cisco Live!

Let's go

The bridge to possible

Catalyst 9000 Virtual Topology Simulation

and Configuration Management Solutions

Jeremy Cohoe, Technical Marketing
@jeremycohoe

DEVNET-2525

cisco Live!

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.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

About Jeremy

WxT/jcohoe@cisco.com

FYI

- From Vancouver, BC, Canada
- Amateur Radio Operator, VA7NSA
- Canadian Forces Army – Signals Operator – 4 yrs
- UBC – Wireless Infrastructure – 7 yrs
- Cisco – Enterprise Networks - 6 yrs





© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Agenda

- Intro
- Architecture
- Introduction to C9KV
- C9KV for Programmability
  - Demo
- Usecases: EVPN, SDA
  - Demo
- C9KV with ESXi
- C9KV with CML
- C9KV with ContainerLab
  - Demo
- Resources

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

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



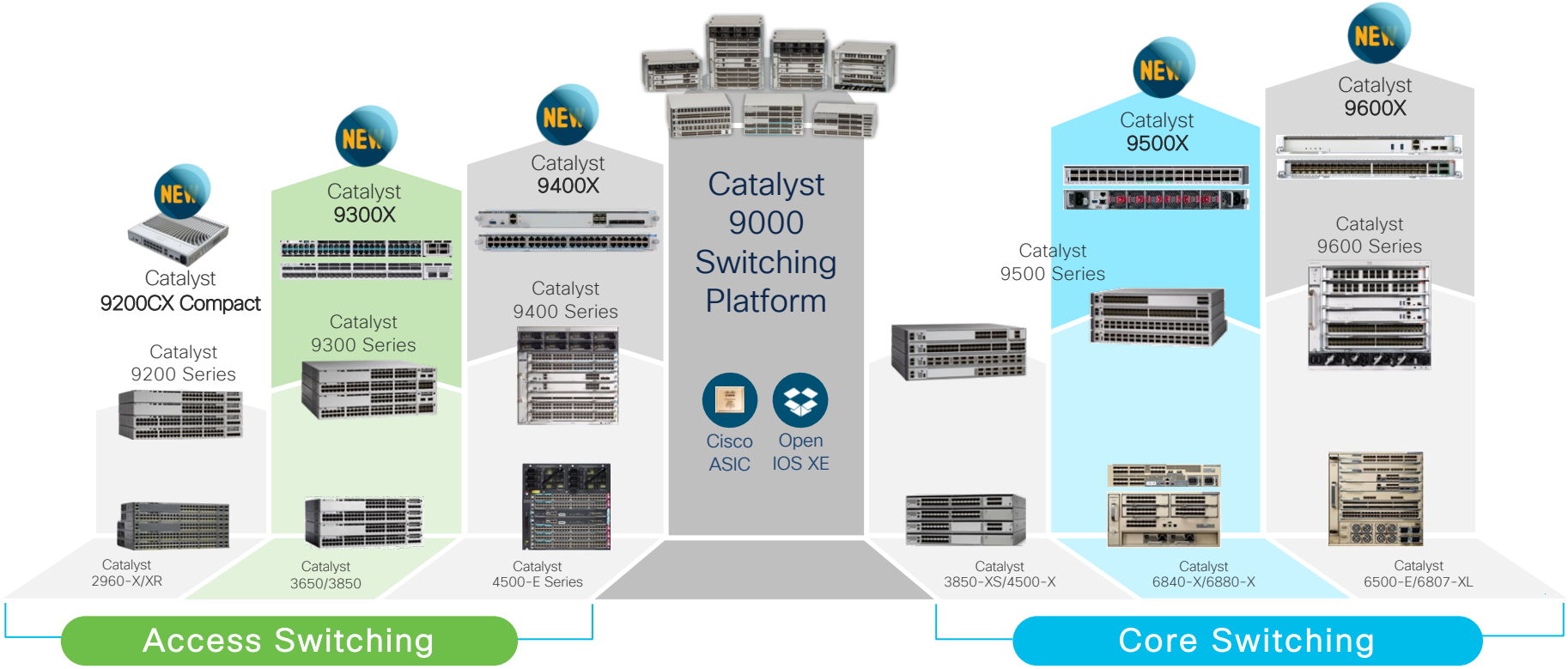
The Cat9KV is in public beta and available in CML and within Cisco dCloud

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



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Catalyst 9000 Series – Common Building Blocks



Programmable x86 Multi-Core CPU

Application Hosting
Secure Containers



Open IOS XE®

Model-Driven APIs
Modular Patching



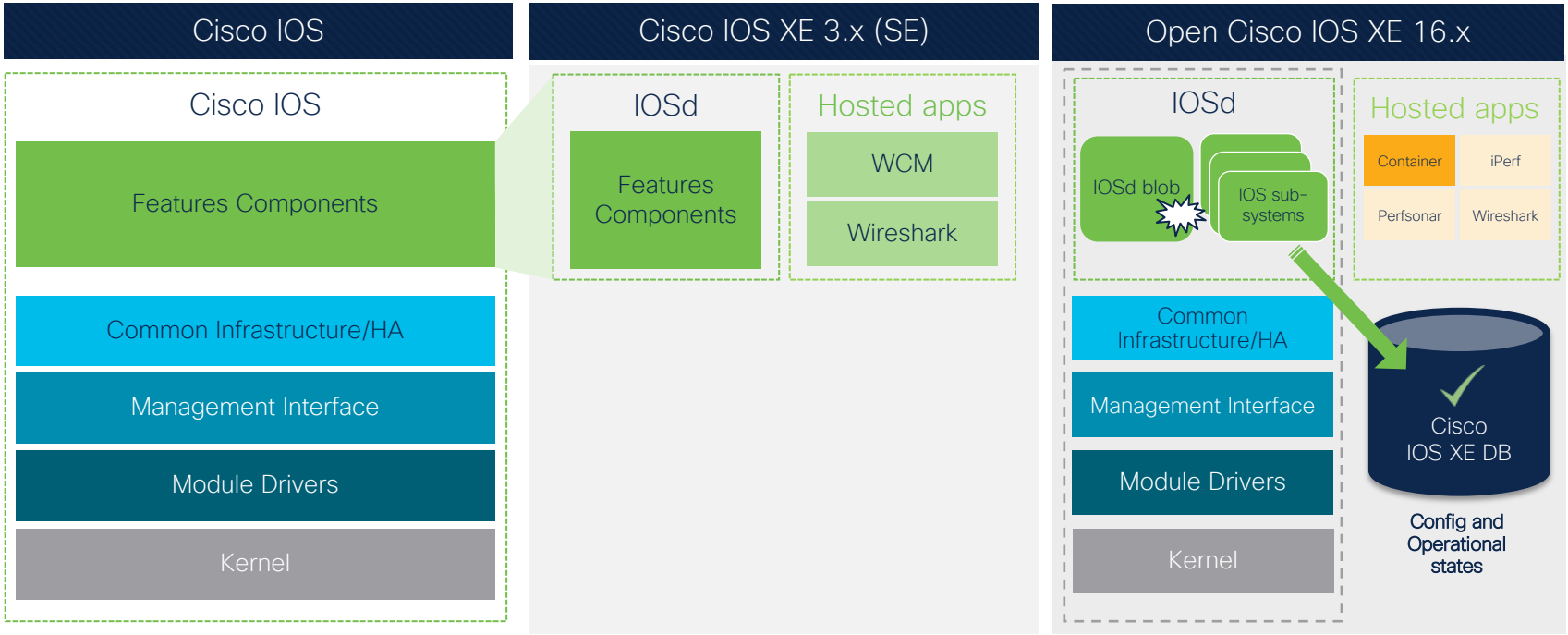
Cisco UADP & Silicon One™

Programmable Pipeline
Flexible Tables

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

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

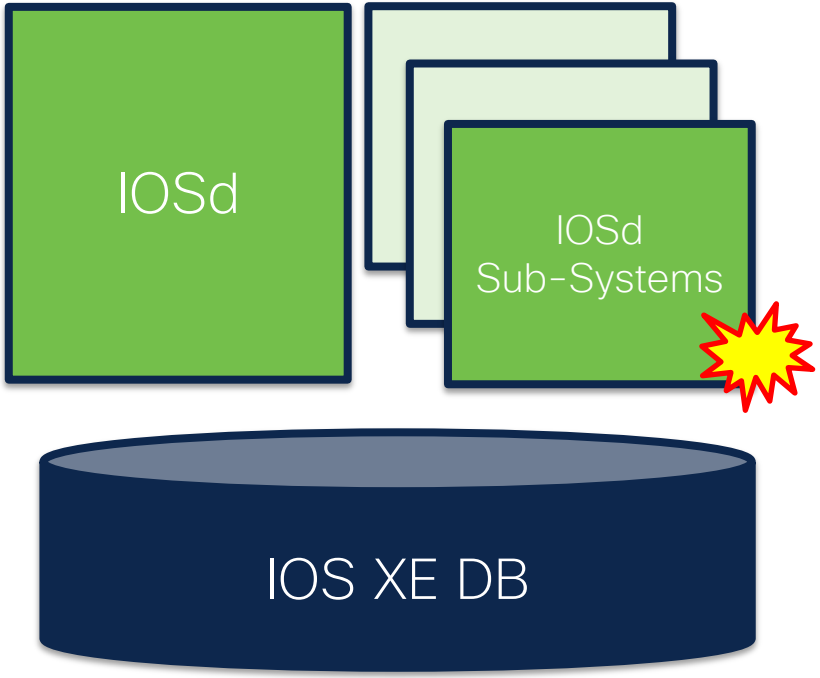
What is “Open” Cisco IOS XE?



Modern Software Architecture - with the same look and feel

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Open IOS XE – IOS Sub Systems



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

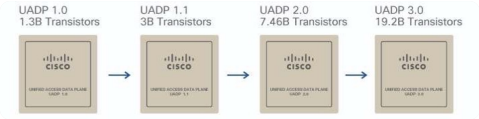
IOSd Sub-Systems enhance IOS Resiliency

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Custom ASICs – Programmable Silicon

Access Data-Plane





Cisco **Silicon One**™





Flexible Pipelines
Investment Protection

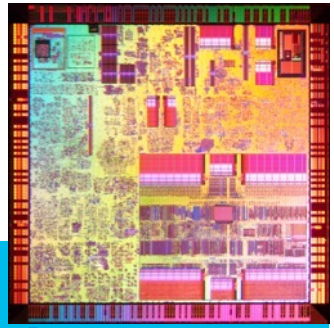
Adaptable Tables
Universal Deployment

Scalable Resources
Enhanced Scale and Buffering

Flexible & Programmable ASICs – Adapt to New Technologies

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

What is an ASIC?



An Application Specific Integrated Circuit (ASIC) is a silicon microchip designed for a specific task ...

... rather than ‘general-purpose’ processing in a CPU.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Why do we need ASICs?

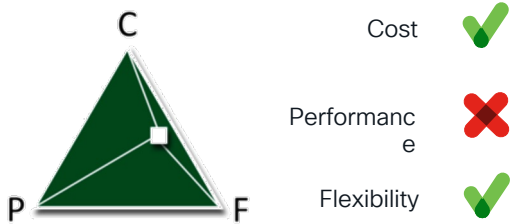
CPUs are Flexible but Slow

A ‘general-purpose’ CPU may be fast at running random-access applications, on a laptop or server, but **processing and forwarding ‘network traffic’** is a different matter.

Network traffic requires **constant searching of large memory tables** (e.g. L2 tables for MAC addresses, L3 tables for IP routes, L4 ACLs for Security and QoS, etc.)

In a CPU - there are **limited data paths** and tables are held in **off-chip memories** (e.g. DRAM) that can incur significant performance penalties for frequent access.





Remember, this is **Millions - Billions** of packets per second

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

How C9KV is different than other container NOS?

- The C9KV does ASIC simulation…
  - Use of asic pipelines in software
  - Different than a process thread
- <u>Simulation of what happens with actual hardware</u>
- <u>Same</u> behavior as what you would expect with physical hardware
- <u>Same</u> feature testing between C9K and C9KV

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





Cisco **Silicon One**™



Flexible Pipelines
Investment Protection

Adaptable Tables
Universal Deployment

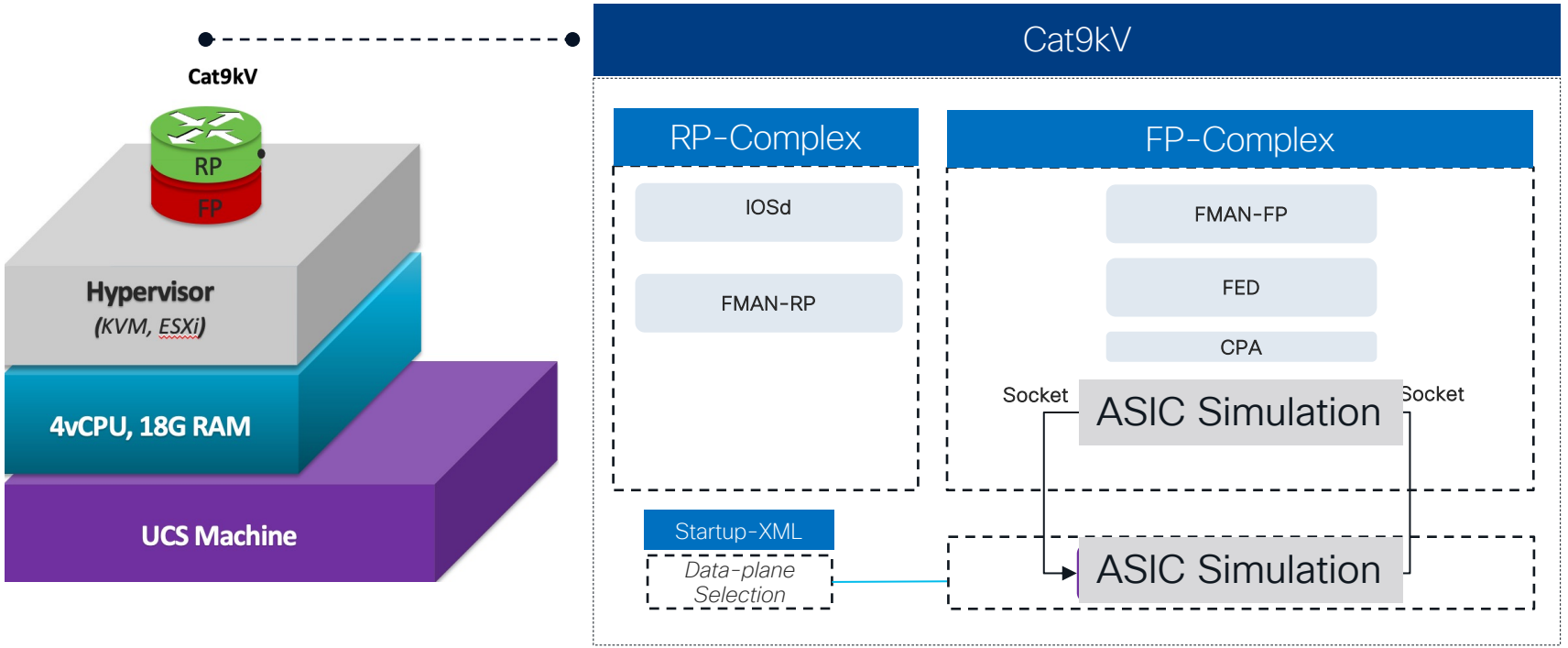
Scalable Resources
Enhanced Scale and Buffering

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Catalyst 9000 Virtual Switch

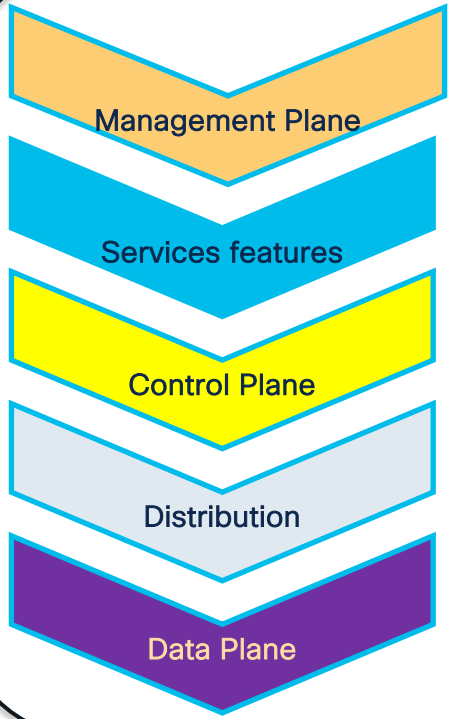


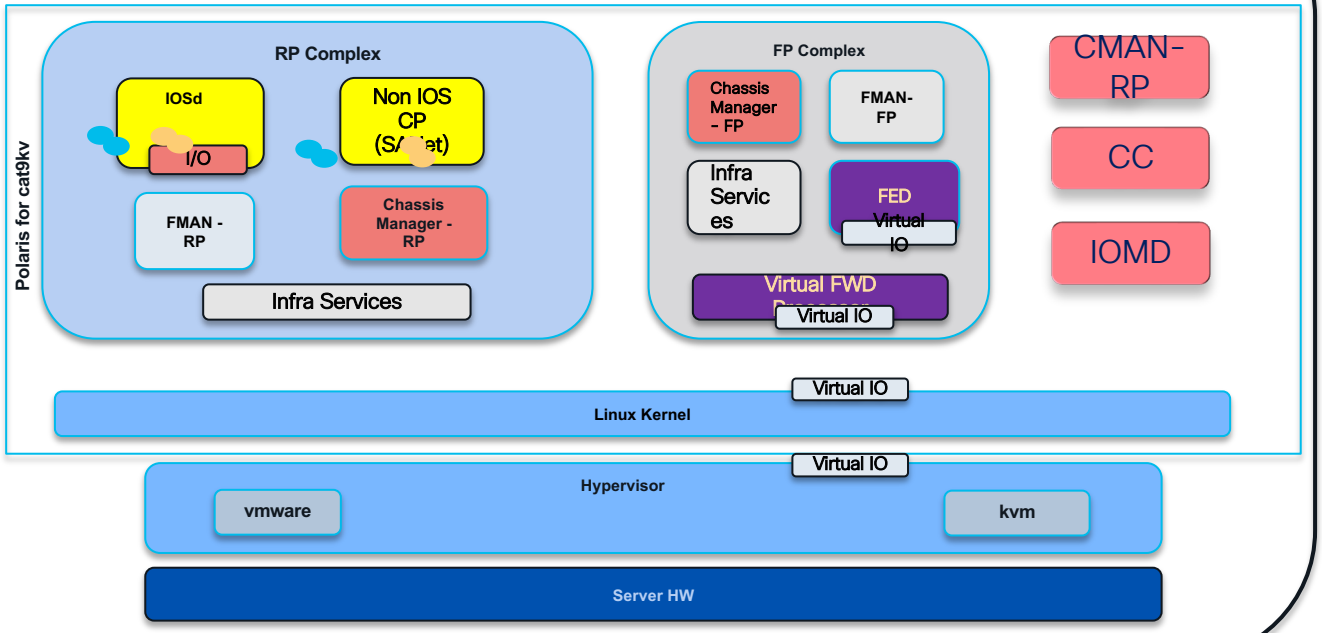
Architecture



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

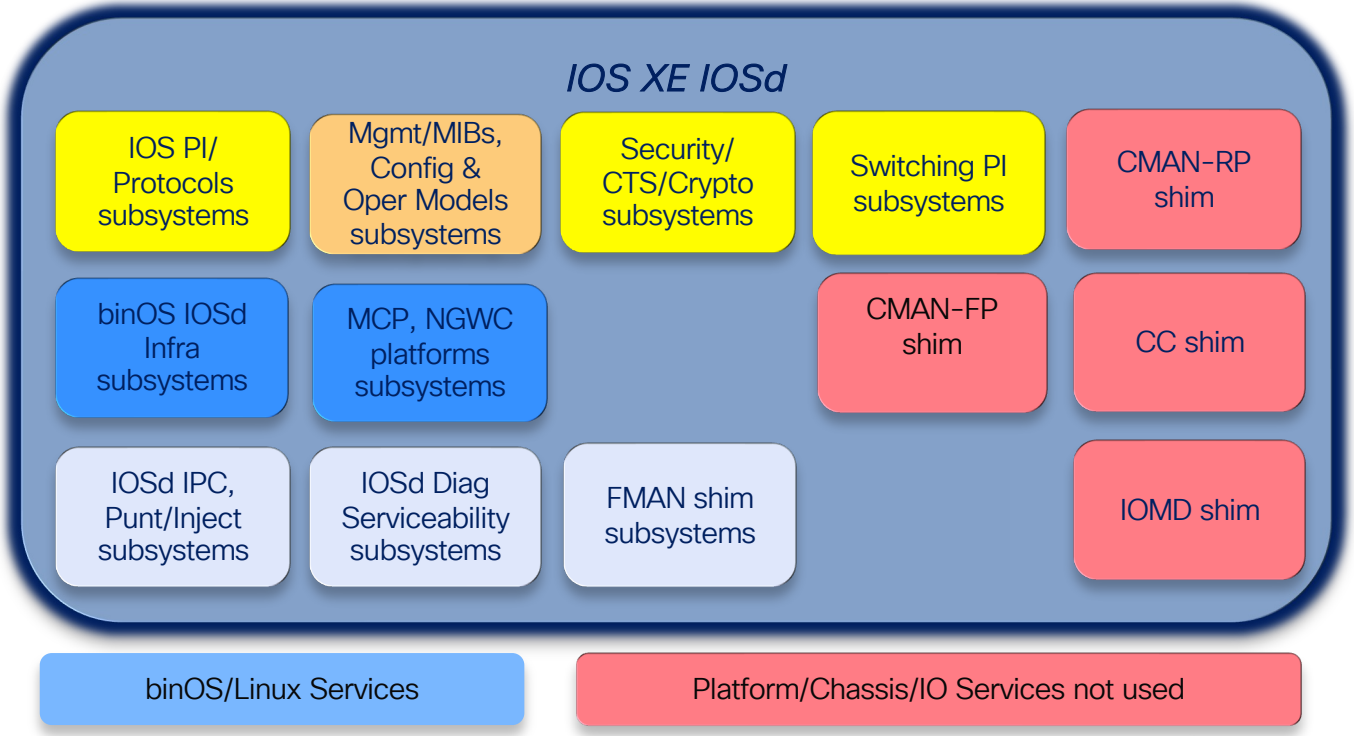
Capabilities





© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

IOSd components and services



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Introducing Cat9kV



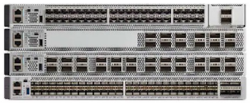
Catalyst 9200
Series Switches



Catalyst 9300
Series Switches



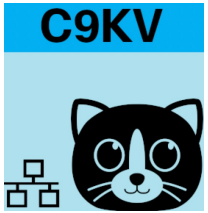
Catalyst 9400
Series Switches



Catalyst 9500
Series Switches



Catalyst 9600
Series Switches



Cat9000 Virtual Switch
An Independent PID & SKU



Increase Productivity



Enable Hybrid workplace



Facilitate Technology Adoption



Convenient Access

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Cat9kV Goals, Vision



Increase Productivity

- *Faster feature development unhindered by board platform dev*
- *Richer data-path debug*
- *Flexible data-path choice*
- *Flexible regression platform*



Enable Hybrid workplace

- *Construct standard network topologies using point-and-click*
- *Eliminate hardware wiring problems (no lab trips)*



Facilitate Technology Adoption

- *Customer Modelling Labs, Solutions showcase*
- *Enable GTM Modelling*
- *Ready access to available switching services features*
- *Customer what-if scenario analysis*



Access with Ease

- *Deployment to private/public cloud supported via*
  - *CML integration*

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Cat9kV High Level Specification

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
Deployment capabilities	• 8 / 24interfaces + 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 pps
	  • Silicon 1 – 300 pps

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

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:~$
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Data Plane & Features

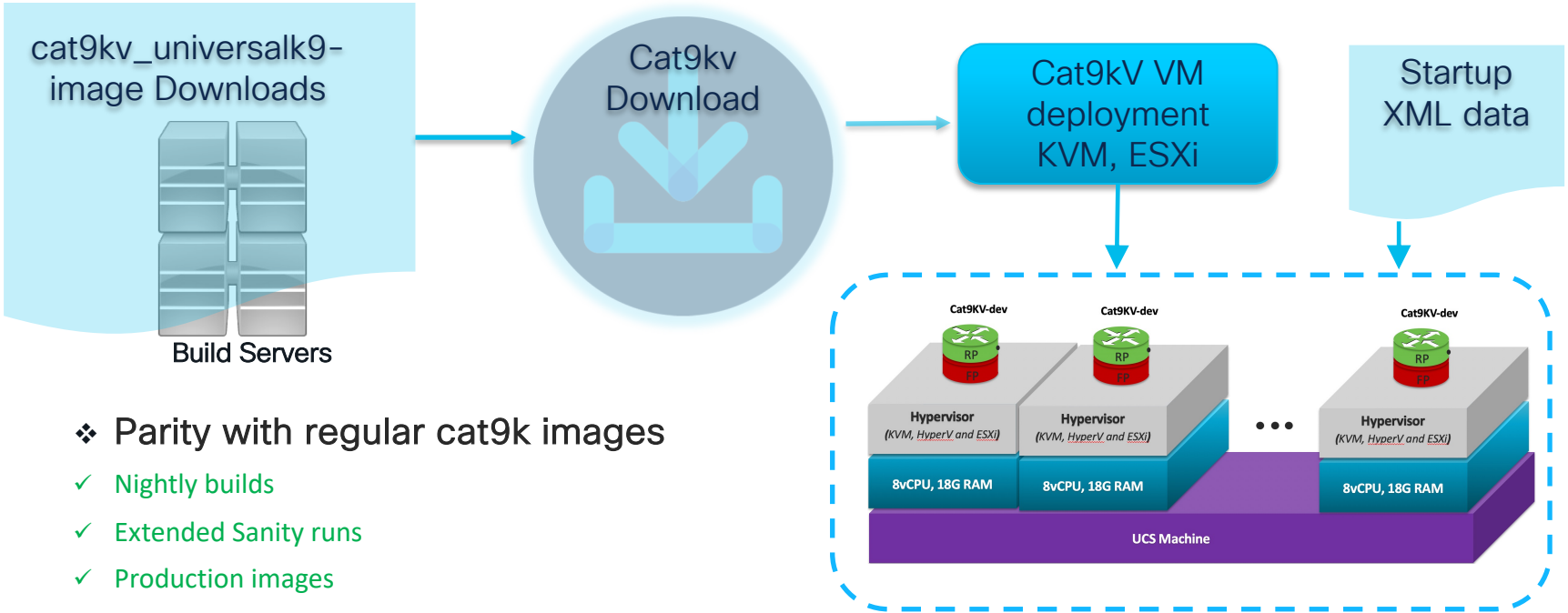
The 8 / 24 interfaces use the data plane and have PPS limitations
Recommend to use Gi0/0 which uses the ESXi management plane

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. Siilcon 1

Features	• Services feature parity for data plane constructs
	• Low packet throughput but sufficient for functional feature behavior
	  • UADP – 200 pps
	  • Silicon 1 – 300 pps

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Unified Single Image – virtual twin of Cat9k



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

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

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, MAC Learning/Aging, MACSEC rekey etc.
GOLD (HW DIAG)	Typically achieved using loopback in hardware at MAC/SERDES.
App hosting	No Guest shell (ZTP), IOX, or Application Hosting infra (IDK
TM, QoS	Traffic Management, Queuing actions
FNF, Netflow	Currently unsupported
Stacking/SVL	Currently unsupported
Embedded WLC	Currently unsupported
Bundle install mode	Only install mode supported

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

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

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

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

Monitoring
Syslog
SNMP
gNMI
NETCONF
RESTCONF
gRPC Dial-Out

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

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 ☺

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Ethernet Renaming

.YAML file to define interface names

Simplified naming coming as part of IOS XE 17.12 release
Gi1/0/1 can be set to Ethernet1 as part of bootstrapping

IOS XE 17.11 >

```
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 unkn
Gi1/0/3
Gi1/0/4
Gi1/0/5
Gi1/0/6                        connected    1          a-full a-1000 unkn
Gi1/0/7                        connected    1          a-full a-1000 unknown
Gi1/0/8                        connected    1          a-full a-1000 unknown
jcohoe-c9kv-1#
```

IOS XE 17.12 +

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

Port        Name               Status       Vlan       Duplex  Speed Type
  Ethernet1                    connected    1          a-full a-1000 unknown
  Ethernet                     connected    1          a-full a-1000 unknown
  …                            connected    1          a-full a-1000 unknown
  …                            connected    1          a-full a-1000 unknown
  …                            connected    1          a-full a-1000 unknown
  Ethernet                     connected    1          a-full a-1000 unknown
  Ethernet                     connected    1          a-full a-1000 unknown
  Ethernet24                   connected    1          a-full a-1000 unknown

jcohoe-c9kv-1#
```

* Marketing example ;)

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Programmability & Automation

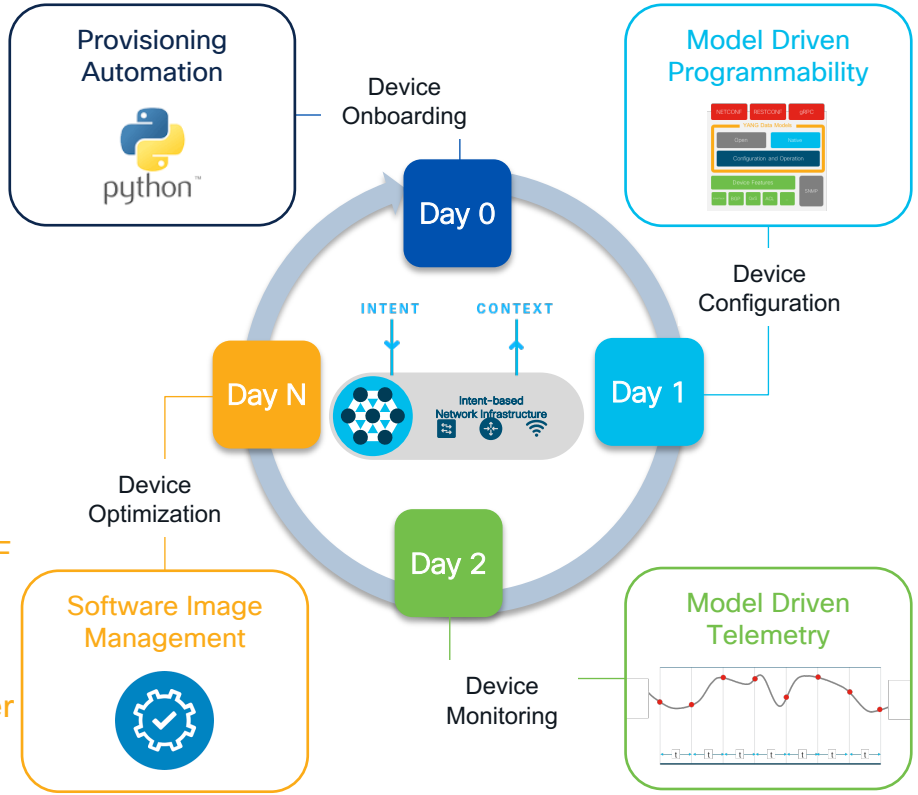


IOS XE Programmability & Automation Lifecycle

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 XML

TIG_MDT container + guide

YANG On-Change support

gRPC Dial-Out + DNS + mTLS

gNMI/NETCONF Dial-In

cisco Live!

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Programmable Interfaces

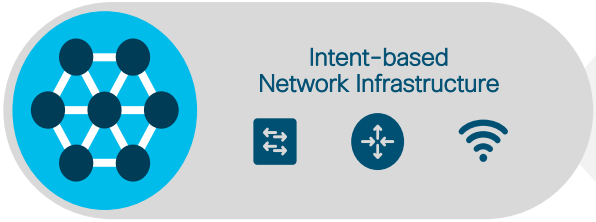
CLI

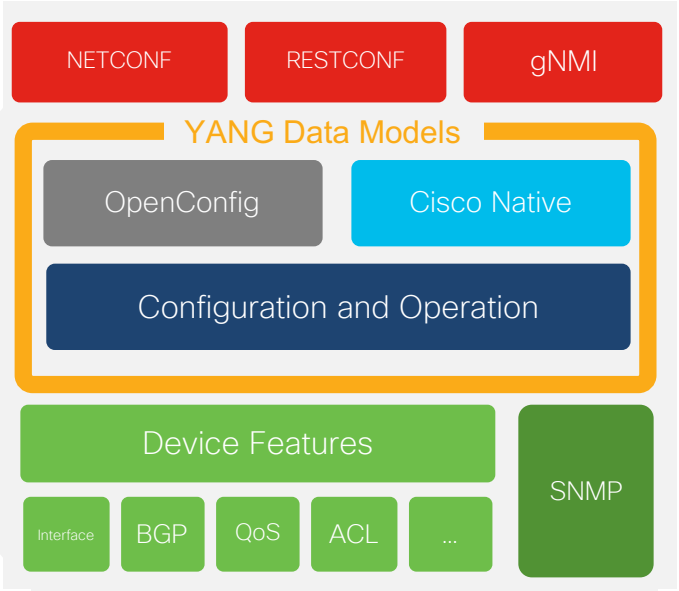
SNMP

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

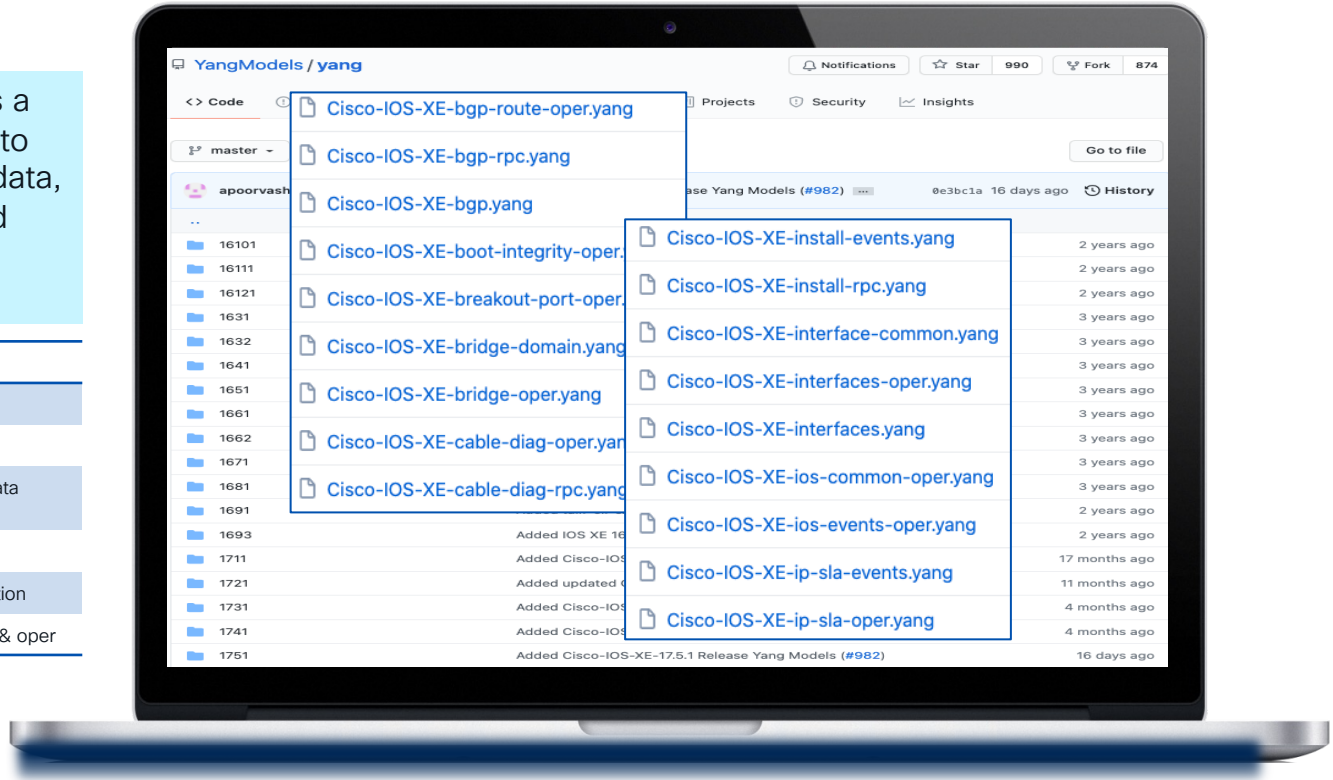
CISCO

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

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

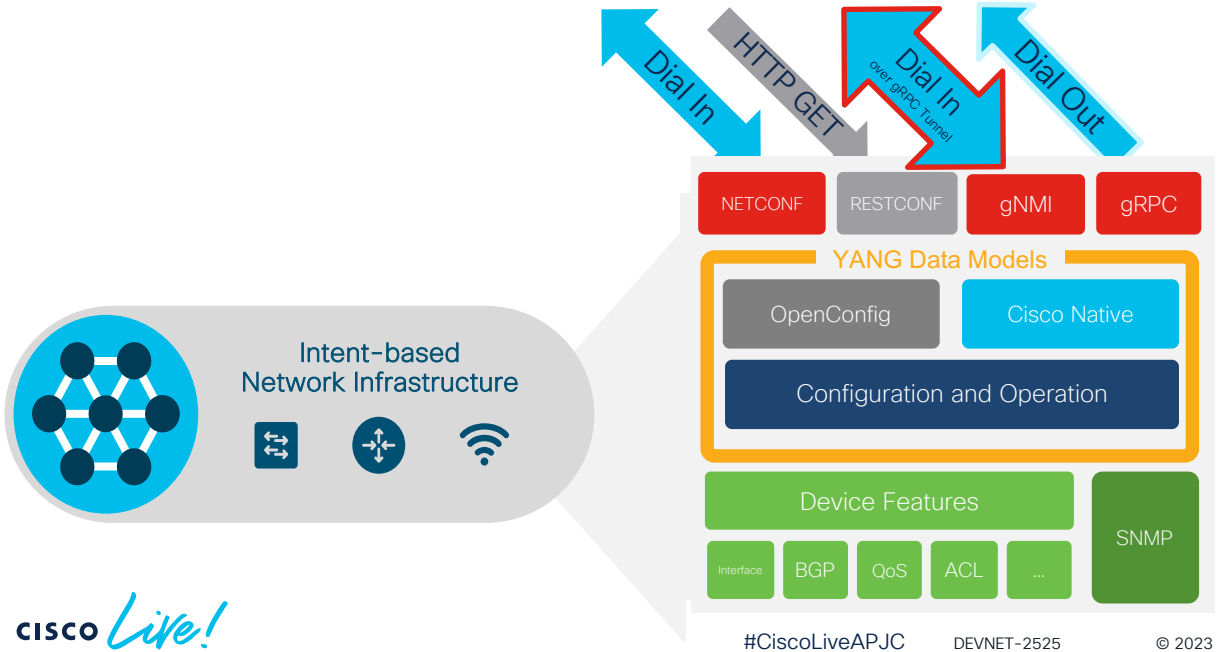
© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

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

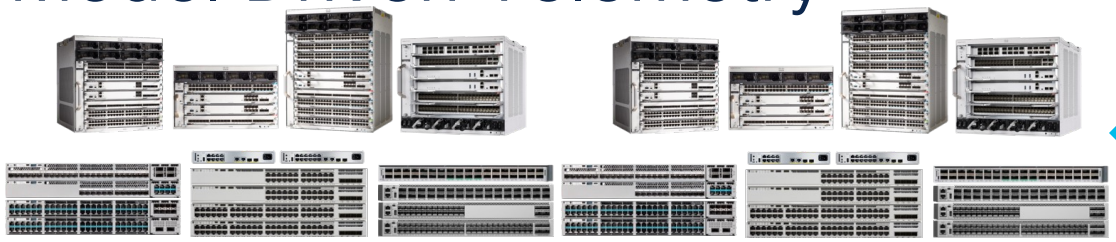
On-change event and time-based publication options

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

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

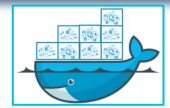
InfluxDB

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



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Updated TIG_MDT container now available!

Updated April 2023

Upgrade coming to Telegraf, Influx, and Grafana Model Driven Telemetry (TIG_MDT) Docker container
Making it easier to consume telemetry in production

Upgraded Telegraf, InlfuxDB, and Grafana tools
Additional dashboards for
Device Health, Wireless Client, Wireless AP, RF etc
Examples for device CLI configuration for telemetry
Details of scale and data storage requirements



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

Collector/Receiver
Decodes to text

telegraf™

Storage
Time Series Database
Monitoring
and Visualizations

InfluxDB
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

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Demo time...



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

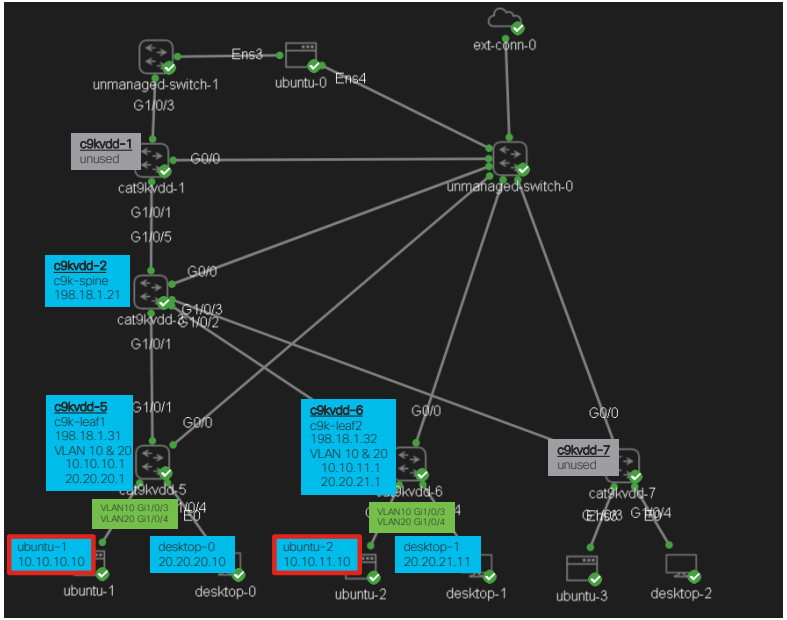
Terraform/RESTCONF
- Declarative EVPN fabric management

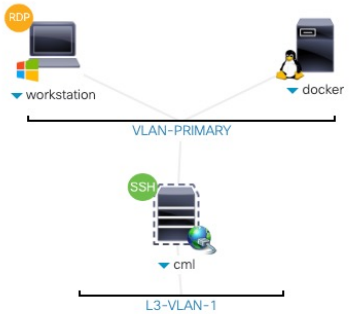
Ansible
- EVPN solution enablement using CLI

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

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

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

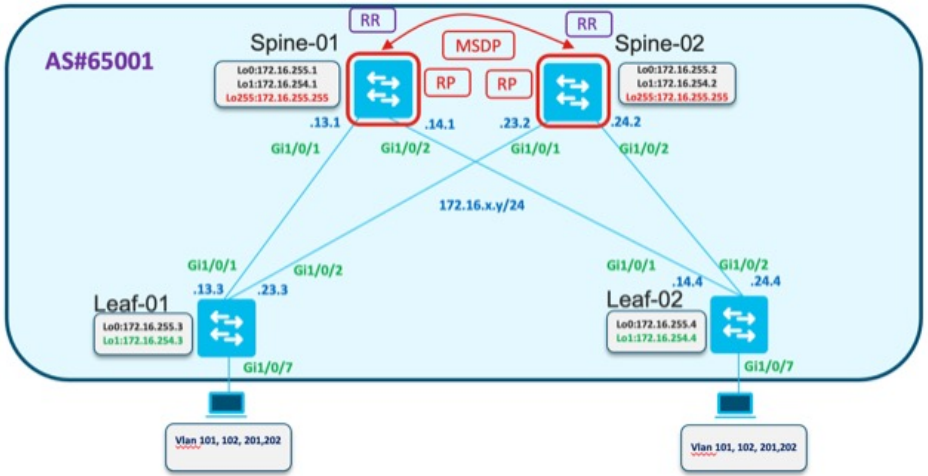
Usecase: EVPN



Now published on Github

BGP EVPN solution management with Ansible

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



Try it out yourself and engage directly on GitHub:
https://github.com/CiscoDevNet/Cat9kEVPN-Ansible

Quick start

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

Step 1

On this step Underaly is provisioned.

Step 1a

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

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

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

Detailed information could be found here

Step 1b

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

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

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

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

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

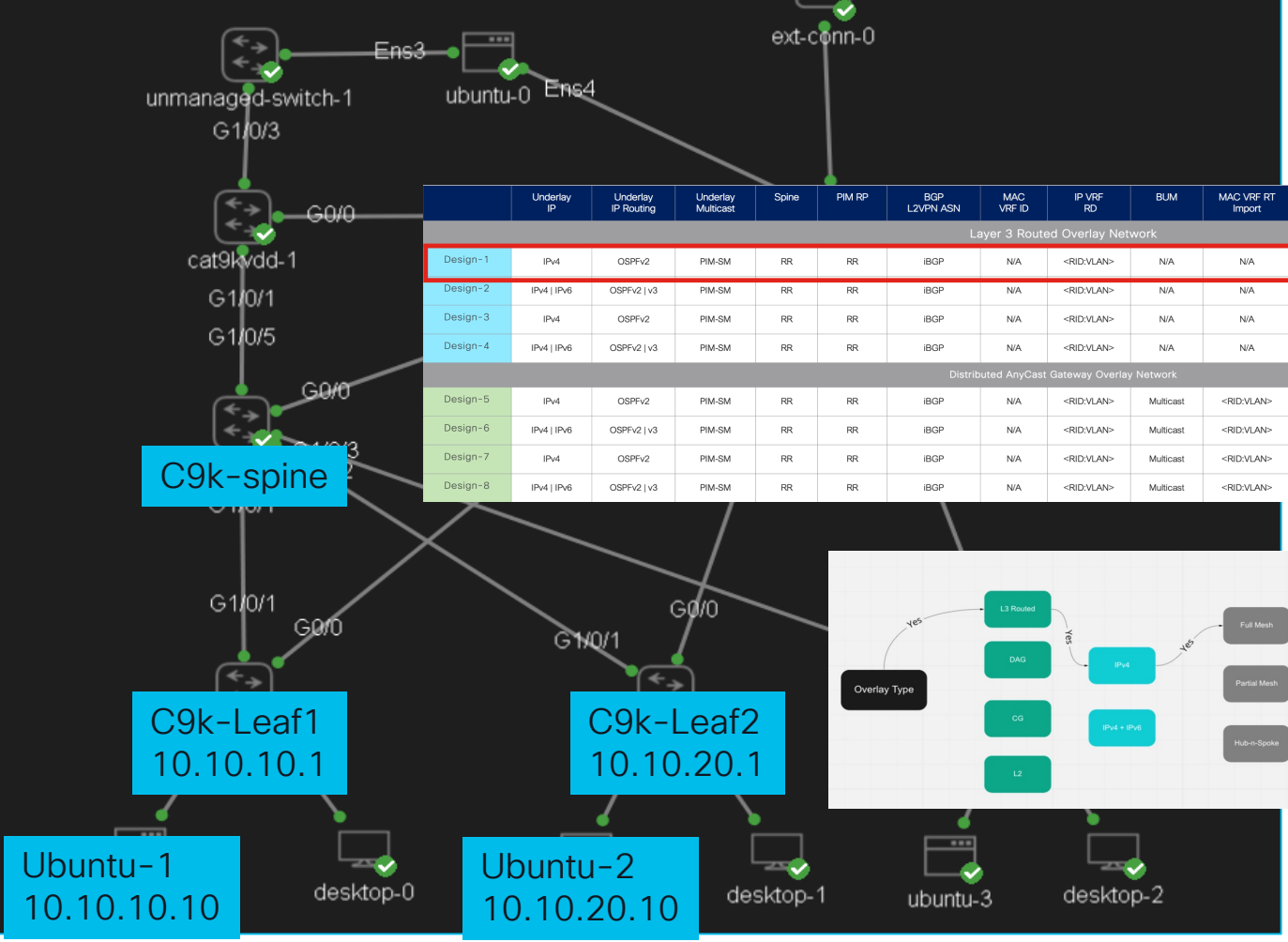
Detailed information could be found here

Step 1c

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

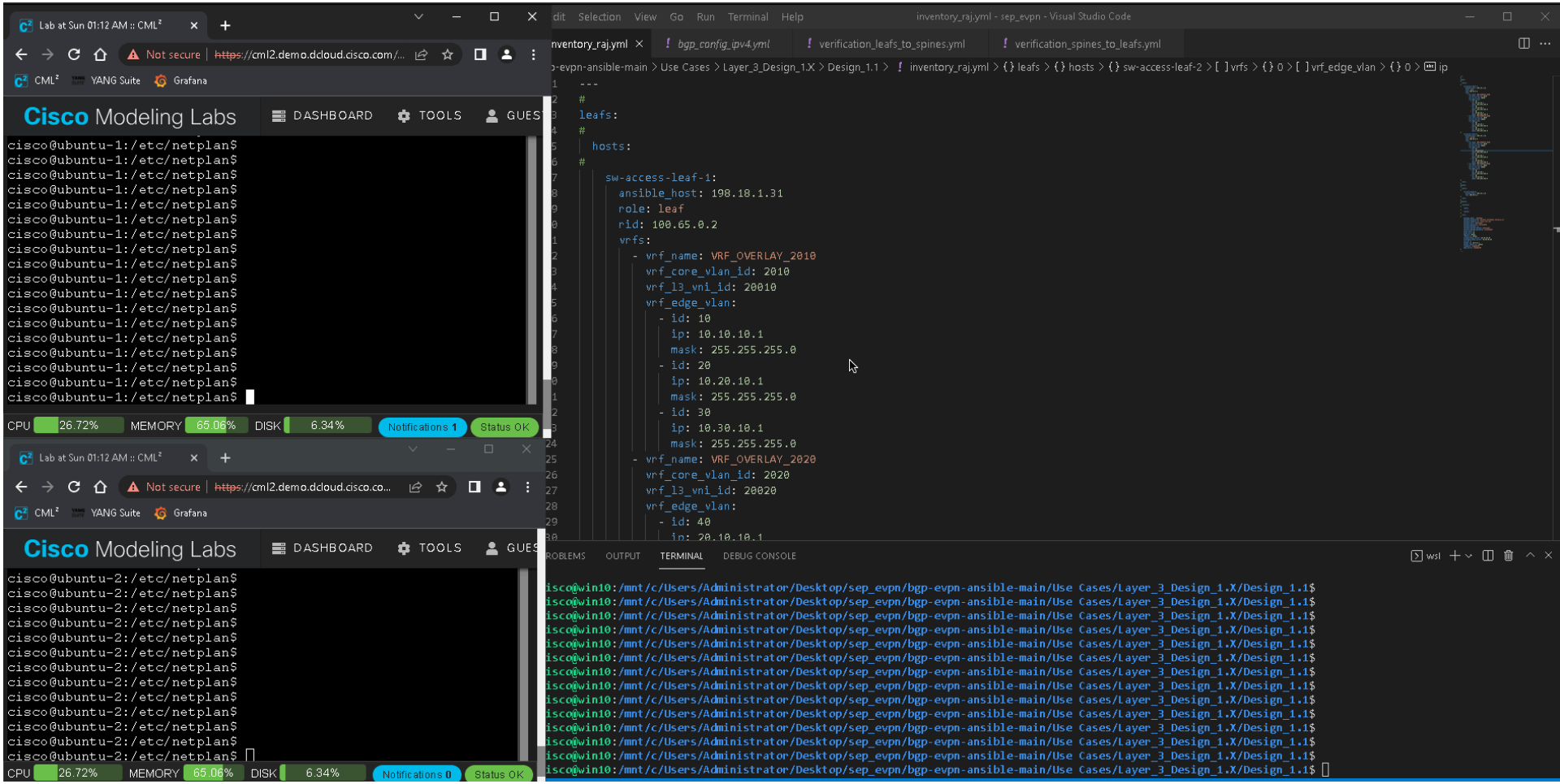
© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Ansible BGP EVPN Design-1 solution enablement



	Underlay IP	Underlay IP Routing	Underlay Multicast	Spine	PIM RP	BGP L2VPN ASN	MAC VRF ID	IP VRF RD	BUM	MAC VRF RT Import	MAC VRF RT Export	IP VRF RT Import	IP VRF RT Export	Layer 3 Multicast	Border Handoff
Layer 3 Routed Overlay Network															
Design-1	IPv4	OSPFv2	PIM-SM	RR	RR	iBGP	N/A	<RID:VLAN>	N/A	N/A	N/A	<ASN:VLAN>	<ASN:VLAN>	N/A	Layer 3 VRF
Design-2	IPv4 | IPv6	OSPFv2 | v3	PIM-SM	RR	RR	iBGP	N/A	<RID:VLAN>	N/A	N/A	N/A	<ASN:VLAN>	<ASN:VLAN>	N/A	Layer 3 VRF
Design-3	IPv4	OSPFv2	PIM-SM	RR	RR	iBGP	N/A	<RID:VLAN>	N/A	N/A	N/A	<ASN:VLAN>	<ASN:VLAN>	TRMv4	Layer 3 VRF
Design-4	IPv4 | IPv6	OSPFv2 | v3	PIM-SM	RR	RR	iBGP	N/A	<RID:VLAN>	N/A	N/A	N/A	<ASN:VLAN>	<ASN:VLAN>	TRMv4 | v6	Layer 3 VRF
Distributed AnyCast Gateway Overlay Network															
Design-5	IPv4	OSPFv2	PIM-SM	RR	RR	iBGP	N/A	<RID:VLAN>	Multicast	<RID:VLAN>	<RID:VLAN>	<ASN:VLAN>	<ASN:VLAN>	N/A	Layer 3 VRF
Design-6	IPv4 | IPv6	OSPFv2 | v3	PIM-SM	RR	RR	iBGP	N/A	<RID:VLAN>	Multicast	<RID:VLAN>	<RID:VLAN>	<ASN:VLAN>	<ASN:VLAN>	N/A	Layer 3 VRF
Design-7	IPv4	OSPFv2	PIM-SM	RR	RR	iBGP	N/A	<RID:VLAN>	Multicast	<RID:VLAN>	<RID:VLAN>	<ASN:VLAN>	<ASN:VLAN>	TRMv4	Layer 3 VRF
Design-8	IPv4 | IPv6	OSPFv2 | v3	PIM-SM	RR	RR	iBGP	N/A	<RID:VLAN>	Multicast	<RID:VLAN>	<RID:VLAN>	<ASN:VLAN>	<ASN:VLAN>	TRMv4 | v6	Layer 3 VRF

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Usecase:
Auto-vSDA
... in dCloud



What is Auto-vSDA?

Automated Virtual SDA

- 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 DNA 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

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Components

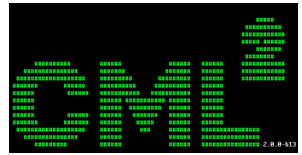
Playbooks

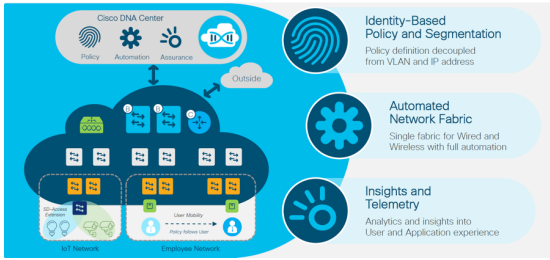
- Restore VM Snapshots specific to testbed
- Build CML2 testbed
- Build Fabric *

ANSIBLE

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

CML2 Lab
- cat9kv
- IOSv
- Ubuntu

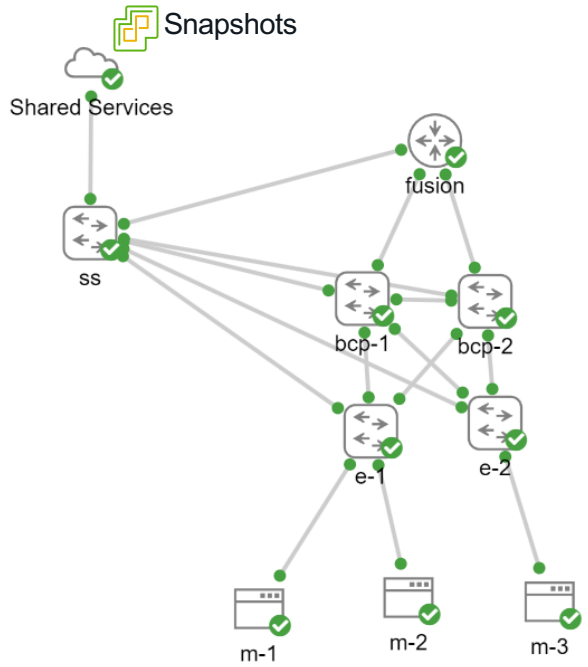




* Using DNAC APIs if needed

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

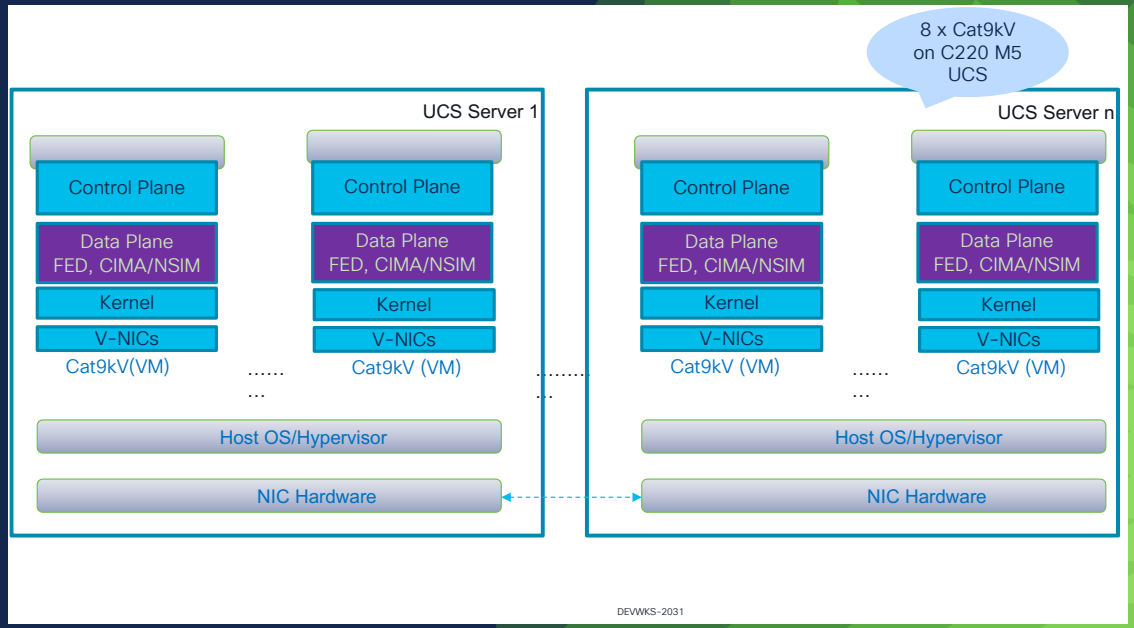
Testbed example “Dual-Border-Edge”



1. Restore VM snapshots
   - vDNAC 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

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

With ESXi

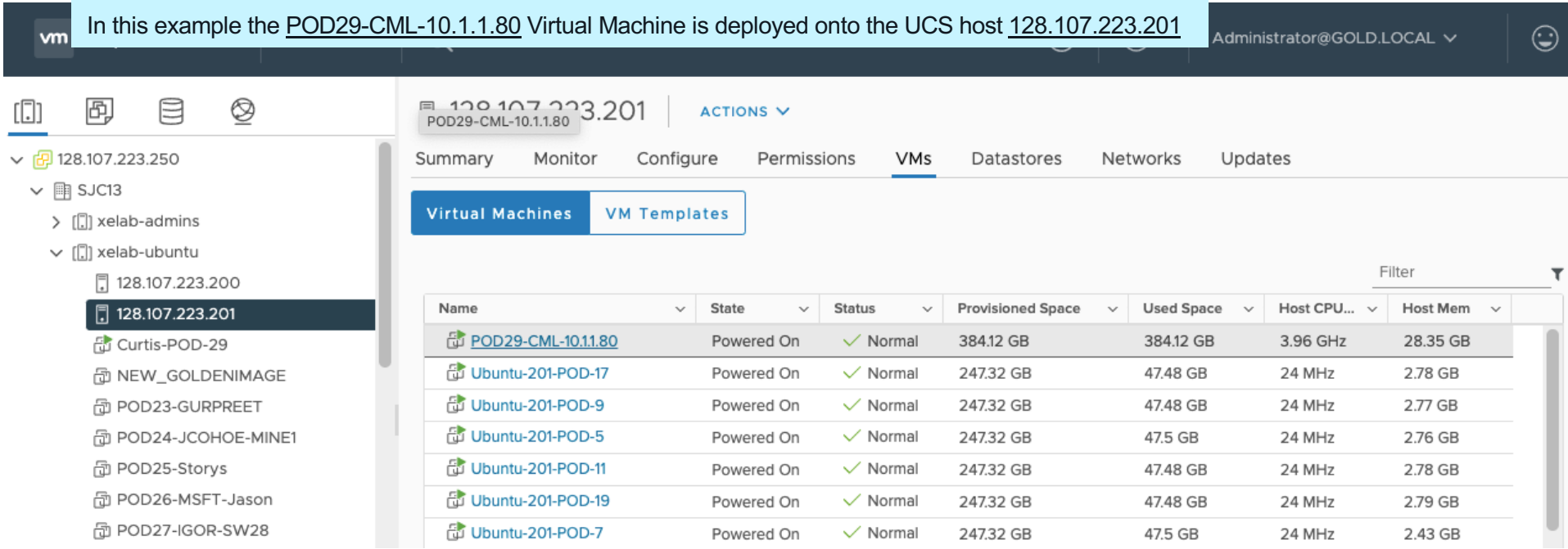

8 x Cat9kV
on C220 M5
UCS
UCS Server 1
UCS Server n
Control Plane
Data Plane
FED, CIMA/NSIM
Kernel
V-NICs
Cat9kV(VM)
Cat9kV (VM)
Host OS/Hypervisor
NIC Hardware
DEVWKS-2031

UCS and VMWare

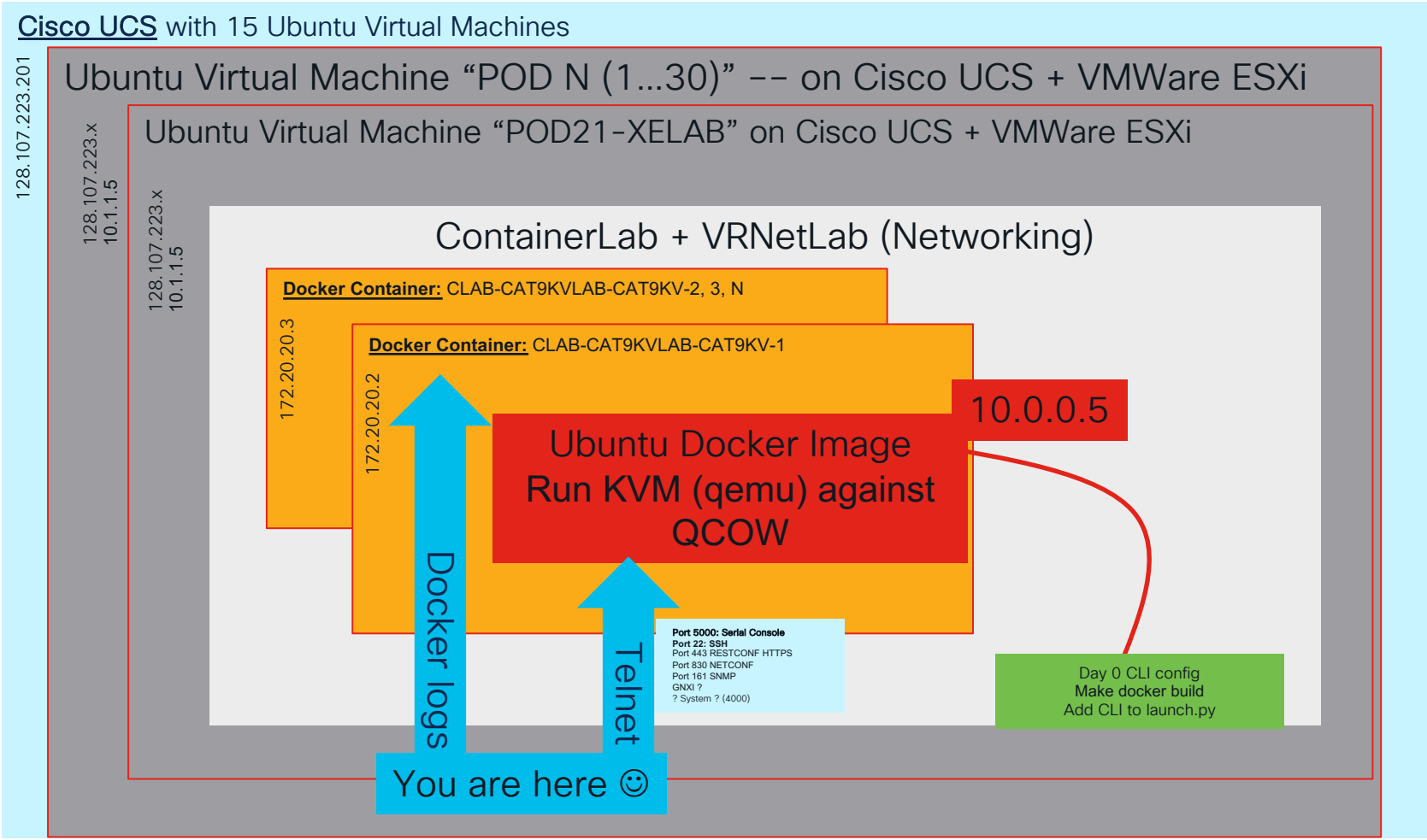
An existing UCS with vmware ESXI + VCSA is being used to run the CML

Instead of UCS + vmware, CML can be install directly onto the UCS as it runs CentOS 8 Linux

In this example the POD29-CML-10.1.1.80 Virtual Machine is deployed onto the UCS host 128.107.223.201



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public



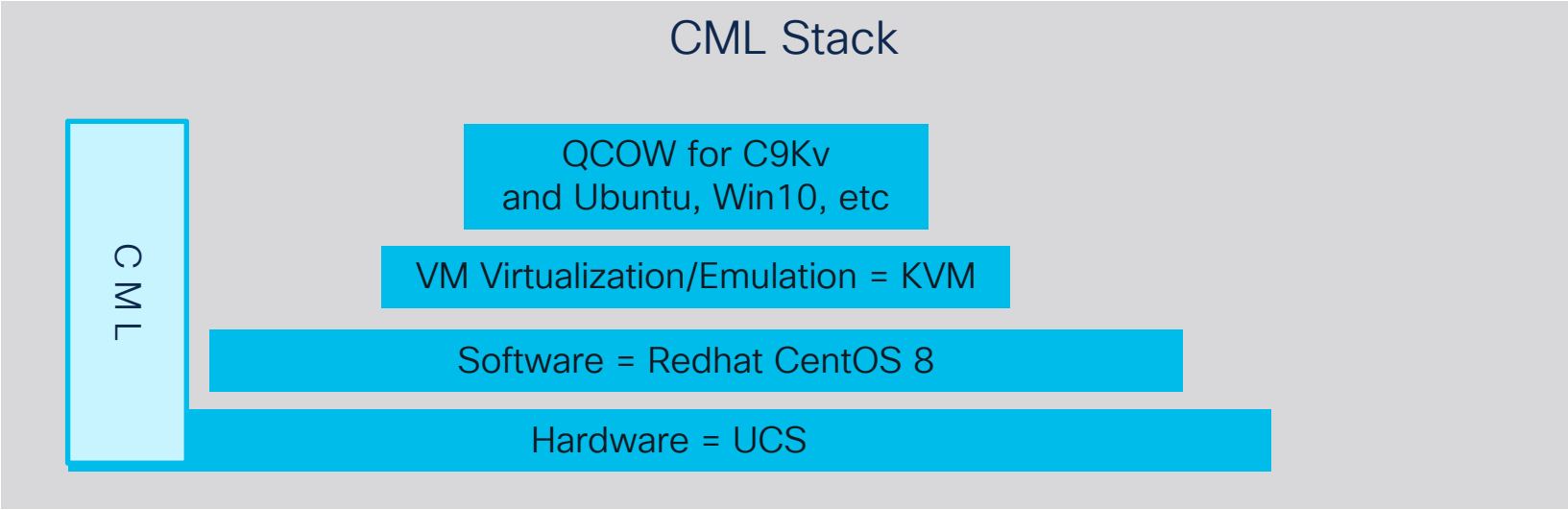
© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

C9KV with CML



What is CML ?

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



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

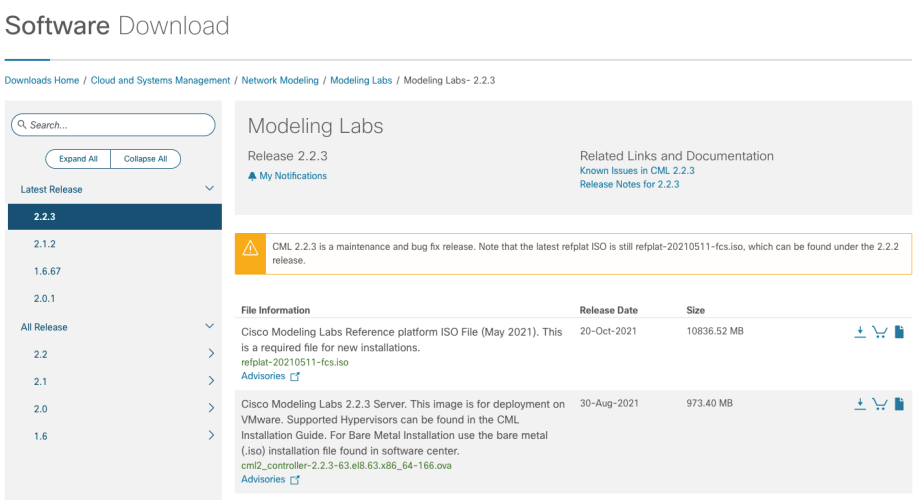
Download CML OVA and reference platforms ISO

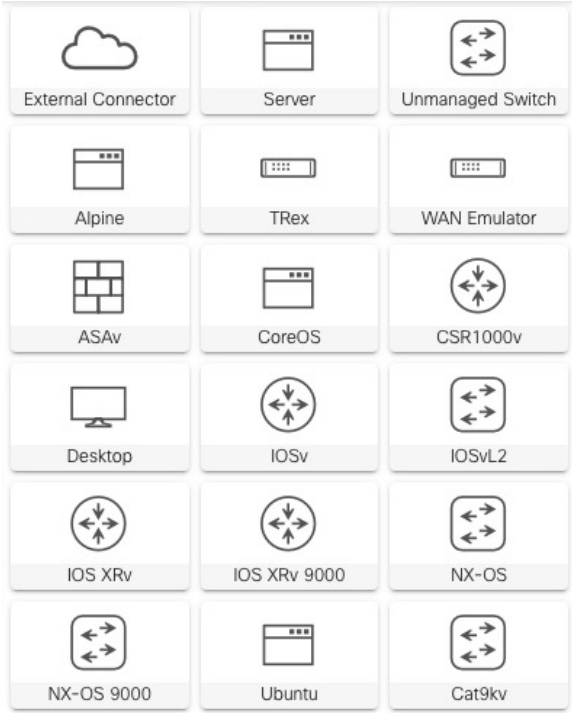
CML 2.5 is available from software.cisco.com

The C9KV is included with the CML 2.5 Reference Platforms

CML run's on-prem from ISO or OVA

CML with C9KV is also available in dCloud





© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Accessing Cisco Modelling Labs

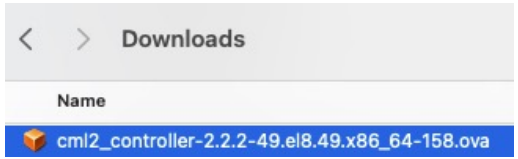
CML 2.5 is available from software.cisco.com

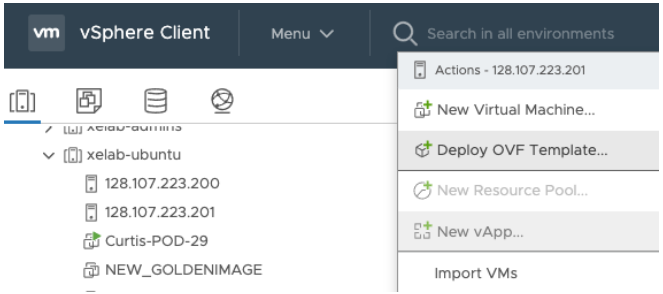
The C9KV is included with the CML 2.5 Reference Platforms

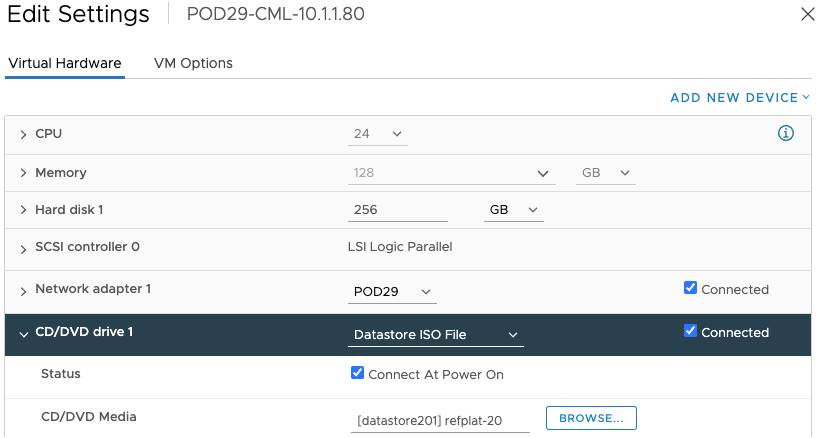
CML run's on-prem from ISO or OVA

CML with C9KV is also available in dCloud





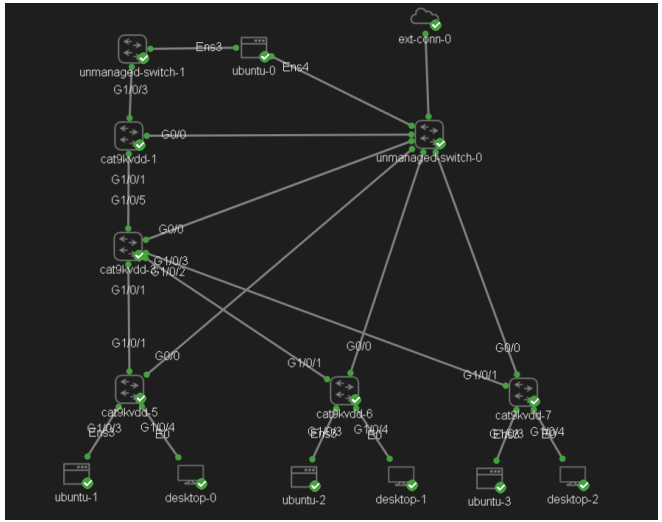


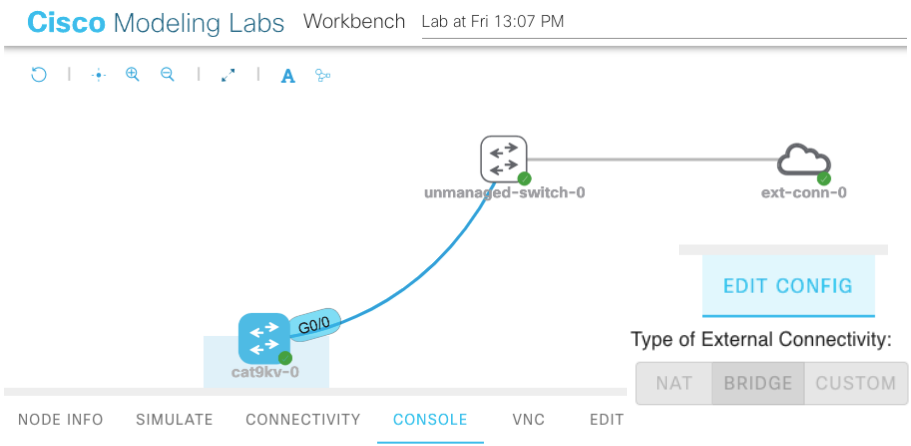


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

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 and configure as needed
Use Day0 config.txt to bootstrap






```
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



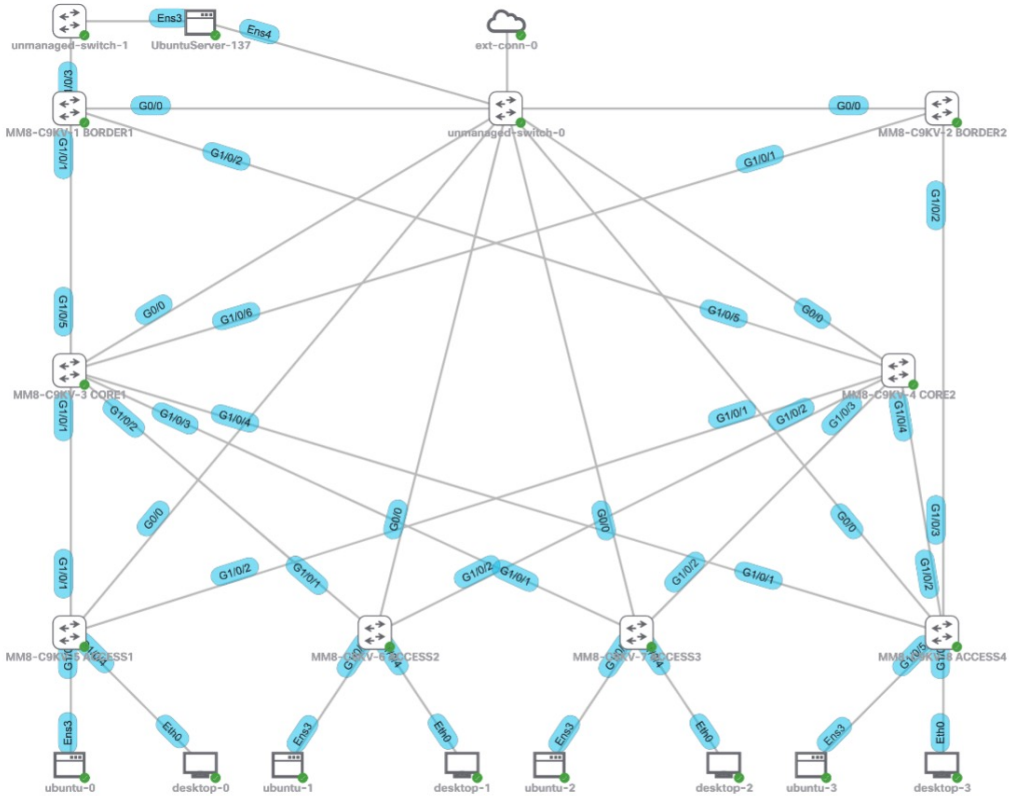
        --- System Configuration Dialog ---

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

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


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Virtualized Topology with CML and C9Kv



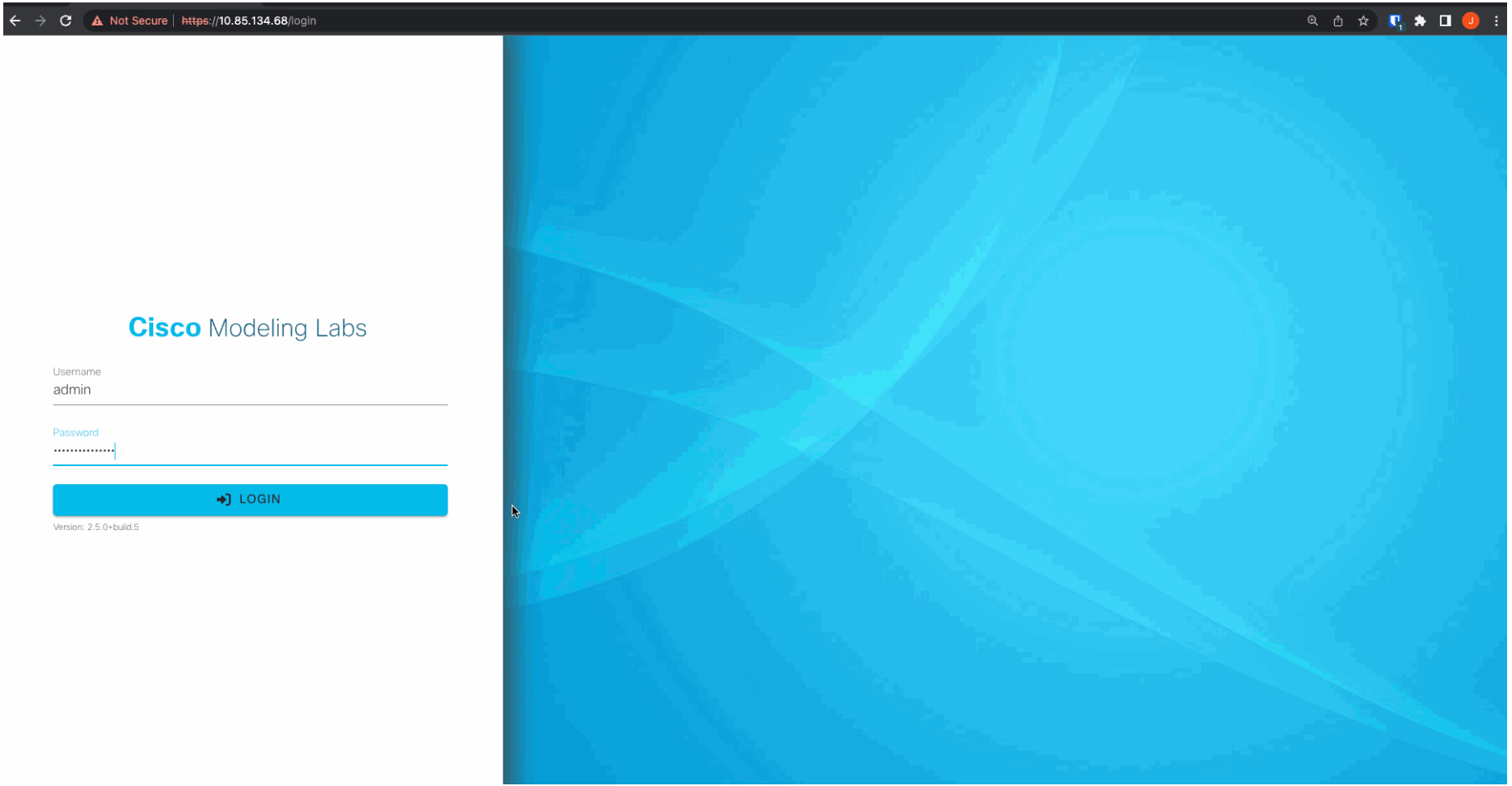

```
MM8-C9KV-7#sh ver
Cisco IOS XE Software, Version 2021-05-18_06.48_vikagarw
Cisco IOS Software [Bengaluru], Catalyst L3 Switch Software (CAT9K_IOSXE), Experimenta
vikagarw/git_ws/polaris_dev 100]
Copyright (c) 1986-2021 by Cisco Systems, Inc.
Compiled Tue 18-May-21 06:43 by vikagarw


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
```

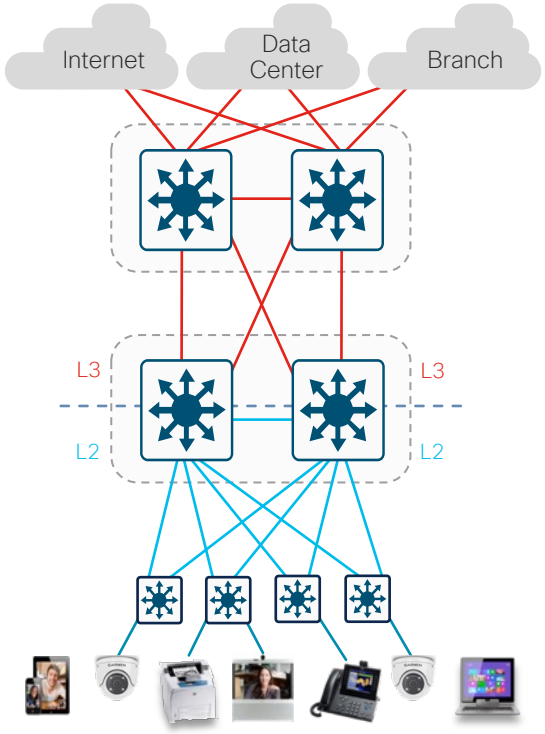
Running in RTP Lab
On UCS/ESXi
Uptime: 1+ years

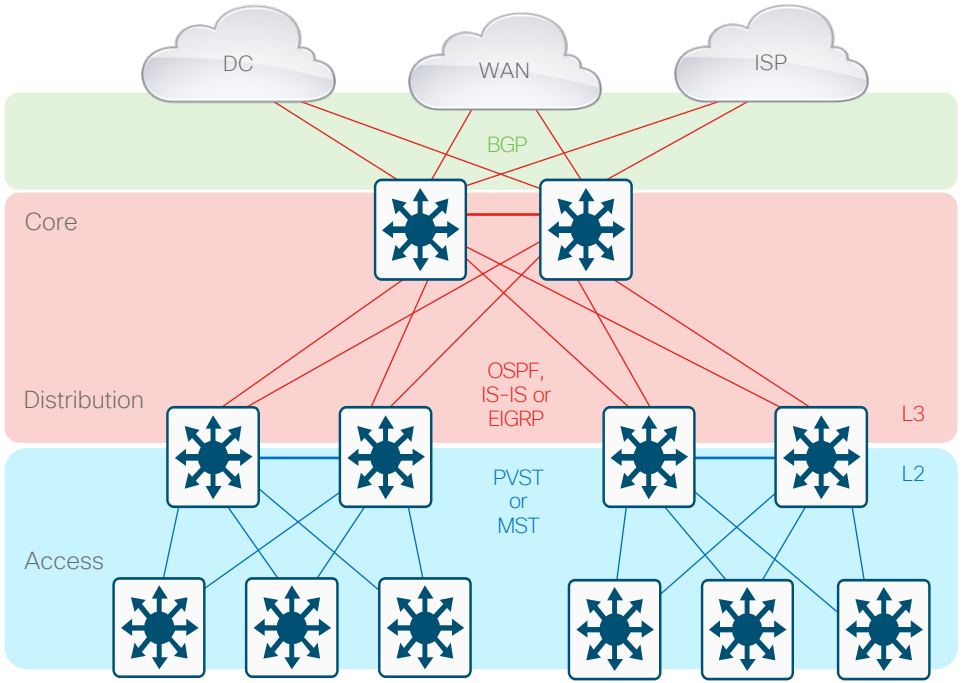
© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

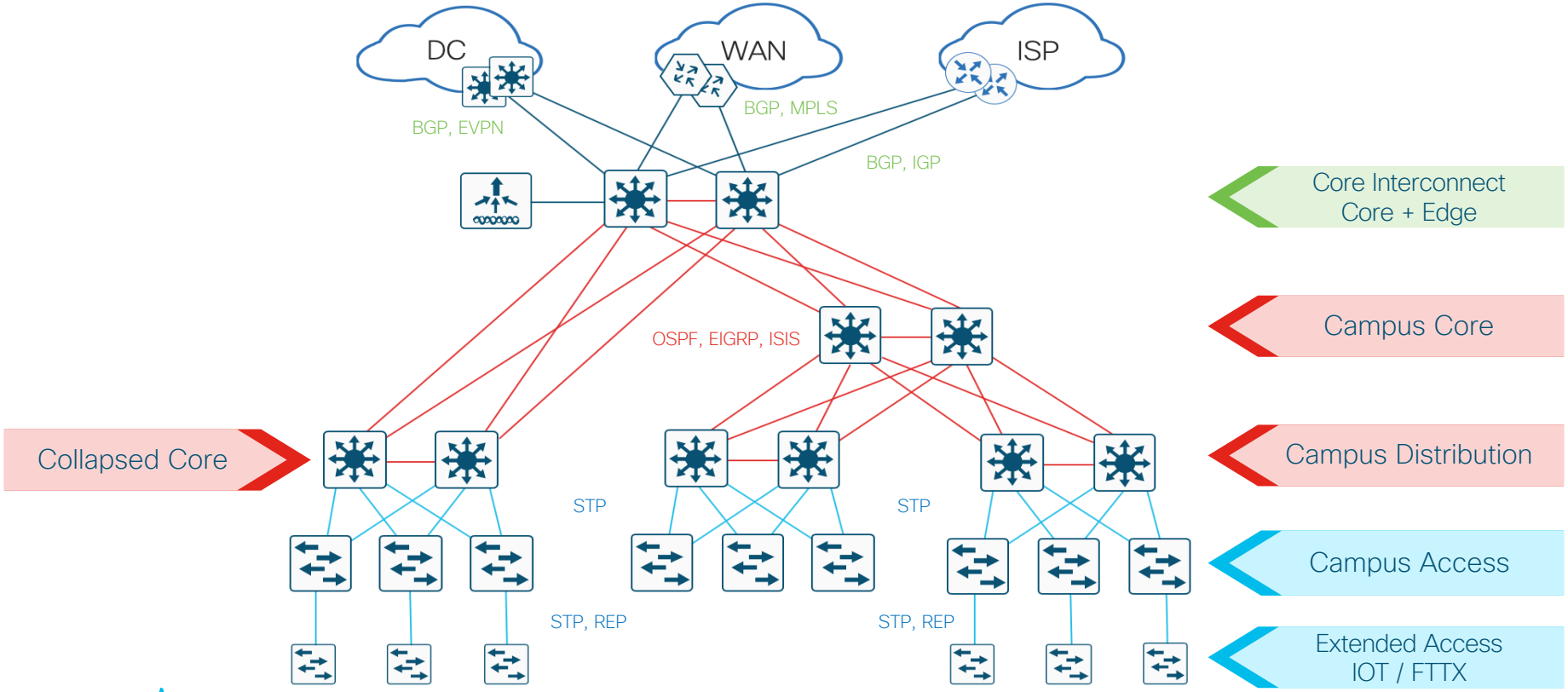
Topology





© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Campus PINs & Topology



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

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.

API White Paper

http://cs.co/apiwp







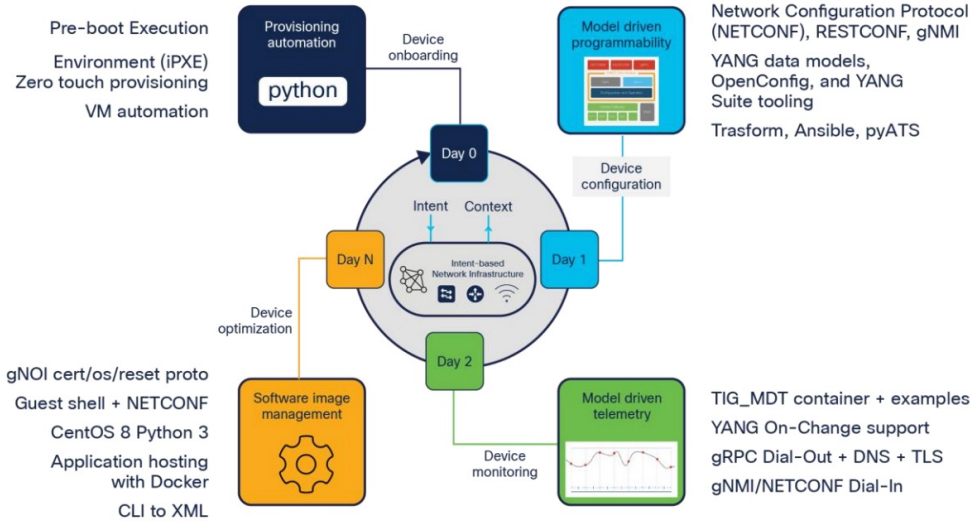
http://cs.co/apiwppdf

Programmability and auto...

Table of Contents

- Programmability and automatio...
  - Day 0: Provisioning automation
  - Day 1: Model-driven programmability
  - Day 2: Model-driven telemetry
  - Day N: Device optimization
  - Cisco IOS XE operational consistency
- Yet Another Next Generation (Y...
- Day 1: Model-driven program...
- Tooling: Cisco YANG Suite
- Day 2: Model-driven telemetry
- Day N: Device optimization
- Conclusion
- Additional resources
- Blogs

Catalyst Programmability and Automation



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

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Cisco IOS XE Programmability – Booksprint Book

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

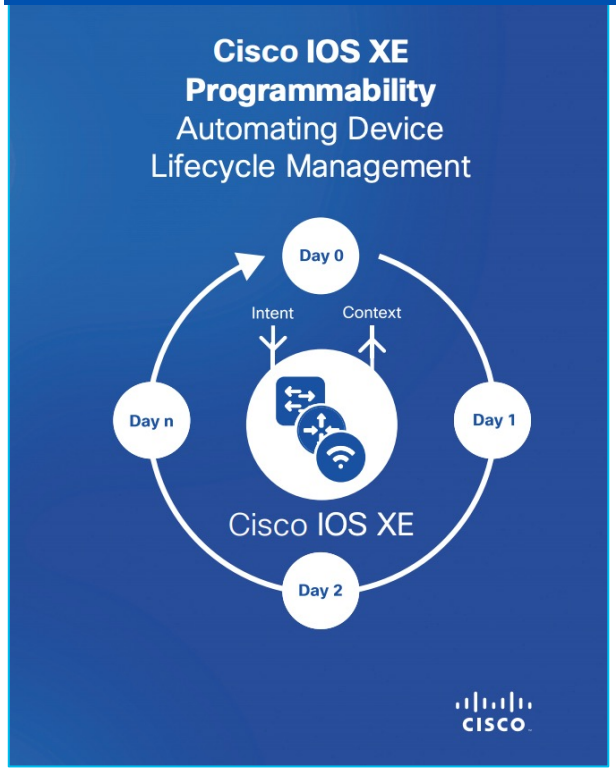


Table of Contents
- Authors
- Acknowledgments
- About this Book

Introduction
- Why Programmability Matters
- Lifecycle of Network Device Operations
- Use Cases
- Operational Approaches
- Next Steps

General Concepts
- Cisco IOS XE
- What is Programmability?
- Application Programming Interfaces (APIs)
- Programming Languages
- Structured Data
- Data Encoding Formats

Day 0 Device Onboarding
- Introduction
- Zero-Touch Provisioning (ZTP) Scenarios
- Basic ZTP Workow
- Advanced ZTP Workows
- Considerations
- Next Steps

YANG
- Overview
- YANG Concepts
- YANG Native vs Open Data Models
- YANG Data Model Highlights
- YANG Tools

Network Device APIs
- Overview
- NETCONF
- RESTCONF
- Comparison of NETCONF and RESTCONF
- Next Steps

Telemetry
- Overview
- Operational Data
- Flow Data
- Use Cases
- Subscription Tools
- Data Collectors

Python
- Overview
- Python WebUI Sandbox
- On-Box Python
- Advanced On-Box Python
- Common Issues

Guest Shell
- Introduction
- Security
- Confuration and Updates
- Resource Allocation
- Use Cases
- Next Steps

Application Hosting
- Introduction
- Cisco Application-Hosting Framework
- Containers and Virtual Machines
- Use Case
- Next Steps

Controllers
- Introduction
- Common Controllers
- Why Use a Controller?

DevOps and NetDevOps
- Introduction
- Continuous Integration and Delivery
- DevOps Tools
- Next Steps

Appendices
- Additional Resources
- Acronyms

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Programmability Configuration Guide

Book Table of Contents

- Preface
- New and Changed Information
- Provisioning
  - Zero-Touch Provisioning
  - IPXE
- Shells and Scripting
  - Guest Shell
  - Python API
  - EEM Python Module
- Model-Driven Programmability
  - NETCONF Protocol
  - RESTCONF Protocol
  - NETCONF and RESTCONF Service-Level ACLs
  - gNMI Protocol
  - gRPC Network Operations Interface
  - Model Based AAA
  - Model-Driven Telemetry
  - In-Service Model Update
- Application Hosting
  - Application Hosting
- OpenFlow
  - OpenFlow
  - High Availability in OpenFlow Mode

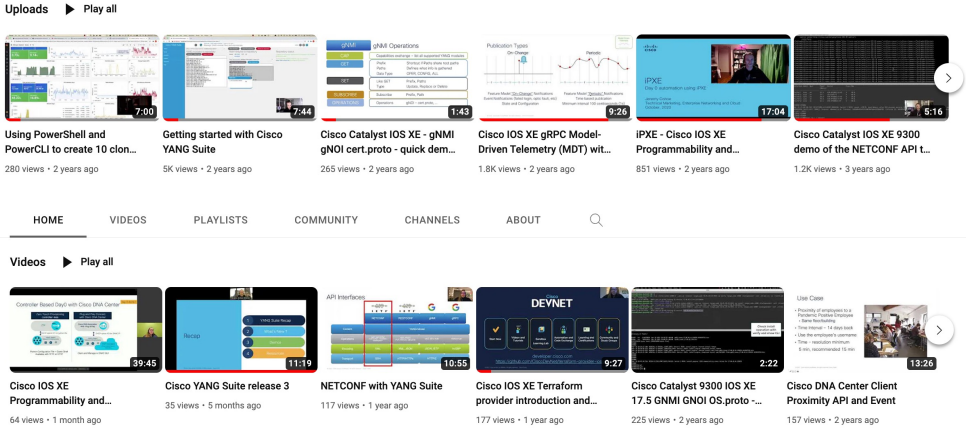


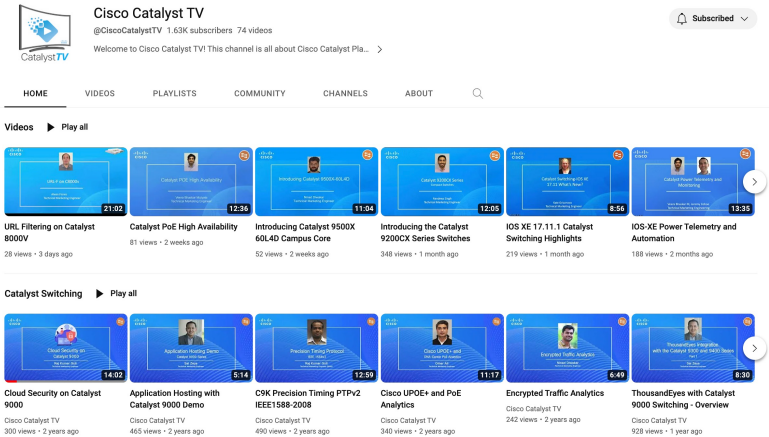
https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/1711/b_1711_programmability_cg.html

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Cisco Programmability on YouTube

https://www.youtube.com/jeremycohoe1
https://www.youtube.com/ciscoenprogrammability
https://www.youtube.com/ciscocatalysttv





© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

cisco Live!

Did you know?

You can have a
one-on-one session with
a technical expert!

Visit Meet the Expert in The HUB
to meet, greet, whiteboard & gain
insights about your unique questions
with the best of the best.



Meet the Expert Opening Hours:

Tuesday	**3:00pm – 7:00pm**
Wednesday	**11:15am – 7:00pm**
Thursday	**9:30am – 4:00pm**
Friday	**10:30am – 1:30pm**

Session Surveys

We would love to know your feedback on this session!

- Complete a minimum of four session surveys and the overall event surveys to claim a Cisco Live T-Shirt

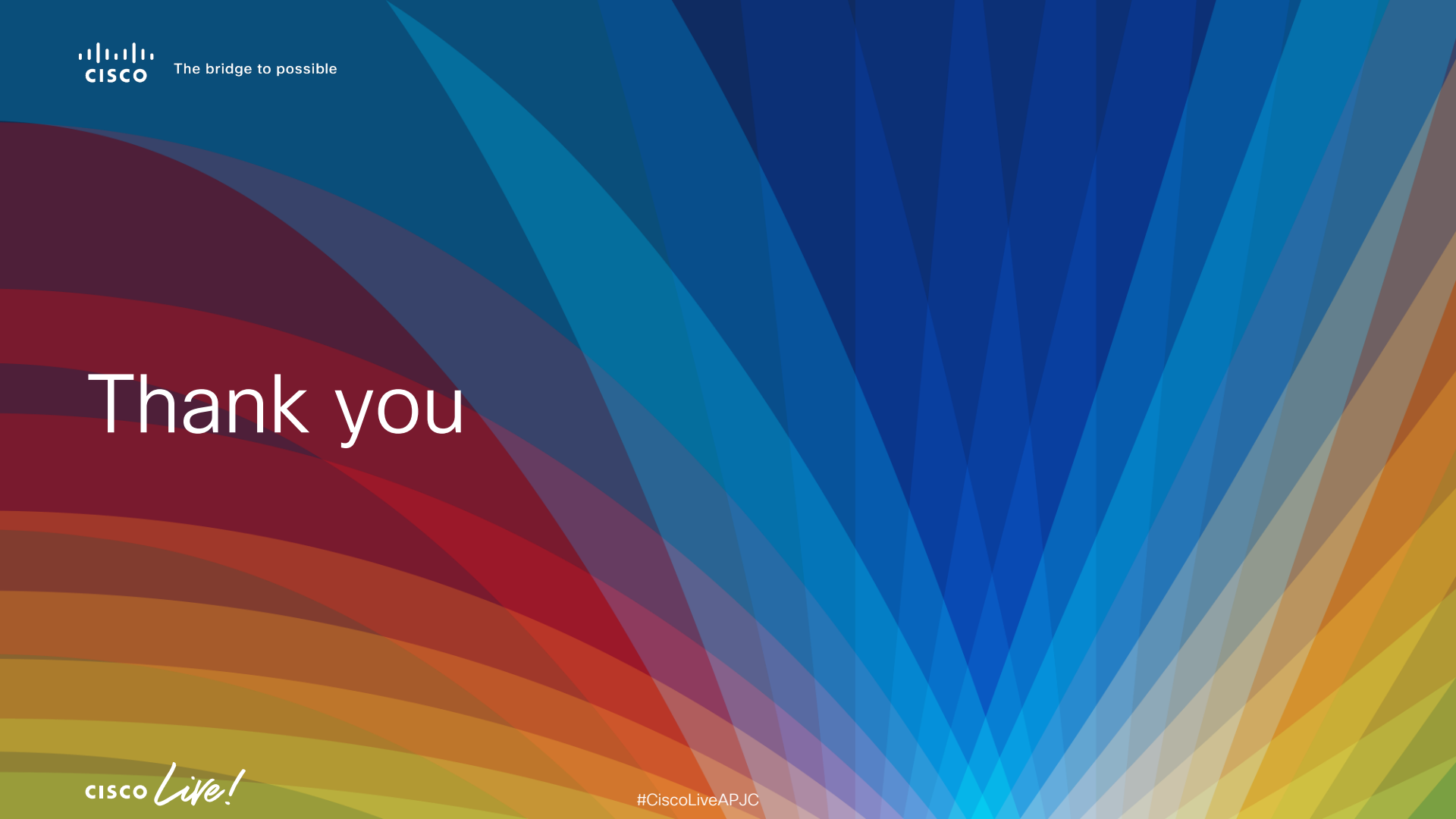


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Continue your education

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Expert 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

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public


cisco
The bridge to possible
Thank you
cisco Live!
#CiscoLiveAPJC



Let's go