

CISCO *Live!*

GO BEYOND

#CiscoLiveAPJC





Lab Up for Success with Cisco Modeling Labs

Ralph Schmieder, Senior Tech Lead, CCIE #9680
@rschmied_cisco
BRKCRT-2059

Cisco Webex App

Questions?

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

How

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

Webex spaces will be moderated by the speaker until November 15, 2024.



<https://ciscolive.ciscoevents.com/ciscolivebot/#BRKCRT-2059>



Hey ChatGPT, how do I pass my Cisco certification exam?

Hey ChatGPT, how do I pass my Cisco certification exam?

-  Passing a Cisco certification exam requires a combination of study, hands-on **practice**, and effective preparation...

Hands-on practice:

- Set up a home lab or use online simulators to gain hands-on experience with Cisco devices. Practical experience is crucial for reinforcing theoretical knowledge and improving your troubleshooting skills.

Practice!

Get the goods

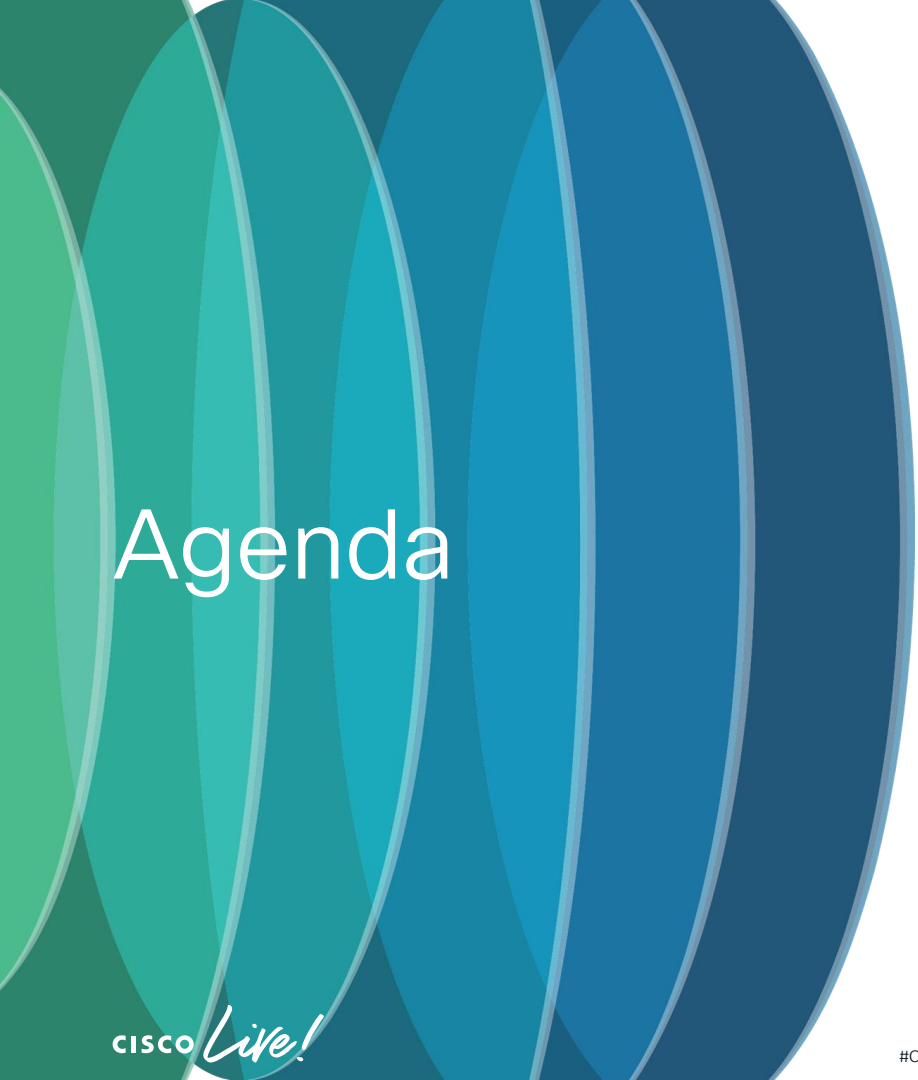
See this icon



- Examples and code available on GitHub

<https://github.com/CiscoLearning/ciscolive-brkcrt-2059>





Agenda

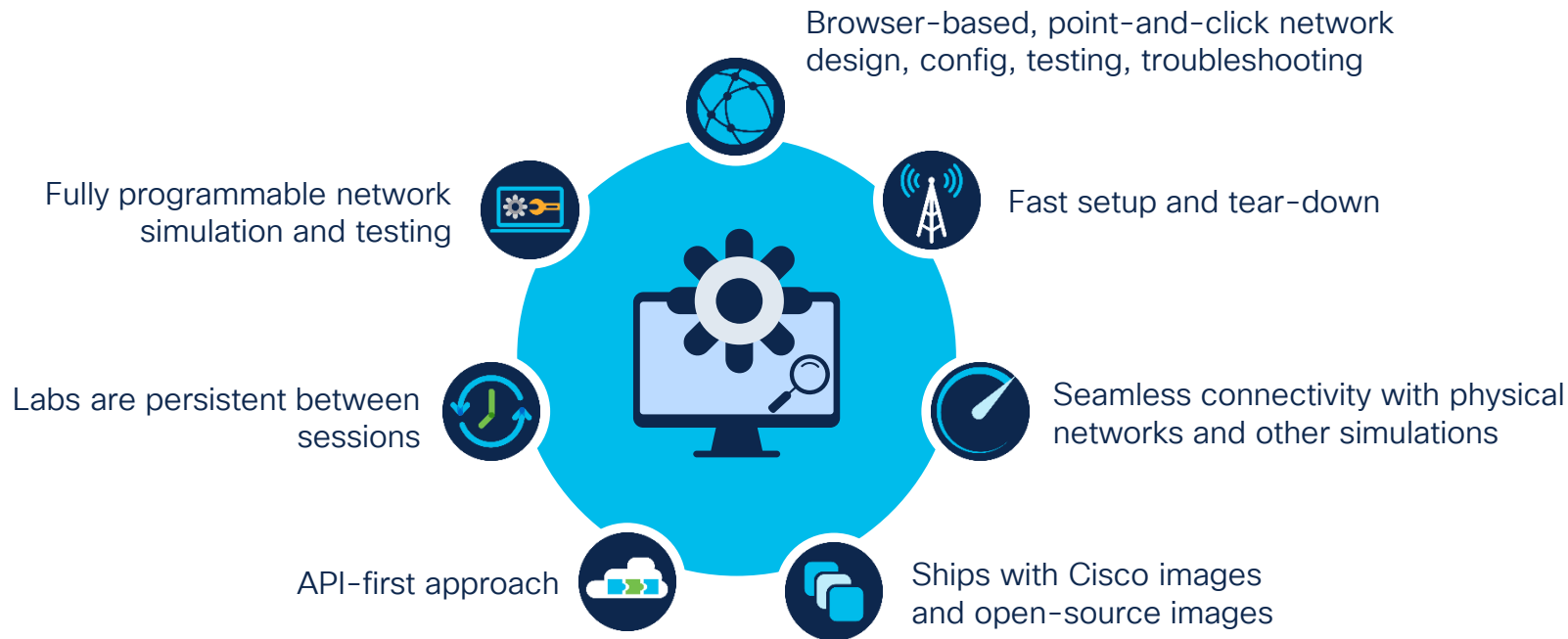
- Introduction to CML
- Advanced CML features
 - PATty
 - External connectivity
 - Day 0 configs
 - Packet capture (and link conditioning)
 - Traffic generation
- Making friends with exam topics
- Sample labs

Introduction to CML



Meet Cisco Modeling Labs

Your lab anywhere



Flavours of CML

Personal

- Single user
- Supports up to 20 concurrent running nodes

Personal Plus

- Single user
- Supports up to 40 concurrent running nodes

Enterprise

- Multi-user
- Clusterable
- Supports up to 320 concurrent running nodes

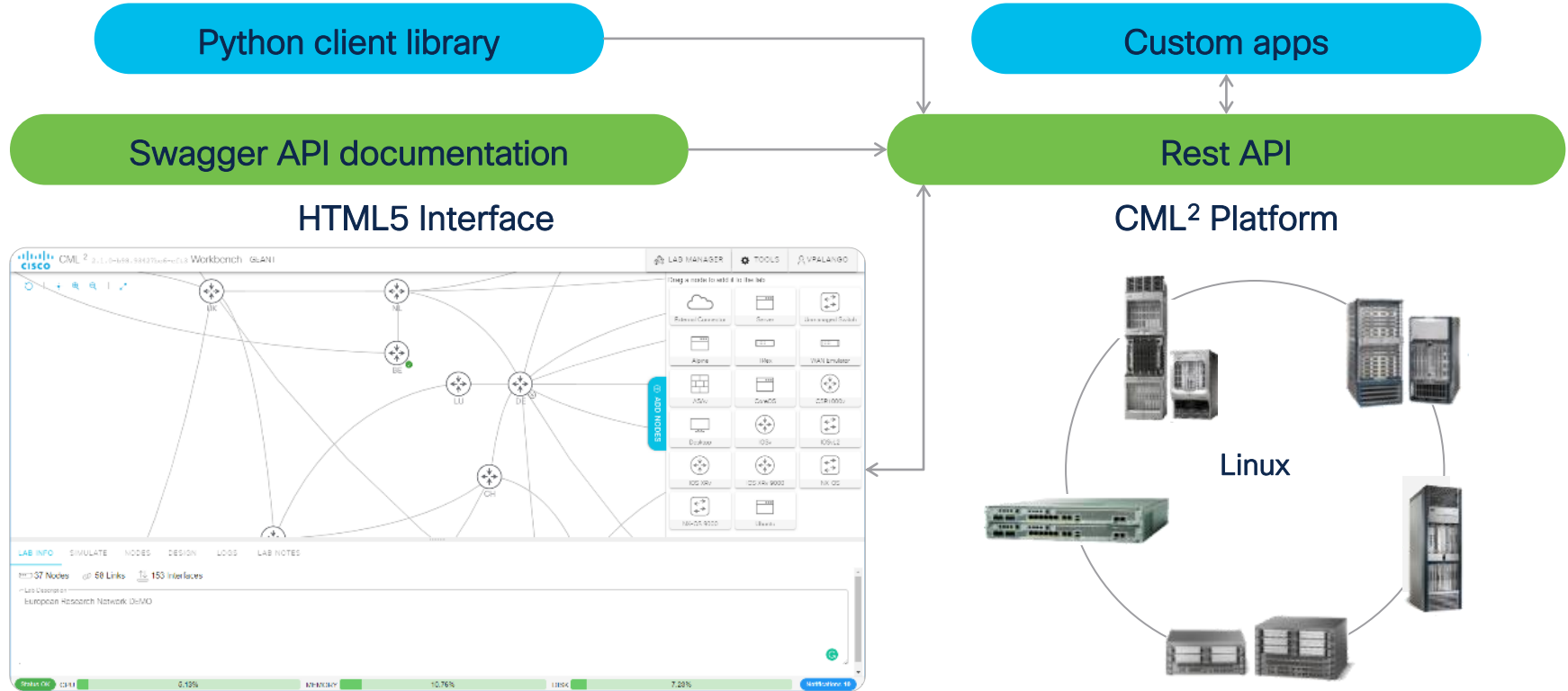
Education

- Multi-user
- Clusterable
- Licensed based on number of students

CML requirements

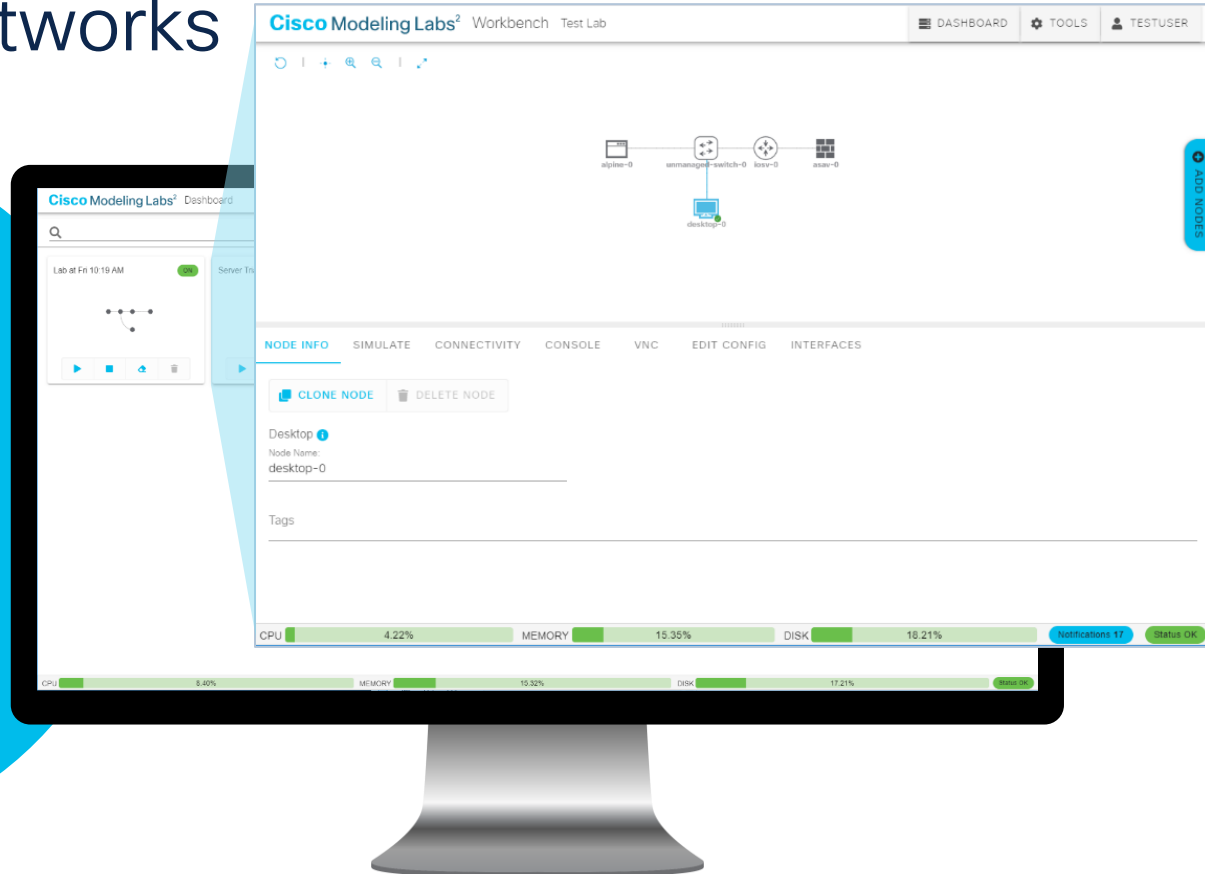
- Intel x86_64 machine
 -  AMD does work, but some device types (e.g., XR9Kv may not run)
- Number of CPUs/cores and memory will vary based on the node types and count you want to run
- Installation
 - Bare metal (i.e., directly on a physical machine)
 - VMware
 - Workstation
 - Fusion
 - ESXi

Architecture overview



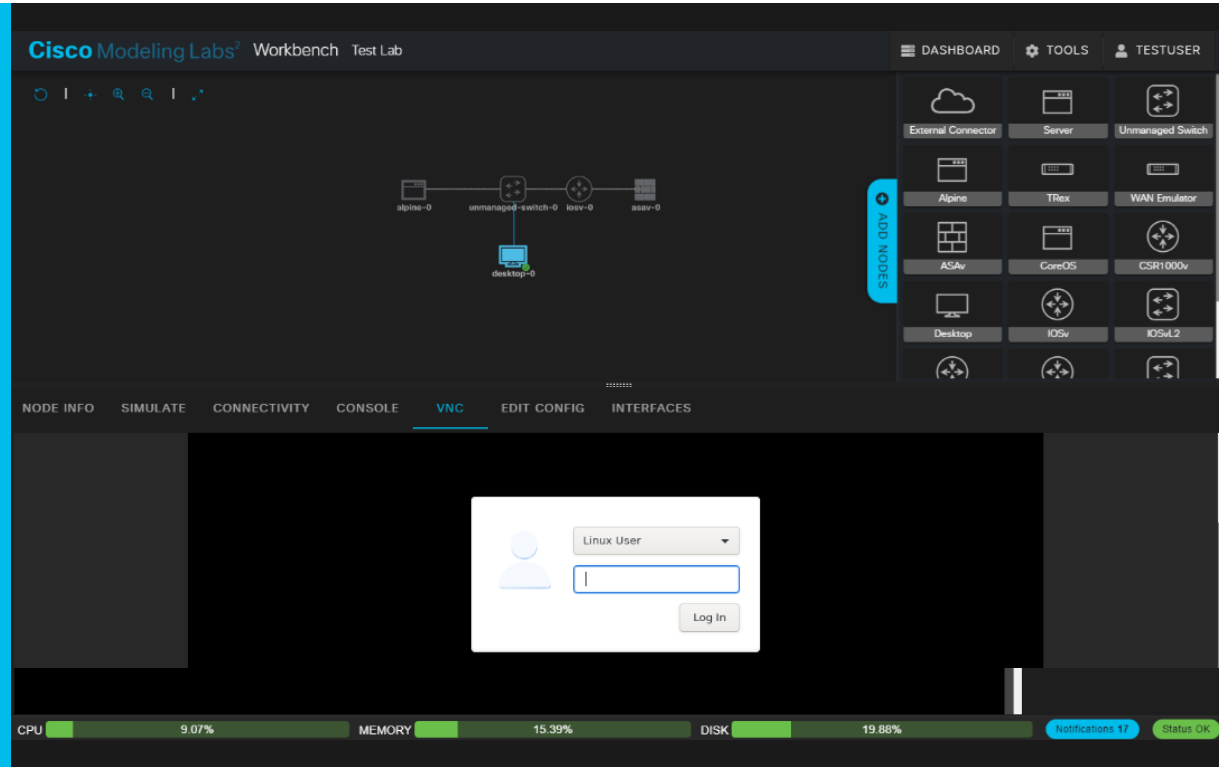
Labs are your networks

- Each lab is its own virtual network
- You can have independent networks or interconnected networks
- You can start up or shut down whole labs or individual nodes



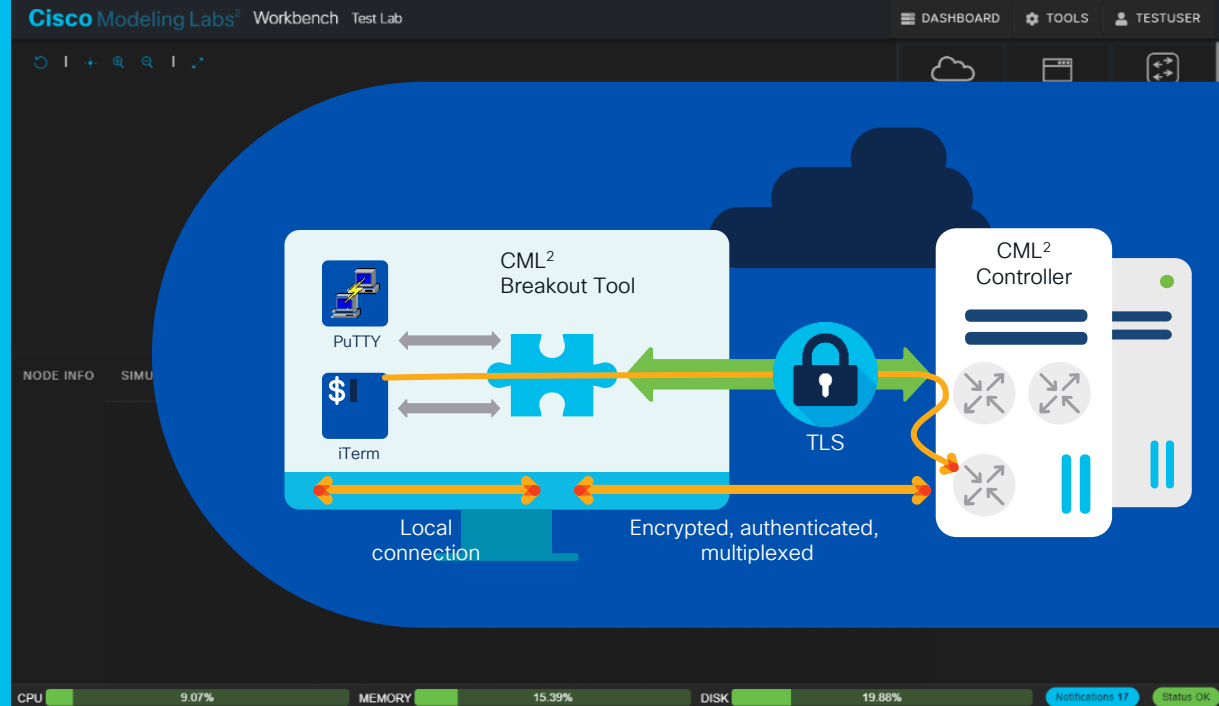
The canvas and palette – the art of network design

- Drag nodes from the palette, and drop them anywhere on your diagram
- Connect nodes by dragging a link between them
- Use graphic or command-line consoles
- Add, remove, and configure nodes in running labs



Access simulated devices many ways

- Use the Breakout tool or PuTTY to map devices to any port on your underlying Cisco Modeling Labs server
- Use external connectors to attach your entire lab to your physical network
- Use the Open Source cmlutils app to easily access devices from outside your Cisco Modeling Labs system



Practice on the same kit

Question

Refer to the topology. Using the network 192.168.100.0/24, *configure* subnetting on interfaces Gi1, Gi2, and Gi3 of R1 to accommodate the client requirements.

The performance-based *labelets* use CML.

You'll be practicing on the same kit.

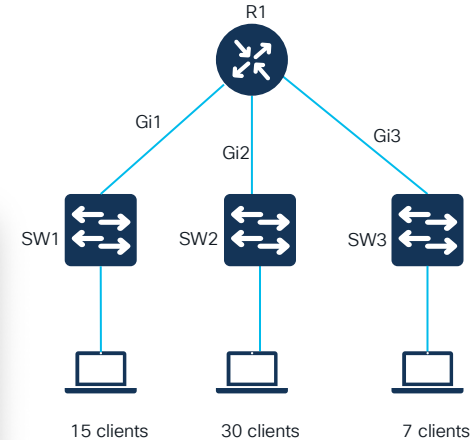


```
ssh R1
R1#show run int gi3
Building configuration...

Current configuration : 119 bytes
!
interface GigabitEthernet3
 no ip address
 shutdown
 negotiation auto
 cdp enable
 no mop enabled
 no mop sysid
end

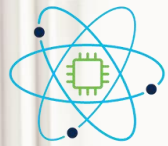
R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#
R1(config)#
R1(config)#
R1(config)#
R1(config)#
R1(config)#
R1(config)#
R1(config)#
```

Topology



CML CCNA Topology Contest

The top three entries with the most votes will win a 1-year CML-
Personal Plus subscription.



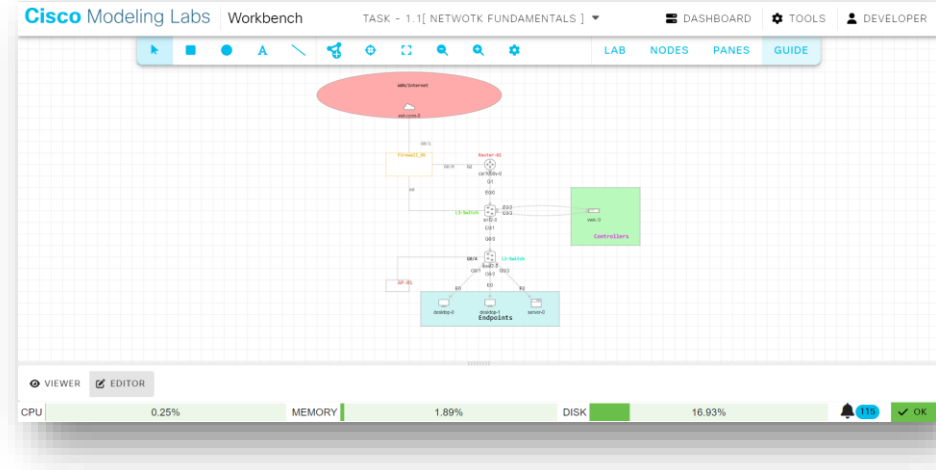
Powered by
Cisco Modeling Labs

CISCO *Live!*



Now with sample labs!

- We recently ran a contest to collect CML labs supporting CCNA exam topics
- We received 19 valid submissions!
- Kick-start your studying at <https://github.com/CiscoDevNet/cml-community>





Practice, practice, practice

Labbing to prepare for cert exams

Quickly build and destroy
virtual networks without impacting
a physical network

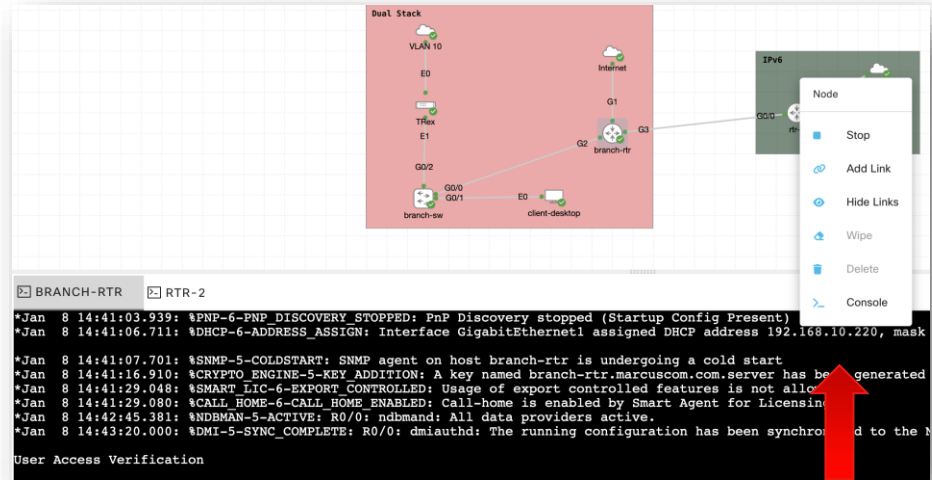
Share your network configurations
with your friends and co-workers
to test each other's skills

Advanced CML Features



PATty

- The console opens from a tab in the lower lab pane
- You can organise multiple consoles next to each other in the browser
- When studying, you might want to use your own terminal app and have multiple consoles handy
- You may also want to allow network access to services through NAT
- Enter PATty



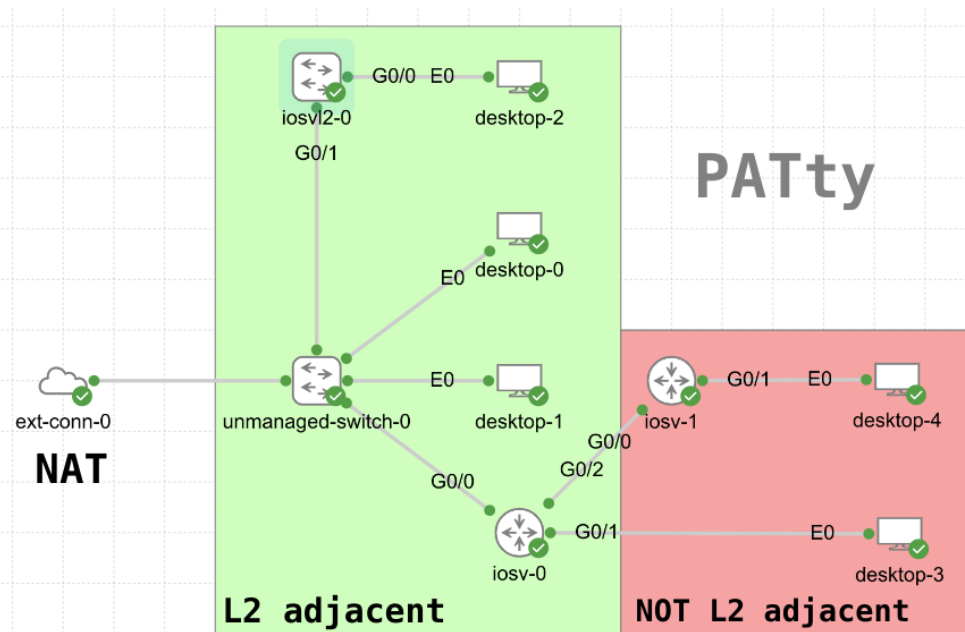
PATty – requirements

- **Service** must be configured and enabled (it's disabled by default for security reasons)
- **Nodes inside a lab** which should be made accessible via PAT must be L2 connected to an external connector in NAT mode
- These nodes **must be "seen" by the IP address snoop**er for the NAT bridge running on the CML controller
- Always works for device access (serial, VNC)



PATty – requirements for port access

- ext-conn is configured for NAT 
- desktop-0, desktop-1 and iosv-0 are layer 2 connected to ext-conn 
- desktop-2 is layer-2 connected via iosv12-0 to ext-conn 
- iosv-1, desktop-3 and desktop-4 are on different subnets, not directly accessible from the CML host 



PATty – enable and configure the service

- Config file is in
/etc/default/patty.env
- POLL time can be reduced if
only user and impatient (a)
- Remove # from OPTS to
enable device support (b)



```
sysadmin@cml-26:~$ cat /etc/default/patty.env
# CML2 / PATty options, sourced by systemd
# service unit virl2-patty.service
#
# this needs to go into /etc/default
#
# log level, use debug/info/warn/error
# recommended is warn
LEVEL="warn"

# poll intervall in seconds, recommended 60
POLL=60

# additional options
# the below allows access to devices, both serial
# and VNC. To enable these, uncomment the line
# or edit to only allow one of them, if needed
# OPTS="-vnc -serial"
sysadmin@cml-26:~$
```

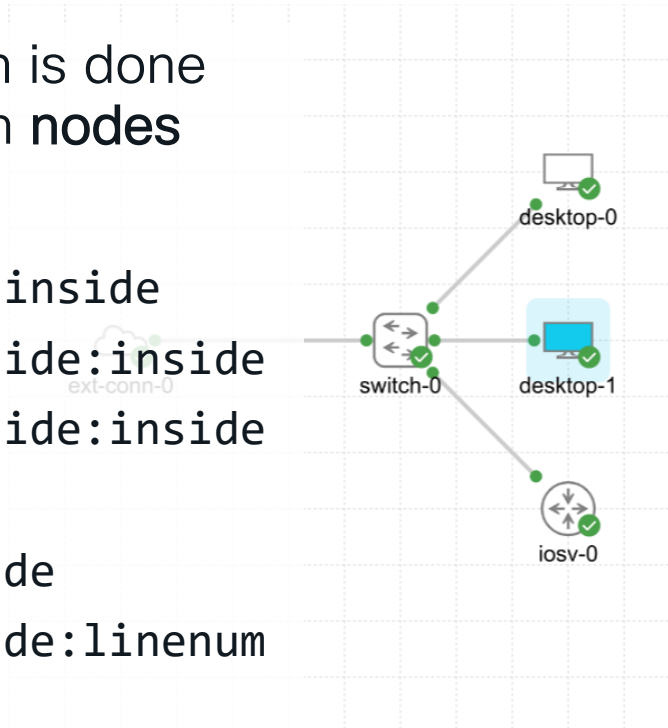
PATty – how to configure nodes

- Configuration is done using **tags** on **nodes**
- Available are
`pat:outside:inside`
`pat:tcp:outside:inside`
`pat:udp:outside:inside`

`serial:outside`

`serial:outside:linenum`

`vnc:outside`



The screenshot shows the 'SETTINGS' tab in the PATty interface. At the top, there are tabs for 'SETTINGS', 'CONNECTIVITY', 'CONFIG', and 'INTERFACES'. Below these are four action buttons: 'STOP' (blue square), 'WIPE' (trash icon), 'CLONE' (blue square with plus), and 'DELETE' (trash icon). The main section is titled 'Desktop' with an information icon. It shows the 'Node Name' as 'desktop-1' with a copy icon and a table with columns 'NAME' and 'ID'. Below this, the 'Tags' section shows two tags: 'pat:tcp:2022:22' and 'vnc:5900', each with a close icon. At the bottom, there is an 'Image Definition' section with the text 'Image Definition Automatic' and a dropdown menu with an information icon.

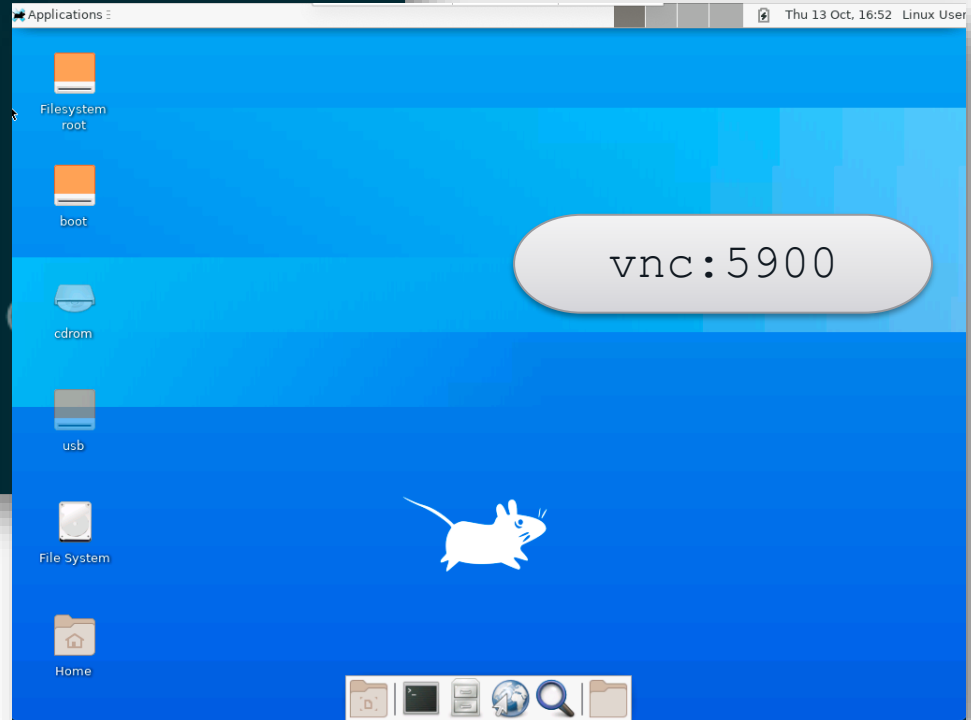
Using PATty capabilities

serial:9004

```
⇒ telnet localhost 9004
Trying ::1...
Connected to localhost.
Escape character is '^]'.

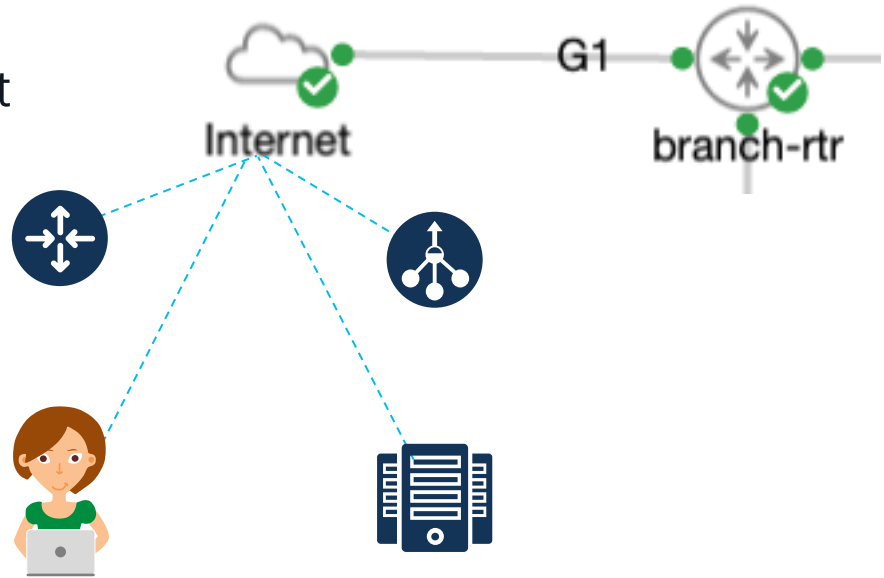
Welcome to Alpine Linux 3.12
Kernel 5.4.43-1-virt on an x86_64

localhost login: █
```



External connectivity

- By default, all labs in CML are isolated from each other and the outside world
- *External connectivity* changes that
- Connect your labs to the world
 - NAT (L3) – think home networking
 - Bridge (L2) – share the segment on which the CML server sits
 - Custom (L2) – connect your labs to *any* layer 2 segment



- Download files (e.g., patches, scripts, etc.)
- Manage devices with external apps (e.g., NMS, controller, etc.)
- Augment your physical home lab with CML virtual nodes



Augment Your Home Lab



Cisco 1941



Type of External Connectivity:

[illegible][illegible]

Day 0 configs

As you make changes, you can pull in the latest config

Day 0 configs are saved to a lab's YAML file so you can download and share



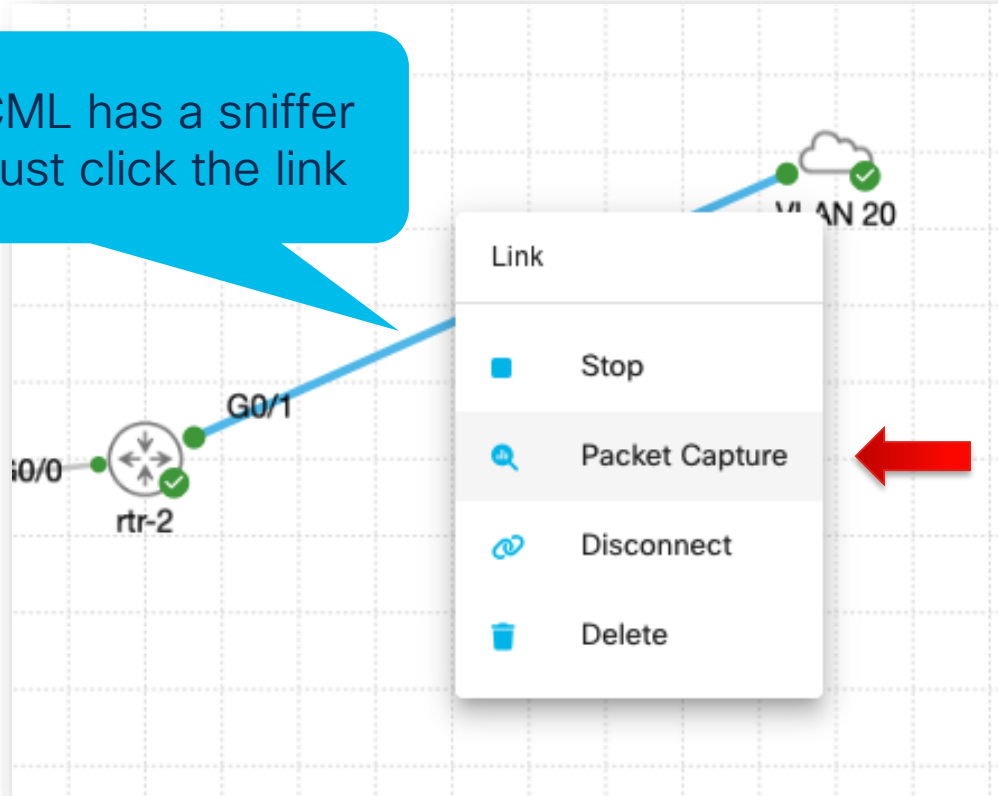
Config to load the *first* time the device boots

Fun with links



Packet captures everywhere!

Every link in CML has a sniffer built right in; just click the link



Packet capture features

LINK INFO

SIMULATE

PACKET CAPTURE

▶ START

🗑️ CLEAR

⬇️ DOWNLOAD

⚙️ SETTINGS

🔍 Search

Max Packets:
50

Max Time:

BPF Templates:

DHCP

Spanning Tree

ip (src)

ip (dst)

ethernet (src)

ethernet (dst)

BPF:
udp port 68 or port 67



Search for packets

Download as a PCAP file

Berkeley Packet Filtering

Packet capture features

[LINK INFO](#) [SIMULATE](#) [PACKET CAPTURE](#)

[▶ START](#) [🗑️ CLEAR](#) [📄 DOWNLOAD](#) [⚙️ SETTINGS](#)

No.	Time	Source ▼	Destination ▼	Protocol ▼	Length ▼	Info ▼
Click start to begin capturing data						

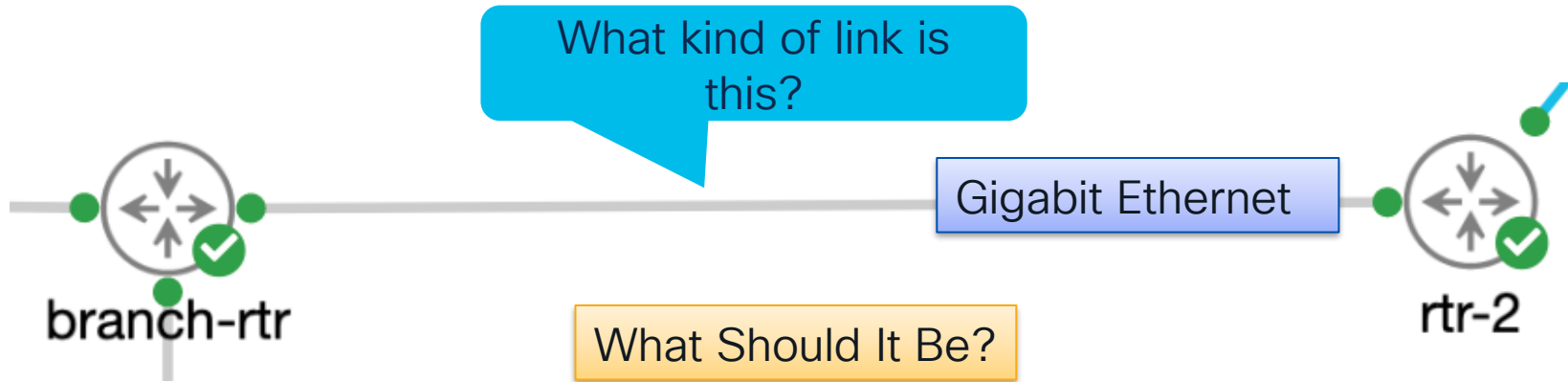
Capture not running

Rows per page: 10 ▼ - < >

Packet Details

No packet selected

Not all links are equal





Public service announcement: dataplane gotchas

- All links on CML are Gigabit Ethernet in *type*
 - N9Kv shows them as being 10 Gbps
- You cannot have interfaces named TenGE, HunGE, etc.
 - This is more a limitation of the underlying network elements
 - The Cat9Kv *does* allow you to change to TenGE
- While some devices like the Linux nodes can achieve ~ 1 Gbps speed, most have a rate-limited dataplane
 - See <https://developer.cisco.com/docs/modeling-labs/#!vm-images-for-cml-labs> for per-node dataplane limitations

Link conditioning: make a link “real”



RTR-EUR-01

rtr-eur-01#

LINK INFO SIMULATE PACKET CAPTURE

STOP DISCONNECT

Link Statistics

Node	Interface	Running ¹	Packets	Bytes
rtr-eur-01	GigabitEthernet0/0	☒	67	11470
rtr-us-01	GigabitEthernet0/0	☒	65	11337

Link Condition

Bandwidth kbps Latency ms Jitter ms Loss %

APPLY RESET

Note: Link condition settings are not persisted in the topology data.

CPU

36.50%

MEMORY

36.97%

DISK

78.69%



OK

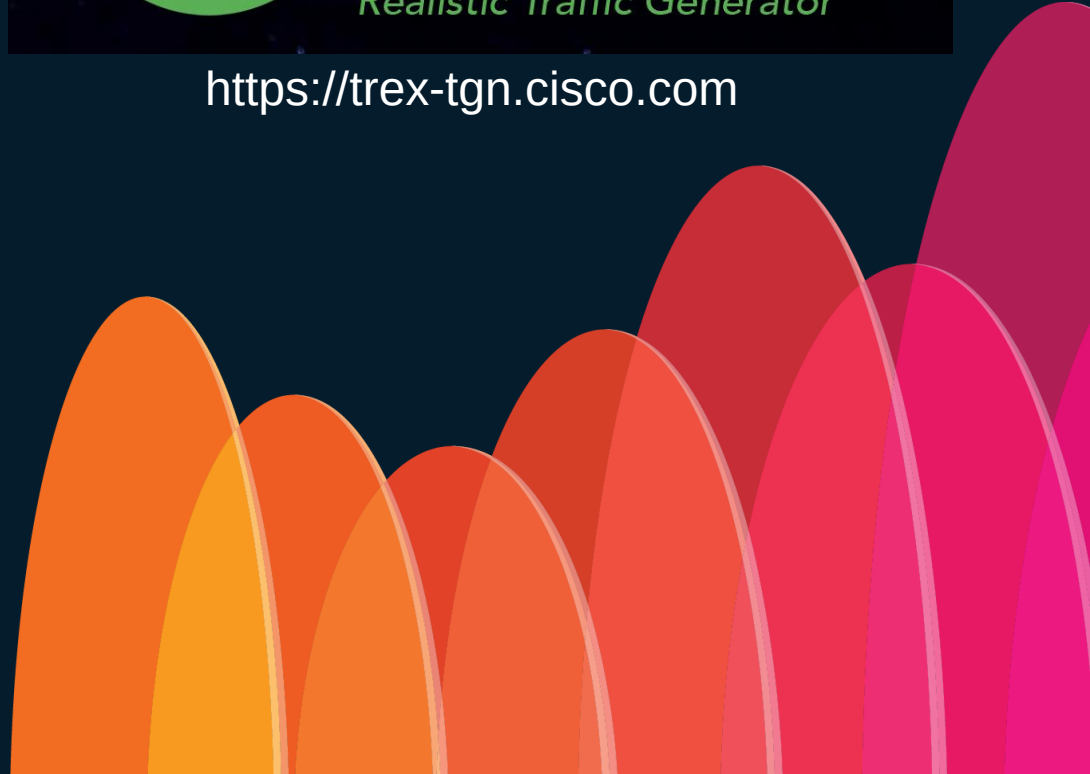


<https://trex-tgn.cisco.com>

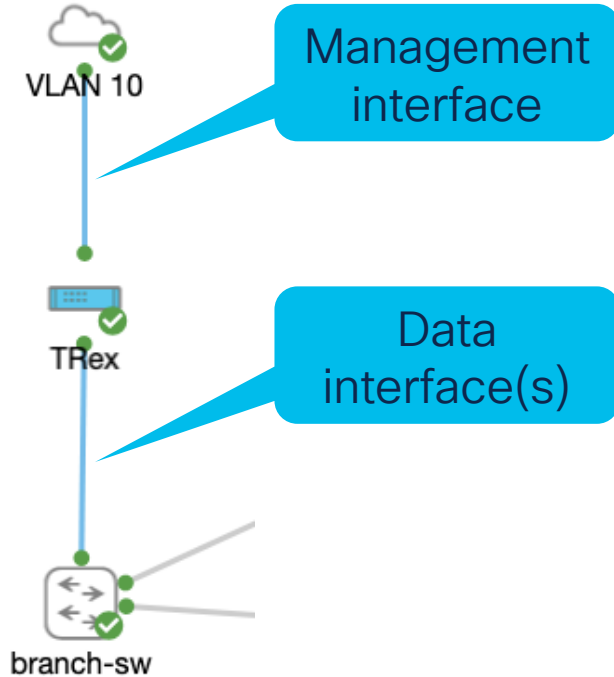
Traffic Generation

Ping is great, but what if you want...more?

CISCO *Live!*



Using TRex



```
GUI https://github.com/cisco-system-traffic-generator/trex-stateless-gui
eth0 IP addresses
# ### ### # ### ### # ### ### ### # ### #
## # # # ## # # # ## # # # # # # #
# ### ### # ### ### # # # ### ### # ### ###
# # # # # # # # # # # # # # # # #
### ### ### # ### ### ### # ### ### # ### ### # ### #

192.168.10.220/24
2600:1700:b00:b239:5054:ff:felf:26b7/64
fe80::5054:ff:felf:26b7/64

[trex] 0:SHELL* 1:SCAPY 2:TRex- "localhost" 18:56 26-Oct-22
```

Control+B <num> to jump terminals

The TRex GUI

Action buttons

Data (or dump) Port(s)

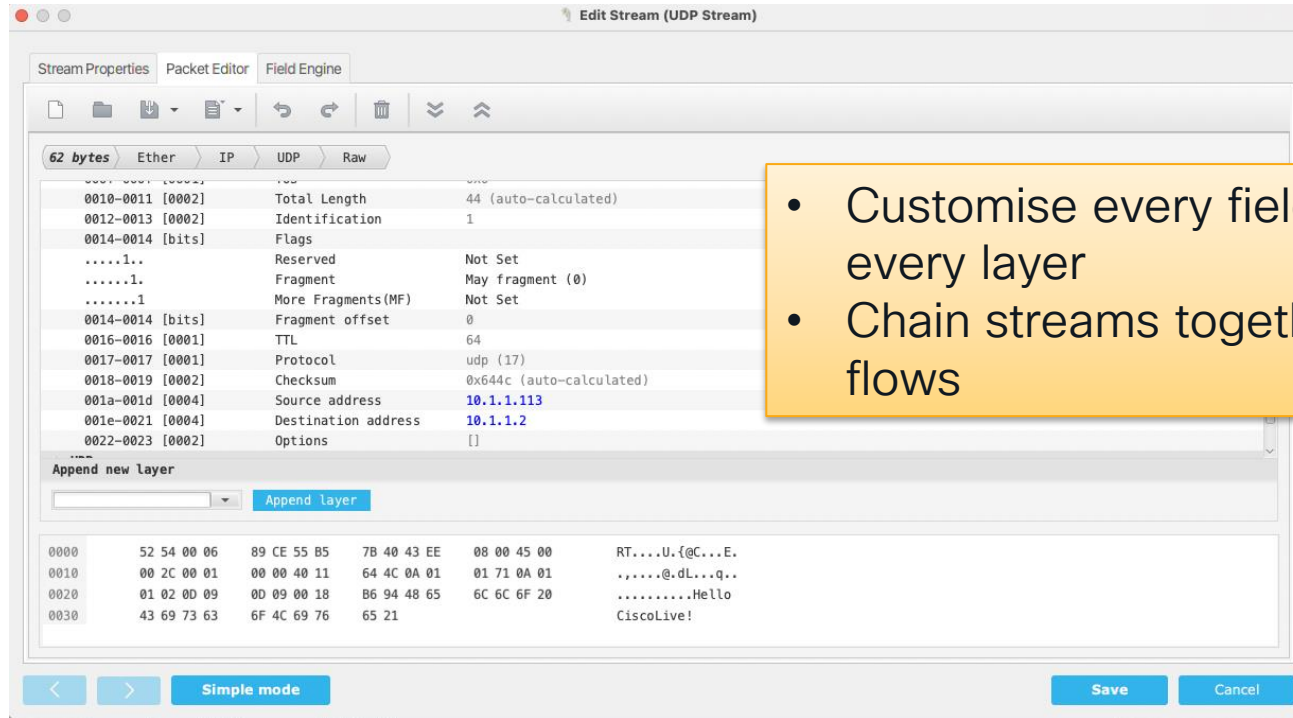
Customisable streams

The screenshot displays the TRex GUI interface. At the top, there is a menu bar with options: File, Traffic Profiles, Stats, Capture, TRex, and Help. Below the menu is a toolbar with various icons for play, pause, stop, and other actions. The main area is divided into two panels. The left panel shows a tree view with 'Trex-localhost' expanded, revealing 'Port 0 (jclarke)' and a 'Profile' button. The right panel, titled 'Profile details', shows the selected profile 'UDP Flow.yaml'. It includes a 'Bandwidth' slider set to 100, and radio buttons for '% L1', 'L1 bps', 'L2 bps', and 'pps' (selected). Below these are input fields for values: 0.0, 3.4K, 2.6K, and 5. A 'Duration' field is set to 0. At the bottom of the right panel, there are buttons for '+ Build Stream', 'Edit Stream', 'Duplicate Stream', 'Delete Stream', 'Import Pcap', 'Export Pcap', and 'Export To Yaml'. Below these buttons is a table with the following data:

Index	Name	Packet Type	Length	Mode	Rate	Next Stream
1	UDP Stream	Ethernet/IPV4/UDP	62	Continuous	10 pps	-

At the bottom of the GUI, there is a 'Log View' section showing a series of log messages. A 'Copy to clipboard' button is located to the right of the log view. The status bar at the bottom right indicates 'Connected' with a green dot.

Building streams



- Customise every field of every layer
- Chain streams together into flows

Viewing the fruits of your labour

No.	Time	Src MAC	Dest MAC	Source	Destination	Protocol	Length	Info
1	0.000000	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
2	0.009647	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
3	0.170074	RealtekU_06:89:ce	Broadcast	RealtekU_06:89:ce	Broadcast	ARP	42	Who has 10.1.1.113?
4	0.199286	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
5	0.205014	RealtekU_1e:0c:16	Spanning-tree-(for-brid...	RealtekU_1e:0c:16	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/
6	0.301245	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
7	0.399614	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
8	0.500175	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
9	0.599750	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
10	0.701559	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
11	0.799846	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
12	0.900547	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
13	0.999956	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
14	1.100590	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
15	1.200497	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
16	1.209665	RealtekU_06:89:ce	Broadcast	RealtekU_06:89:ce	Broadcast	ARP	42	Who has 10.1.1.113?
17	1.301033	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
18	1.402244	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
19	1.501441	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
20	1.600853	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
21	1.700612	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
22	1.799872	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
23	1.902575	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
24	1.999926	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
25	2.100792	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
26	2.201843	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
27	2.300404	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
28	2.314849	RealtekU_1e:0c:16	Spanning-tree-(for-brid...	RealtekU_1e:0c:16	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/
29	2.400524	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!
30	2.501070	55:b5:7b:40:43:ee	RealtekU_06:89:ce	10.1.1.113	10.1.1.2	Syslog	60	Hello CiscoLive!

> Frame 25: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0

> Ethernet II, Src: 55:b5:7b:40:43:ee (55:b5:7b:40:43:ee), Dst: 01:00:00:00:00:00

> Internet Protocol Version 4, Src: 10.1.1.113, Dst: 10.1.1.2

> User Datagram Protocol, Src Port: 3337, Dst Port: 5432

0000 52 54 00 06 89 ce 55 b5 7b 40 43 ee 08 00 45 00 RT...U: @C...E:
0010 00 2c 00 01 00 00 40 11 64 4c 0a 01 01 71 0a 01 ,...@ dL...q...
0020 01 02 0d 09 0d 09 00 18 b6 94 48 65 6c 6c 6f 20Hello
0030 43 69 73 63 6f 4c 69 76 65 21 00 00 CiscoLive!e!..

Syslog message (syslog), 16 bytesPackets: 537 · Displayed: 537 (100.0%)Profile: Default

Join at
slido.com
#1706 743

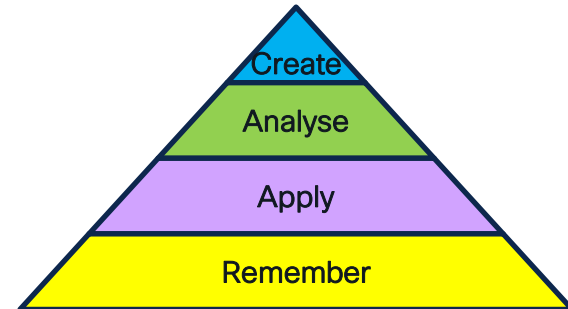


Making Friends with Exam Topics

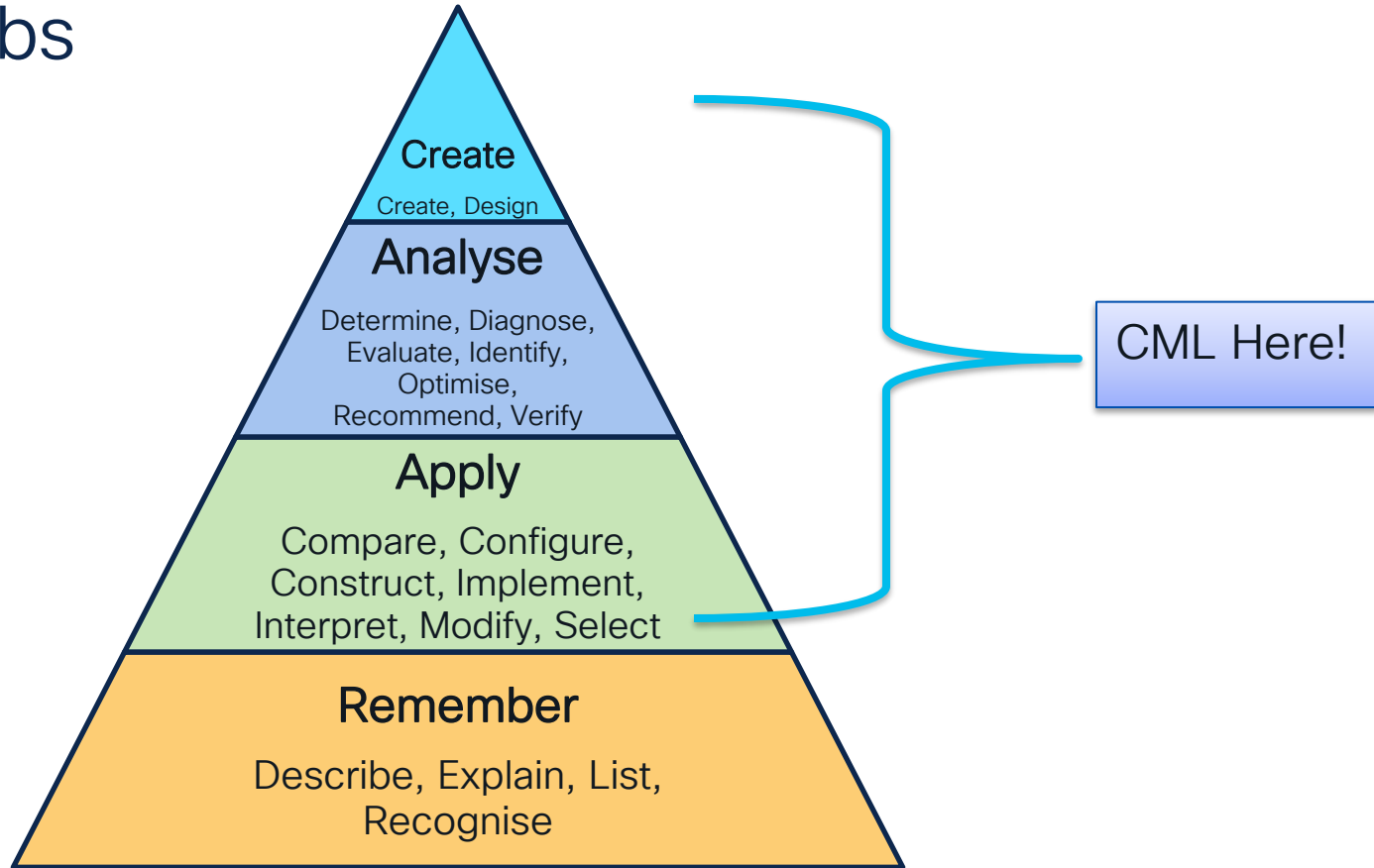


How to read exam topics

- We use Bloom's Taxonomy to convey the *cognitive complexity* of tasks
- Cognitive complexity is a hierarchy of thought processes used to order and classify objectives (e.g., blueprint tasks).
- Levels of cognitive thinking are ordered from low to high using descriptive **verbs** aligned to the taxonomy.
 - low levels of the taxonomy represent objectives that are simpler or less difficult
 - high levels of the taxonomy represent objectives that are more abstract and complex
- Each level of the hierarchy builds on the previous level(s), incorporating the previous level(s) of thought to proceed to the higher level.



Cisco's verbs



Consider ENCOR

350-401 ENCOR Exam Topics

Exam Description

To earn your CCNP Enterprise, CCIE Enterprise Infrastructure, or CCIE Enterprise Wireless certification, you must pass the **350-401 ENCOR** exam. This exam tests your knowledge of:



Architecture



Virtualization



Infrastructure



Network Assurance



Security



Automation

Expand each item below to view related exam topics.

1.0 Architecture	15%	▼
2.0 Virtualization	10%	▼
3.0 Infrastructure	30%	▼
4.0 Network Assurance	10%	▼
5.0 Security	20%	▼
6.0 Automation	15%	▼

<https://learningnetwork.cisco.com/s/encor-exam-topics>

3.1 Layer 2

- 3.1.a **Troubleshoot** static and dynamic 802.1q trunking protocols
- 3.1.b Troubleshoot static and dynamic EtherChannels
- 3.1.c Configure and verify common Spanning Tree Protocols (RSTP and MST)

3.2 Layer 3

- 3.2.a Compare routing concepts of EIGRP and OSPF (advanced distance vector vs. linked state, load balancing, path selection, path operations, metrics)
- 3.2.b **Configure** and verify simple OSPF environments, including multiple normal areas, summarization, and filtering (neighbor adjacency, point-to-point and broadcast network types, and passive interface)
- 3.2.c Configure and verify eBGP between directly connected neighbors (best path selection algorithm and neighbor relationships)

3.3 Wireless

- 3.3.a Describe Layer 1 concepts, such as RF power, RSSI, SNR, interference noise, band and channels, and wireless client devices capabilities
- 3.3.b Describe AP modes and antenna types
- 3.3.c Describe access point discovery and join process (discovery algorithms, WLC selection process)
- 3.3.d Describe the main principles and use cases for Layer 2 and Layer 3 roaming
- 3.3.e Troubleshoot WLAN configuration and wireless client connectivity issues

3.4 IP Services

- 3.4.a Describe Network Time Protocol (NTP)
- 3.4.b Configure and **verify** NAT/PAT
- 3.4.c Configure first hop redundancy protocols, such as HSRP and VRRP
- 3.4.d Describe multicast protocols, such as PIM and IGMP v2/v3

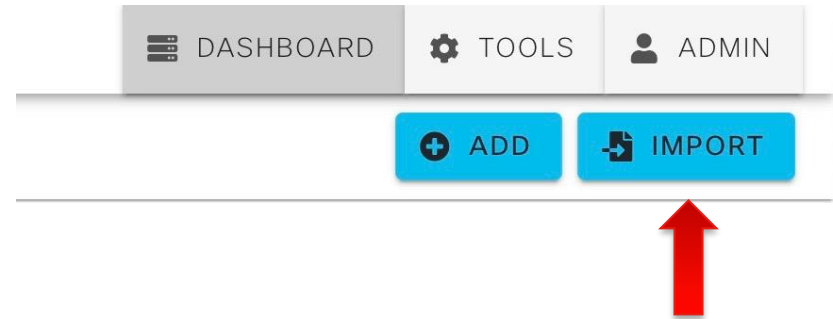
Sample Labs





Loading the sample labs

- Get the `topology.yaml` files from the respective sample lab directories in the git repository
- Import them into CML
 - **NOTE:** CML 2.5 or higher is required
- A `topology-solution.yaml` is also provided with the lab exercises completed



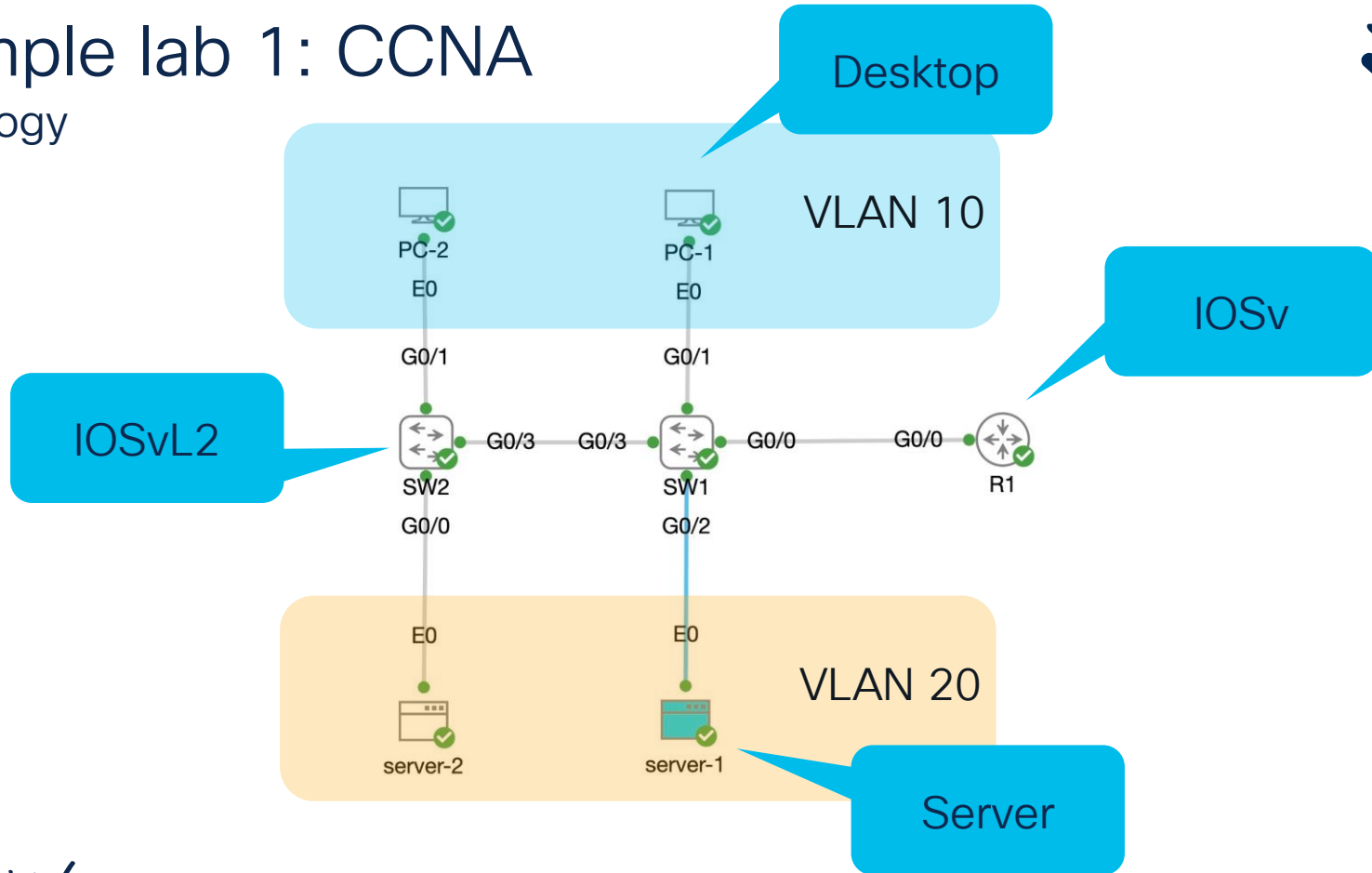
Sample lab 1: CCNA

Implement Multiple VLANs and Basic Routing Between the VLANs

- Implement VLAN assignments on two switches
- Configure a dot1q trunk between those switches
- Configure a dot1q trunk between one switch and a router that will handle the routing for the VLANs
- Configure VLAN trunking on that router to act as a gateway between VLANs
- Verify that connectivity exists between hosts in the lab on each VLAN, and that the connectivity goes through the router

Sample lab 1: CCNA

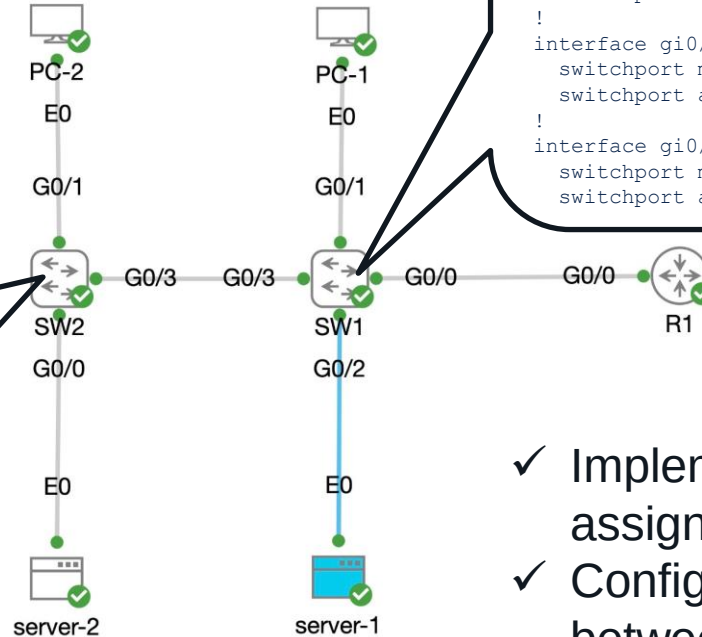
Topology



Sample lab 1: CCNA

Config

```
vlan 10
 name Users
!
vlan 20
 name Servers
!
interface gi0/0
 switchport mode access
 switchport access vlan 20
!
interface gi0/1
 switchport mode access
 switchport access vlan 10
!
interface gi0/3
 switchport trunk encap dot1q
 switchport mode trunk
```

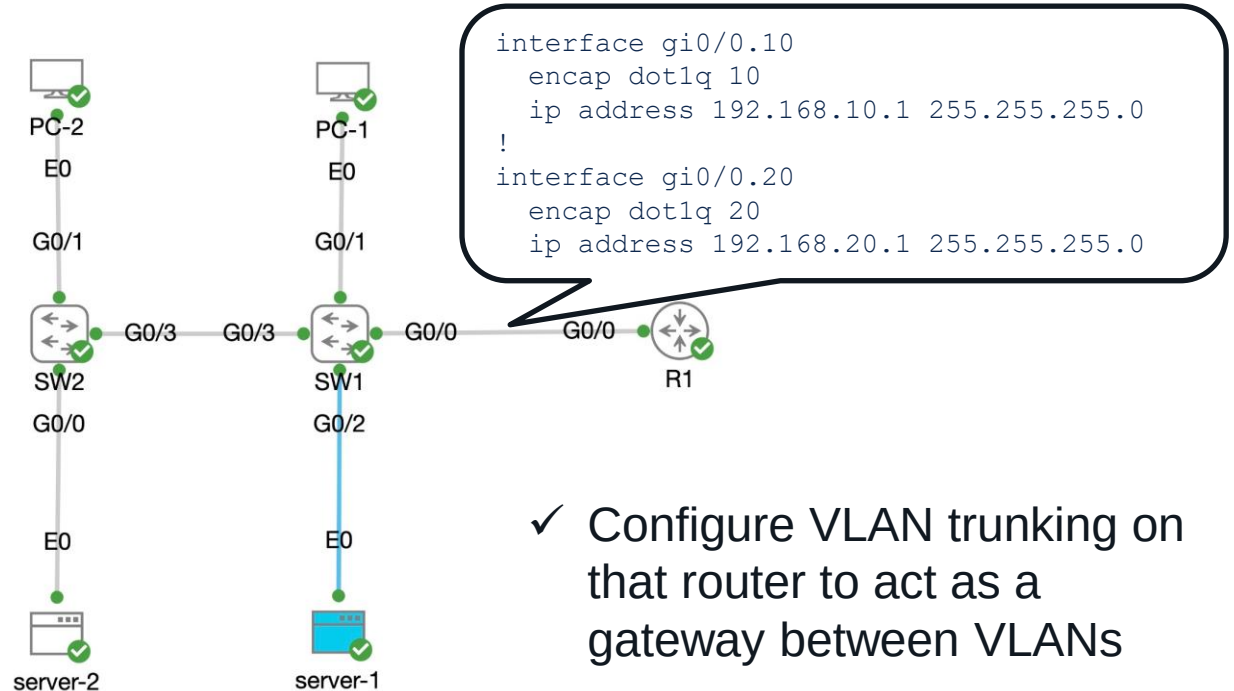


```
vlan 10
 name Users
!
vlan 20
 name Servers
!
Interface gi0/0
 switchport trunk encap dot1q
 switchport mode trunk
!
interface gi0/1
 switchport mode access
 switchport access vlan 10
!
interface gi0/2
 switchport mode access
 switchport access vlan 20
```

- ✓ Implement VLAN assignments on two switches
- ✓ Configure a dot1q trunk between those switches

Sample lab 1: CCNA

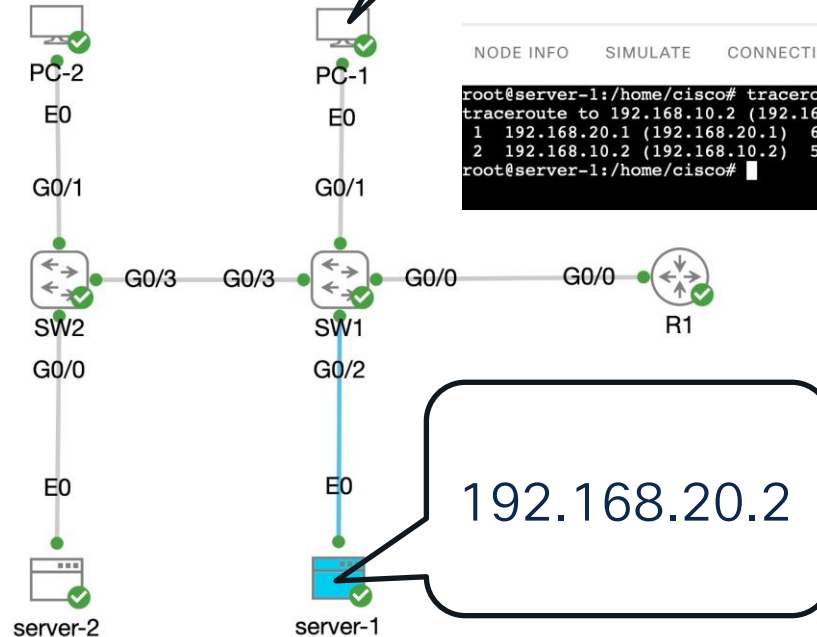
Config



Sample lab 1: CCNA

Verify

- ✓ Verify that connectivity exists between hosts in the lab on each VLAN, and that the connectivity goes through the router



192.168.10.2

NODE INFO SIMULATE CONNECTIVITY CONSOLE VNC EDIT CONFIG

```
root@server-1:/home/cisco# traceroute 192.168.10.2
traceroute to 192.168.10.2 (192.168.10.2), 30 hops max, 38 byte packets
 1  192.168.20.1 (192.168.20.1)  6.701 ms  3.892 ms  3.430 ms
 2  192.168.10.2 (192.168.10.2)  5.713 ms  8.352 ms  7.340 ms
root@server-1:/home/cisco#
```

192.168.20.2

Sample lab 2: Enterprise Advanced Routing and Switching (ENARSI)

Troubleshooting BGP

Full-mesh IBGP peering in AS 65001 uses Loopback1 interfaces instead of directly connected interfaces. R2 and R3 advertise correct next hop information to their IBGP peers. EBGP peering between R2 and R4 uses Loopback1 interfaces instead of directly connected interfaces.

R6 should prefer the path through R7 to reach the 10.1.1.0/24 network.

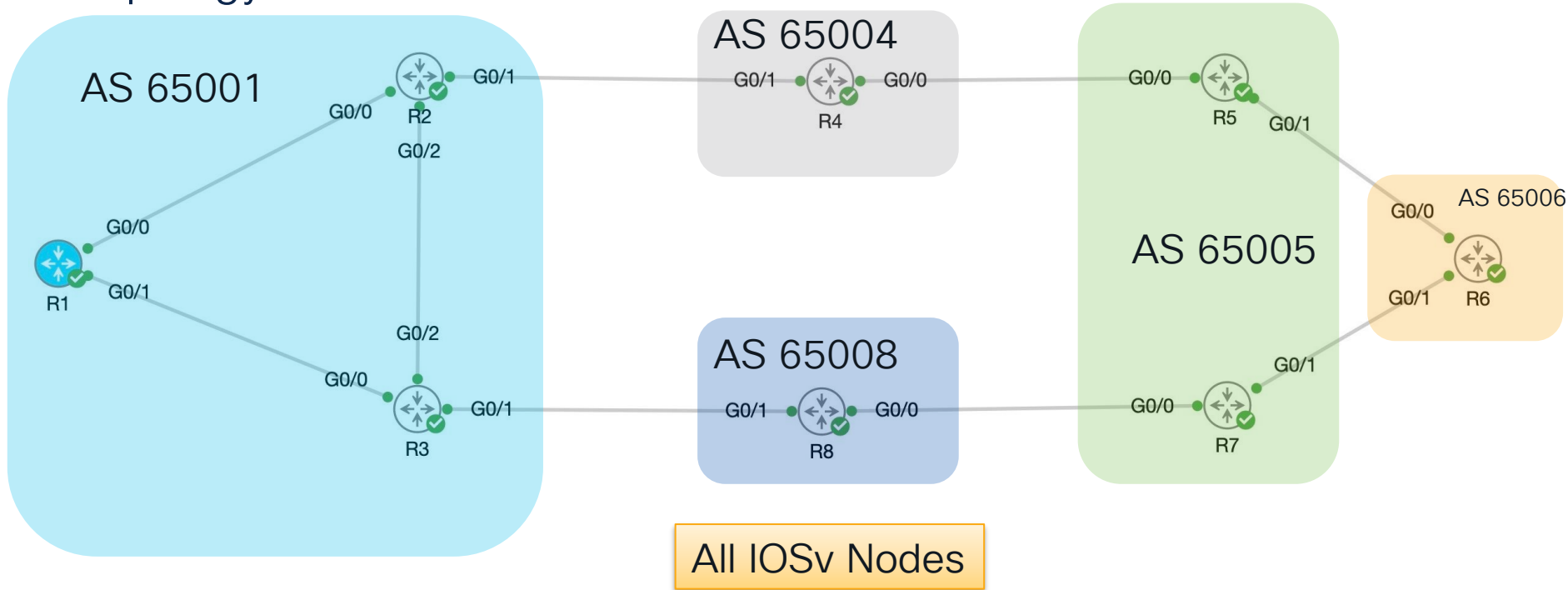
Why isn't BGP working throughout? There are five neighbor relationship issues, one routing issue, and one path selection issue.

Find and fix the problems and make sure all loopbacks are pingable!



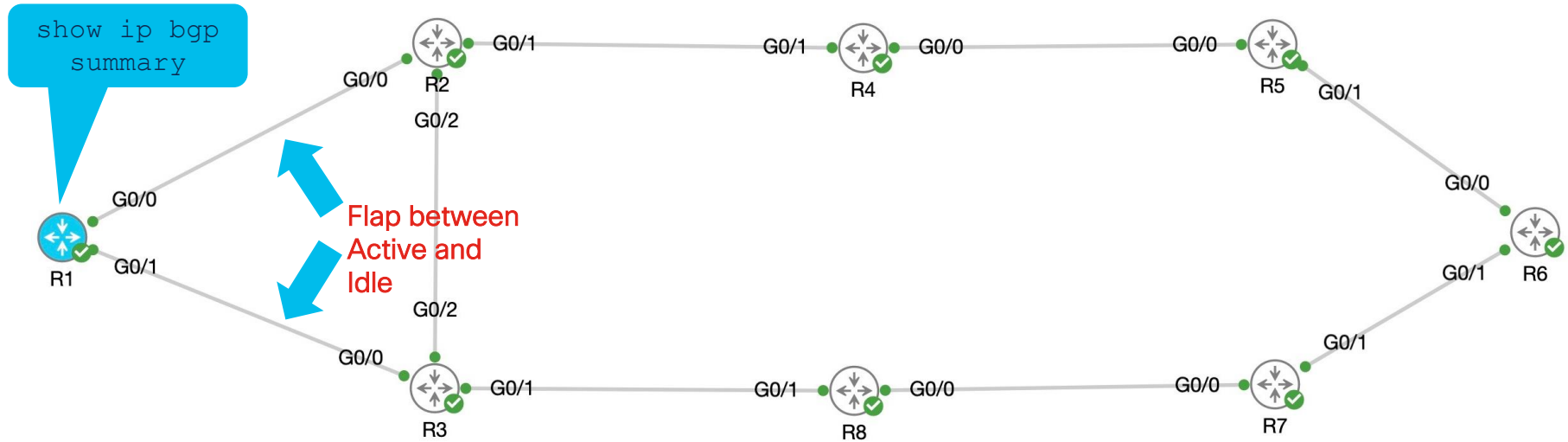
Sample lab 2: Enterprise Advanced Routing and Switching (ENARSI)

Topology



Sample lab 2: Enterprise Advanced Routing and Switching (ENARSI)

Diagnose at R1



Sample lab 2: Enterprise Advanced Routing and Switching (ENARSI)

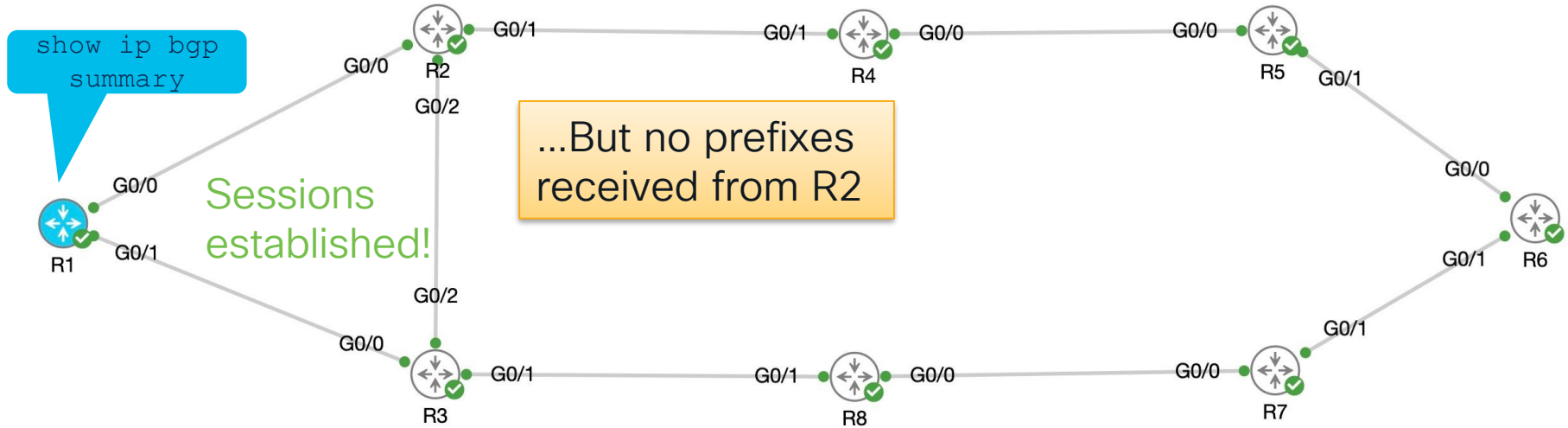
R1 Config

```
router bgp 65001
  synchronization
  bgp log-neighbour-changes
  network 10.1.1.0 mask 255.255.255.0
  neighbour 172.16.1.2 remote-as 65001
  neighbour 172.16.1.2 update-source Loopback0
  neighbour 172.16.1.3 remote-as 65001
  neighbour 172.16.1.3 update-source Loopback0
  neighbour 172.16.1.3 route-map WEIGHT in
```

Requirements state we need to use Loopback1

Sample lab 2: Enterprise Advanced Routing and Switching (ENARSI)

Fix R1

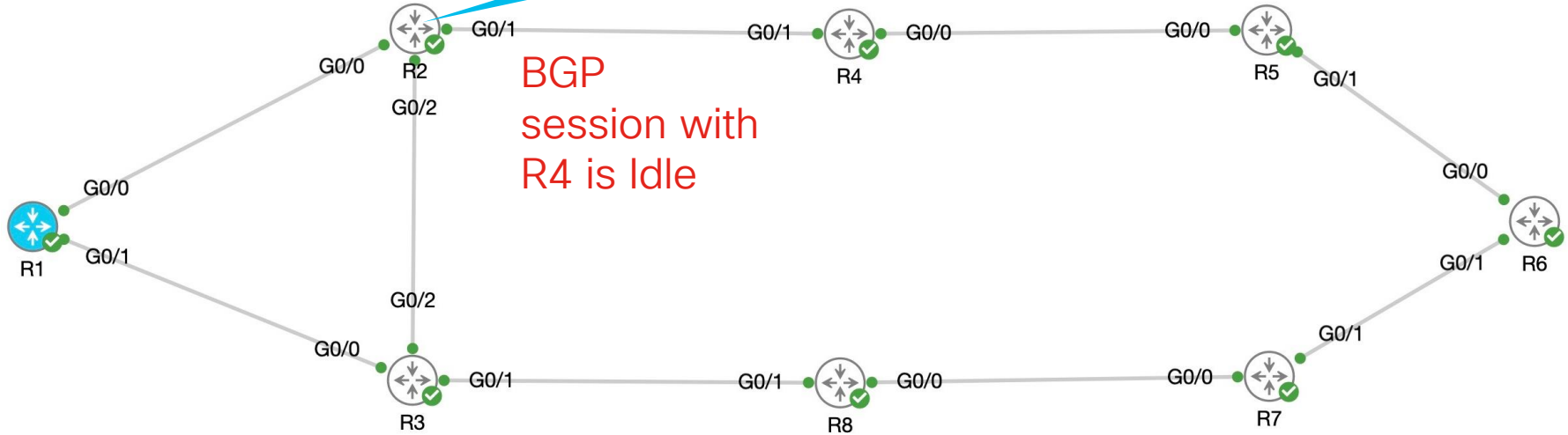


Sample lab 2: Enterprise Advanced Routing and Switching (ENARSI)

Diagnose R2

show ip bgp
summary

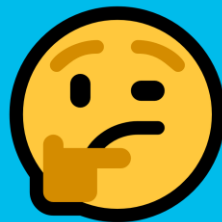
BGP
session with
R4 is Idle



Sample lab 2: Enterprise Advanced Routing and Switching (ENARSI)

Verify BGP Peering with R4

```
R2#show ip bgp neighbours 172.16.4.4 | inc BGP neighbour  
BGP neighbour is 172.16.4.4, remote AS 65004, external link  
External BGP neighbour not directly connected.  
External BGP neighbour configured for connected checks (single-  
hop no-disable-connected-check)
```



Sample lab 2: Enterprise Advanced Routing and Switching (ENARSI)

Fix R2 Config

```
R2 (config) #router bgp 65001
```

```
R2 (config-router) #neighbor 172.16.4.4 ebgp-multihop 2
```

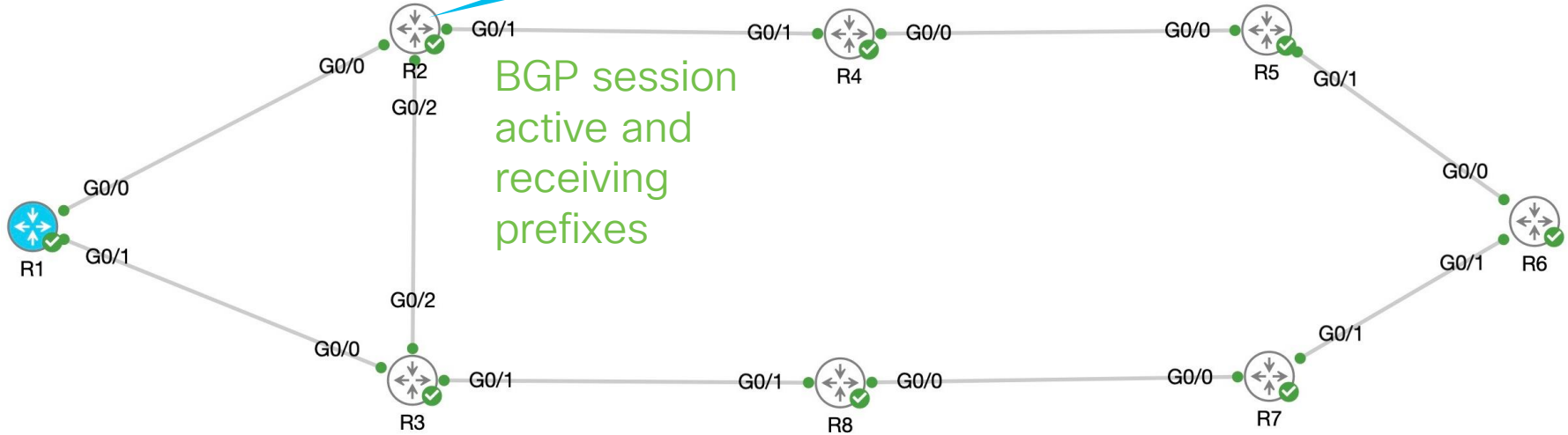
Allow peer to use BGP multihop with a TTL of at least 2 (i.e., two hops to get to the peer)

Sample lab 2: Enterprise Advanced Routing and Switching (ENARSI)

Diagnose R2

`show ip bgp
summary`

BGP session
active and
receiving
prefixes

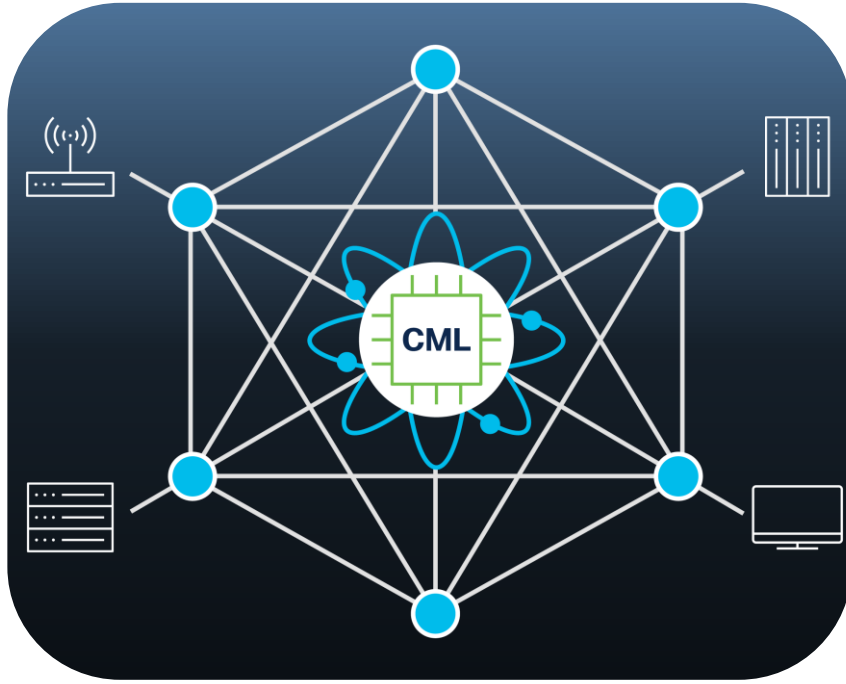


There are a few more problems
left for your own practice 😊

Join at
slido.com
#1706 743



Get CML, and then...



- **Obtaining:**
<https://developer.cisco.com/docs/modeling-labs/#!downloading-files-for-cml-installation>
- **Docs:**
<https://developer.cisco.com/docs/modeling-labs/#!introduction>
- **Sandbox:**
<https://devnetsandbox.cisco.com/RM/Topology> (Reserve “Cisco Modeling Labs Enterprise”)



Cisco Certifications

>> Your name here <<

has successfully completed the Cisco certification exam requirements and is recognized as a

Cisco Certified DevNet Associate



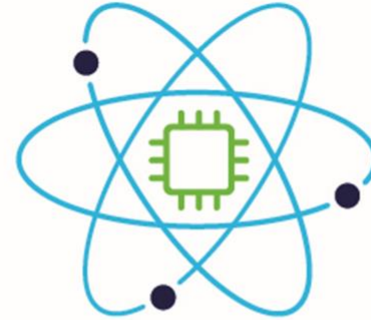
Become a power user

Cisco Live Melbourne Exclusive:
Save 50% on CML-Personal and
CML-Personal Plus.

Visit the Learning & Certification
booth to learn more.



CISCO *Live!*



Powered by
Cisco Modeling Labs

Complete Your Session Evaluations



Complete a minimum of 4 session surveys and the Overall Event Survey to claim a **Cisco Live T-Shirt**.



Complete your surveys in the **Cisco Live mobile app**.



Continue your education



- Visit the Cisco Stand 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



Thank you

CISCO *Live!*

#CiscoLiveAPJC

CISCO *Live!*

GO BEYOND

#CiscoLiveAPJC

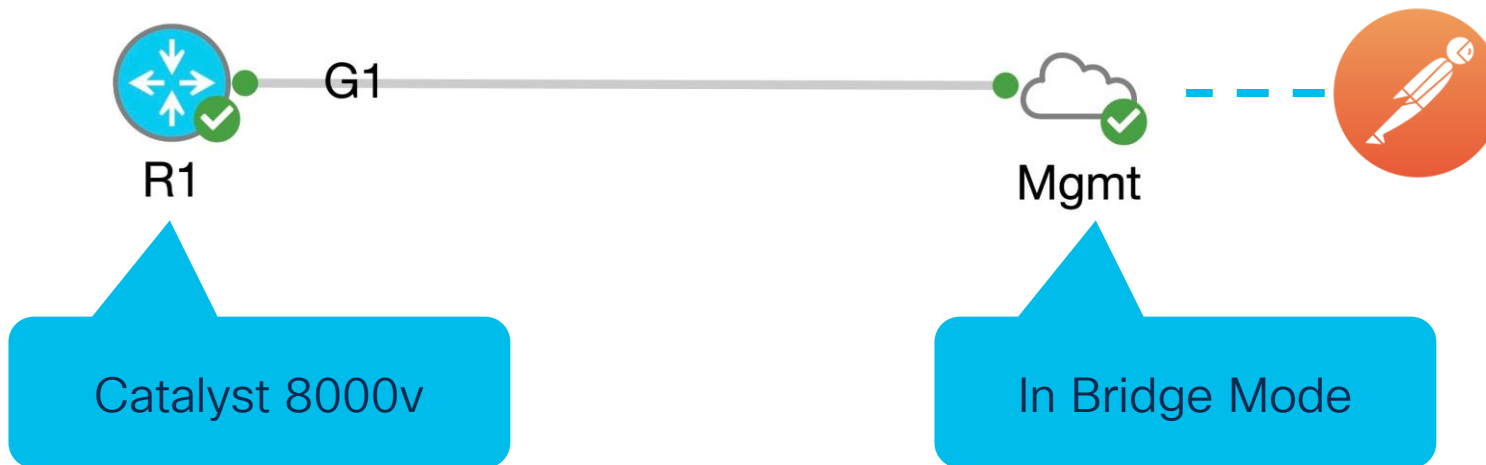
Sample Lab 3: DevNet Technology Core (DEVCOR)

Utilise RESTCONF for network configuration

Retrieve the configuration with RESTCONF using Postman, and then add an IP address to Loopback0

Sample lab 3: DevNet Technology Core (DEVCOR)

Topology



Sample Lab 3: DevNet Technology Core (DEVCOR)

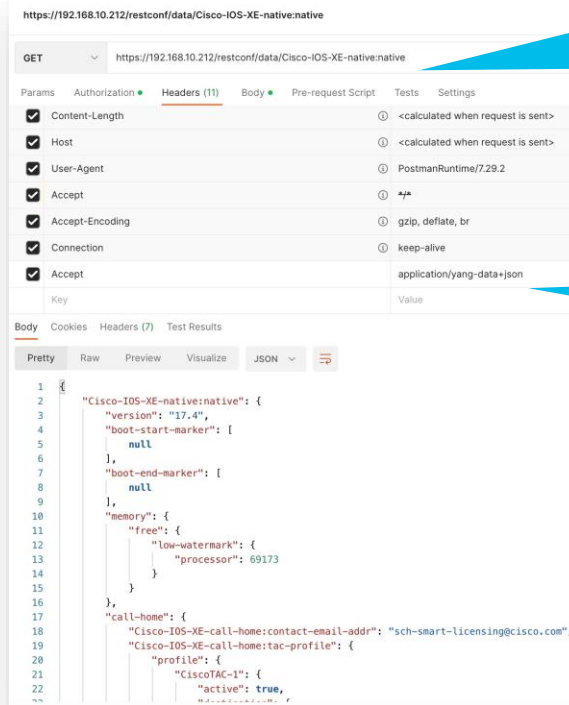
Enable RESTCONF

```
R1 (config) #restconf
```

```
R1 (config) #username cml privilege 15 secret cleur
```

Sample Lab 3: DevNet Technology Core (DEVCOR)

Retrieve the Config



The model for the native (i.e., “show run”-like config is Cisco-IOS-XE-native:native

Get JSON output with application/yang-data+json

Note: RESTCONF uses Basic auth

Sample Lab 3: DevNet Technology Core (DEVCOR)

Build a Payload to Configure IP on Loopback0

```
{  
  "address": "10.1.1.0",  
  "mask": "255.255.255.255"  
}
```

Sample Lab 3: DevNet Technology Core (DEVCOR)

Retrieve the Config

The screenshot shows a REST client interface with the following details:

- URL:** `https://192.168.10.212/restconf/data/Cisco-IOS-XE-native:native/interface/Loopback=0/ip/address/primary`
- Method:** POST (highlighted with a red box)
- Body:**

```
1 {  
2   "address": "10.1.1.0",  
3   "mask": "255.255.255.255"  
4 }
```
- Status:** 201 Created (indicated by a green icon and text)
- Time:** 784 ms
- Size:** 424 B
- Buttons:** Save Response, Beautify, Cookies

A blue callout bubble points to the status 201, containing the text: "201 means the resource was created".

Sample Lab 3: DevNet Technology Core (DEVCOR)

Validate the Change

https://192.168.10.212/restconf/data/Cisco-IOS-XE-native:native/interface/Loopback=0

GET https://192.168.10.212/restconf/data/Cisco-IOS-XE-native:native/interface/Loopback=0

Params Authorization Headers (12) Body Pre-request Script Tests Settings Cookies

☒	Host	<calculated when request is sent>	
☒	User-Agent	PostmanRuntime/7.29.2	
☒	Accept	*/*	
☒	Accept-Encoding	gzip, deflate, br	
☒	Connection	keep-alive	
☒	Accept	application/yang-data+json	
☒	Content-Type	application/yang-data+json	
	Key	Value	Description

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1  {
2    "Cisco-IOS-XE-native:Loopback": {
3      "name": 0,
4      "ip": {
5        "address": {
6          "primary": {
7            "address": "10.1.1.0",
8            "mask": "255.255.255.255"
9          }
10       }
11     }
12   }
13 }
```

Status: 200 OK Time: 788 ms Size: 460 B Save Response