
cisco Live!
Let's go

The bridge to possible

Infrastructure as Code

with Cisco Catalyst 9000 Virtual

Ama Owusu-Hammond & Jeremy Cohoe
Technical Marketing Engineering
@jeremycohoe

cisco Live!

BRKOPS-2455

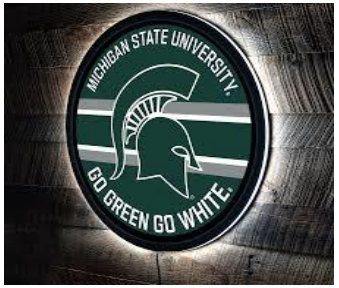
Abstract...

Cisco Modelling Labs, Container Lab, KVM and Dockers... oh my! The way in which we can model networks and create infrastructure from code is evolving. In this session we will use virtualization technologies like CML and Docker in order to virtualize and simulate common network topologies and configurations. We will gain confidence in configuration management processes and tooling in the simulated lab and make the transition into production easier. The complete IOS XE programmability and automation lifecycle will be demonstrated and discussed using the virtual form factor alongside YANG Suite for testing and validation.

About Ama

Let’s Connect: eowusuha@cisco.com

- From Accra, Ghana
- Michigan State University, BSc Electrical Engineering
- Cisco – Enterprise Networks TME – ~4 yrs.
- Focus areas: C9200, Switching Security Solutions, C9Kv
- Hobbies: Travelling, Reading, Photography



About Jeremy

WxT/jcohoe@cisco.com

FYI

- From Vancouver, BC, Canada
- Amateur Radio Operator VA7NSA & BCWARN.net volunteer
- Canadian Forces Army Signals
- UBC – Wireless Infrastructure
- Cisco – Campus & Enterprise Network Automation
- Automation enthusiast, home assistant, raspberry pi, etc
- Supporter of Electronic Frontier Foundation (EFF) with a passion for digital rights

Agenda

- Intro to Catalyst 9000 Virtual (C9KV)
- Catalyst 9000 Family
- Cisco Modelling Labs
  - CML + C9KV
- C9KV Capabilities
  - Demo: Interface Template
- Operating in CML
- C9KV for Programmability Usecases
- Use Cases: EVPN, SDA
- Resources

Cisco Catalyst 9000 Virtual… aka the C9KV

1. The C9KV is available within Cisco CML 2.5+ in beta format
2. There is no BU or TAC support for the C9KV node or “product”
3. It has been released as “beta” for any early adopters to trial and provide feedback and input
4. About a dozen customers are also using this as part of unofficial beta program with Cisco Engineering
5. Their usecases are not limited to CML and expand in scope to include KVM, Docker & ContainerLab deployments

Using Catalyst 9000v Nodes in CML

Starting with CML 2.5, the ISO with reference platforms for CML includes the CAT 9000v IOS-XE VM image version 17.10.01prd7.

Both dataplane flavors of the Catalyst 9000v require 4 vCPUs and 18 GB of RAM *per node*. The memory is critical, memory oversubscription may result in nodes crashing on bootup or during operations. The CAT 9000v requires CML 2.3 or higher.

Reference Platforms and Images

Disk Image Version: `refplat-20230211-fcs.iso`

Ref. Platform/Image	Description	Version
CAT 8000V	IOS-XE Catalyst 8000V Edge Software Router	17.09.01a
CSR 1000v	IOS-XE Cloud Services Router	17.03.06
CAT 9000v	IOS-XE Catalyst 9000 Virtual Switch (beta)	17.10.01prd7

https://developer.cisco.com/docs/modeling-labs/#!reference-platforms-and-images
https://developer.cisco.com/docs/modeling-labs/#!cat-9000v

How can I use the C9KV ?

CML

- Build entire virtual networks with different virtual OS
- Support with deployment from CML team
- Fully customizable workspace with memory/CPU resources

dCloud

- Demo environments with C9Kv in CML instances
- dCloud for Programmability, EVPN, SDA use cases
- Available on-demand with set resources

EFT with Cisco Engineering

- Specific requirements for virtual network operating testing and validation
- With guidance from your local Account Team

Your C9KV feedback is important as we continue to bring this product to market

Webex Teams: https://eurl.io/#h6Pe-Qgmo
E-mail: c9kv-interest-ext@external.cisco.com
Form: https://forms.office.com/r/b36aeEsjyA

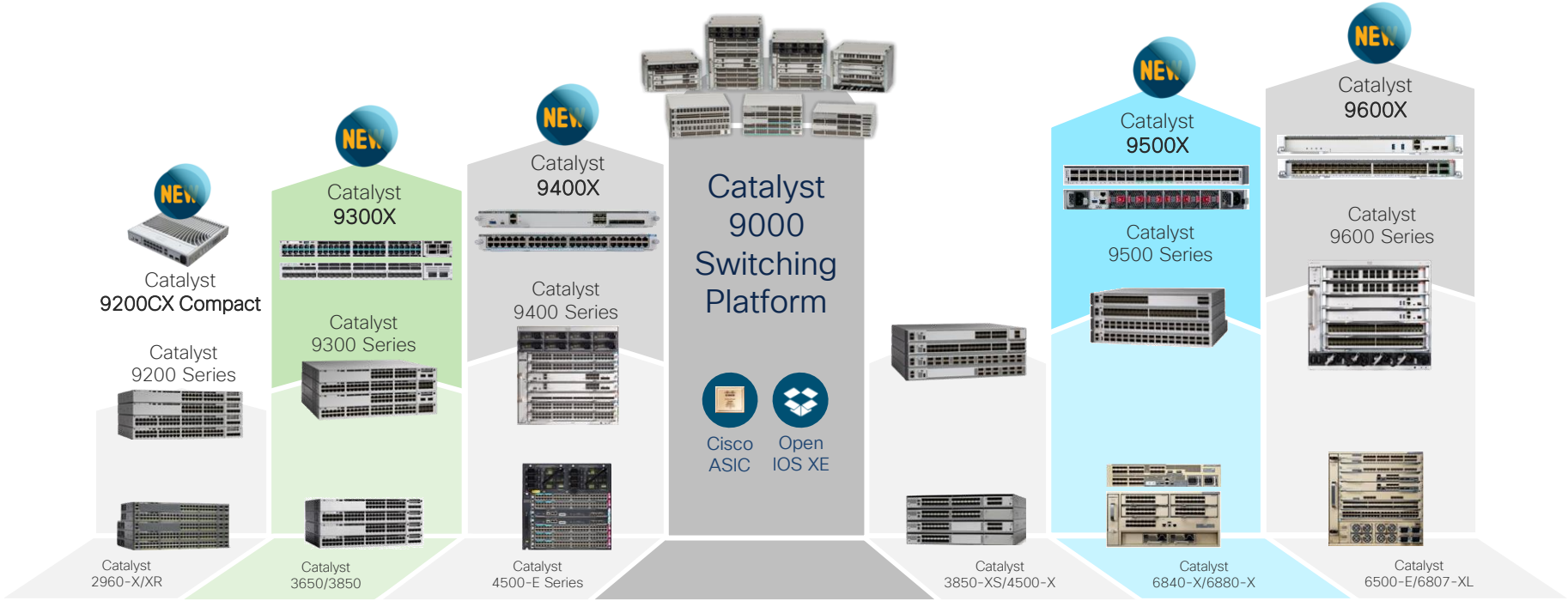
The Cat9KV is in public beta and available in CML and within Cisco dCloud

We want to understand your use cases and to ensure success with topology validation and configuration management, tooling and integrations.

There is no Tac or BU support during the beta.

Cisco Catalyst 9000 Switching Portfolio

Adding the “X factor” to the industry’s leading switching family



What is the C9000 switch family good at doing ?

Transport Security- IPsec, WAN MACsec	BGP/EVPN
Application Hosting – ASAc Firewall	Policy based automation (SD-Access)
Support mGig speeds	Model Driven Telemetry
POE capabilities with support up to 90W	Highest bandwidth for stacking in industry ~1Tb
Trustworthy solutions	PTP for time sensitive networks
Encrypted Traffic Analytics	Secure data wipe for RMA scenarios

What features does the IOS-XE have ?

1 L2 Forwarding

- Private VLANs
- REP
- Flexlinks
- Bonjour / SDG
- CDP Bypass
- Interface Template Configuration

2 L3 Forwarding

- NAT / PAT
- IP FRR / LFA
- Adv. PBR (some features)
- WCCP
- GLBP

3 Multicast & MVPN

- PIM Bidir
- PIM Snooping
- MVPN (Rosen & mLDP)
- Multicast mGRE
- MSR
- MVR

4 Fabric (SDA/EVPN)

- LISP Extranet
- L2 EVPN TRM

5 Management (CLI/GUI/API)

- RSPAN
- ERSPAN + IPsec Encap
- Adv. SPAN (some features)
- Adv. EPC (some features)
- Open Flow 1.3

6 HA & SVL

- L3 NSR
- GIR - L3 EIGRP
- Quad Sup SVL (9600X)

7 QoS & Buffering

- Egress QoS ACL TCAM
- Egress Aggregate Policing
- Egress WTD, WRED Thresholds
- QoS on Portchannel interface
- AVC/NBAR based QoS
- Netflow based Microflow Policing

8 MPLS (L2/L3VPN)

- VPLS & H-VPLS
- EoMPLS over LACP & PaGP
- EoMPLS over GRE & mGRE
- MPLS over GRE & mGRE
- TE FRR (some features)

9 Timing (NTP/PTP)

- PTP
- AVB

10 Security & Crypto

- PACL & VACL
- MAC ACL
- FQDN & Reflexive ACL
- L2 CMD
- ACL Hitless Update
- Adv. 802.1x (some features)
- RADSEC

11 NetFlow & AVC

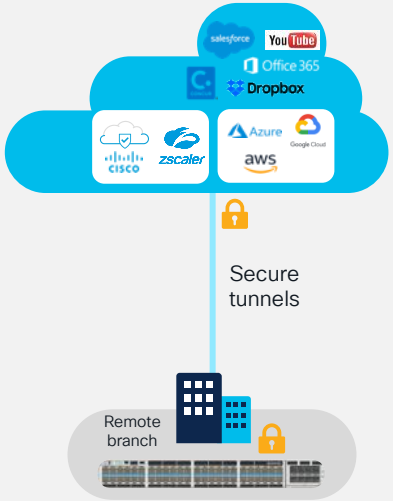
- AVC
- Egress (output) FNF
- BGP NH & AS records
- sFlow

12 Tunnels (GRE/QinQ)

- L2oMGRE
- 802.3ad full QinQ
- L3VPN over GRE
- IPv6 Auto Tunnels

C9K/IOS-XE deployments

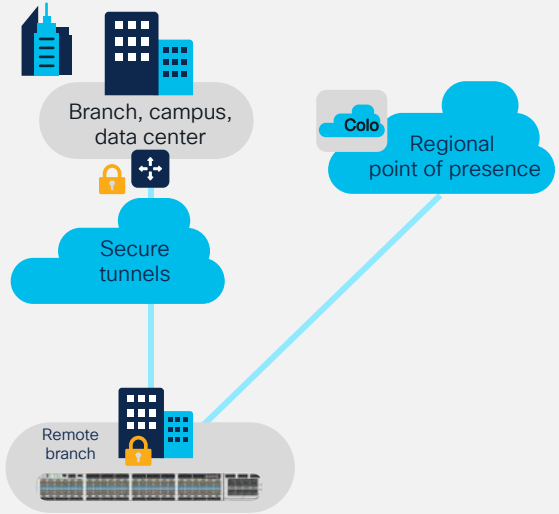
Secure connectivity to anywhere use case with IPsec support on C9300X



Cisco Catalyst 9300X

Site to cloud

Standards-based IPsec for secure direct internet access and cloud-native workloads



Cisco Catalyst 9300X

Site to site

100G line-rate IPsec encryption with low-latency forwarding

Catalyst 9000 Series – Common Building Blocks

Programmable x86 Multi-Core CPU

Application Hosting
Secure Containers

Open IOS XE®

Model-Driven APIs
Modular Patching

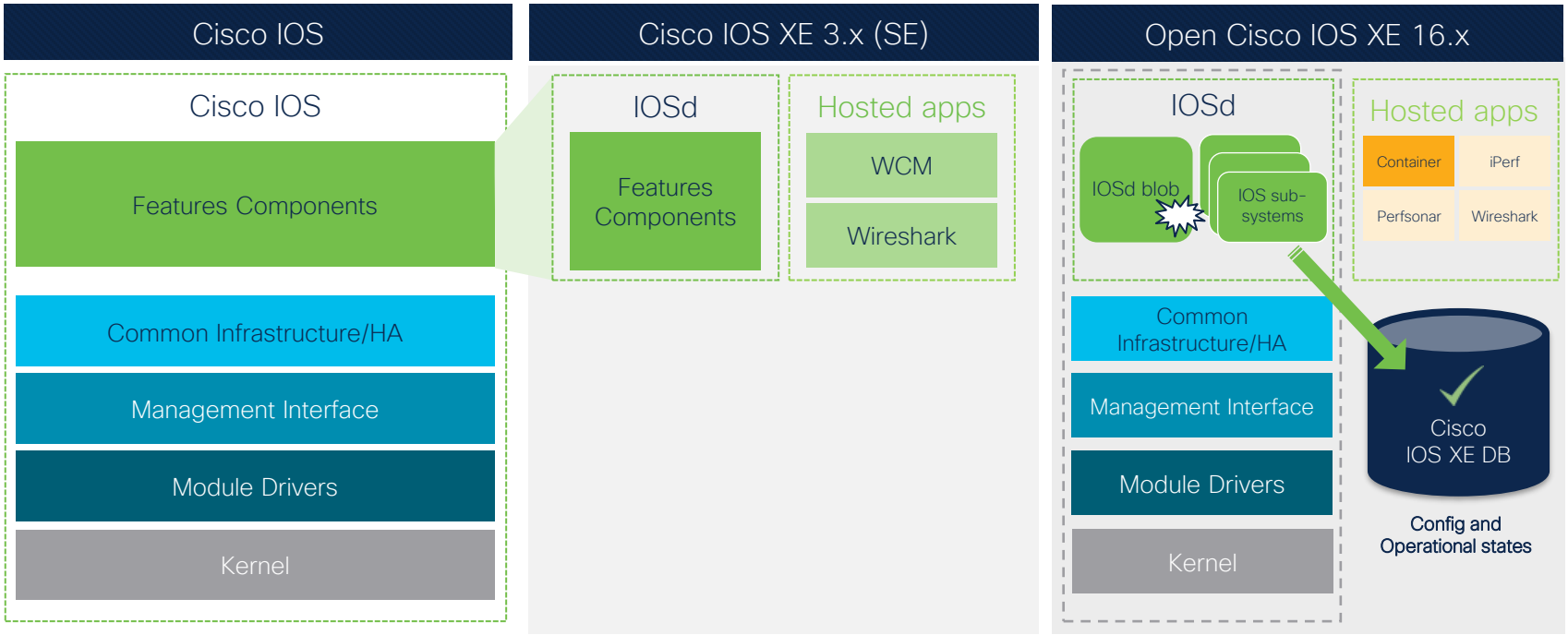
Cisco UADP & Silicon One™

Programmable Pipeline
Flexible Tables

Same IOS XE image for both **UADP*** and **Silicon One** C9K platforms

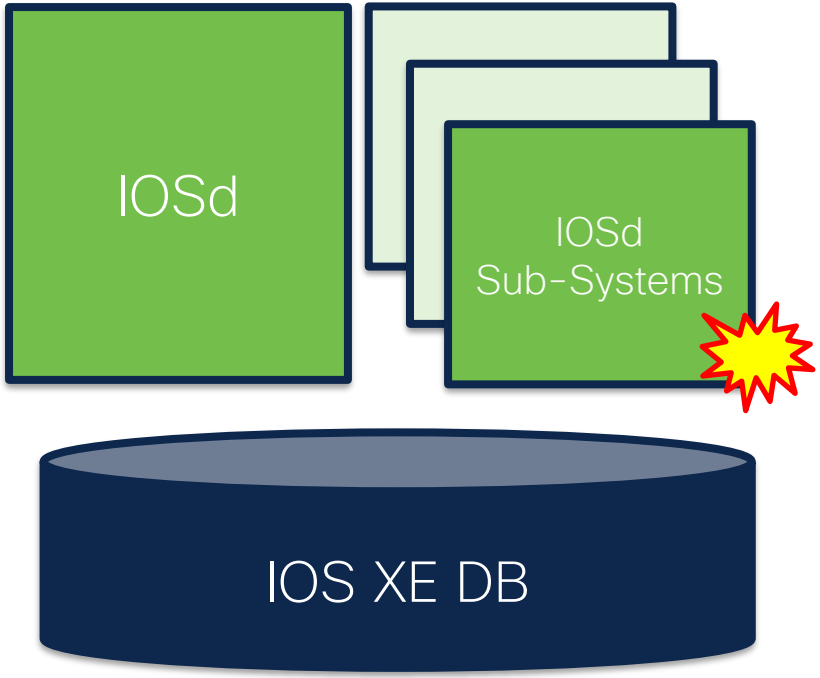
* Catalyst 9200 Series uses IOS-XE Lite

What is “Open” Cisco IOS-XE?



Modern Software Architecture - with the same look and feel

Open IOS-XE IOS Sub Systems



STP
OSPF
BGP
MPLS
etc

Failure of one IOS XE Sub-System(s) - keeps rest of the system intact

IOSd Sub-Systems enhance IOS Resiliency

Custom ASICs – Programmable Silicon

Cisco **Unified UADP**(UADP®)

Cisco **Silicon One**™

Innovations in virtualizing the ASICs on C9K

1. UADP ~ 2017
2. Silicon 1 – Q200. ~2021
3. UADP-IPsec (yet to be made available) ~2023
4. Future innovations ~2026?

Flexible & Programmable ASICs – Adapt to New Technologies

Catalyst 9000

+

Cisco Modelling Labs

Catalyst 9000 + Virtual

Cisco Modeling Labs

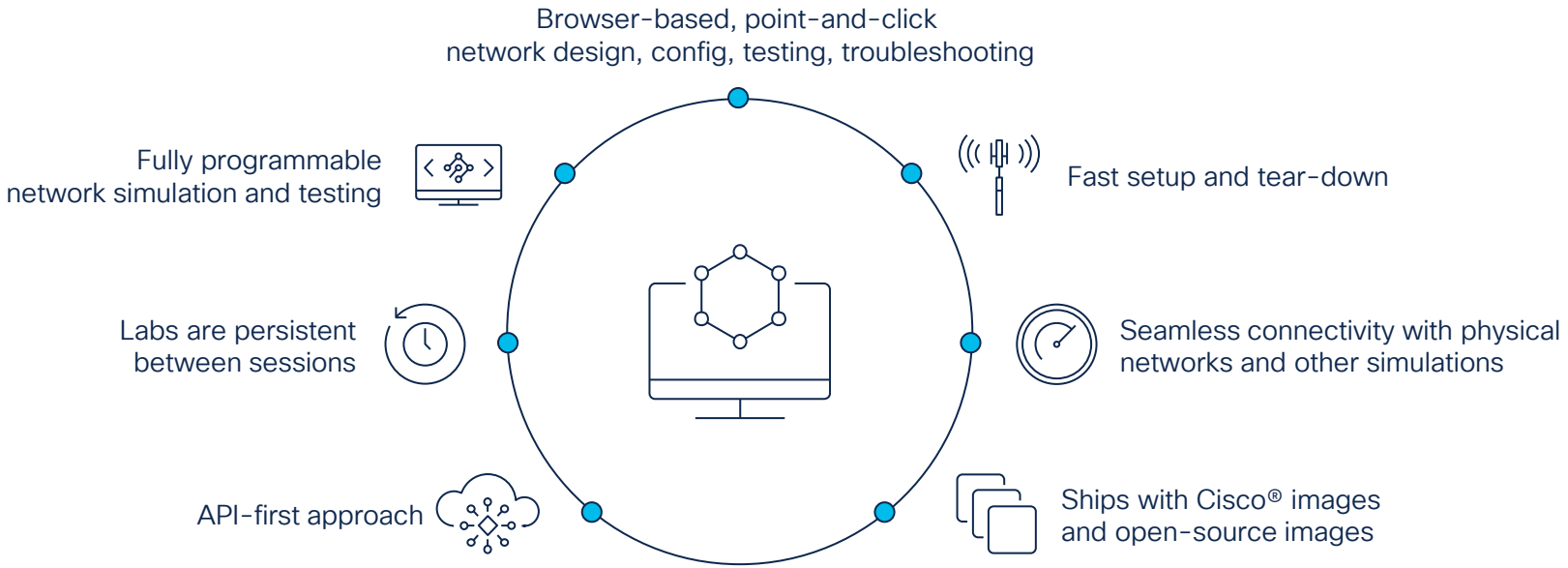
About CML

Cisco Modelling Labs

Meet Cisco Modeling Labs

Cisco's premier platform for network simulation

CML is the Cisco Modelling Labs, it is a virtualization software for network engineers to easily create network topologies, links, and clients in order to simulate and test various topologies and features



CML Architecture overview

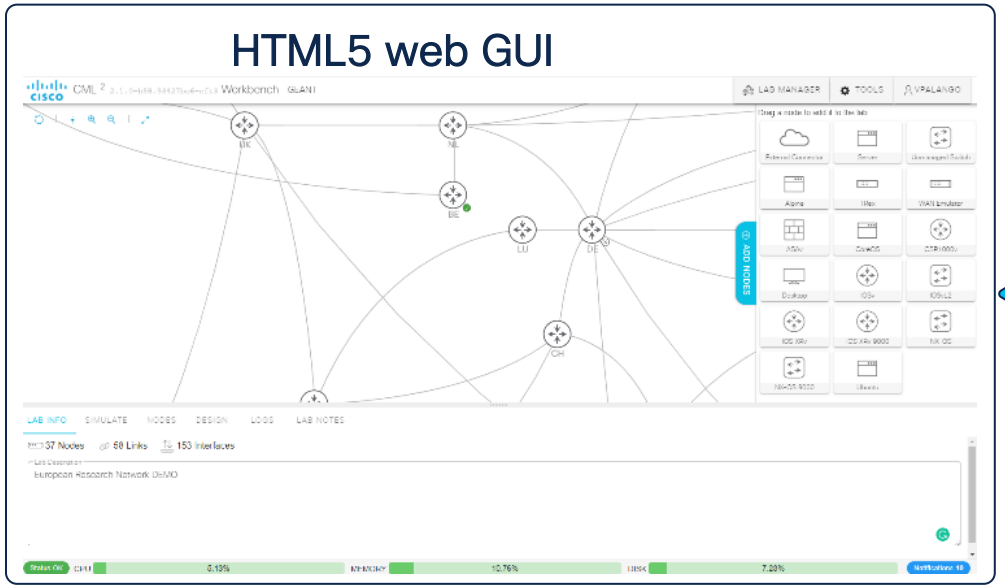
Python client library

Custom apps

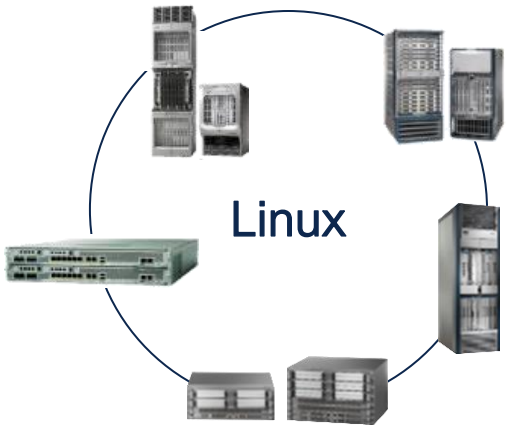
Swagger API documentation

Rest API

HTML5 web GUI

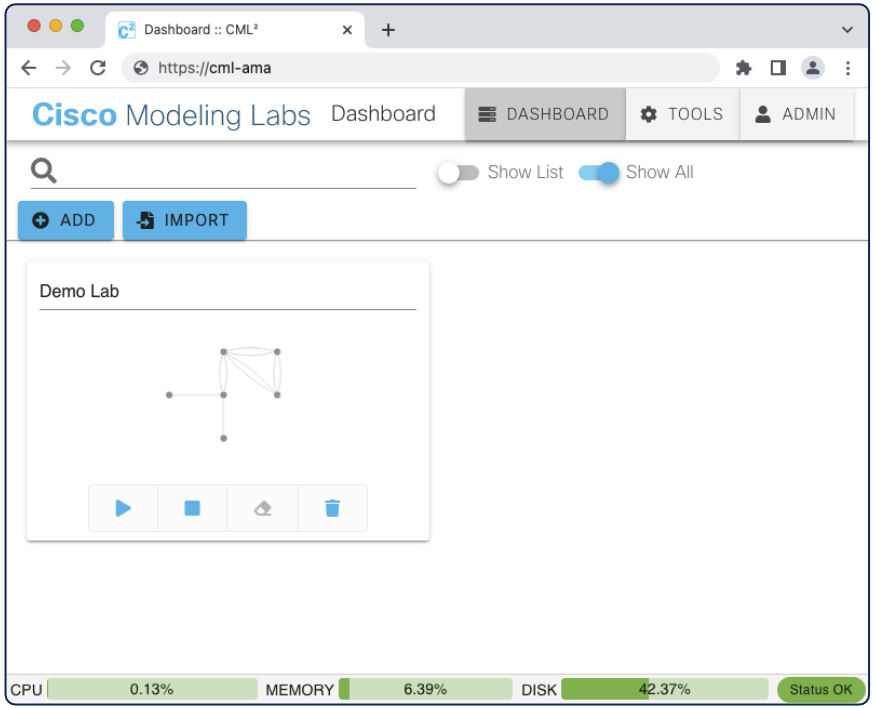


CML Platform



Create and configure labs

- Drag and drop nodes into the canvas
- Create links between nodes
- Node configuration through the console
- Import labs and node image definitions

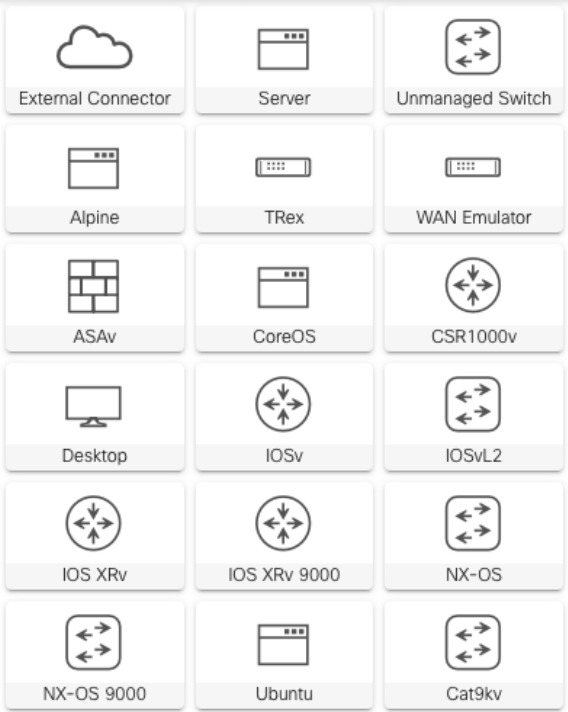


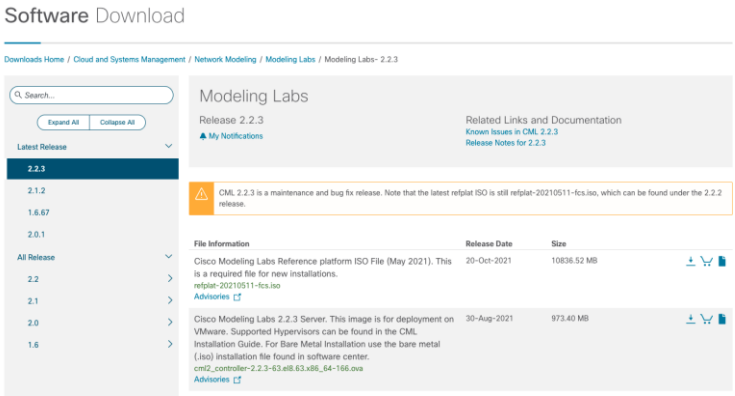
What is a “node” in CML ?

The “node definitions” in CML are the Virtual Machines and Network Devices that are available for use in the lab workspace.

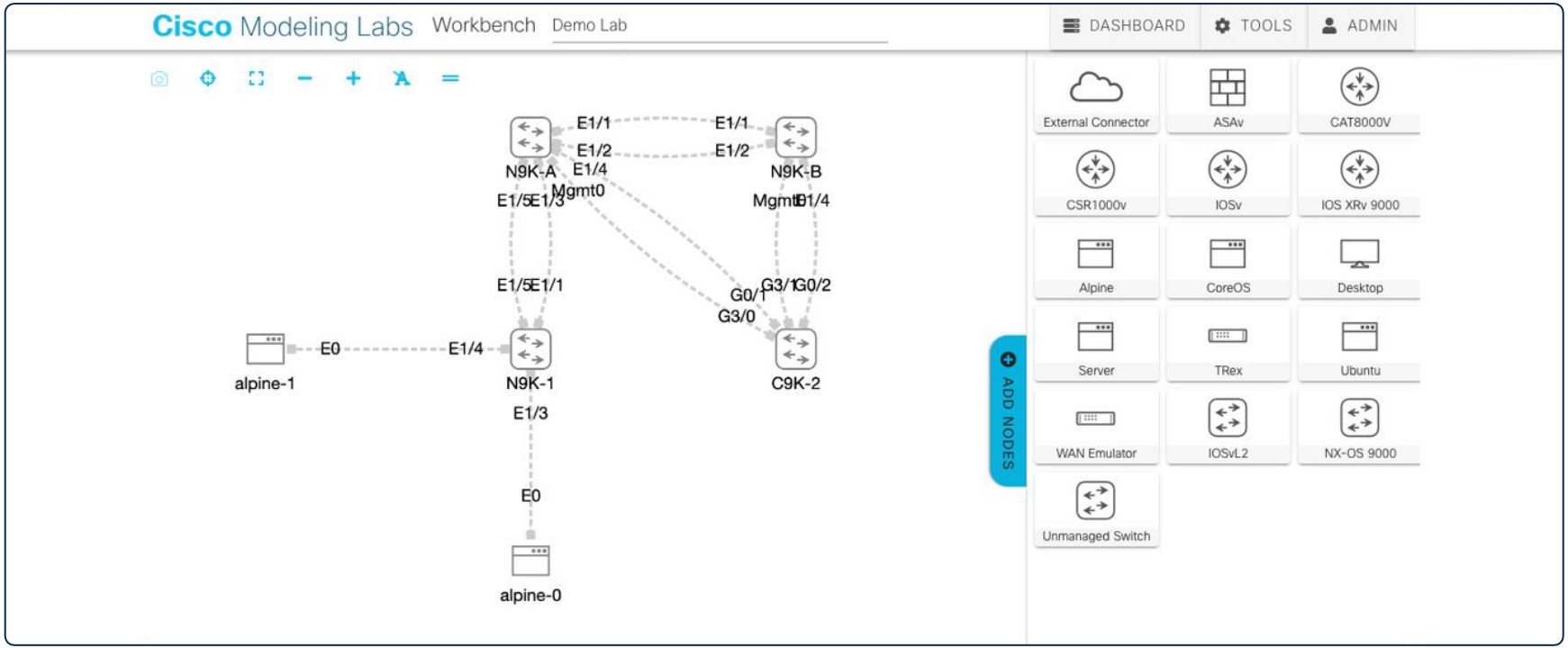
Node examples are: Windows 11 Desktop, Ubuntu VM, Alpine Linux, etc
Node examples are: IOS XRv, NX-OS, cat9kv, IOS, IOL, etc

The node definitions are available along side CML.OVA from software.cisco.com and is called the “reference platforms ISO file”



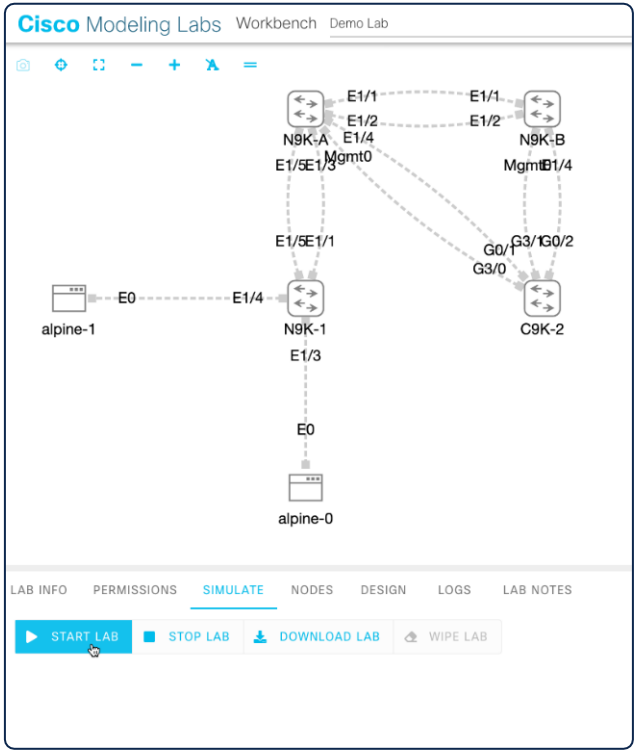


Drag and drop nodes into the canvas

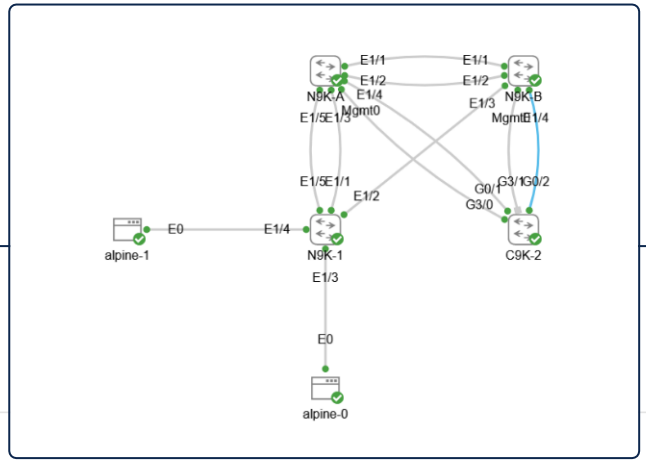


Simulate and analyze network behavior

- Simulate link breaks and network disruptions
- Analyze packet captures on links
- Connect simulated network with external network and physical devices



Simulate link breaks and network disruptions



LINK INFO SIMULATE PACKET CAPTURE

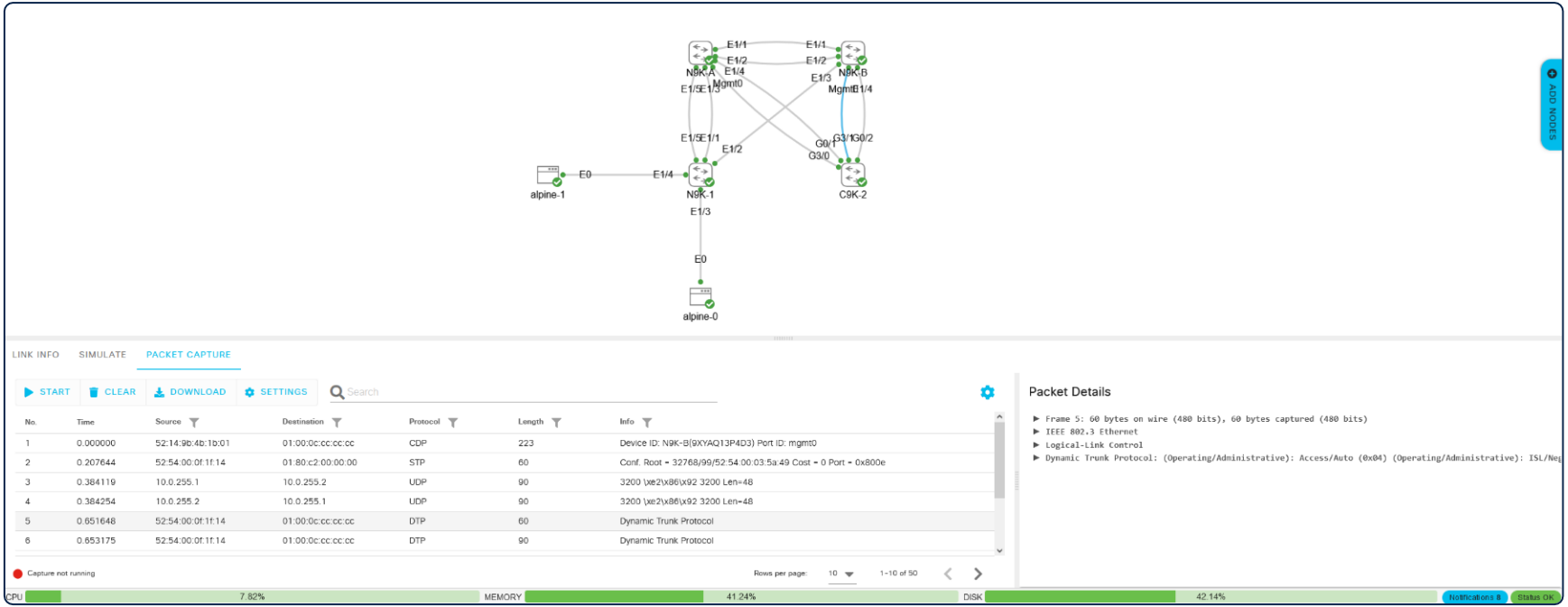
Disconnect stops the interface on each side of the link simulating a cable disconnection

STOP DISCONNECT

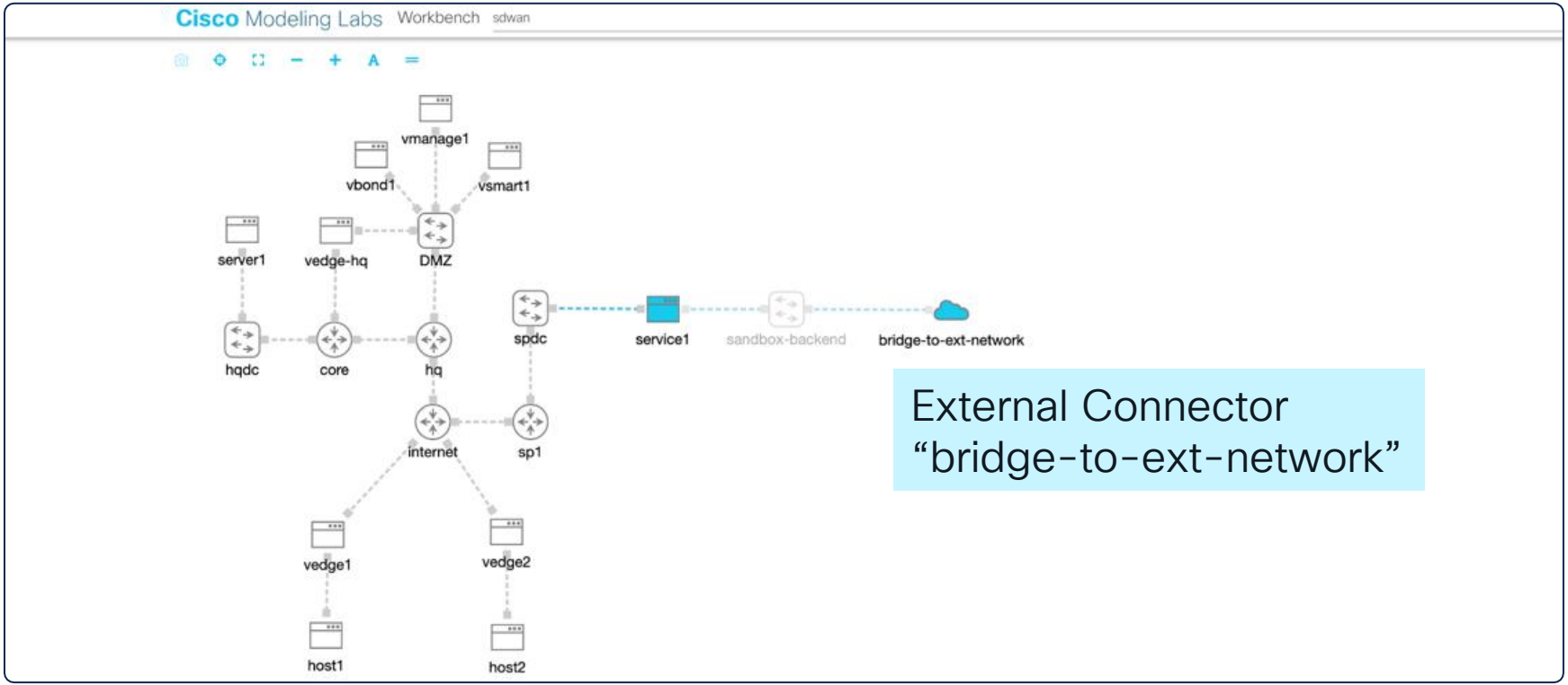
Link Statistics

Node	Interface	Running	Packets	Bytes
N9K-B	Ethernet1/4		561109	83324811
C9K-2	GigabitEthernet0/2		124806	16352462

Analyze packet captures on links

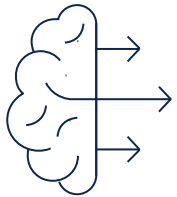


Connect simulation to external networks / hardware



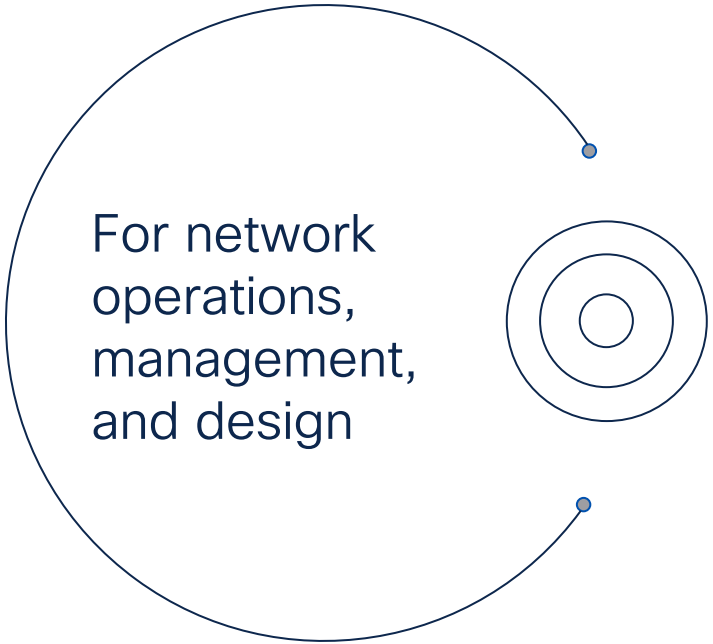
CML usecases

For network architects and engineers



- Validate designs
- Reproduce and test bugs
- What-if scenarios
- Prepare for network changes
- Automation

CML usecases

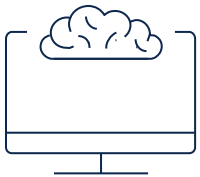


Design virtual or hybrid physical/virtual networks

- What-if scenarios
- Reproduce and test bugs
- Prepare for network changes
- Capture changes and impacts
- Confirm alerts
- Automation

CML usecases

For learning



- Lab practice to prepare for cert exams
- Quickly build and destroy virtual networks without impacting a physical network
- Share your network configurations with your friends and co-workers to test each other's skills

Become a power user

Cisco Live Amsterdam Exclusive:
Save 25% on CML-Personal and CML-Personal Plus.

Visit the Learning & Certification booth to learn more.

cisco Live!

Powered by
Cisco Modeling Labs

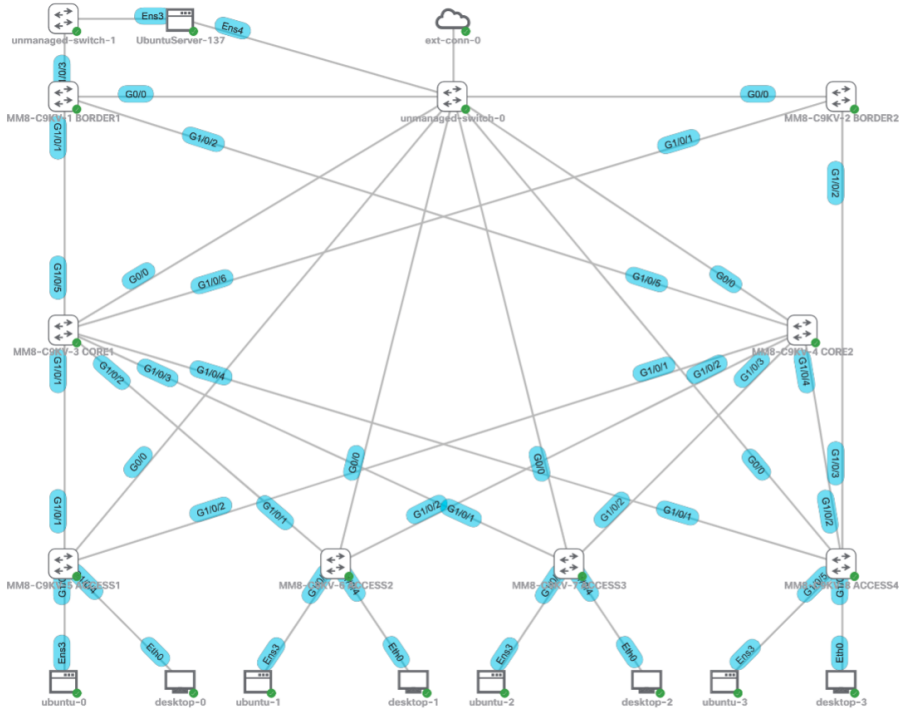
CML Personal users
CML Support community:
CML Personal Community Support

CML Enterprise users
Contact TAC for support

CML and C9KV demo

Larger Virtualized Topology

CML on UCS with 3.2TB RAM = 256 C9KV nodes tested

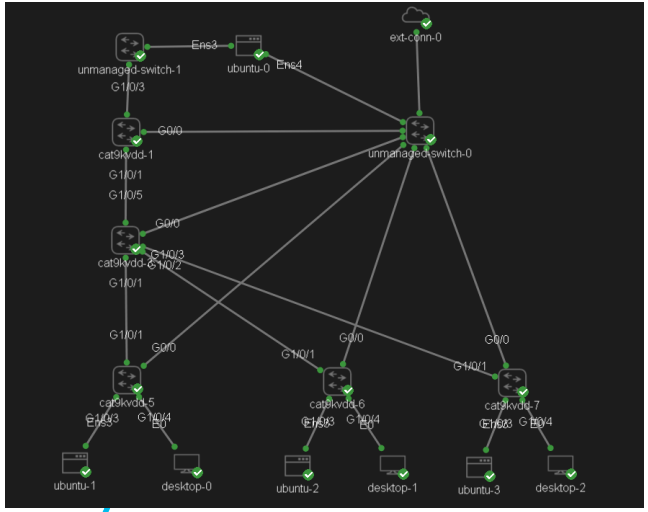


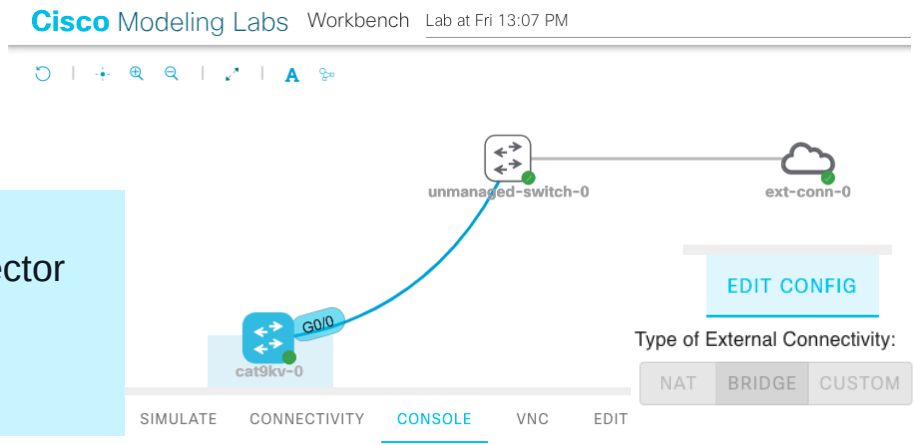

```
MM8-C9KV-7#sh ver
Cisco IOS XE Software, Version 2021-05-18_06.48_vikagarw
Cisco IOS Software [Bengaluru], Catalyst L3 Switch Software (CAT9K_IOSXE), Experimental
vikagarw/git_ws/polaris_dev 100]
Copyright (c) 1986-2021 by Cisco Systems, Inc.
Compiled Tue 18-May-21 06:43 by vikagarw


Cisco IOS-XE software, Copyright (c) 2005-2021 by cisco Systems, Inc.
All rights reserved.  Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0.  The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY.  You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0.  For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.


ROM: IOS-XE ROMMON
BOOTLDR:
MM8-C9KV-7 uptime is 1 year, 21 weeks, 2 days, 2 hours, 32 minutes
Uptime for this control processor is 1 year, 21 weeks, 2 days, 2 hours, 35 minutes
```

RTP “MM8” Lab
With UCS/ESXi
Uptime: 1+ years
IOS XE 17.4

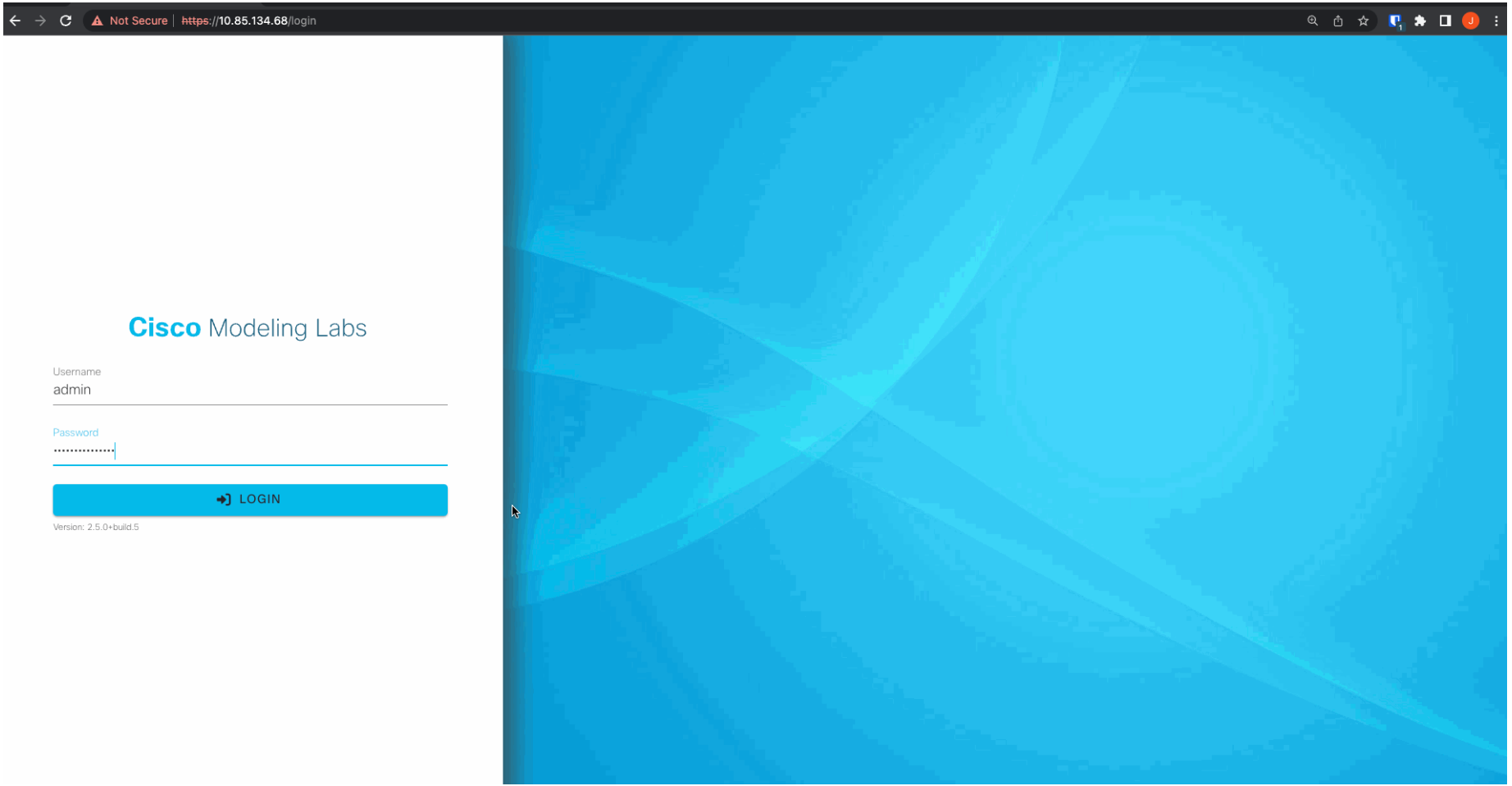
Topology with C9KV

- Drag and Drop the C9KV node onto the dashboard
- Create and unmanaged switch and an external connector
- External Connector should be set to bridge the traffic
- Create links between the nodes
- Access C9KV console to manually configure






```
cisco C9KV-UADP-8P (VXE) processor (revision VXE) with 762097K/3075K bytes of memory.
Processor board ID 9UOQ3K9ZBS0
32768K bytes of non-volatile configuration memory.
18874368K bytes of physical memory.
6201343K bytes of virtual hard disk at bootflash:.

Base Ethernet MAC Address          : 52:54:00:12:f7:b5
Motherboard Assembly Number        :
Motherboard Serial Number          :
Model Revision Number              :
Motherboard Revision Number        :
Model Number                       :
System Serial Number               : 9UOQ3K9ZBS0
CLEI Code Number                   :


No startup-config, starting autoinstall/pnp/ztp...

Autoinstall will terminate if any input is detected on console

Autoinstall trying DHCPv6 on GigabitEthernet0/0



        --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]:
Autoinstall trying DHCPv4 on GigabitEthernet0/0

Autoinstall trying DHCPv6 on GigabitEthernet0/0,Vlan1
```

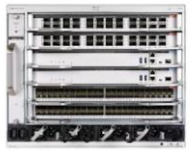


Catalyst 9000 Virtual Switch

Capabilities

C9KV Positioning within C9K Family

Catalyst 9400
Series Switches



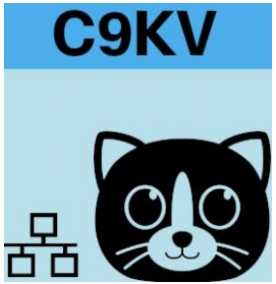
Catalyst 9600
Series Switches

Catalyst 9500
Series Switches

Catalyst 9200
Series Switches

Catalyst 9300
Series Switches

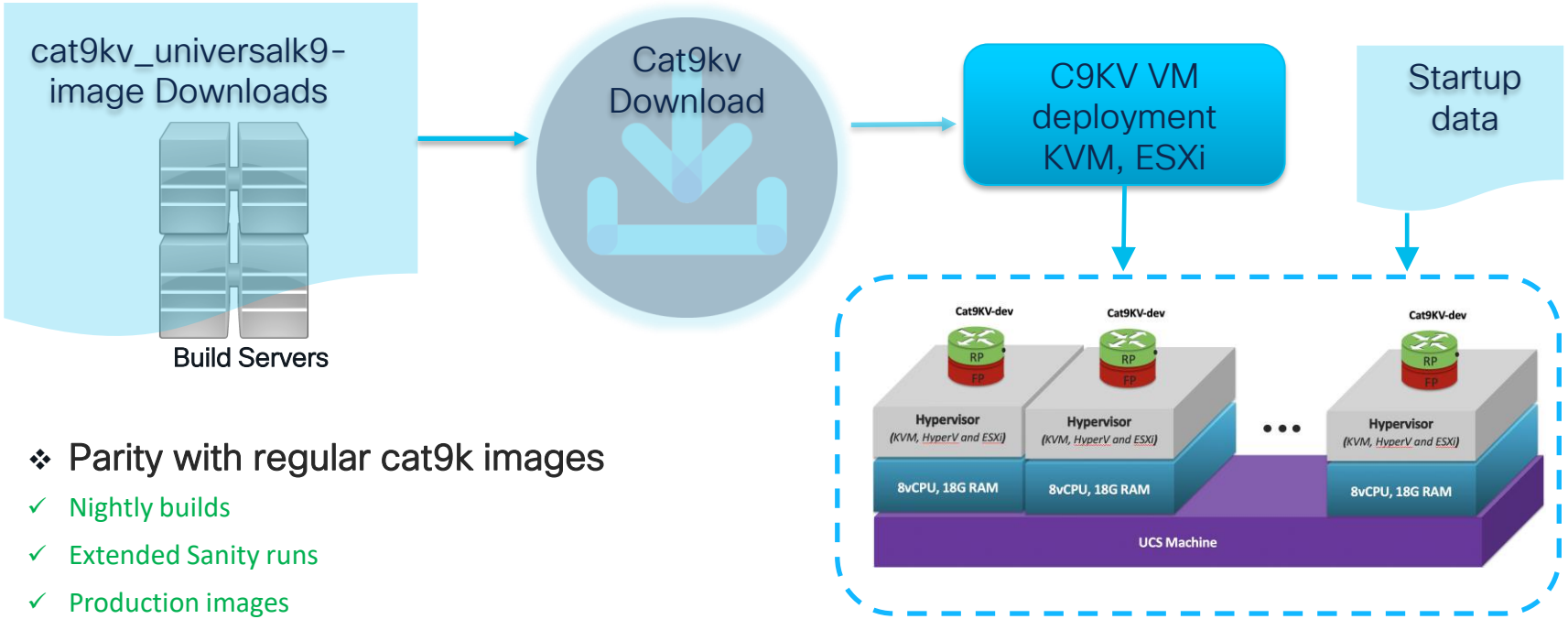
Catalyst 9000 Series Switches



An Independent PID & SKU!

Catalyst 9000 Virtual Switch

Unified Single Image – Virtual twin of catalyst 9000



- Parity with regular cat9k images
  - ✓ Nightly builds
  - ✓ Extended Sanity runs
  - ✓ Production images

How C9KV is different than other container NOS?

- The C9KV does ASIC simulation…
  - ✓ Use of ASIC pipelines in software
  - ✓ High fidelity reproduction of forwarding feature behavior
  - ✓ Single code base between physical and virtual
- Simulation of what happens with actual hardware ASIC
- Same forwarding behavior as what you would expect with physical hardware
- Same functional feature testing between C9K and C9KV

Cisco **Unified Access Data-Plane** (UADP®)

Cisco **Silicon One**™

Flexible Pipelines
Investment Protection

Adaptable Tables
Universal Deployment

Scalable Resources
Enhanced Scale and Buffering

Goals & Vision

Increase Productivity

- Faster feature development unhindered by board platform dev
- Simplifies spinning up demo environments
- Flexible data-path choice on same unified image

Enable Hybrid workplace

- Construct standard network topologies using point-and-click
- Eliminate hardware wiring requirements (no lab visit needed to change cabling)

Facilitate Technology Adoption

- Enable GTM Modelling
- Ready access to available switching services features
- Proof of concepts and value
- what-if scenario analysis

Access with Ease

- Deployment to private & public cloud
- Readily available VMs & Virtual Networks save time

High Level Specification

Packaging Format	• Single IOS-XE Image • Independent PID compared to hardware SKUs • Available as ISO & OVA
Resource Requirement	• 4vCPU, 18G RAM (UADP) • 4vCPU, 12GB RAM (Silicon 1) • Hypervisors: KVM, ESXi • Internet reachability for user telemetry
Deployment capabilities	• 8/24 interfaces+ one mgmt. port • Day 0 config support • Customization parameters (serial#)
Data Plane	• Software data plane is one of: 1. UADP 2. Silicon 1
Features	• Services feature parity for data plane constructs • Low packet throughput but sufficient for functional feature behavior • UADP – 200-300 pps • Silicon 1 – 500-1300 pps

Comparing UADP and Silicon 1 C9KV Variants

UADP C9KV

- ✓ 18GB memory required
- ✓ More feature set
- ✓ Provides support for Software Defined Access fabric, EVPN
- ✓ Can be managed in Catalyst Center

Silicon 1 C9KV

- ✓ Memory efficient with 12GB required
- ✓ Can be managed in Catalyst Center
- ✓ Higher throughput for testing

BETA CAT9000v
UADP

Both node types are available in CML 2.7

BETA CAT9000v
Q200

Speeds, Feeds, and Licensing

CML version	C9Kv version	PPS Thruput UADP/S1	Thruput
CML 2.5	17.10.01prd7	200/300	250 Kbps
CML 2.6	Same	Same	Same
CML 2.7	17.12	~1300	2 Mbps out 1.4 Mbps in

What license does C9KV come with?
- ✓ Essentials

What about Advantage features ?
- ✓ Update with: "license boot level network-advantage addon dna-advantage"
- ✓ Reboot

C9KV 17.12 in CML 2.7

Hardware Resources	Silicon 1 C9Kv variant now requires 12GB RAM to run
Throughput	Improved rate of 2Mbps inbound and / ~1.4Mbps outbound
Interfaces	Increased number of interfaces supported up to 24 ports in only KVM/CML
Day 0 config file	Path to day 0 config file is in root of CDROM file system. The filename must be iosxe_config.txt
MAC Learning/ Aging	MAC Learning/Aging is now supported in 17.12

Interface Naming Feature Support C9KV 17.12

Interface prefix (Gi, Te, etc) is set in the vswitch.xml prior to first boot by updating the “interface_name_prefix” tag

```
jcohoe-c9kv-1#sh int status

Port        Name               Status       Vlan       Duplex  Speed Type
Gi1/0/1                        connected    1          a-full a-1000 unknown
Gi1/0/2                        connected    1          a-full a-1000 unknown
Gi1/0/3                        connected    1          a-full a-1000 unknown
Gi1/0/4                        connected    1          a-full a-1000 unknown
Gi1/0/5                        connected    1          a-full a-1000 unknown
Gi1/0/6                        connected    1
Gi1/0/7                        connected    1
Gi1/0/8                        connected    1          a-full a-1000 unknown
jcohoe-c9kv-1#
```

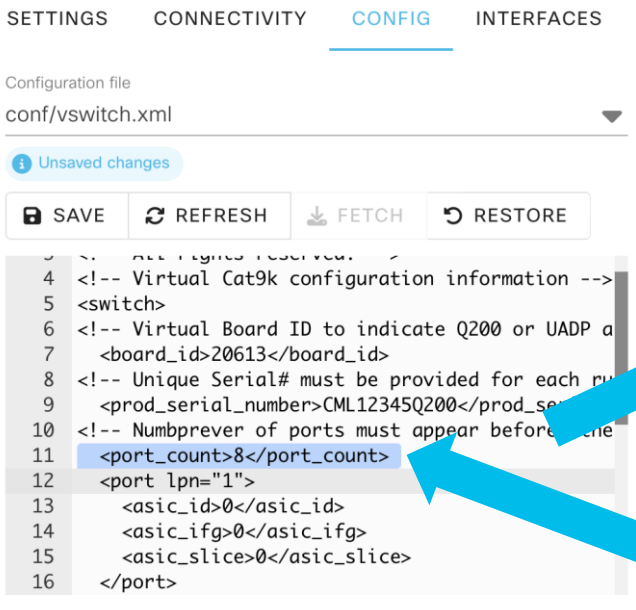
```
jcohoe-c9kv-1#sh int status

Port        Name               Status       Vlan       Duplex  Speed Type
  TenGigEthernet1/0/1          connected    1          a-full a-1000 unknown
  TenGigEthernet1/0/2          connected    1          a-full a-1000 unknown
  ……                           connected    1          a-full a-1000 unknown
  TenGigEthernet1/0/6          connected    1          a-full a-1000 unknown
  TenGigEthernet1/0/7          connected    1          a-full a-1000 unknown
  TenGigEthernet1/0/8          connected    1          a-full a-1000 unknown
                               connected    1          a-full a-1000 unknown
jcohoe-c9kv-1#
```

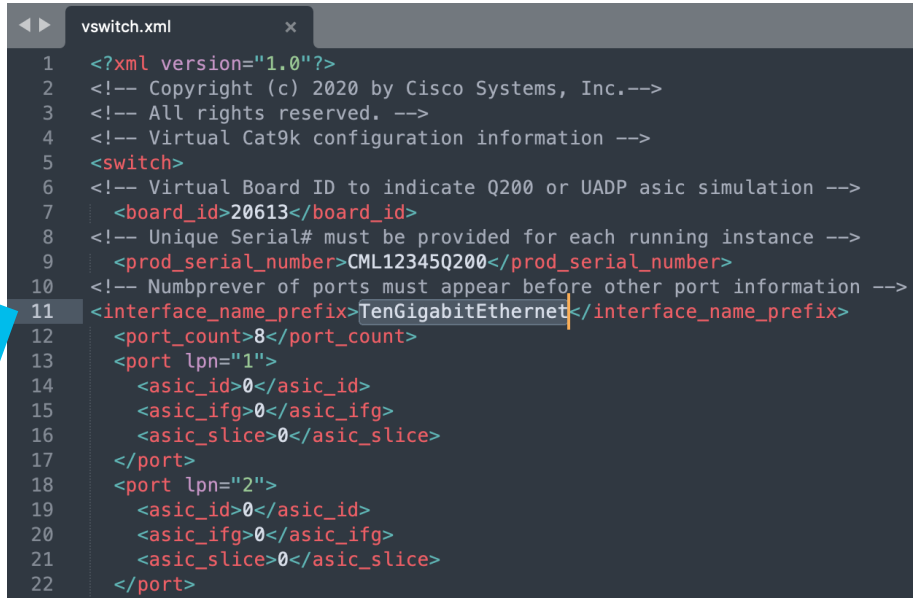
- Renaming of interfaces from Gi1/0/1 to TenGig1/0/1 or even Ethernet1
- This enables closer to production network topologies

There is no support for the concept of “Network Modules” or uplinks ports
There is no support for the concept of “Line cards” or “SFP/optics”

Increasing the interface count from 8 to 24

Switchport count can be increased from 8 up to 24 by modifying the vswitch.xml prior to first boot by updating the “port_count” tag





Unsupported Functionality

Does X/Y/Z feature work? Yes.
Unless it's listed below...

Unsupported Functionality	Comments
POE	Not supported
Hardware component simulation	No PSU, Fans, Temperature Sensor, SFP, MAC/SERDES, linecards, FRUs etc. No MIB for board or components
Interrupt, DMA based features	Interrupt based features needs to be done via polling such as punt interrupt, MACSEC rekey etc.
Hardware Diagnostics	Typically achieved using loopback in hardware at MAC/SERDES.
App hosting	No Guest shell (ZTP), IOX, or Application Hosting infra
TM, QoS	Traffic Management, Queuing actions
FNF, Netflow	Currently unsupported
Stacking/SVL	Currently unsupported
Network Modules, Line cards, SFP, optics	Currently unsupported
Embedded WLC	Currently unsupported
Bundle install mode	Only install mode supported

Controller Management

Catalyst Center

- ✓ Unique serial number required
  - ✓ (if adding more than 1)
- ✓ Image management, SWIM, etc
- ✓ Device 360, Health score checks, etc

Meraki

- ❑ Not supported due to hardware SUDI certificate requirement

Maximum Transmission Unit (MTU) = 1500

- ESXi support MTU up to 9000 however 22 units are reserved for ESXi
  - C9KV on ESXi can support an MTU up to 8978 (9000-22)
  - Larger MTU required “for fabric testing”

- CML (KVM) supports up to 1500 MTU only.
  - The e1000 NIC adapter will see resets if MTU set larger

```
Switch(config)#system mtu ?
  <1500-8978>  MTU size in bytes

Switch(config)#system mtu 1900
Global Ethernet MTU is set to 1900 bytes.
Note: this is the Ethernet payload size, not the total
Ethernet frame size, which includes the Ethernet
header/trailer and possibly other tags, such as ISL or
802.1q tags.

Switch(config)#system mtu 1900
*Jan 29 21:43:17.882: %IOSXE-3-PLATFORM: Switch 1 R0/0: kernel: e1000 0000:00:04.0 eth2: Reset adapter
```

What are some features are being tested ?

Interfaces
Loopbacks
SVI
Routed P2P (/127)
Layer 2 Access Port
Layer 2 Trunk Ports
Layer 3 Sub-interfaces DHCP relay

Security
VTY ACL
SNMP ACL
NAC / RADIUS
CoPP
DHCP Trusted Port
IPv6 RA
DHCPv6 Guard
PVLAN

Monitoring
Syslog
SNMP
gNMI
NETCONF
RESTCONF
gRPC Dial-Out

Management
Hostname
Boot / Image
DNS
LLDP
SSH
Alias
Banner
Local User
TACACS
AAA
NTP

Overlay Control and Data Plane
BGP EVPN VxLAN BGP EVPN
MAC-VRF Type 2 MAC
Type 2 MAC IP
Type 3 IMET
Type 5 IP-Prefix
Route target import / export
Import / export route-maps
VxLANv6
Loopback - VTEP Source
L2VNI
L3VNI
VXLAN SGT Inline Tagging to CMD
SGT Inline Tagging

Underlay Routing
eBGP
Redistribute
Peer-Groups
ECMP
Prefix List
Route Maps
Community List
AS PATH Access list

Solutions:
SDA with Cisco DNA Center
BGP EVPN VXLAN with YANG

This is not an extensive list of features !
Solution level testing also in use ☺

Demo
C9KV in CML
Feature:
Interface Template CLI

Interface Template Feature

- ✓An interface template is a container of configurations or policies that can be applied to specific ports.
- ✓Built-in templates are created by the system.
- ✓You can also create specific user templates with the commands that you want to include.
- ✓Currently, not all configuration commands are supported with user interface templates type
- ✓17.12 had added feature support of including DHCP guard policy to the interface template
  - ✓Focus of upcoming demo…

Demo: Interface Template Support for IPv6 DHCP Guard
IOS-XE 17.12 Feature Testing on C9Kv

1. Creating DHCP Guard Policy

```
bcp-1(config-dhcp-guard)#ipv6 dhcp guard policy DHCP_GUARD_POLICY
bcp-1(config-dhcp-guard)#device-role server
bcp-1(config-dhcp-guard)#pref
bcp-1(config-dhcp-guard)#preference max 250
bcp-1(config-dhcp-guard)#end
bcp-1#
```

2. Creating DHCP Guard Policy

```
bcp-1(config)#
bcp-1(config)#template TEST
bcp-1(config-template)#ipv6 dhcp guard att
bcp-1(config-template)#ipv6 dhcp guard attach-policy DHCP_GUARD_POLICY
bcp-1(config-template)#end
bcp-1#
```

2. Creating DHCP Guard Policy

```
bcp-1(config)#int
bcp-1(config)#interface gi
bcp-1(config)#interface gigabitEthernet 1/0/1
bcp-1(config-if)#source tem
bcp-1(config-if)#source template TEST
bcp-1(config-if)#
```

Interface Template Demo Detailed Workflow

1. Login to CML
2. Access C9KV console
3. In interface template config mode, showcase list of supported commands
3. Create a DHCP guard policy
4. Create a user interface template and add DHCP guard policy to template
5. On an interface, attach the source template containing the DHCP guard policy

Demo Recording

Cisco Modeling Labs | Dashboard

DASHBOARD | TOOLS | ADMIN

Show List | Show All

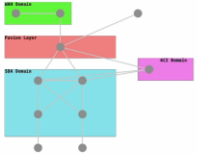
ADD | IMPORT

LAX endpoints ON

multi_sda ON

sda_l2

sda_fw



single

Lab at Mon 18:49 PM

Lab at Mon 18:50 PM ON

Lab at Thu 00:00 AM

CPU 19.59% | MEMORY 21.59% | DISK 5.39% | OK

ContainerLab

Deployment

Packaging Format

- Single IOS-XE Image
- Independent PID compared to hardware SKUs
- Available as ISO & OVA

Resource Requirement

- 4vCPU, 18G RAM
- Hypervisors: KVM, ESXi
- Internet reachability for user telemetry

Many customers use ContainerLab for Network Operating System lab virtualization

In addition to KVM/ESXi – customers can run the C9KV in a variety of hypervisors and software including:
Docker, ContainerLab, CML, KNE (K8S Networking), … etc

https://containerlab.dev/ provides a CLI for orchestrating and managing container-based networking labs. It starts the containers, builds a virtual wiring between them to create lab topologies of users choice and manages labs lifecycle.

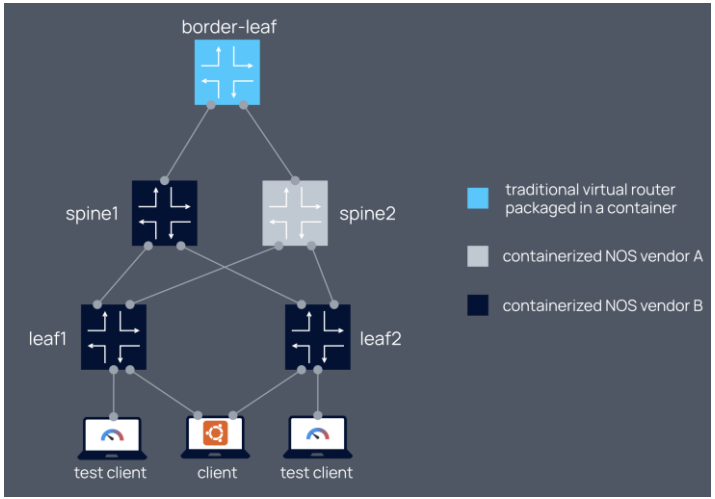
```
tme@tme-yangsuite:~$ docker ps
CONTAINER ID   IMAGE                          COMMAND                  CREATED        STATUS                  PORTS                                                                                          NAMES
ad465bf74a54   vrnetlab/vr-c9kv:17.11.01      "/launch.py --userna…"   3 hours ago    Up 3 hours (healthy)    22/tcp, 830/tcp, 5000/tcp, 10000-10099/tcp, 161/udp                                            clab-cat9kvlab-cat9kv-7
808bdbae8322   vrnetlab/vr-c9kv:17.11.01      "/launch.py --userna…"   3 hours ago    Up 3 hours (unhealthy)  22/tcp, 830/tcp, 5000/tcp, 10000-10099/tcp, 161/udp                                            clab-cat9kvlab-cat9kv-3
57b6f363257e   vrnetlab/vr-c9kv:17.11.01      "/launch.py --userna…"   3 hours ago    Up 3 hours (healthy)    22/tcp, 830/tcp, 5000/tcp, 10000-10099/tcp, 161/udp                                            clab-cat9kvlab-cat9kv-4
a83d675fe08b   vrnetlab/vr-c9kv:17.11.01      "/launch.py --userna…"   3 hours ago    Up 3 hours (healthy)    22/tcp, 830/tcp, 5000/tcp, 10000-10099/tcp, 161/udp                                            clab-cat9kvlab-cat9kv-6
0c5812d83cbb   vrnetlab/vr-c9kv:17.11.01      "/launch.py --userna…"   3 hours ago    Up 3 hours (healthy)    22/tcp, 830/tcp, 5000/tcp, 10000-10099/tcp, 161/udp                                            clab-cat9kvlab-cat9kv-8
9562399d9056   vrnetlab/vr-c9kv:17.11.01      "/launch.py --userna…"   3 hours ago    Up 3 hours (healthy)    22/tcp, 830/tcp, 5000/tcp, 10000-10099/tcp, 161/udp                                            clab-cat9kvlab-cat9kv-5
7a59d4ce051b   vrnetlab/vr-c9kv:17.11.01      "/launch.py --userna…"   3 hours ago    Up 3 hours (healthy)    22/tcp, 830/tcp, 5000/tcp, 10000-10099/tcp, 161/udp                                            clab-cat9kvlab-cat9kv-1
f857a73db00d   vrnetlab/vr-c9kv:17.11.01      "/launch.py --userna…"   3 hours ago    Up 3 hours (healthy)    22/tcp, 830/tcp, 5000/tcp, 10000-10099/tcp, 161/udp                                            clab-cat9kvlab-cat9kv-2
f8e327fdf268   yangsuite-internal:latest      "/start"                 4 months ago   Up 5 days               0.0.0.0:58500-58501->58500-58501/tcp, :::58500-58501->58500-58501/tcp, 0.0.0.0:8481->8480/tcp, :::8481->8480/tcp   yangsuite
fddbf9fb2514   portainer/portainer-ce:2.9.3   "/portainer"             8 months ago   Up 5 days               0.0.0.0:8000->8000/tcp, :::8000->8000/tcp, 0.0.0.0:9443->9443/tcp, :::9443->9443/tcp, 9000/tcp   portainer
tme@tme-yangsuite:~$
```

Container Lab: How EFT customers use C9KV

https://ContainerLab.dev provides a CLI for orchestrating and managing container-based networking labs. It starts the containers, builds a virtual wiring between them to create lab topologies of users choice and manages labs lifecycle.

- Nokia SR Linux
- Arista cEOS
- Cisco XRd
- Azure SONiC
- Juniper cRPD
- Cumulus VX
- Keysight IXIA-C
- RARE/freeRtr

- Nokia virtual SR OS (vSim/VSR)
- Juniper vMX
- Juniper vQFX
- Juniper vSRX
- Juniper vJunos Switch
- Juniper vJunos Evolved
- Cisco IOS XRv9k
- Cisco Nexus 9000v
- Cisco CSR 1000v
- Cisco FTDv
- Dell FTOS10v
- Arista vEOS
- Palo Alto PAN
- IPInfusion OcNOS
- Check Point Cloudguard
- Aruba AOS-CX
- OpenBSD



ContainerLab is an abstraction to Docker – it helps manage the container state and networking

ContainerLab is just Dockers (docker containers)

```
dcloud@dcloud:~$ docker ps
CONTAINER ID   IMAGE                              COMMAND                  CREATED      STATUS                 PORTS                                                                NAMES
c26e82b961b6   hailumeng/ubuntu-desktop:focal     "/startup.sh"            7 days ago   Up 7 days (healthy)    22/tcp, 80/tcp, 443/tcp                                              clab-fabric-client1
628bb320d050   hailumeng/ubuntu-desktop:focal     "/startup.sh"            7 days ago   Up 7 days (healthy)    22/tcp, 80/tcp, 443/tcp                                              clab-fabric-client2
bd007468e338   vrnetlab/vr-cat9kv:17.10.01        "/launch.py --userna…"   7 days ago   Up 7 days (healthy)    22/tcp, 830/tcp, 5000/tcp, 10000-10099/tcp, 161/udp                  clab-fabric-fe2
f21443cf9954   vrnetlab/vr-cat9kv:17.10.01        "/launch.py --userna…"   7 days ago   Up 7 days (healthy)    22/tcp, 830/tcp, 5000/tcp, 10000-10099/tcp, 161/udp                  clab-fabric-fe1
75c0443e0fe4   grafana/grafana:latest             "/run.sh"                7 days ago   Up 7 days              0.0.0.0:3000->3000/tcp, :::3000->3000/tcp                            clab-testdrive-grafana
44f39faf798e   telegraf:1.21.4-alpine             "/entrypoint.sh tele…"   7 days ago   Up 7 days              8092/udp, 8125/udp, 8094/tcp                                         clab-testdrive-telegraf
5182db2e0ebf   hailumeng/ubuntu-desktop:focal     "/startup.sh"            7 days ago   Up 7 days (healthy)    22/tcp, 80/tcp, 443/tcp                                              clab-testdrive-client1
bb79720c65d2   hailumeng/ubuntu-desktop:focal     "/startup.sh"            7 days ago   Up 7 days (healthy)    22/tcp, 80/tcp, 443/tcp                                              clab-testdrive-client2
8d7781f553f3   vrnetlab/vr-cat9kv:17.10.01        "/launch.py --userna…"   7 days ago   Up 7 days (healthy)    22/tcp, 830/tcp, 5000/tcp, 10000-10099/tcp, 161/udp                  clab-testdrive-access
20af66fb7f36   vrnetlab/vr-cat9kv:17.10.01        "/launch.py --userna…"   7 days ago   Up 7 days (healthy)    22/tcp, 830/tcp, 5000/tcp, 10000-10099/tcp, 161/udp                  clab-testdrive-core
548947fc58e1   hailumeng/vr-c8000v:17.08.01       "/launch.py --userna…"   7 days ago   Up 7 days (healthy)    22/tcp, 830/tcp, 5000/tcp, 9339/tcp, 10000-10099/tcp, 50052/tcp, 161/udp   clab-testdrive-wan
91f7b1fdddbc   hailumeng/yangsuite-solo:latest    "/yangsuite/migrate_…"   7 days ago   Up 7 days                                                                                   clab-testdrive-yangsuite
8323d7d90a85   hailumeng/nginx:1.23               "/docker-entrypoint.…"   7 days ago   Up 7 days              22/tcp, 80/tcp, 443/tcp                                              clab-testdrive-server1
84ecddacde35   influxdb:latest                    "/entrypoint.sh infl…"   7 days ago   Up 7 days              0.0.0.0:8086->8086/tcp, :::8086->8086/tcp                            clab-testdrive-influxdb
dcloud@dcloud:~$
```

Docker containers that have some additional wiring / connectivity:

```
dcloud@dcloud:~$ docker exec -it clab-fabric-fe2 telnet localhost 5000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

fe2>en
fe2#sh ver | i 17
Cisco IOS XE Software, Version 17.10.01prd7
Cisco IOS Software [Dublin], Catalyst L3 Switch Software (CAT9K_IOSXE), Version 17.10.1prd7, RELEASE SOFTWARE (fc1)
*    1 8      C9KV-UADP-8P       17.10.01prd7       CAT9K_IOSXE           INSTALL
fe2#sh int statu
fe2#sh int status

Port         Name               Status       Vlan       Duplex  Speed Type
Gi1/0/1                         connected    1          a-full  a-1000 unknown
```

In this example port 5000 inside the container is mapped to the C9KV serial console

C9KV in CML

What do I need to know ?

Differences between Gig0/0 and Gig1/0/1?

Gig0/0

- Higher packet rate
- Uses CPU data plane of ESXI
- A lot more PPS because of the CPU datapath
- Management plane and capabilities only though

Gig1/0/1

- Lower packet rate compared to Gig0/0
- Uses the CPU + ASIC data-path simulation
- 200/500/1300 PPS throughput
- Support for front panel configurations
- Data plane connectivity for clients and endpoints

Previous recommendation to connect Catalyst Center via Gi0/0 is no longer true

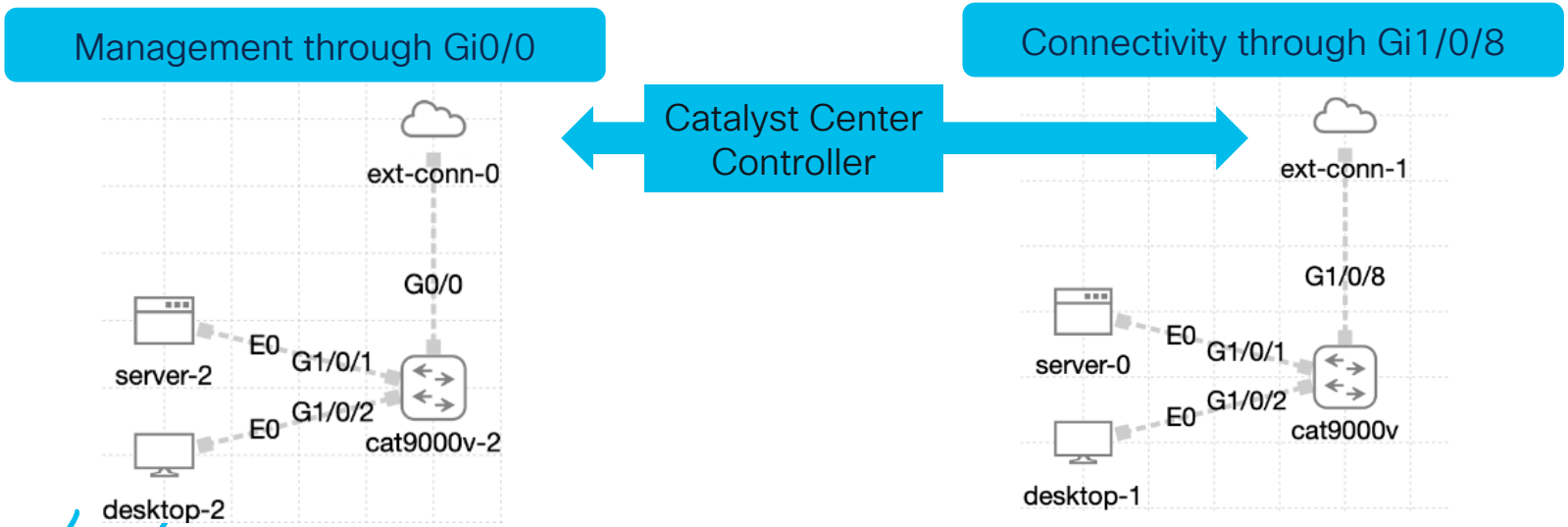
The controller can now manage C9KV either Management or Front panel due to the increased throughput

Front panel connectivity

How can we have connectivity from C9KV clients to the rest of the physical network?

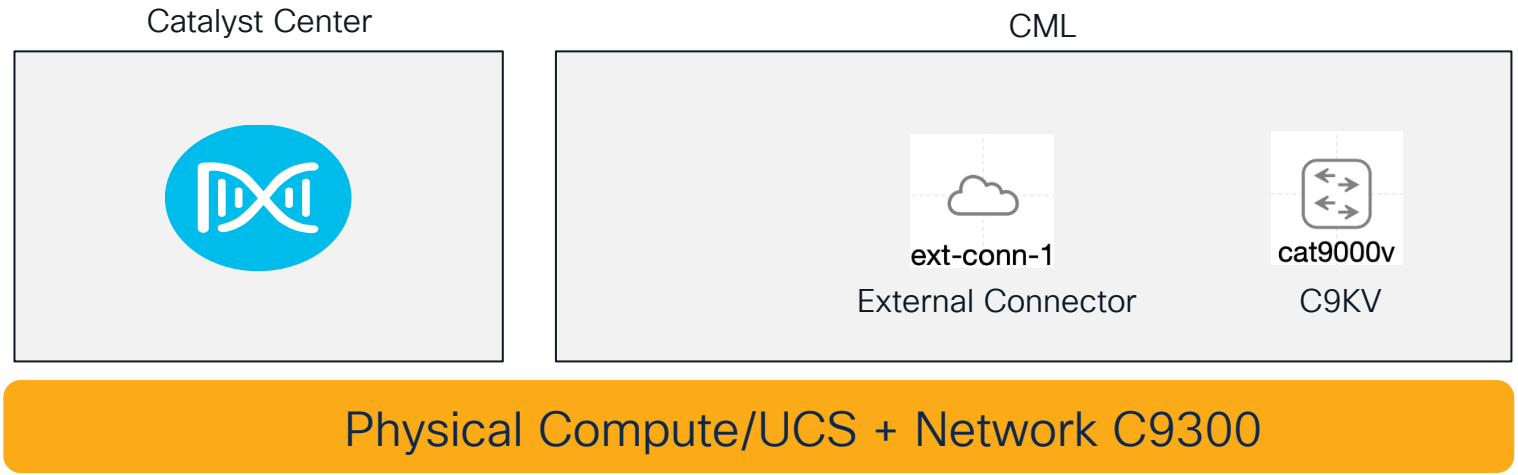
Previous recommendation was to connect C9KV to DNAC through Gi0/0 however with the increase of PPS this is no longer the case

We now recommend to connect C9KV to DNAC through any one of the front panel ports

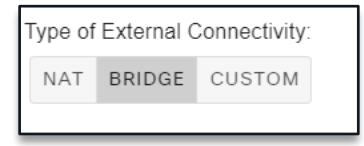


Bridging the Physical and Virtual Gap

The C9Kv switch can be connected to the Catalyst Center by using the External Connector node within CML



- Configure the external connector to type: **BRIDGE**
- Once external connector is configured for bridge then packets to/from C9KV will be able to reach the Catalyst Center controller and other devices on the LAN network



Catalyst Center

C9KV presents the same as C9K in Catalyst Center

Detail Information

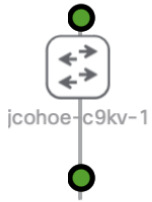
Device Info | Interfaces

All Interfaces (12) 3 out of 12 interface(s) are down

PORT TYPE: All | Access | Auto | Routed | Trunk

Search Table

Interface Name	Interface Description	Operational Status	Admin Status	Port Channel ID	Type	VLAN
GigabitEthernet0/0	--	●	●	--	Routed	--
GigabitEthernet1/0/1	--	●	●	--	Access	311
GigabitEthernet1/0/2	--	●	●	--		



-- /10 DEVICE DETAILS

Model: C9KV-UADP-8P Management IP: 10.85.134.67 Location: Global / Vancouver Canada / VNC2

Uptime: -- Reachability Status: Unreachable View All Details

Issues (1) Feb 5, 2024 6:13 PM

P1 Availability

Network Device jcohoe-c9kv-1 is unreachable from Cisco DNA Center

Instance Count: 8

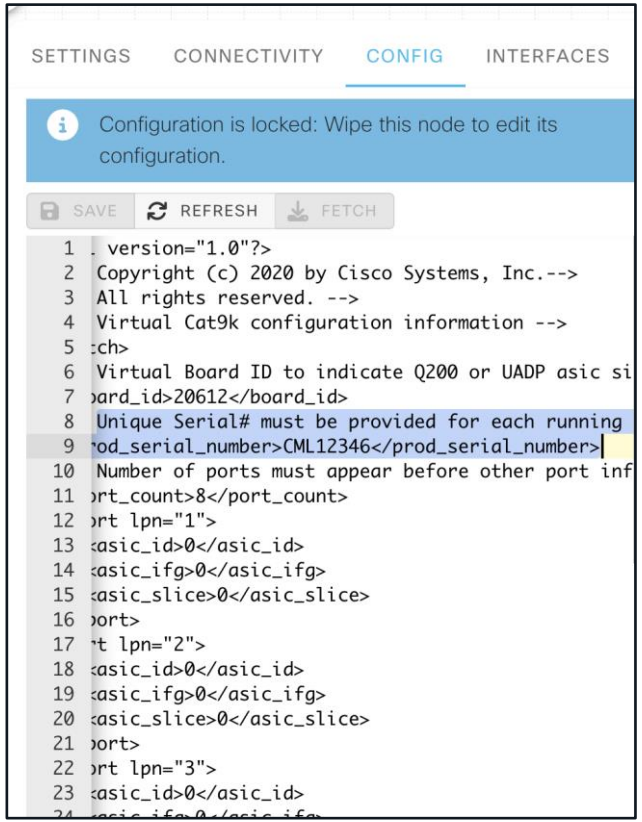
1 Records

Custom Serial Number

In CML 2.6 with C9KV 17.10 the device serial number could not be easily changed So having multiple C9KV in Catalyst Center controller was not supported

Cisco Catalyst Center requires unique device serial number to be added to inventory and managed for Assurance insights

- Change each instance's serial number by editing the XML under the Edit Config tab after adding an instance to a lab.
- This must be done prior to booting the node for the first time.
- Change the <prod_serial_number> property to be the desired unique serial number

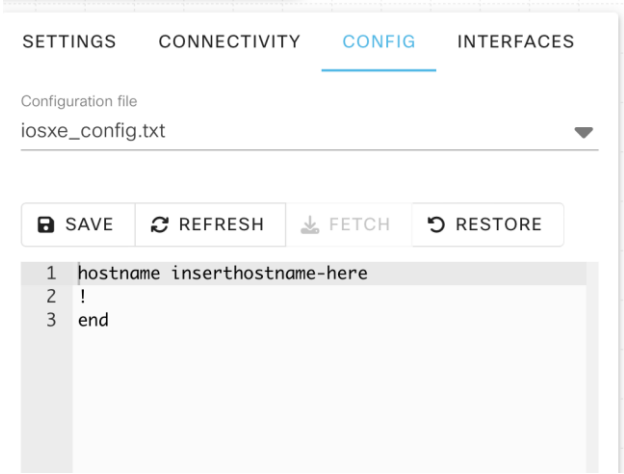


Bootstrapping the Day 0 config

Day 0 configuration can be added before first boot

The “config” section shows the day0 iosxe_config.txt
Default is to set hostname but any CLI’s can be added

- Select device
- Select “config” tab
- Select “iosxe_config.txt” configuration file
- Set CLI’s
- Select “Save”
- Power on C9KV



Day 0 Guest Shell / ZTP with embedded CentOS is currently not supported on C9KV

Programmability & Automation

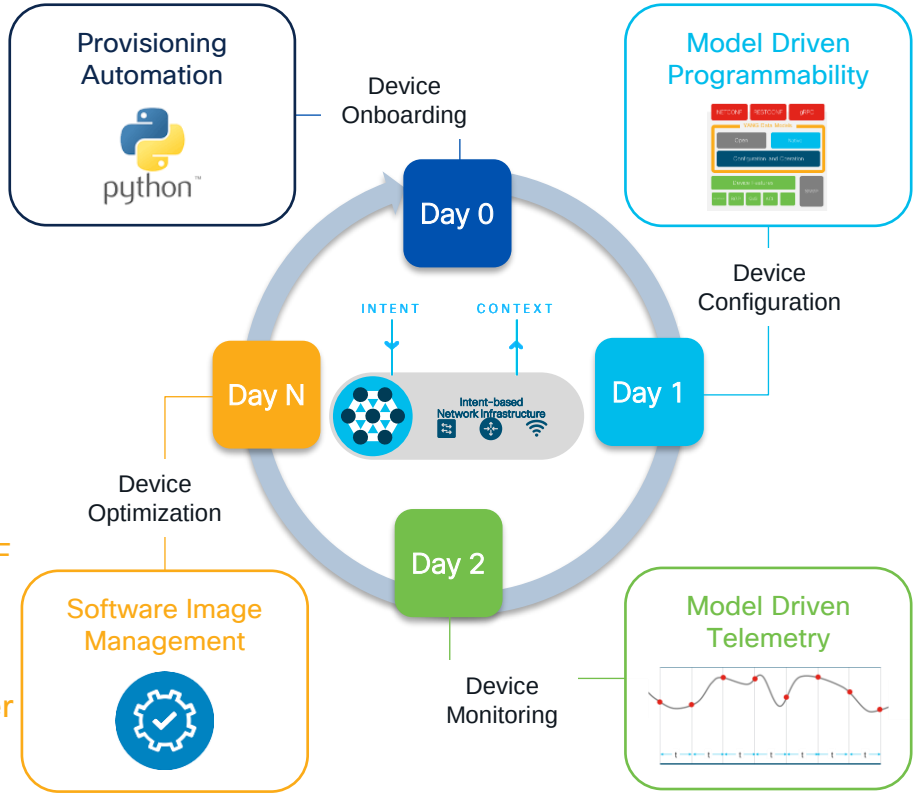
Some examples of using C9KV

IOS XE Programmability & Automation Lifecycle

Pre-boot Execution Environment (iPXE)

RD8572 Secure Zero Touch Provisioning ZTP

VM Automation



Network Configuration Protocol (NETCONF), RESTCONF, gNMI

YANG “native” Data Models, OpenConfig,

YANG Suite, Terraform, Ansible, pyATS tooling

gNOI cert/os/reset proto

Guest Shell + Python/NETCONF

CentOS 8 Python 3

Application Hosting with Docker

“show run” CLI to YANG

TIG_MDT container + guide

YANG On-Change support

gRPC Dial-Out + DNS + mTLS

gNMI/NETCONF Dial-In

cisco Live!

Programmable Interfaces

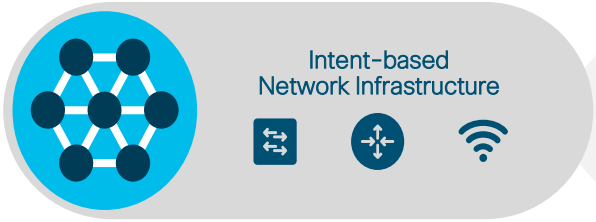
CLI

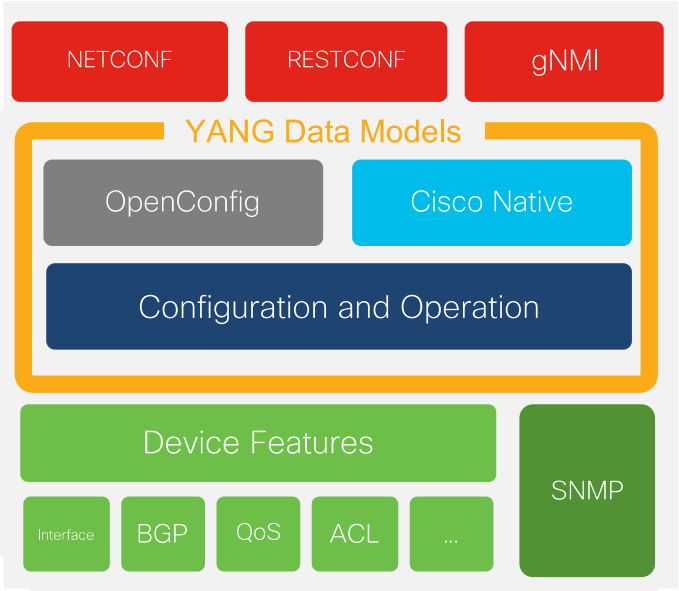
SNMP

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

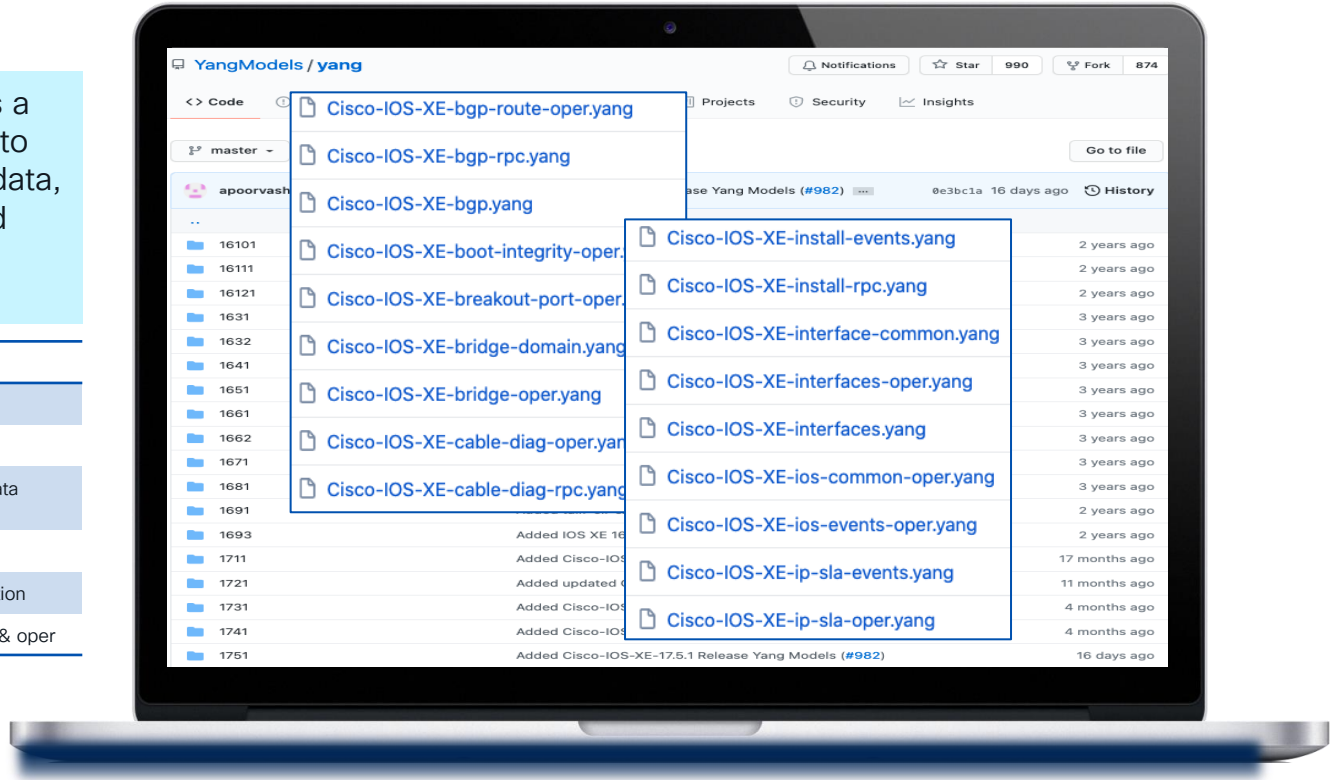
IEEE

CISCO

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



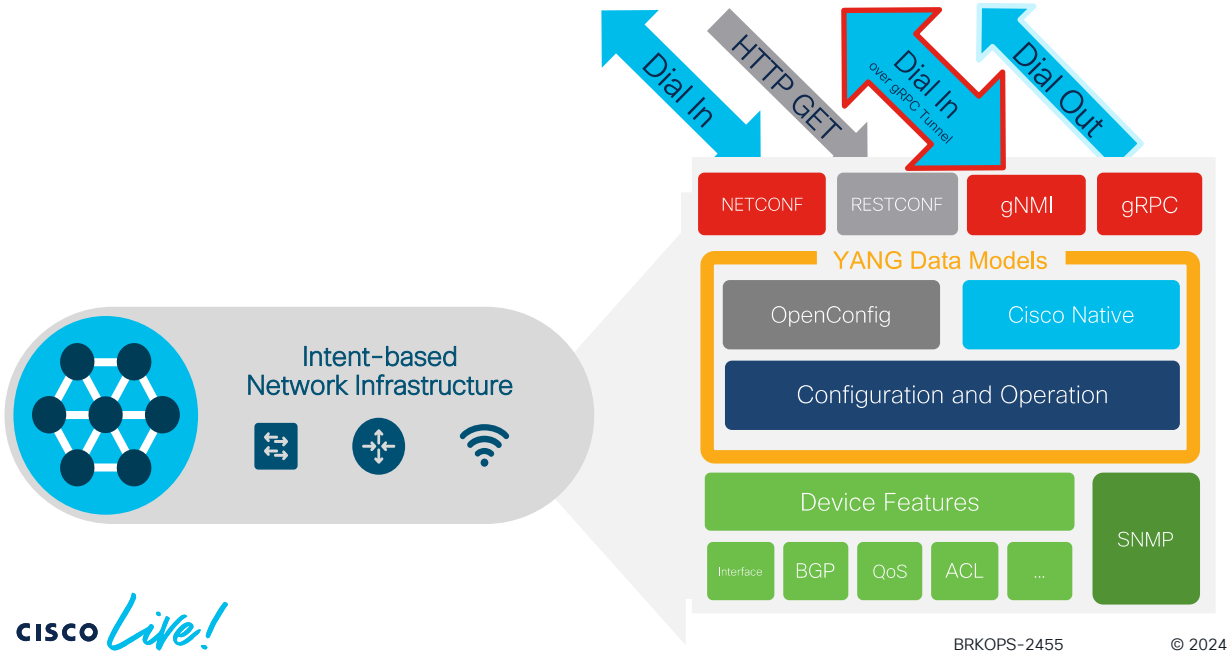
https://github.com/YangModels/yang/tree/master/vendor/cisco/xe

Model Driven Telemetry Interfaces

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

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

Publication / Subscription



XML, JSON, proto and kvGPB encoding

Consistent YANG data models between interfaces

On-change event and time-based publication options

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

dCloud for 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

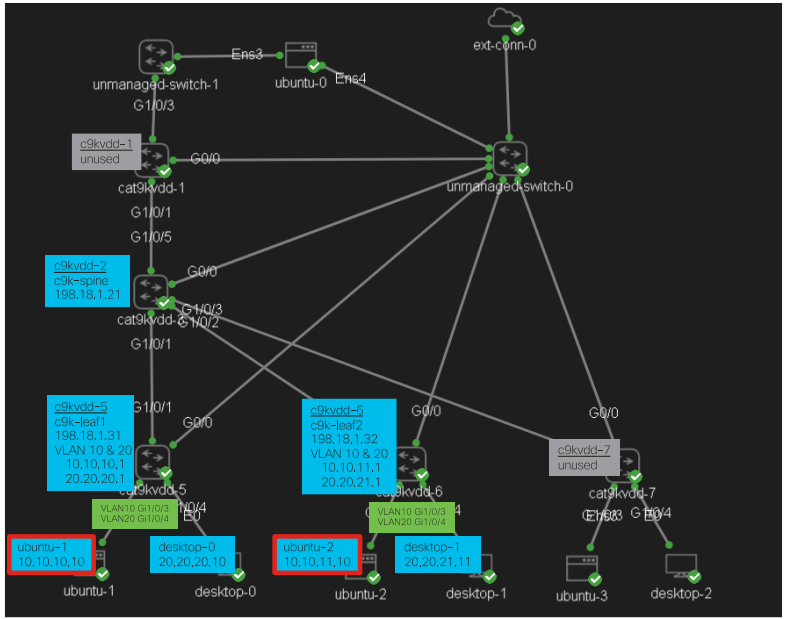
Terraform/RESTCONF
- Declarative EVPN fabric management

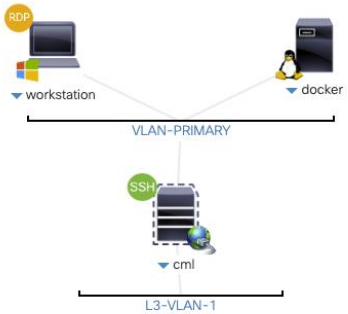
Ansible
- EVPN solution enablement using CLI

Ubuntu VM Details:
Syslog receiver from all switches
TFTP config backup

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

3 configured C9KV 17.10
2 un-configured C9KV 17.10





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

Usecase:
Auto-vSDA
... in dCloud

+ BGP EVPN VXLAN

What is Auto-vSDA?

Automated Virtual Software-Defined Access

- Ansible-based framework for automating vSDA testbeds deployments
- Consisting of a set of reusable playbooks for the building of different SDA scenarios
- It is expected to have a repository of testbeds created and validated by engineers to be reused by the community
- The primary objective is to save lot of time and effort in the testbed building whenever an SDA related test/validation is needed

Cisco Catalyst Center

Cisco DNA Center, with its platform and software, provides the network automation and secure access capabilities at the core of SD-Access.

Cisco Identity Services Engine (ISE)

Policy, segmentation, and enforcement are central to an effective zero-trust workplace solution. Cisco ISE is the engine for defining and enforcing segmentation SD-Access.

Cisco DNA Software subscriptions

Cisco DNA Software subscriptions give you the tools you need in a single, easy-to-consume license. Your SD-Access solution continues to evolve as your needs grow.

Cisco Catalyst 9000 infrastructure

The power of SD-Access is its integration with the network. Cisco Catalyst switches and access points help optimize the depth and breadth of access security.

https://www.cisco.com/c/en/us/solutions/enterprise-networks/software-defined-access/index.html

Auto-vSDA: Components

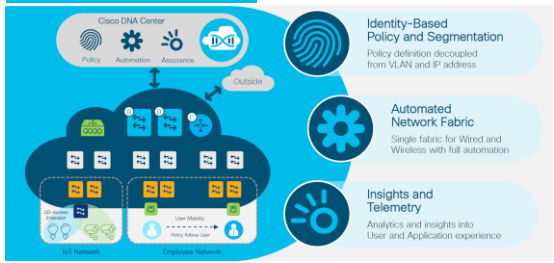
Playbooks

- Restore VM Snapshots specific to testbed
- Build CML2 testbed
- Build Fabric

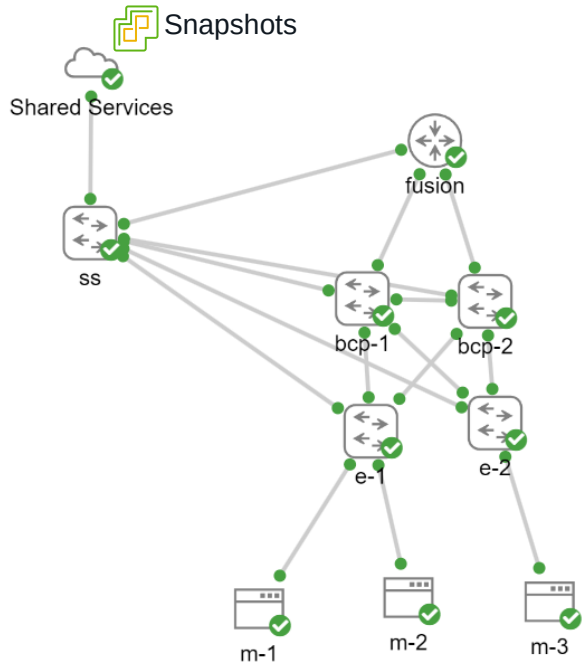
ANSIBLE

VM Snapshots
- vCatalystCenter
- vISE
- Jump Server
- NAT Router

CML2 Lab
- cat9kv
- IOSv
- Ubuntu



Auto-vSDA: “Dual-Border-Edge” example testbed

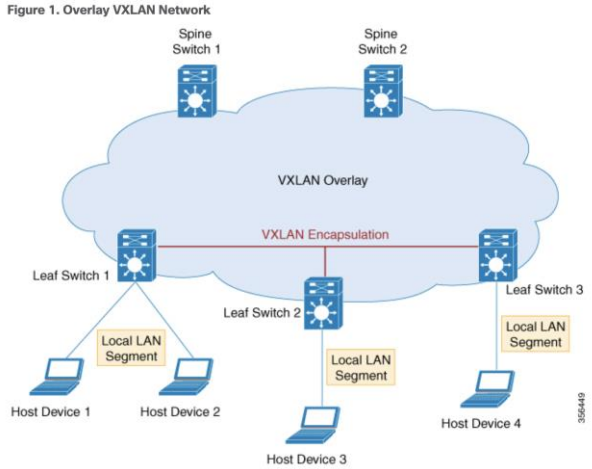


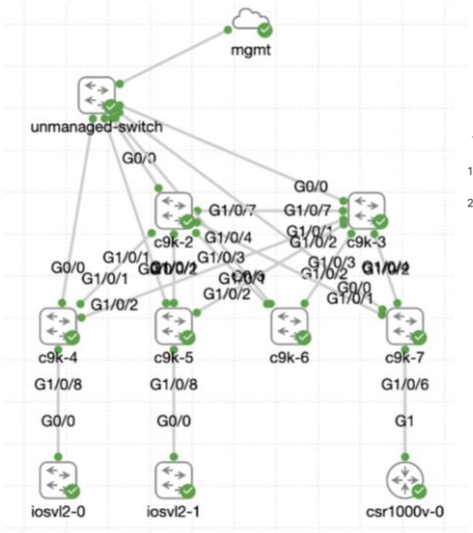
1. Restore VM snapshots
   - vCatalystCenter and vISE snapshots include SDA fabric already configured
   - Router NAT used for OOB management
   - Jump Server has node configs
2. Build CML2
   - CML2 topology is created
   - Nodes are started
3. Build Fabric
   - Cat9kV is initialized
   - Configurations are copied to nodes from Jump server
4. SDA is up

Search dcloud.cisco.com catalog for updated SDA and BGP EVPN VXLAN topologies

dCloud for BGP EVPN VXLAN

“Catalyst 9000 Campus EVPN Fabric Part 1 v1”
@ dcloud.cisco.com
https://github.com/Cat9kEVPN/cat9k-evpn-ansible
https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9300/software/release/17-12/configuration_guide/vxlan/b_1712_bgp_evpn_vxlan_9300_cg/bgp_evpn_vxlan_overview.html





Overview

BGP EVPN VXLAN is a campus network solution for the Cisco Catalyst 9000 Series Switches running Cisco IOS XE software. It provides a unified overlay network solution which enables micro and macro segmentation, mobility and on demand network expansion, and multi tenancy in campus networks. By adopting EVPN VXLAN overlay networking in campus environments with Catalyst 9000 series, customers can simplify network operations, address application consolidation, and IOT needs in a secure and agile manner.

Requirements

- Required: Laptop; router that is registered and configured for dCloud
- Optonal: Cisco AnyConnect

Scenarios

- Distributed Anycast Gateway with Ingress Replication
- Distributed Anycast Gateway with Underlay Multicast

Run Ingress Playbooks on Ansible Host

This section describes how to run playbooks on the Ansible host on the remote desktop.

1. Go back to the Ansible terminal host session on the remote desktop.
2. Enter the following command to run the clean-up playbook.

```
(ansible) cisco@ubuntu:~/cat9k-evpn-ansible/config_examples$ ansible-playbook playbook_cleanup.yml
-i ./inventory.yml --extra-vars "design=dag_unicast_replication_type_ingress"

PLAY [Clean up the config] ******************************************************

TASK [Set playbook name and mode] ***********************************************
ok: [Leaf-15]
ok: [Spine-03]
ok: [Leaf-17]
ok: [Leaf-16]
ok: [Leaf-14]
ok: [Spine-04]
```

Resources and Closing

Your C9KV feedback is important as we continue to bring this product to market

Webex Teams: https://eurl.io/#h6Pe-Qgmo
E-mail: c9kv-interest-ext@external.cisco.com
Form: https://forms.office.com/r/b36aeEsjyA

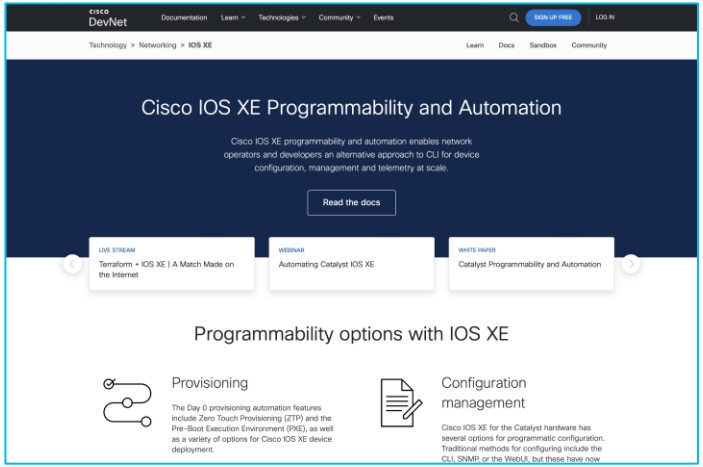
The Cat9KV is in public beta and available in CML and within Cisco dCloud

We want to understand your use cases and to ensure success with topology validation and configuration management, tooling and integrations.

There is no Tac or BU support during the beta.

Programmability Website

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

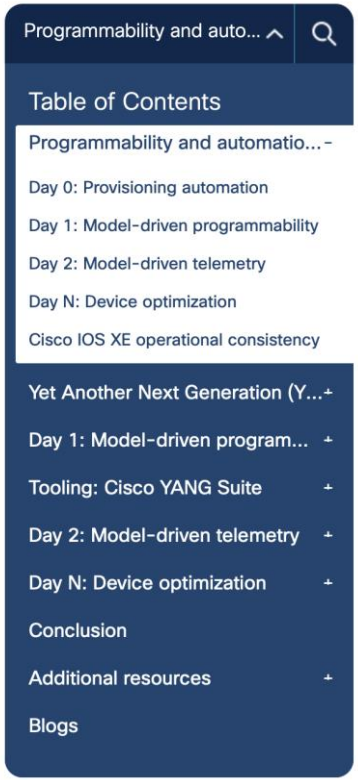


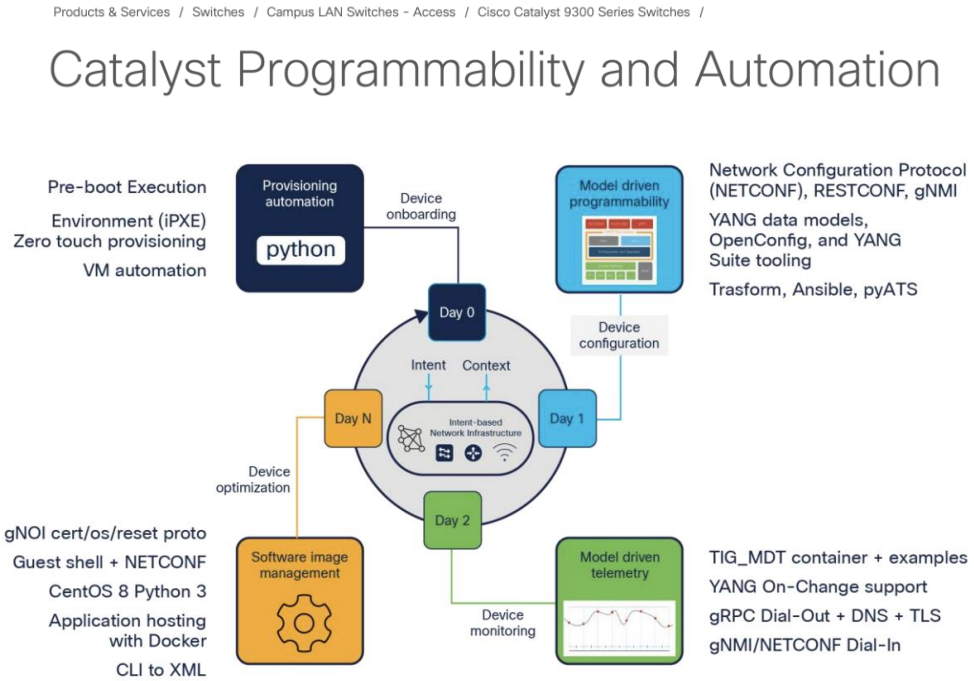
- Community Forum
- IOS XE FAQ
- White Papers
- Code Exchange
- IOS XE Docs & Guide
- Learning Tracks and Labs
- Sandboxes
  - … and more !

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

API White Paper

http://cs.co/apiwp



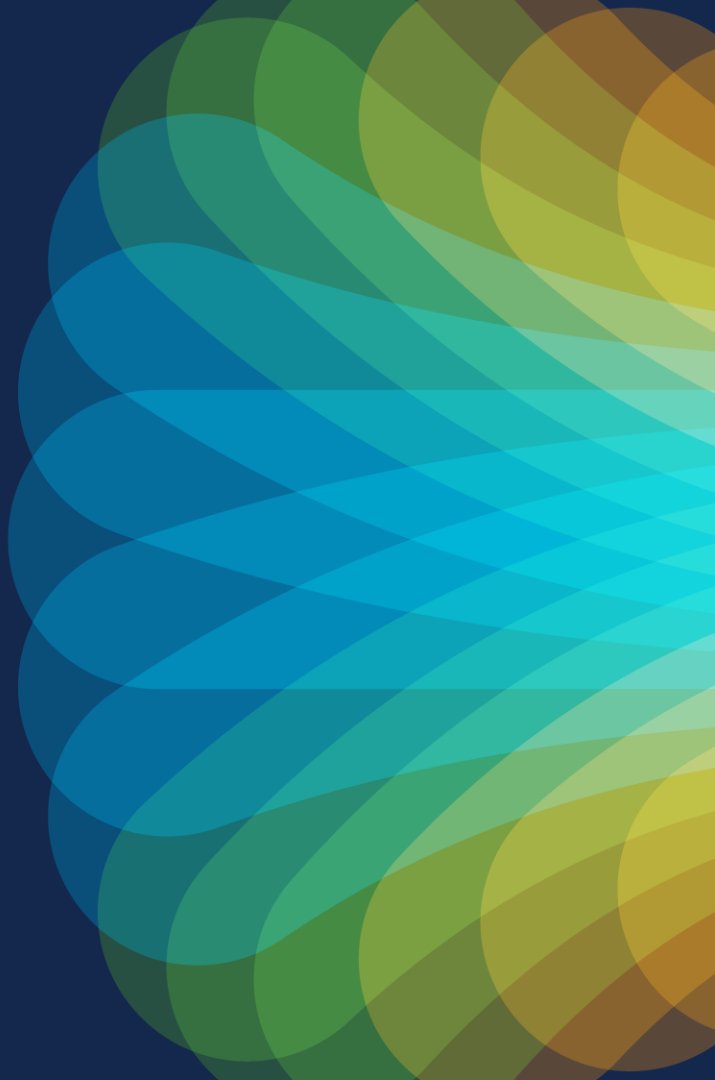


Website: https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9300-series-switches/nb-06-catalyst-programmability-automation-wp.html
PDF: https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9300-series-switches/nb-06-catalyst-programmability-automation-wp.pdf
Webinar with live demos & examples: https://www.youtube.com/watch?v=LdcK5PnPu2I

http://cs.co/apiwppdf

The bridge to possible

Thank you



cisco Live!


cisco
The bridge to possible
Thank you
cisco Live!


cisco Live!
Let's go