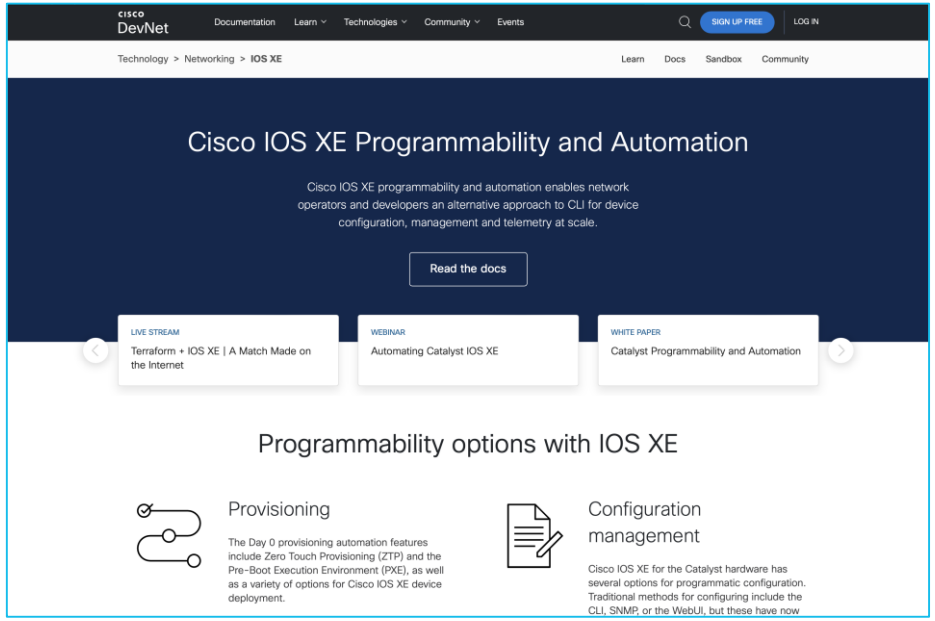
BRKENS-3834 10:30AM
1 to 100: Master All Steps of Automated and Seamless Deployment, Integration, and Migration of Large SDA and SD-WAN Networks

BRKENS-3810 2:30PM
How to Adopt Zero Trust using SD-Access and Default-Deny without Tears

BU-led sessions

Programmability Website

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



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

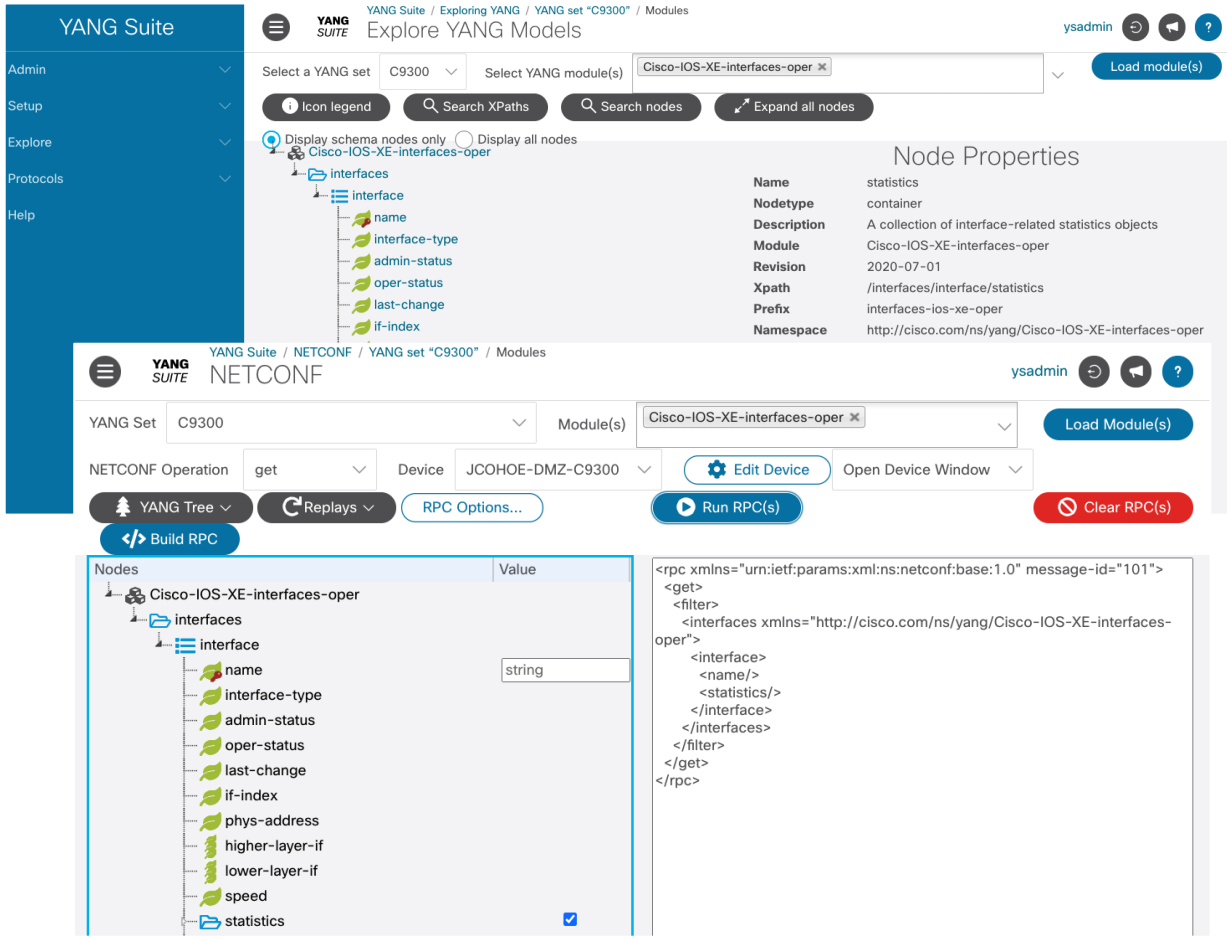
https://developer.cisco.com/iosxe/

Cisco YANG Suite

YANG API Testing and Validation Environment

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

IOS XE / IOS XR / NX OS platforms



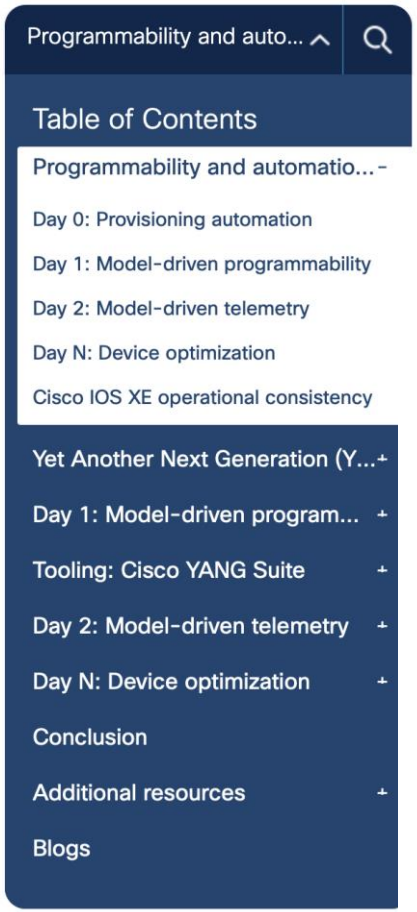
developer.cisco.com/yangsuite

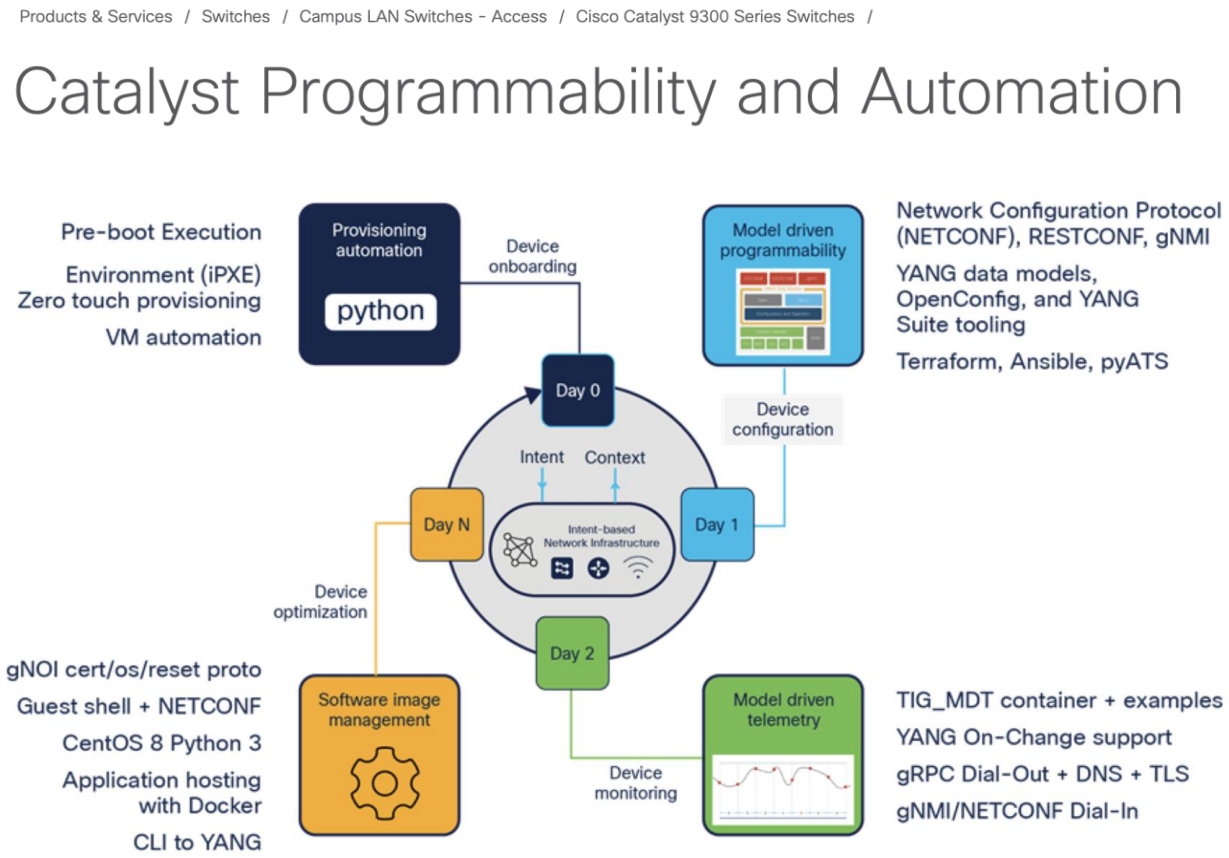
github.com/CiscoDevNet/yangsuite

On-Demand Learning Lab with YANG Suite in Docker
https://developer.cisco.com/learning/labs/intro-yangsuite/

API White Paper

http://cs.co/apiwp





http://cs.co/apiwppdf

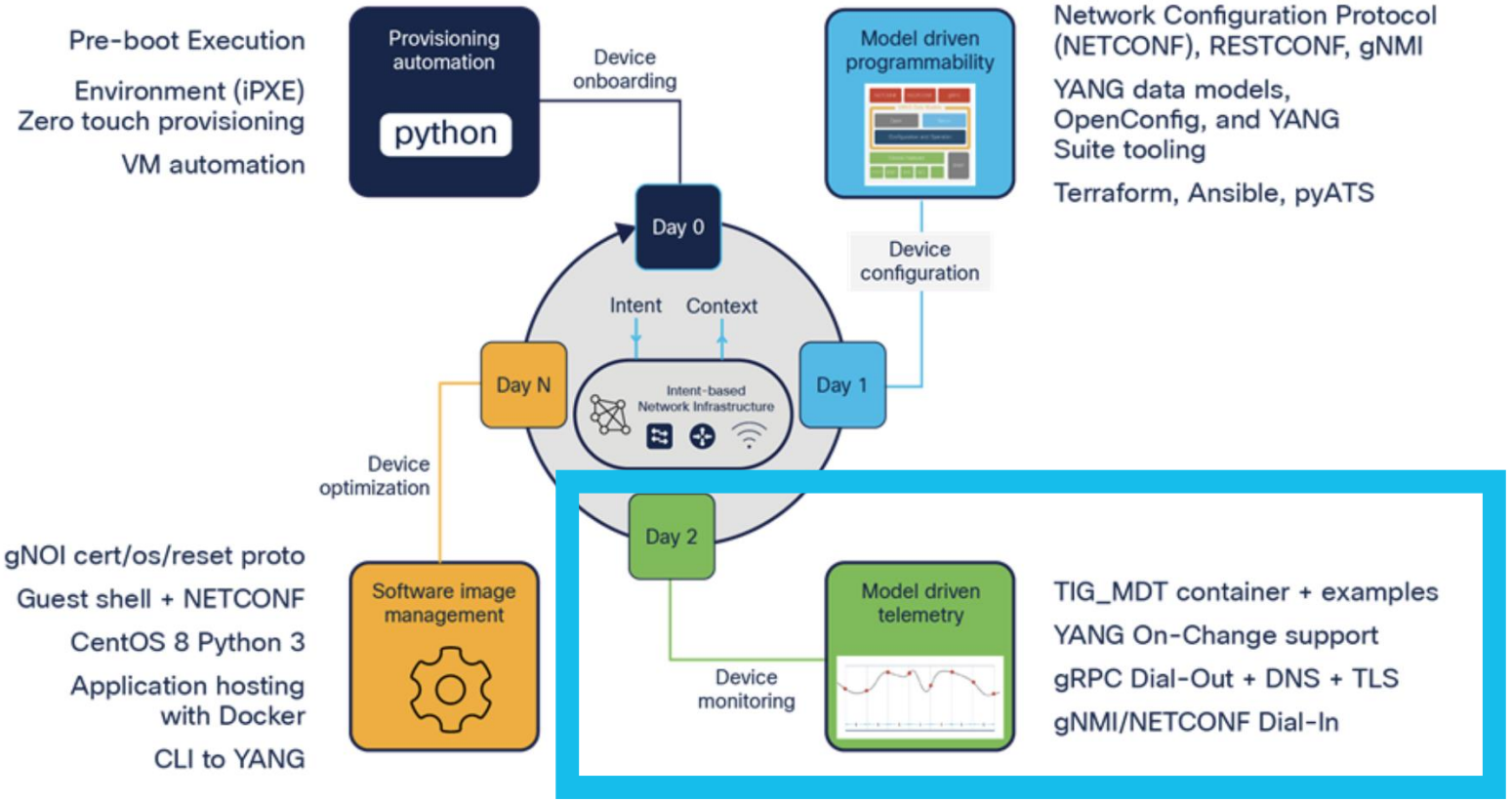
https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9300-series-switches/nb-06-catalyst-programmability-automation-wp.html

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

https://www.youtube.com/watch?v=LdcK5PnPu2I

Model Drive Telemetry (MDT) White Paper

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



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

Introduction to Cisco IOS ...

Table of Contents

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

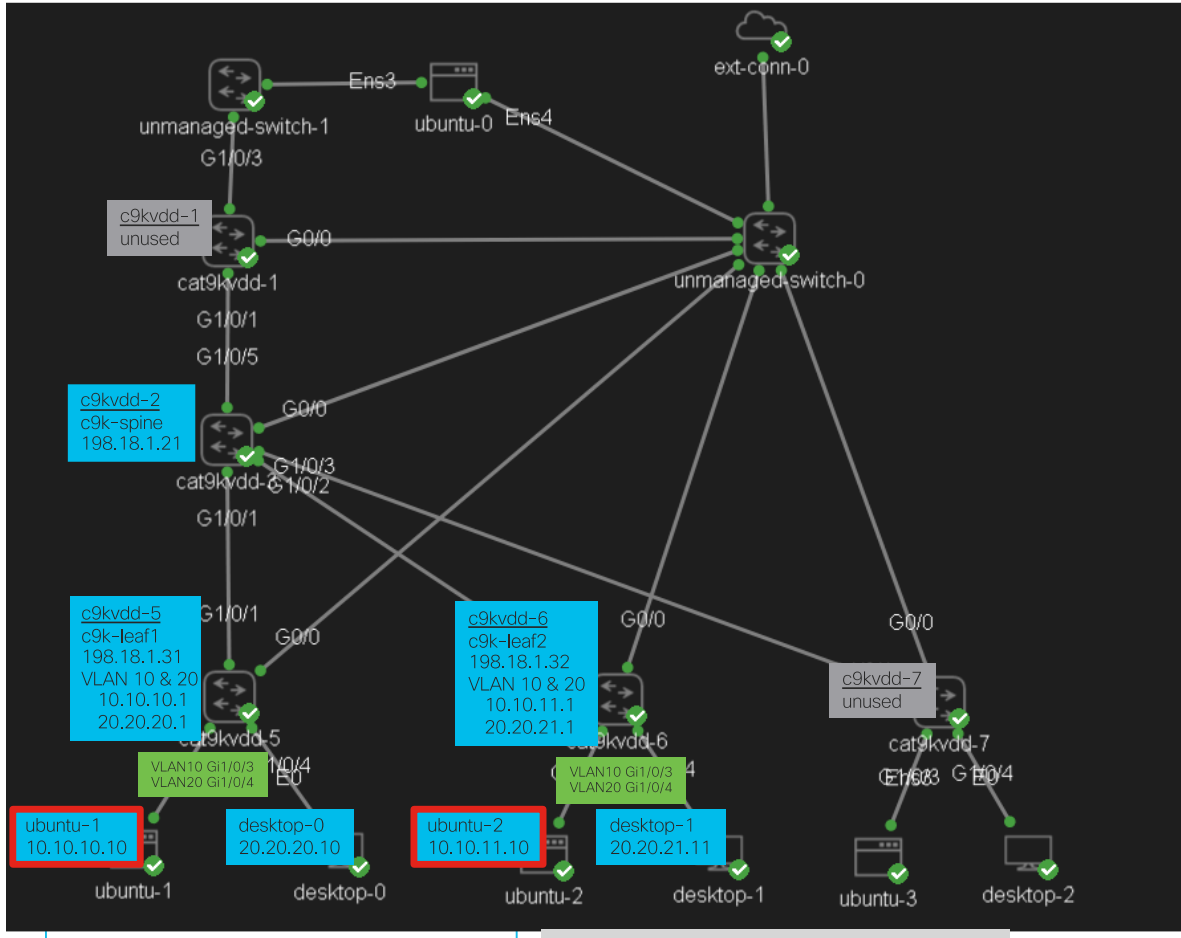
Ubuntu VM Details:

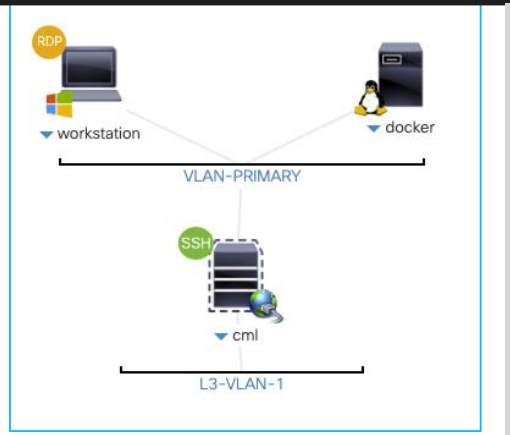
Syslog receiver from all switches
TFTP config backup
See slide

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

C9K VM's





VLAN1
c9k-spine
IP: 198.18.1.21
developer /
C1sco12345
c9k-leaf1
IP: 198.18.1.31
developer /
C1sco12345
c9k-leaf2
IP: 198.18.1.32
developer /
C1sco12345
c9kvdd-1 - unconfigured

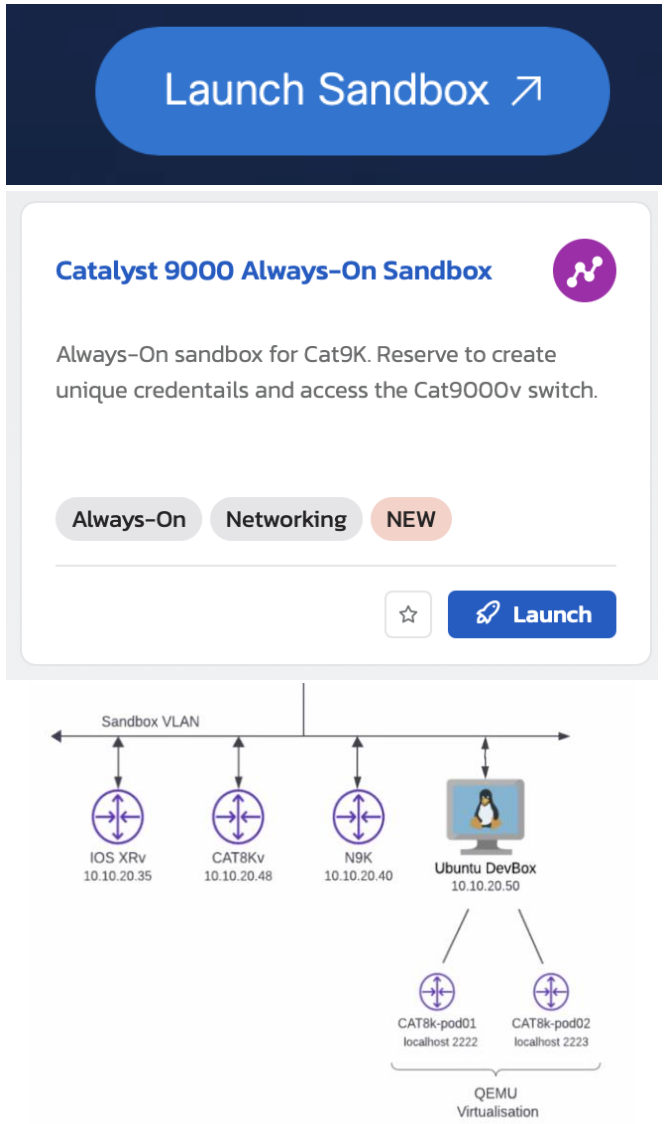
c9kvdd-7 - unconfigured

DevNet Sandbox – overview for Campus and Enterprise

https://developer.cisco.com/site/sandbox/

1. Reservable Physical: C9200, C9300, C9300X including stacks
   About to go into production April 2025
   Usecases: Application Hosting, Power telemetry, etc
2. Reservable Virtual: C8KV Router + NX + XR + Ubuntu VM
   Usecases: Enterprise topology, dual-ZTP
3. Always-On: C9KV
   Usecase: Virtual switch for basic config validation usecases
   DNS: devnetsandboxiosxec9k.cisco.com
   "Launch Sandbox" to get login credentials
4. Always-On: C8KV
   DNS is devnetsandboxiosxe.cisco.com
5. Reservable Catalyst Center (Physical & Virtual)
   CML and C9KV, ISE for SDA

C9KV user is priv15 and has full permissions, there is reset automation for when you break it now too ☺



Additional enablement labs:

1. YANG Suite "Learning Lab 2.0"
   Interactive guide with tool running in Docker container
   "YANG Suite as a service"
2. dCloud Programmability Lab
   EVPN topology with all programmability features enabled

C9KV Always-On with reservation

New C9kV Sandbox now available!

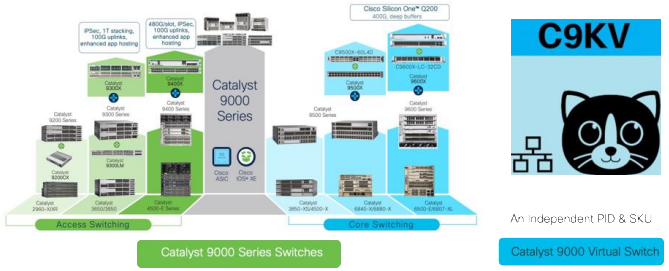
Sandbox access: https://devnetsandbox.cisco.com/DevNet/catalog/Cat9k-Always-On_cat9k-always-on
Hostname: devnetsandboxiosxec9k.cisco.com

C9KV Sandbox Capabilities:

- Cisco IOS XE 17.15.1 Virtual Catalyst 9000 UADP 8 port switch
- “Always-On” outside of the Cisco Network DMZ in a colocation on hardware in VMWARE/OVA
- Accessible with reservation though Cisco’s Developer Enablement platform
- Reservable by anybody including customers, partners and external
- Enabled for **read-only usecases** with SSH/CLI and API: NETCONF/YANG, RESTCONF, gNMI
- Support for Model-Driven Telemetry and basic configuration changes through the API
- Supports 40 concurrent sessions

```
sdeweese@SDEWEESE-M-C20V ~ % ssh sdeweese@devnetsandboxiosxec9k.cisco.com
(sdeweese@devnetsandboxiosxec9k.cisco.com) Password:

CAT9k_AO#
CAT9k_AO#
CAT9k_AO#
CAT9k_AO#
```

```
CAT9k_AO#sh inv
NAME: "Switch 1", DESCR: "Catalyst 9000 UADP 8 Port Virtual Switch"
PID: C9KV-UADP-8P      , VID: V01  , SN: 98DVJUONW1X

NAME: "Switch 1", DESCR: "Catalyst 9000 UADP 8 Port Virtual Switch"
PID: C9KV-UADP-8P      , VID: V01  , SN: 98DVJUONW1X


CAT9k_AO#sh ver
Cisco IOS XE Software, Version 17.15.01
```

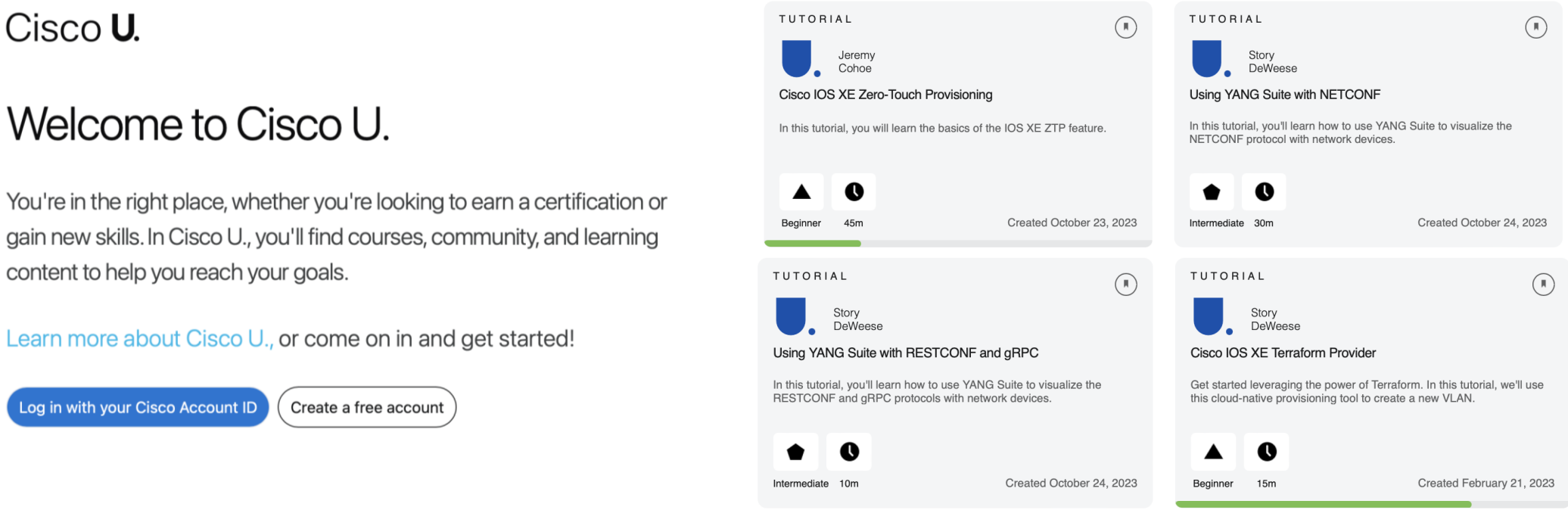


Cisco University (Cisco U) part of L&D

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

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

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



Would You Like to Know More?

Catalyst 9000 Series Enterprise Switches

cisco.com/go/cat9K

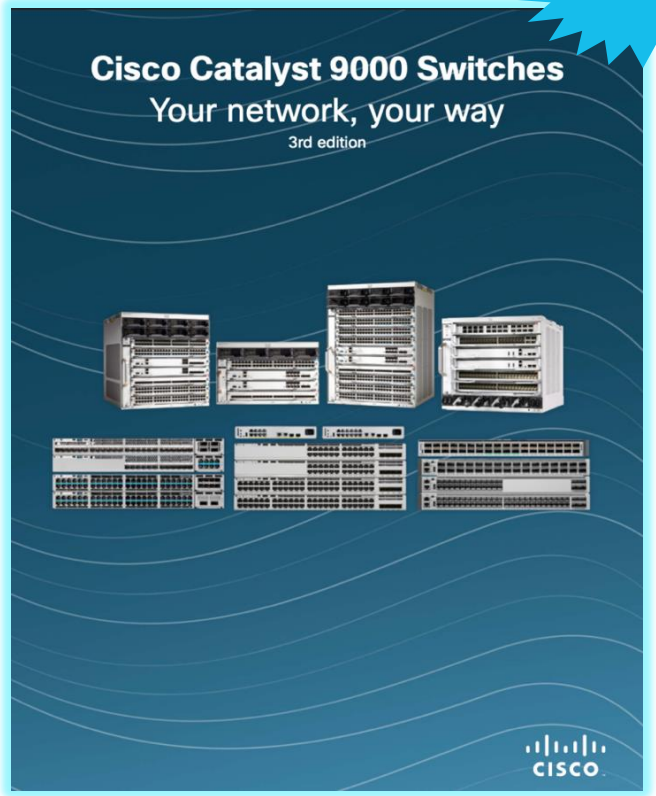
Cisco Catalyst 9000 at-a-Glance

Cisco Catalyst 9000 Family FAQ

Catalyst 9000 Series - Cisco Community

Catalyst 9000 Series – CiscoLive Library

FREE



cs.co/cat9kbook

Cisco Catalyst TV @ YouTube

SUBSCRIBE

This channel is all about Cisco Catalyst Platforms and its services and software solutions. Subscribe and Explore Playlists Catalyst Switching and Catalyst Programmability & Automation for videos and demos by the Technical Marketing Engineers on latest, relevant and exciting topics.

Cisco Catalyst TV
24 subscribers
SUBSCRIBE

HOME VIDEOS PLAYLISTS CHANNELS DISCUSSION ABOUT

Uploads ▶ PLAY ALL

iPXE Day 0 Automation
3 views • 5 days ago

Cisco YANG Suite - The Wait Is Over
No views • 5 days ago

Cisco UPOE+ and PoE Analytics
2 views • 5 days ago

C9K Precision Timing PTPv2 IEEE1588-2008
1 view • 5 days ago

Application Hosting with Catalyst 9000 Demo
2 views • 5 days ago

Introducing Catalyst 9600X
Cisco Catalyst TV

Introducing Catalyst 9400X
Cisco Catalyst TV

Indoor IoT Services for Smart Buildings
Cisco Catalyst TV

Cisco IOS XE Terraform Provider Introduction and Demo
Cisco Catalyst TV

NETCONF with YANG Suite
Cisco Catalyst TV

http://cs.co/CatalystTV

Questions

Complete Your Session Evaluations

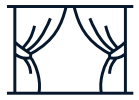
Complete a minimum of 4 session surveys and the Overall Event Survey to be entered in a drawing to win 1 of 5 full conference passes to Cisco Live 2026.

Earn 100 points per survey completed and compete on the Cisco Live Challenge leaderboard.

Level up and earn exclusive prizes!

Complete your surveys in the Cisco Live mobile app.

Continue your education



Visit the Cisco Showcase for related demos

Book your one-on-one Meet the Engineer meeting

Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs

Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

Thank you


CISCO