

Containerlab: A modern approach to networking labs

Saju Salahudeen
NOKIA



- Principal Consulting Engineer
- 19+ years at Nokia
- IP Networking
- Multi-vendor Management
- Network Automation
- Daily user of Containerlab
- Discord (@sajusal)

Containerlab - Introduction



309K

Installations

100+

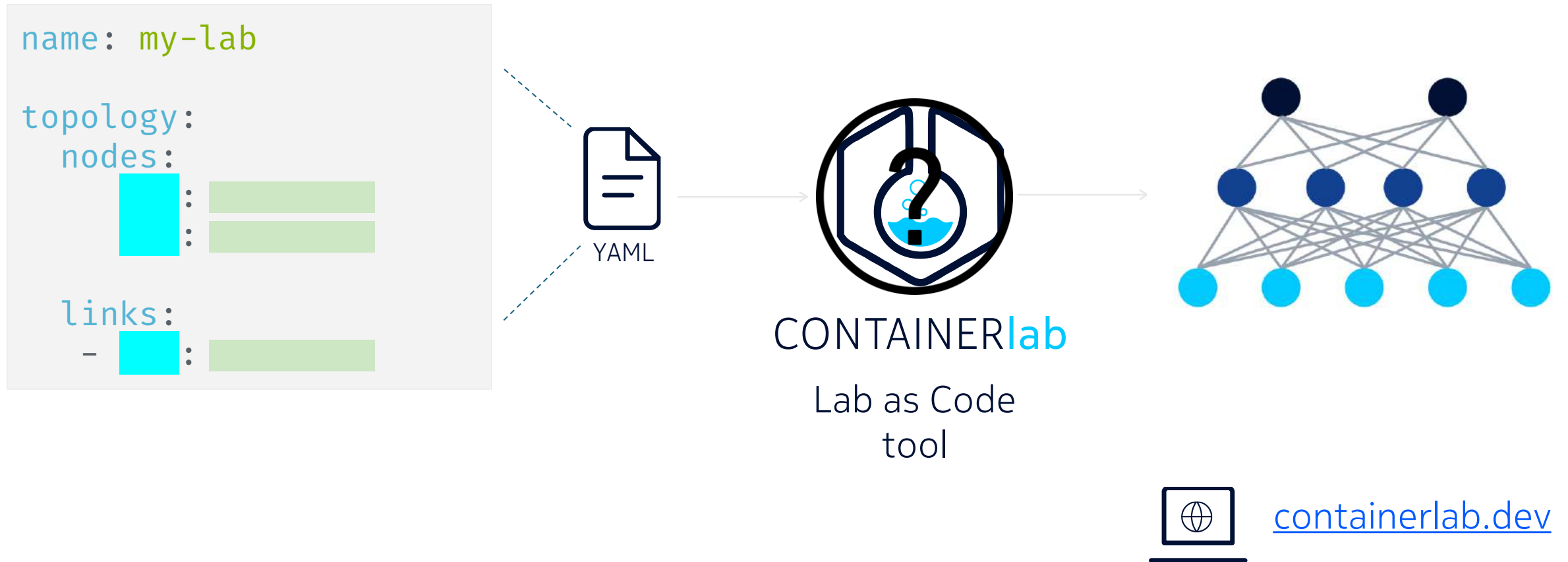
Contributors

30

Supported NOSes

24

Releases in 12mo

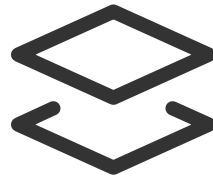


Why “Lab as code”?

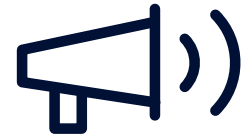
git



Collaboration

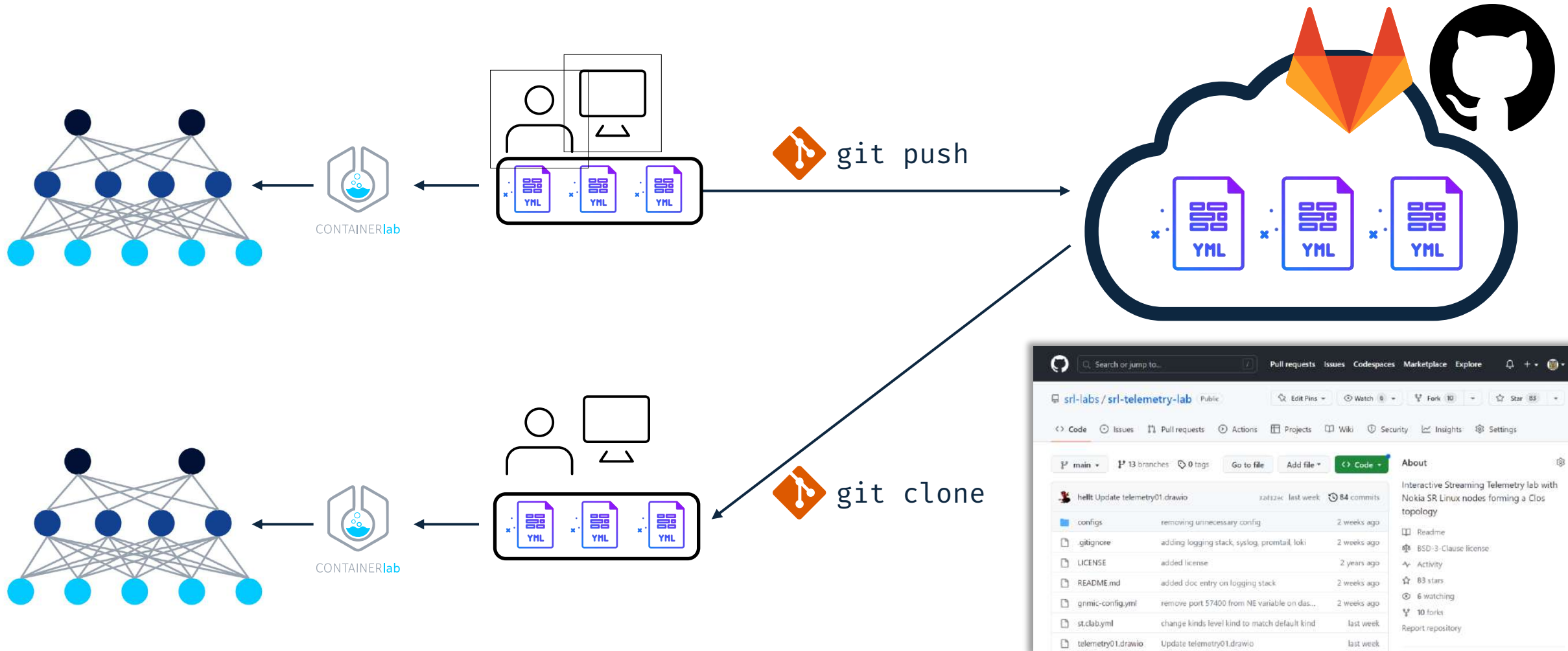


Versioning



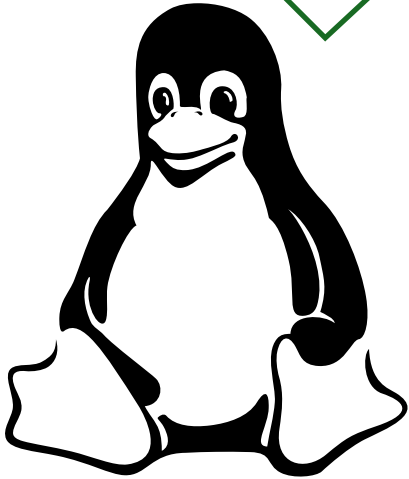
Sharing

Share your Labs





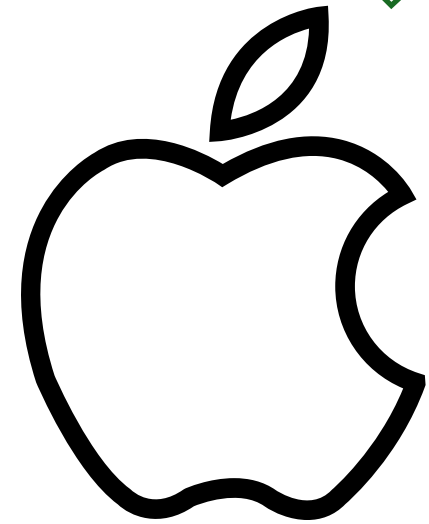
<https://containerlab.dev/install/>



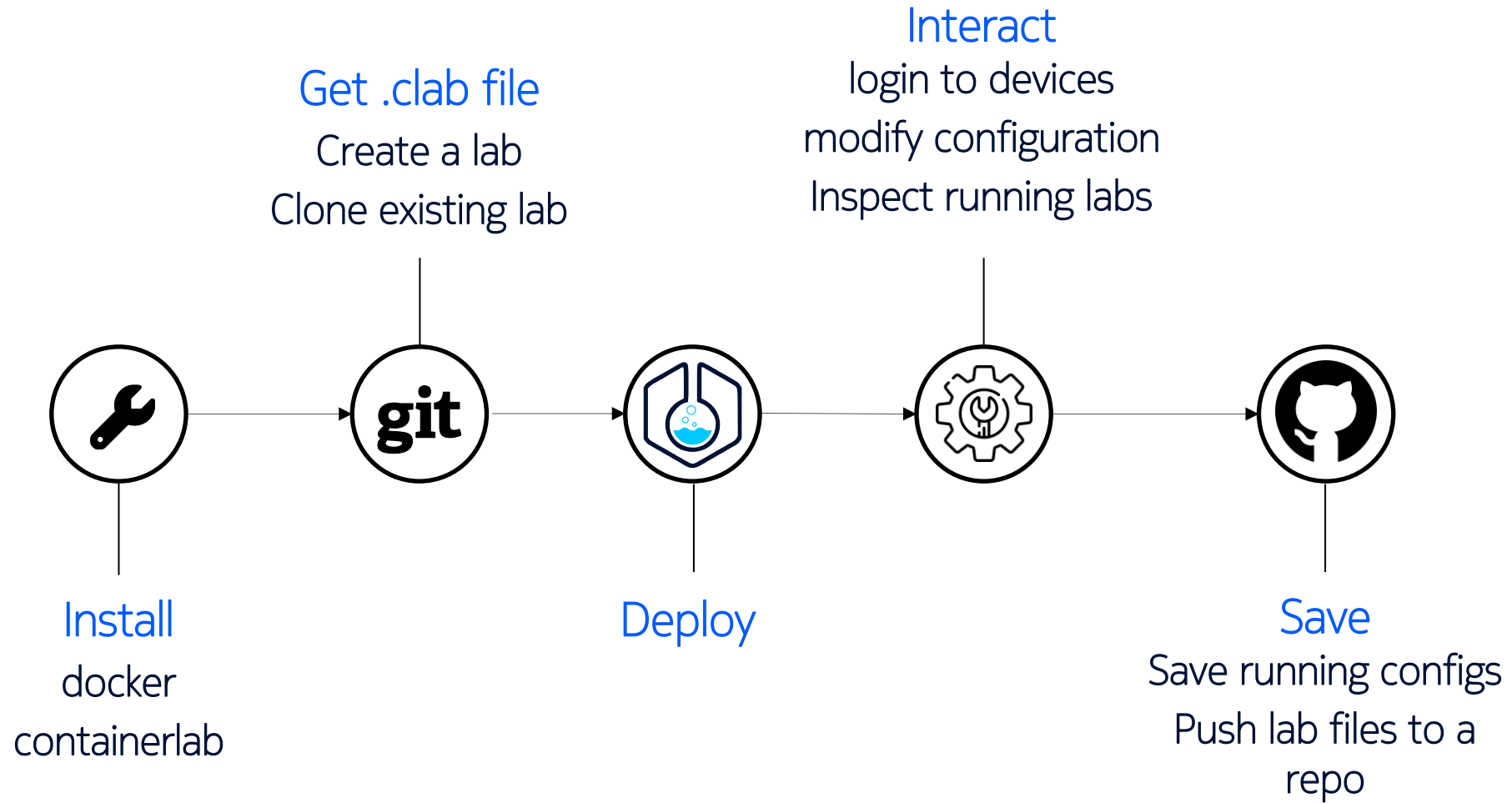
LINUX



WSL2



MacOS



- Sourced by the vendor
- Fast to spin up
- Small footprint
- Shareability and versioning

Current trend is to **move away from VM** packaging towards containers
for new NOSes

NOKIA
SR Linux

JUNIPER
NETWORKS
cRPD

ARISTA
cEOS

CISCO
XRd

NVIDIA
cVX

KEYSIGHT
TECHNOLOGIES
IXIA-c

FRR

and others...

- All available container images
- Emulating clients
- Hundreds of network-focused software
 - Telemetry, logging stacks
 - Peering software
 - Flow collectors



Get / Set / Subscribe / Collect



Containerlab Topology File - Nodes

```
name: basic
```

```
topology:
```

```
  nodes:
```

```
    srl:
```

```
      kind: nokia_srlinux
```

```
      image: ghcr.io/nokia/srlinux
```

1

Node definition container.
Container name will be the node name.

2

Kinds define the flavour of the node,
it says if the node is a specific containerized
Network OS or something else.

3

Image specifies container image to use for this
node.

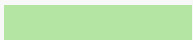
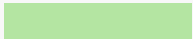


[topology definition file](#)

Topology definition

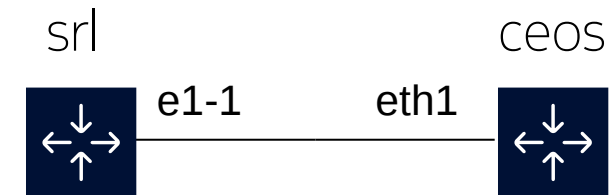
```
name: basic

topology:
  nodes:

    srl:  
    ceos:  

  links:
    - endpoints: [srl:e1-1, ceos:eth1]
```

Logical view



Containerlab Topology File – Startup config

topology definition

```
name: basic

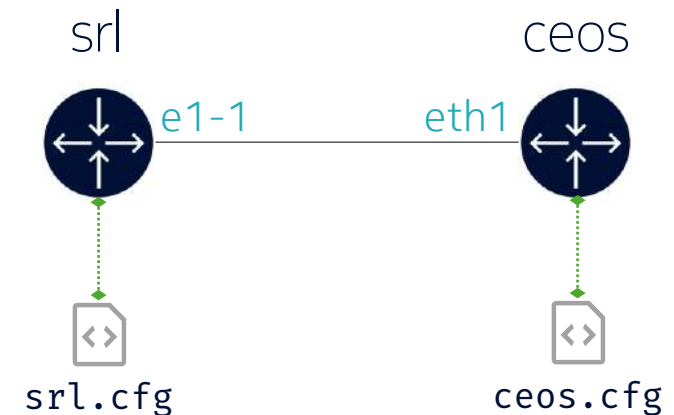
topology:
  nodes:

    srl:
      kind: nokia_srlinux
      image: ghcr.io/nokia/srlinux
      startup-config: srl.cfg

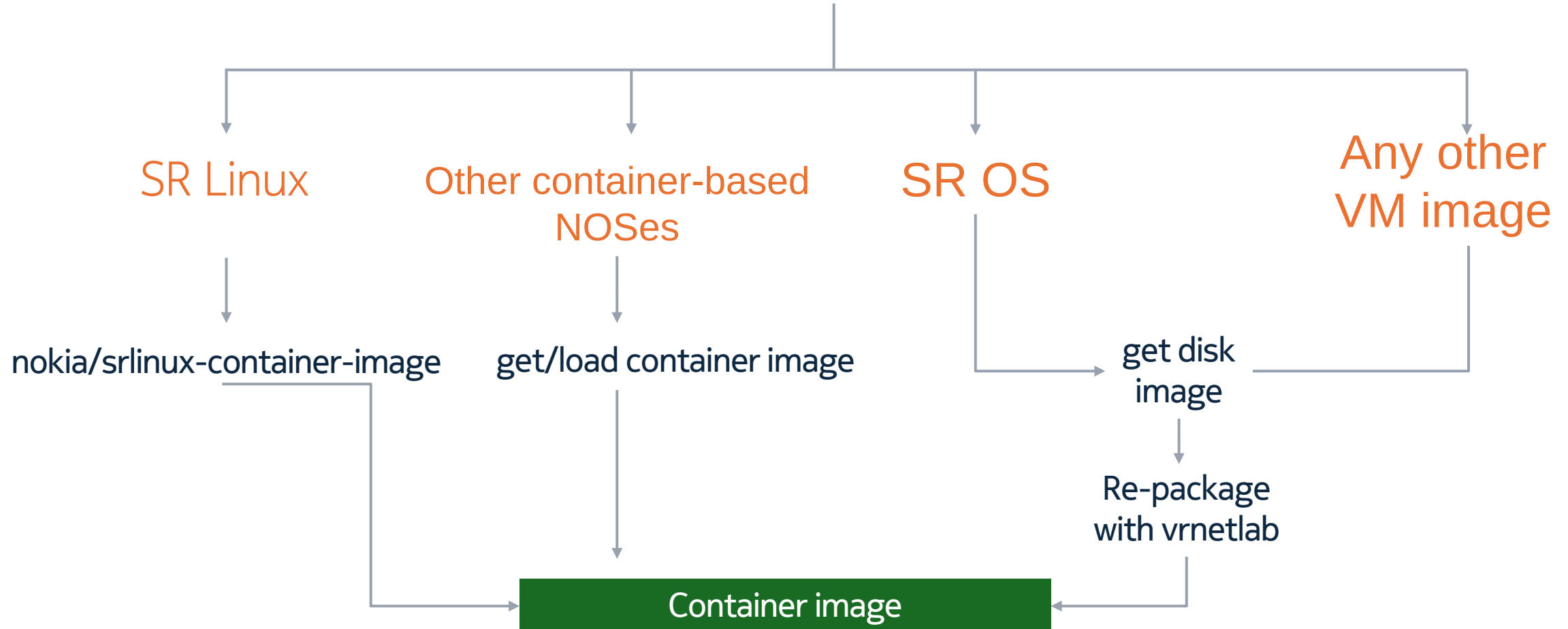
    ceos:
      kind: arista_ceos
      image: ceos:4.33.0F
      startup-config: ceos.cfg
```

mylab.clab.yml

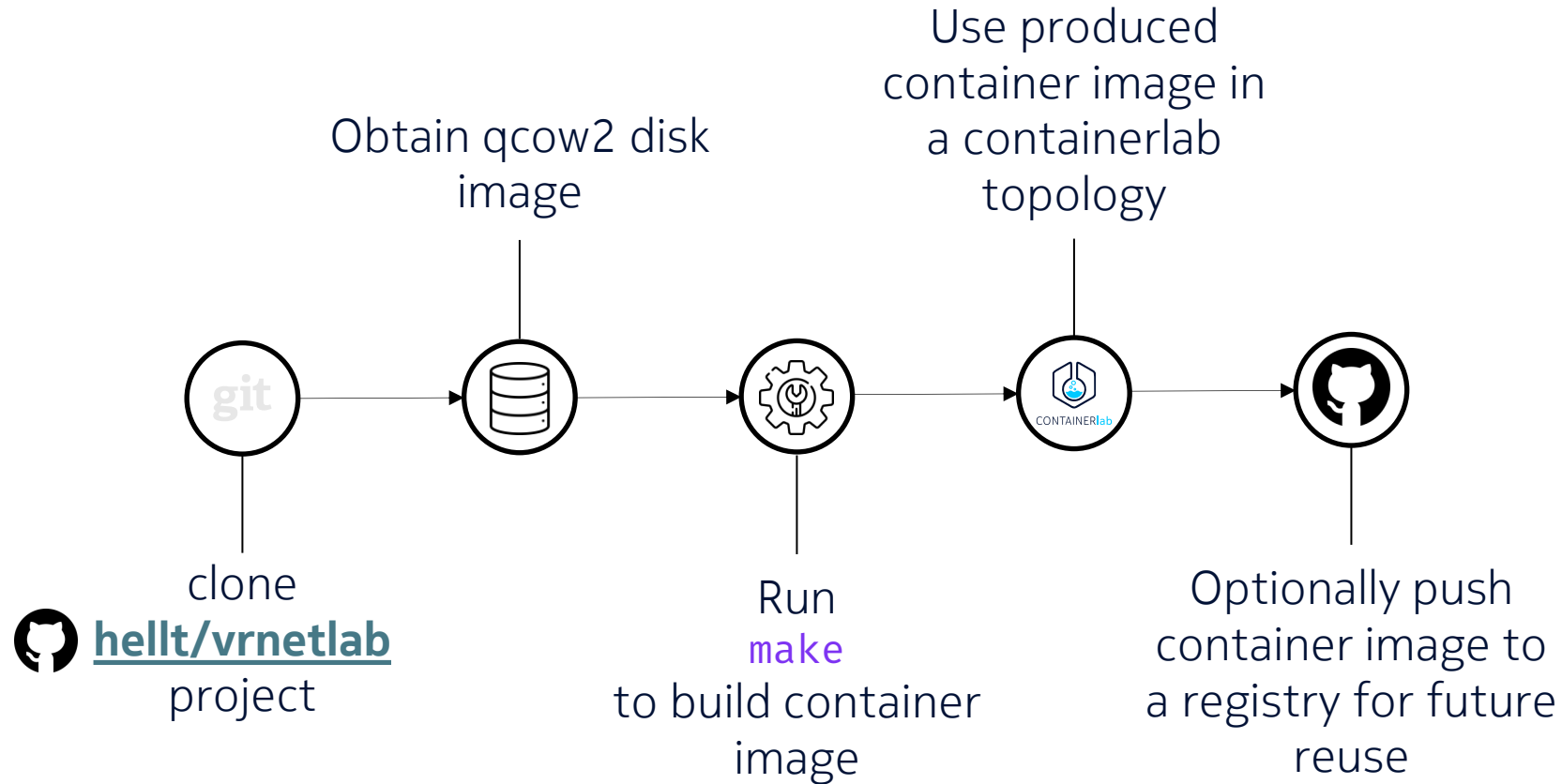
logical view



How do I get an image?



Bringing VM-based nodes to Containerlab



[~/cwrk]

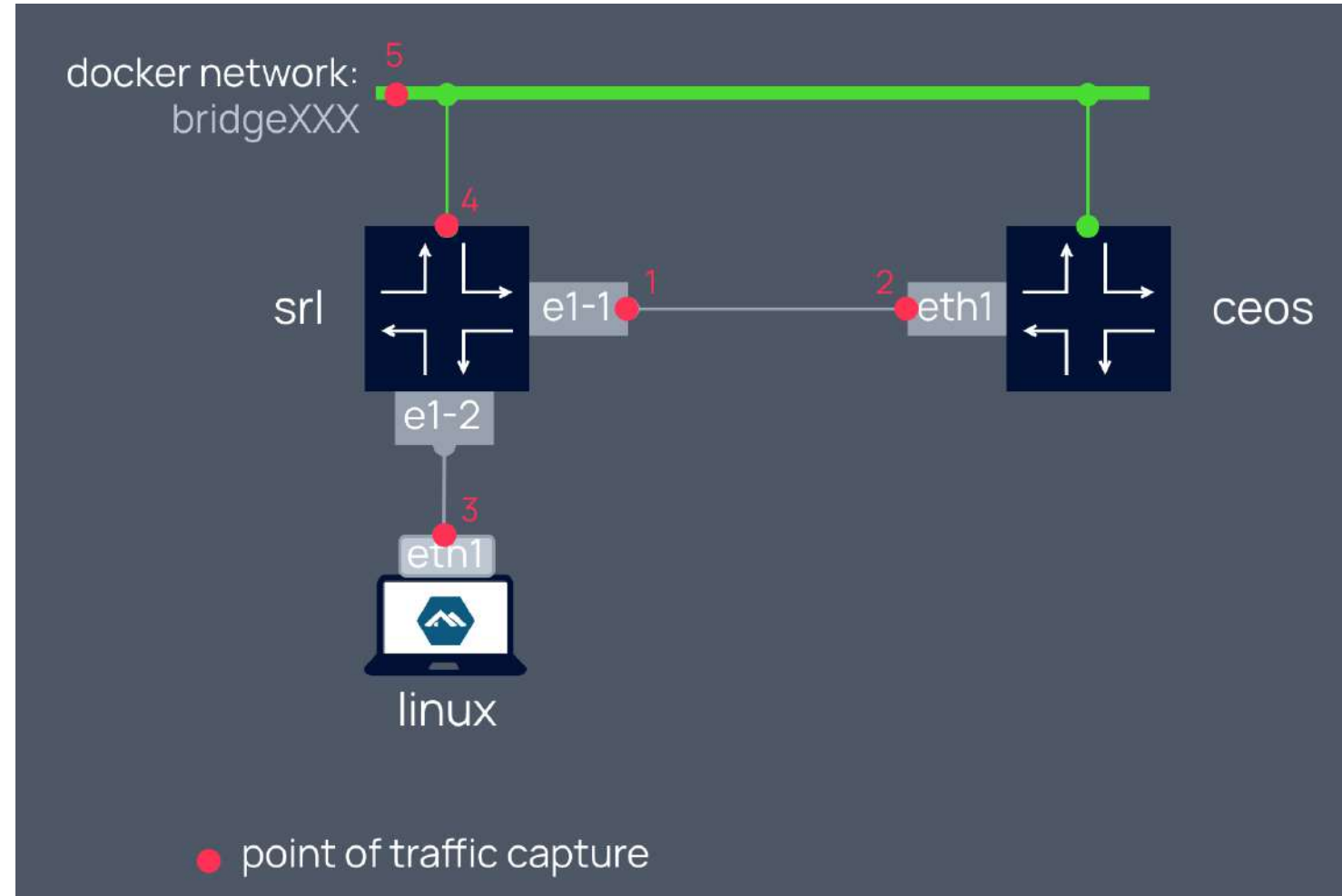
└─> **sudo containerlab deploy -t mylab.clab.yml**

Name	Kind/Image	State	IPv4/6 Address
clab-basic-ceos	arista_ceos ceos:4.33.0F	running	172.20.20.2 3fff:172:20:20::2
clab-basic-srl	nokia_srlinux ghcr.io/nokia/srlinux	running	172.20.20.3 3fff:172:20:20::3

- 1 Local capture
 - Via container's shell
- 2 Remote capture
 - Via ssh and pipes
- 3 Web UI capture
 - Via Edgeshark

Reference:

github.com/srlinuxamericas/n93-clab



containerlab.dev/manual/wireshark

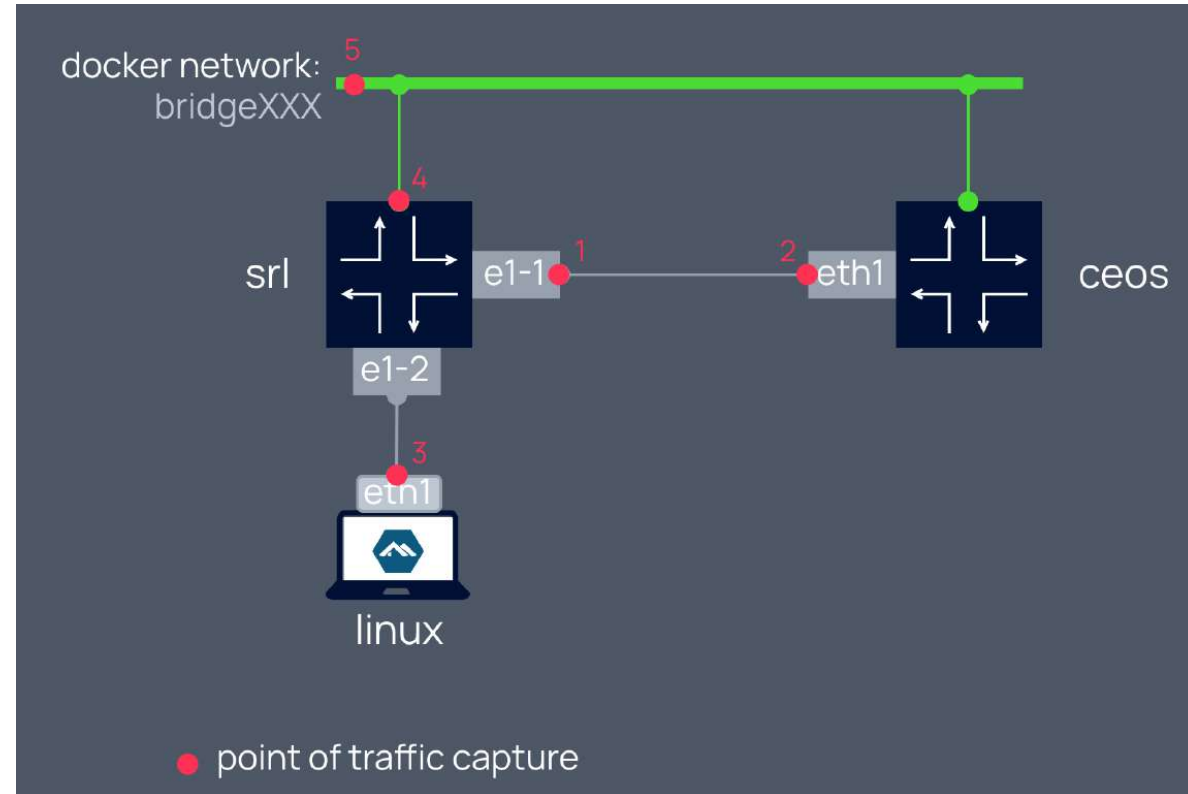
Command to capture at point #1



```
sudo ip netns exec clab-startup-srl tcpdump -nni e1-1
```

Notes:

- Output is displayed on screen after capture is stopped

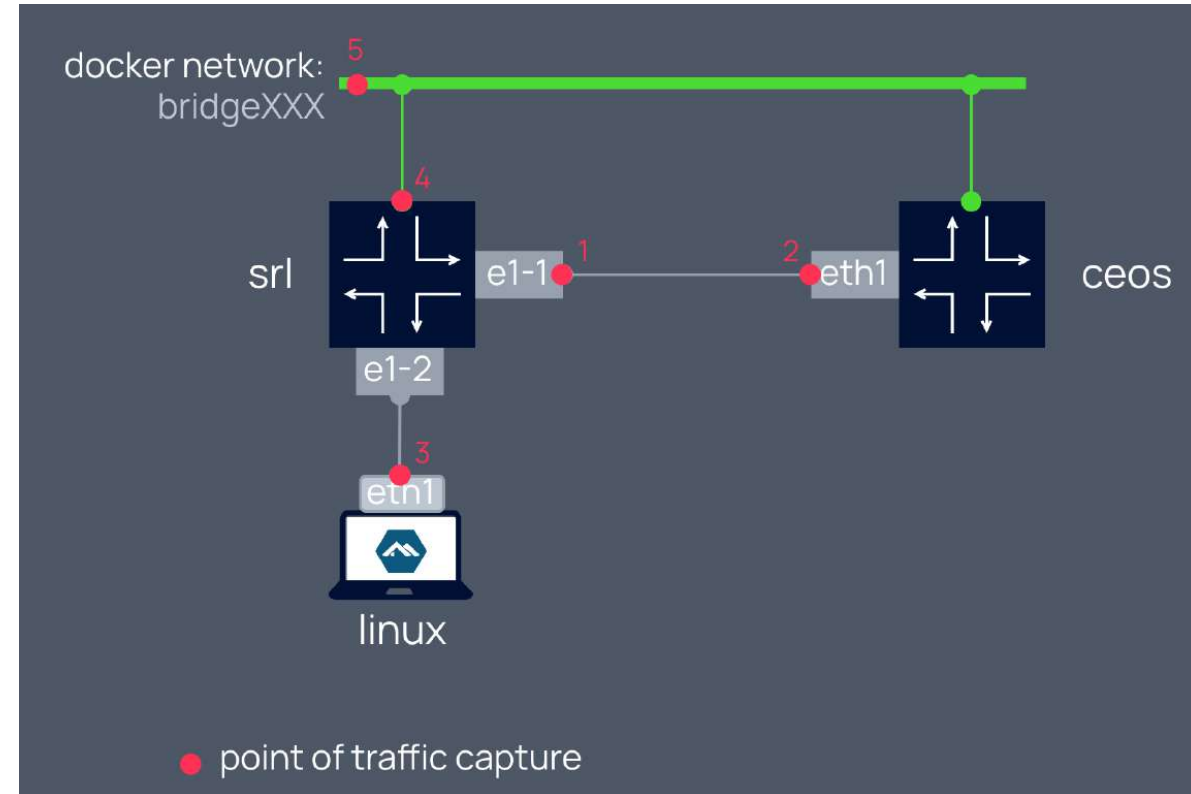


Command to capture at point #1

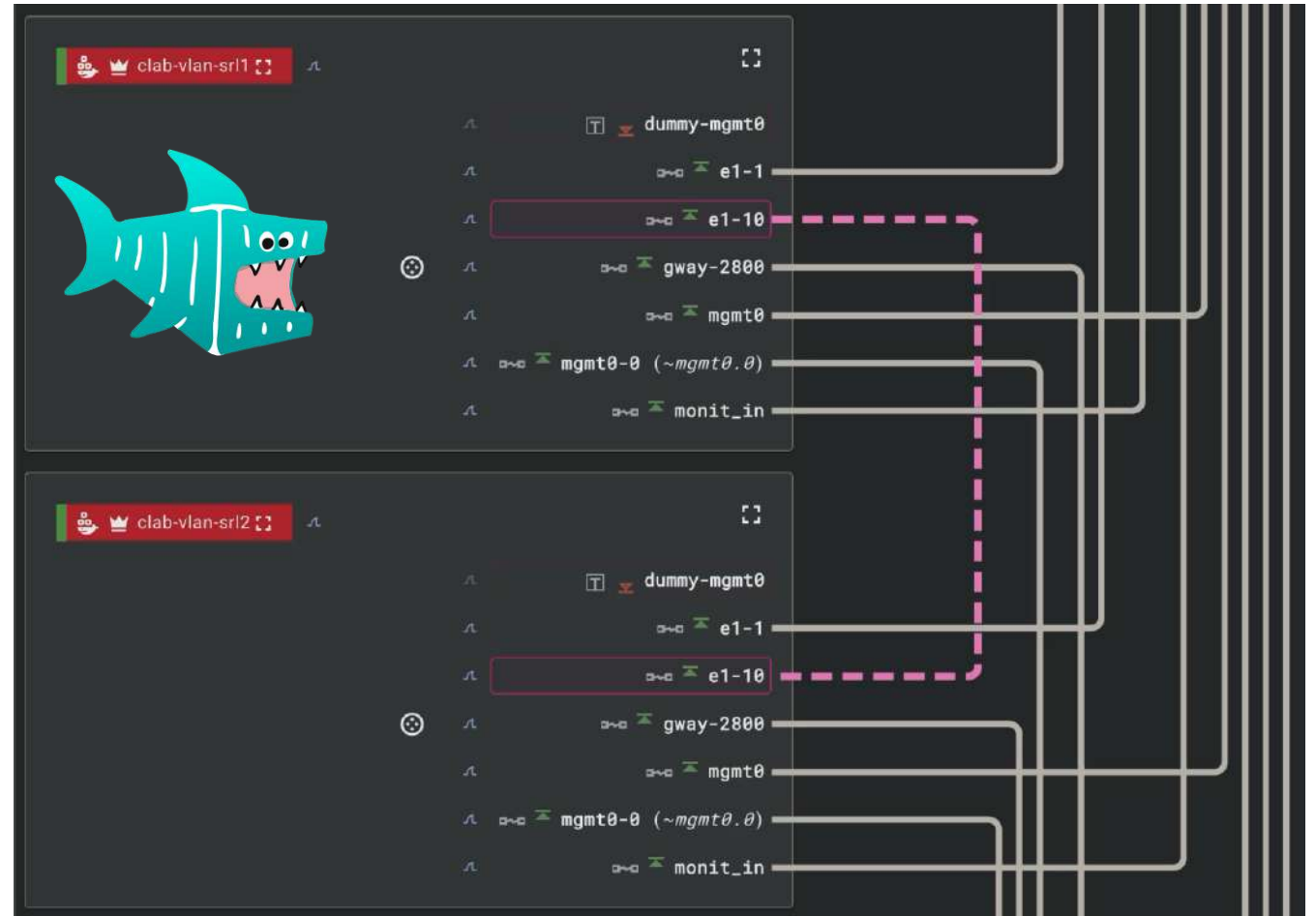
```
ssh $clab_host "ip netns exec srl tcpdump -U -nni e1-1 -w -" | wireshark -k -i -
```

Notes:

- Use WSL on windows
- Wrap this long command in a pcap.sh that takes in the node/interface arguments
- No extra packages/modifications required



- Way more than this, but that's how we use it
- Web-based, no installation required
- Container-based, no need for dependencies
- Uses native Wireshark
- Open-source
- Free



containerlab.dev/manual/wireshark/#edgeshark-integration

1 Deploy Edgeshark service

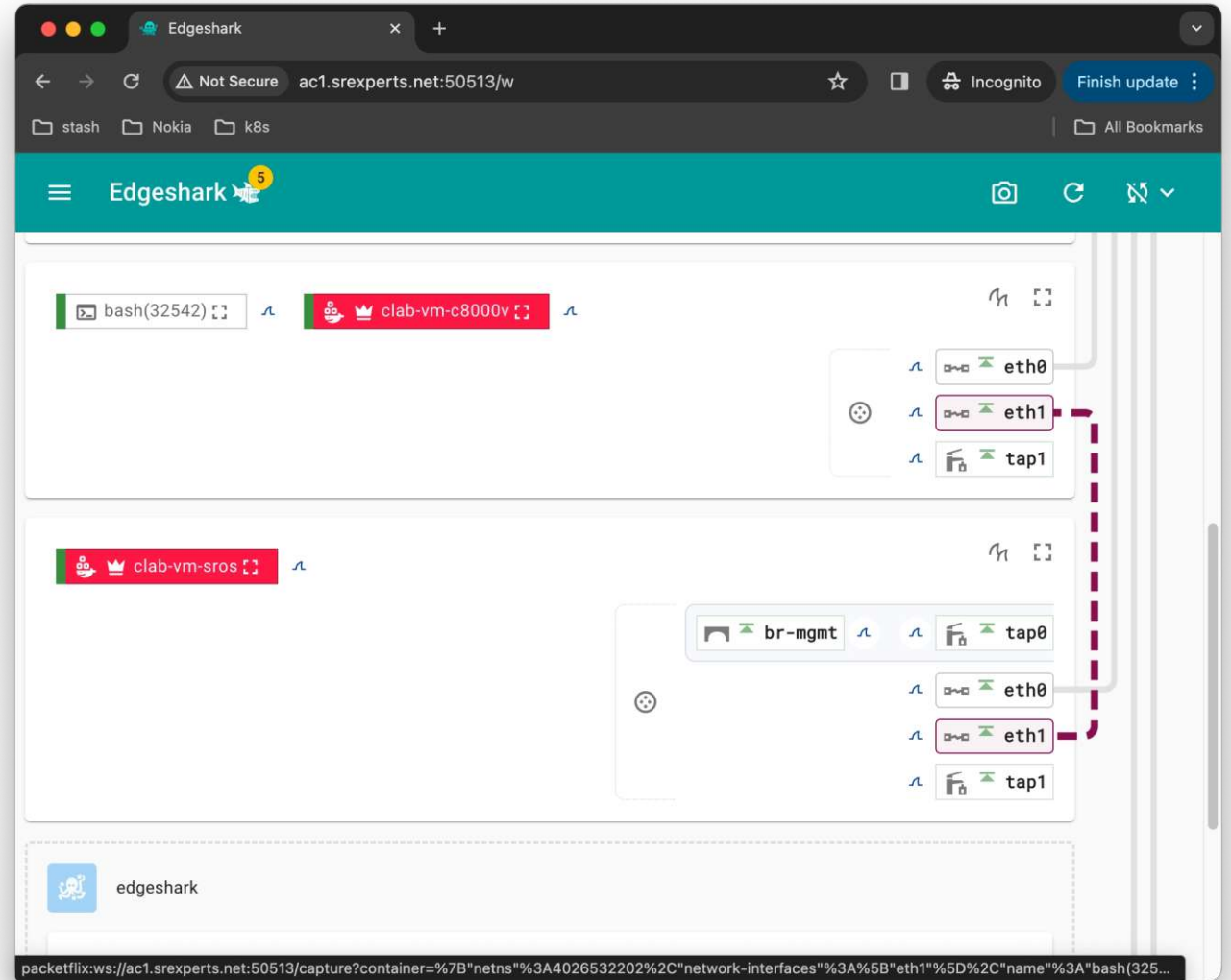
```
[*]-[<ID>]-[~]  
└─> curl -sL \  
https://github.com/siemens/edgeshark/raw/main/deployments/wget/docker-  
compose.yaml | docker compose -f - up -d
```

2 Install Wireshark capture plugin on PC or mac



When the page opens, hit to refresh the list of containers

Click on 'shark' icon against interface to start packet capture



- Microsoft-backed compute cloud
- Generous free tier with a monthly quota reset
- 120cpu/hours a month FREE
- Containerlab integrated with Codespaces
- No \$\$\$ until explicitly committed

Nokia SR Linux Streaming Telemetry Lab

SR Linux has first-class Streaming Telemetry support thanks to [100% YANG coverage](#) of state and config data. The holistic coverage enables SR Linux users to stream any data off of the NOS with on-change, sample, or target-defined support. A discrepancy in visibility across APIs is not about SR Linux.

This lab represents a small Clos fabric with [Nokia SR Linux](#) switches running as containers. The lab topology consists of a Clos topology, plus a Streaming Telemetry stack comprised of [gnmic](#), prometheus and grafana applications.

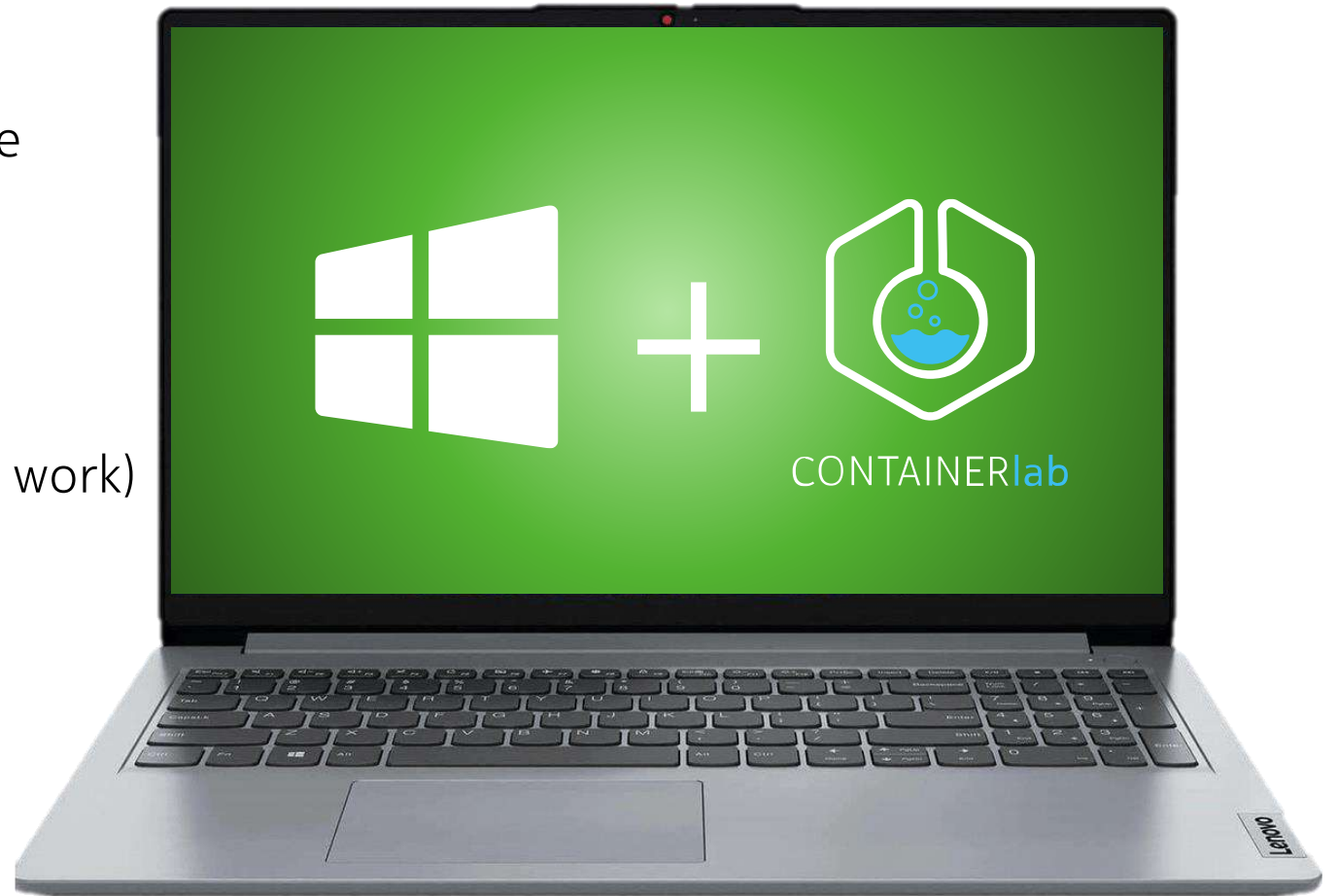


Run In Codespaces

[Run](#) this lab in GitHub Codespaces for free.

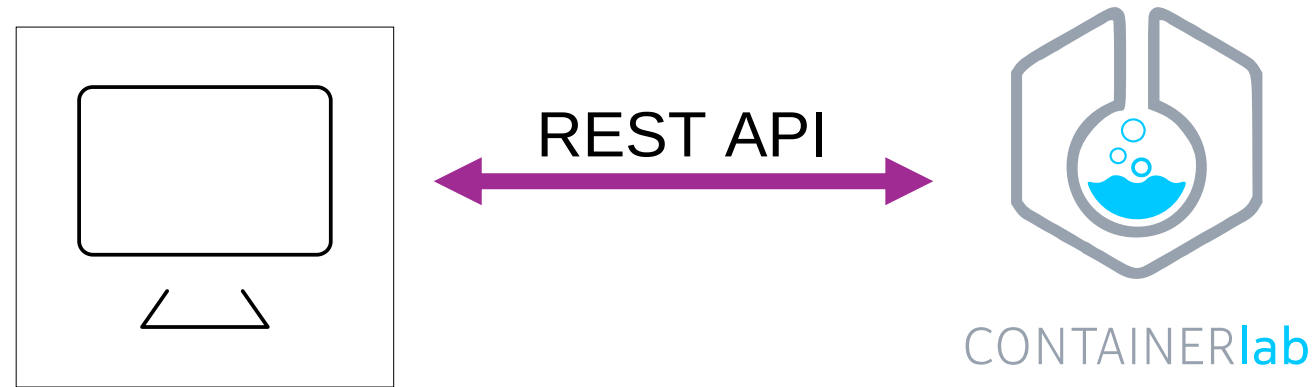
[Learn more](#) about Containerlab for Codespaces.

- Windows Subsystem Linux distributive with Containerlab pre-installed
 - One-click install
 - All pre-requisites. and tooling installed
- Will soon support VMs (pending MSFT work)



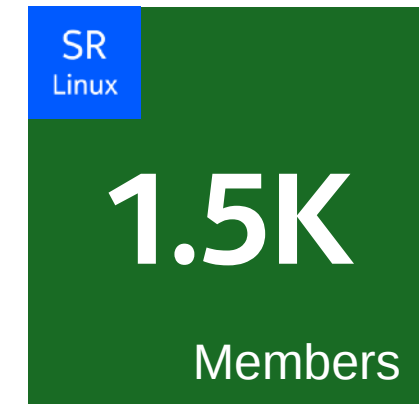
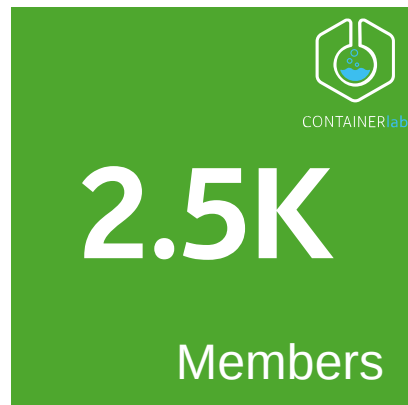
- 





- API server - Standalone component and requires separate installation
- Source - <https://github.com/srl-labs/clab-api-server>
- Can also run REST API server as a container
- Flashpost (alternative to Postman) collection available

Supporting the user community



- Containerlab - <https://containerlab.dev/>
- SR Linux free docker image - <https://github.com/nokia/srlinux-container-image/pkgs/container/srlinux>
- SR Linux Tutorials and Blogs - <https://learn.srlinux.dev/>
- SR Linux and Containerlab Discord - <https://discord.gg/tZvgjQ6PZf>
- SR Linux Documentation - <https://documentation.nokia.com/srlinux/>
- SR Linux Yang models per release- <https://yang.srlinux.dev/>
- EDA – “Try it” install & documentation - <https://docs.eda.dev>
- EDA Discord - (<https://discord.com/invite/wvCujUdp6P>)

Feedback



#sf25us