



CONTAINERlab

Modern Network Labs with Containerlab

Gordon Gidófalvy, Nokia
HUNOG3, Mátraháza

Virtual Lab-scape: What is Containerlab?



CONTAINERlab

What are virtual networking labs used for?

Learning...

...the basics & beyond



...a new network OS

...new protocols

...new concepts

...at school & university

...for the next cert exam

Testing...

...automation tools

...failure scenarios

...configuration changes

...interoperability

...network migrations

...vendor claims

Simulating...

...proof of concepts

...lab exercises

...full network designs

...your production network

...changes in your network

...your future network

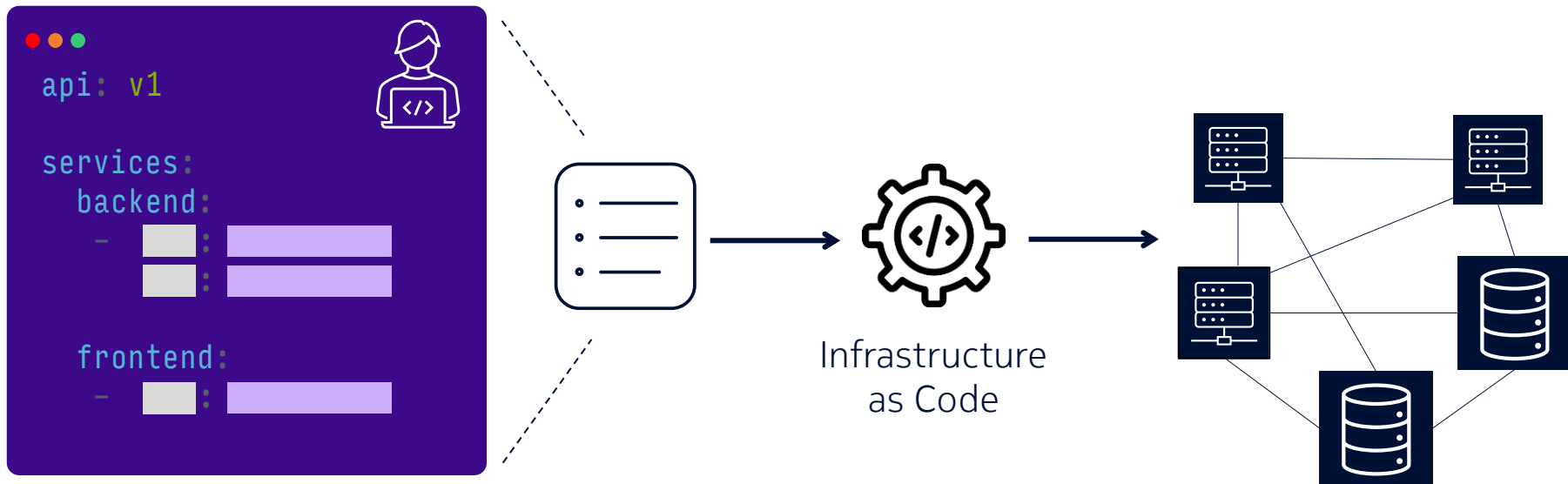
Containerlab

First Class support for containerized Network OSeS

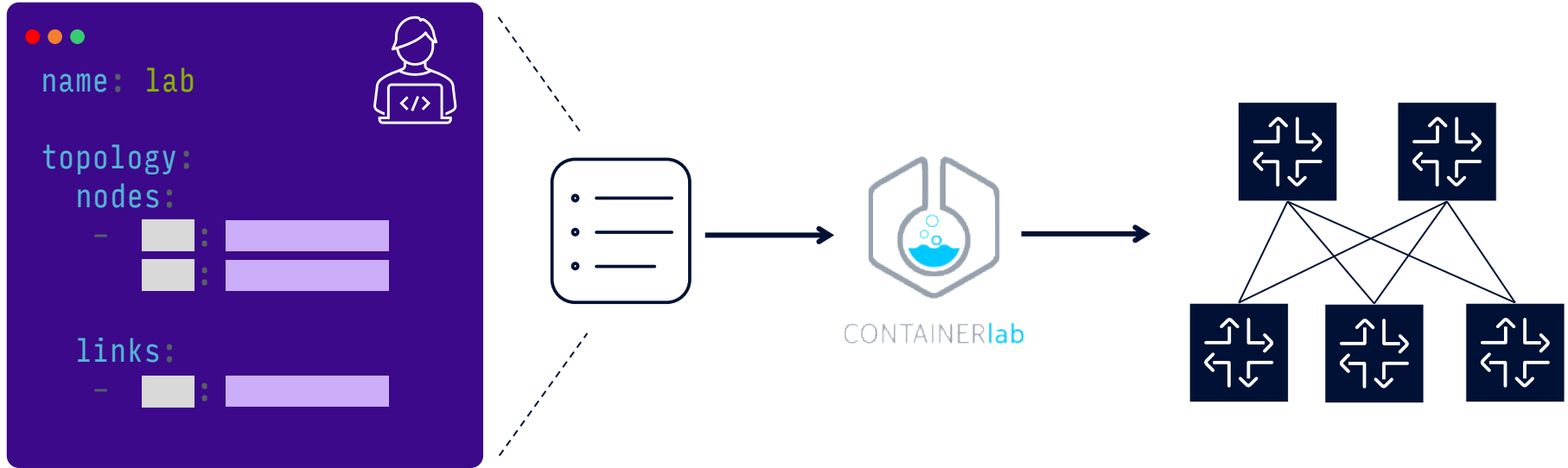
- 1 **Containerlab** provides a simple framework to create and manage container-based networking labs with arbitrary meshed topologies.
- 2 With *multi-vendor* support, integration with VM based routers, scaled topology generation and lab catalog, **Containerlab** delivers a unique Lab-as-Code experience.
- 3 And it runs anywhere where Docker is installed!

How does DevOps deploy services?

Declarative, infrastructure as code approach



Lab as Code with Containerlab



Containerlab

The development story so far



300K

Installations

140+

Open-Source
Contributors

35+

Supported
NOSes

10+

NOSes added
in last 12mo

Containerlab node types: What does Containerlab support?

Containerized Network OSES

- Sourced by the vendor
 - Fast to spin up
 - Small footprint
 - Shareability and versioning
 - Easier to emulate on non-x86
-
- Current trend: **moving away from VMs to containers!**



SR Linux



SR OS (SR-SIM)



cRPD



cEOS



XRd
IOL



cVX



IXIA-c

and others...

Containerlab node types

Regular container images

- **Any container image** can be used in topologies
 - Leveraging existing Docker ecosystem!
- Open-source routing daemons
- Clients & traffic generators
- Hundreds of network-focused software
 - Telemetry, logging stacks
 - DCIM & IPAM
 - Flow collectors
 - Peering management



Get / Set / Subscribe / Collect



Prometheus



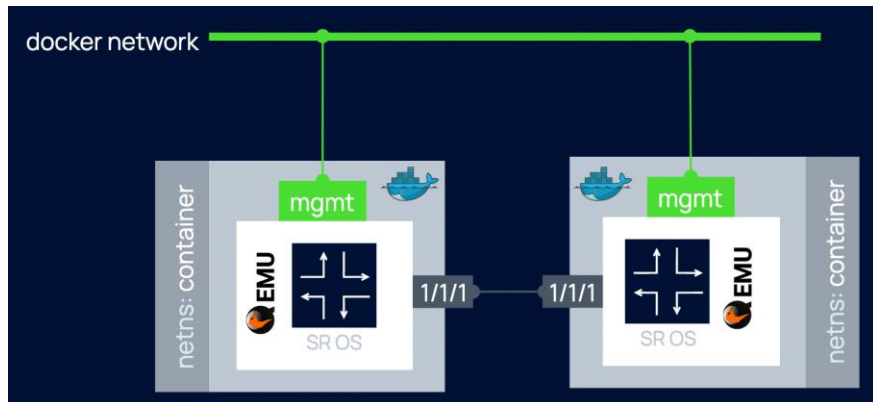
influxdata

Containerlab node types

Virtual machines in a container package

- Traditional Network OS packaged as a VM
- Integrated with Containerlab via vrnetlab* open-source project

* using the github.com/hellt/vrnetlab fork



- Nokia SR OS (vSIM)
- Juniper vMX, vJunosRouter, EVO
- Arista vEOS
- Cisco XRv9k, c8000v, NX-OS, vIOS
- Fortinet Fortigate
- IPInfusion OcNOS
- Palo Alto PAN-OS
- Linux, FreeBSD, OpenWRT, other (N)Oses
- ...and many more!

Getting started with Containerlab

Installation

In a few seconds

Installation options:

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



```
$ curl -sL https://containerlab.dev/setup | sudo -E bash -s "all"
```

• • •

```
$ containerlab version
```

[illegible]

```
version: 0.70.1
```

commit: 3828f87c

```
date: 2025-09-24T10:04:31Z
```

source: <https://github.com/srl-labs/containerlab>

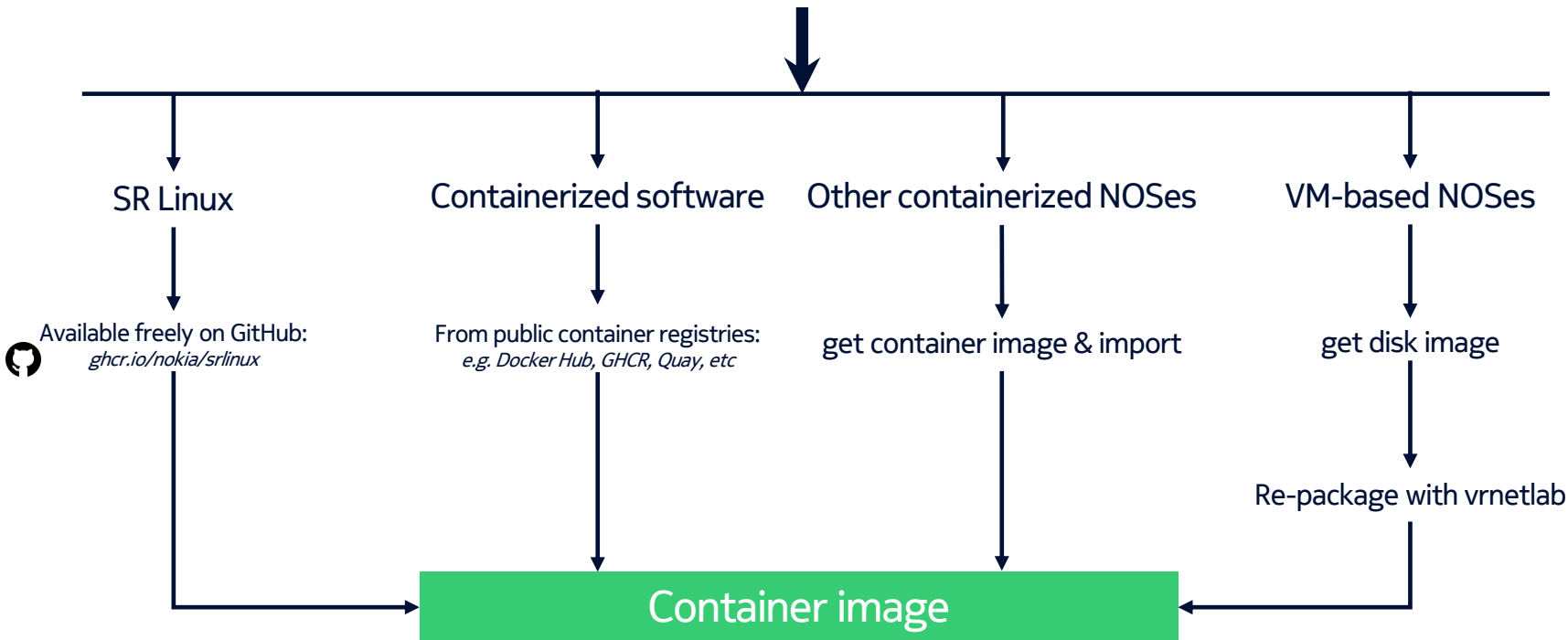
rel. notes: <https://containerlab.dev/rn/0.70.1/#0701>



CONTAINERlab

Container Images

Where do I get one?



Topology file

Writing your own topology

```
name: my-lab
```

```
topology:
```

```
  nodes:
```

```
    leaf1:
```

```
      kind: nokia_srlinux
```

```
      image: ghcr.io/nokia/srlinux:25.7
```

1

Node definition container.
Container name will be the node name.
[Read more](#)

2

Kinds define the flavour of the node, it says if the node is a specific containerized Network OS or something else.
[Read more](#)

3

Image specifies container image to use for this node.
[Read more](#)



[topology definition file](#)

Topology File

Logical view

topology definition

```
demo1.clab.yml
name: demo-frr-srl

topology:
  nodes:
    frr:
      kind: linux
      image: quay.io/frouting/frr:10.2.3
    srl:
      kind: nokia_srlinux
      image: ghcr.io/nokia/srlinux:24.10

  links:
    - endpoints: ["srl:ethernet-1/1", "frr:eth1"]
```

5

logical view



Links connect node interfaces together.
You can generally use NOS-style interface names,
they get automatically translated to Linux interfaces.
[Read more](#)

Let's bring up our new lab!

Deploy and access the nodes



```
$ containerlab deploy -t demo1.clab.yml
```

Name	Kind/Image	State	IPv4/6 Address
clab-demo1-srl	nokia_srlinux ghcr.io/nokia/srlinux:24.10	running	172.20.20.3 3fff:172:20:20::15
clab-demo1-frr	linux quay.io/frrouting/frr:10.2.3	running	172.20.20.2 3fff:172:20:20::14

SR Linux

```
ssh clab-demo1-srl
```

Welcome to the srlinux CLI.

Type 'help' (and press <ENTER>) if you need any help using this.

```
--{ running }--[ ]-
```

```
A:srl#
```

FRR

```
docker exec -it clab-demo1-frr vtysh
```

Hello, this is FRRouting (version 10.2.3_git).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

```
frr#
```

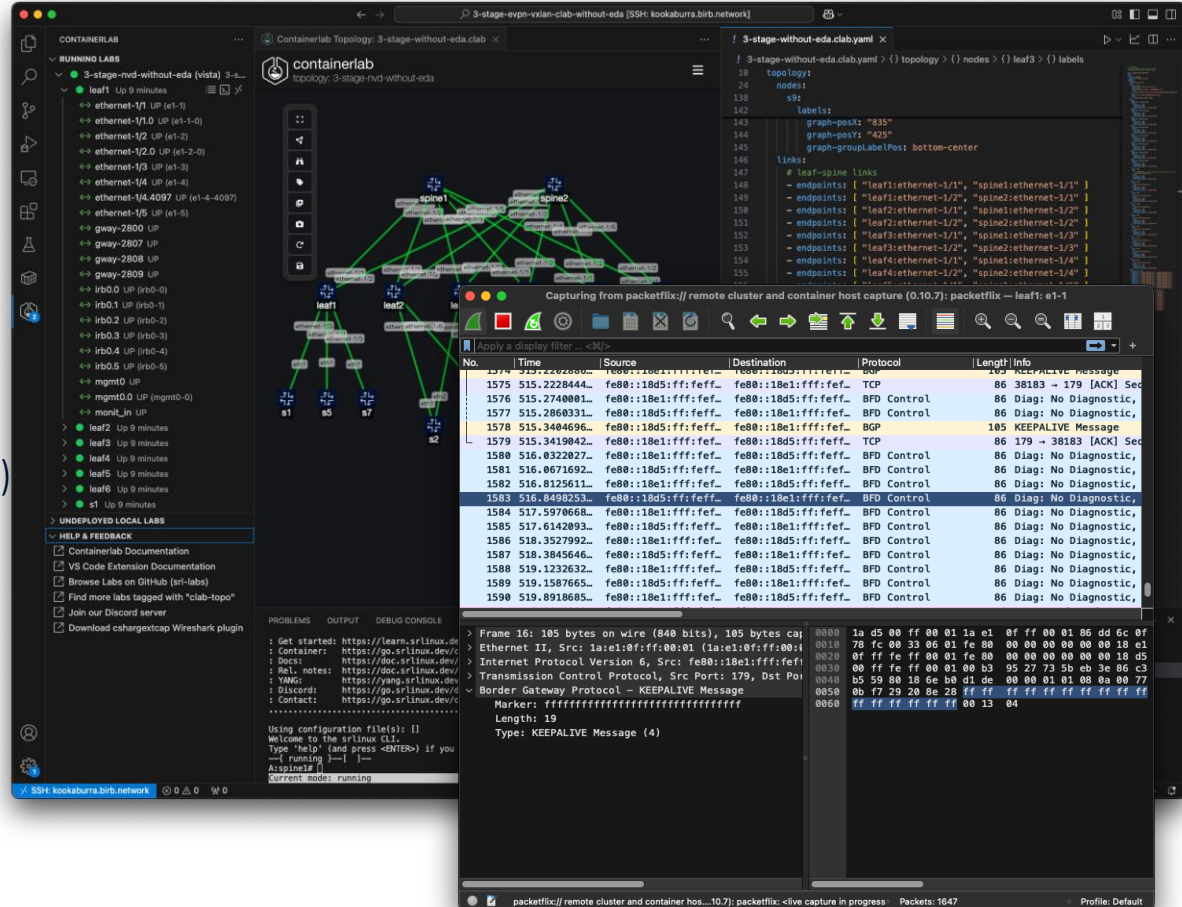

Interacting with Containerlab...

...with a GUI

VS Code Extension: A GUI for Containerlab

Thanks to Florian Schwarz,
Containerlab got a full GUI...
...directly in your code editor!

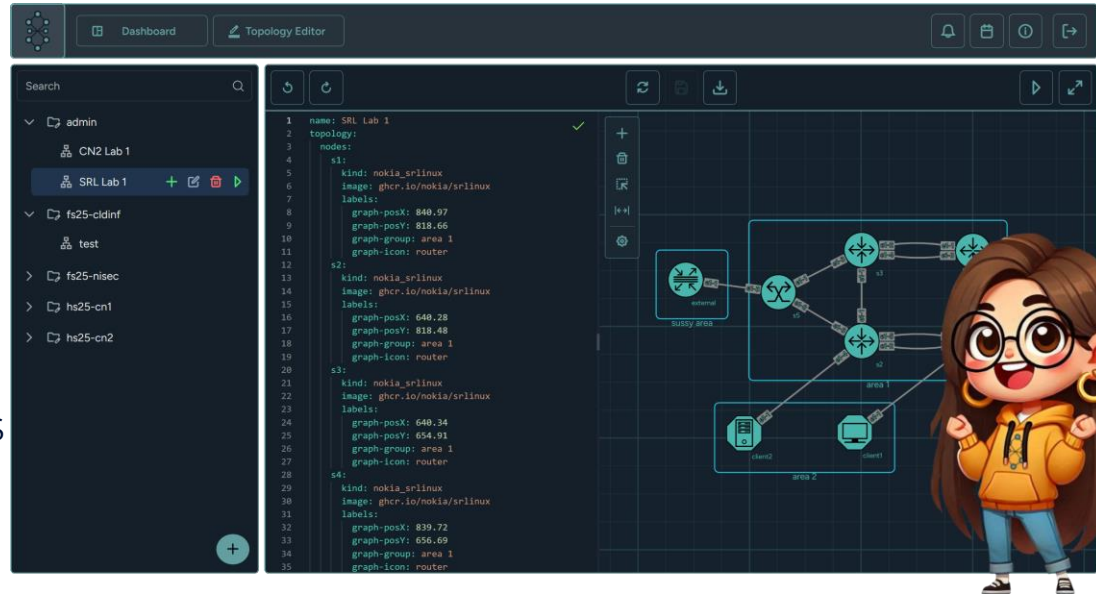
- Available in the VS Code Extension Marketplace
- Feature coverage for most Containerlab functionalities
- Works locally and remotely (SSH)
- Topology viewer AND editor!
- EdgeShark integration for live packet captures



Antimony: A modern, web-based (Container)lab manager

A BSc thesis project by
Kian Gribi and **Tom Stromer** at
Ostschweizer Fachhochschule

- Designed for management of university network labs
- Drag & Drop topology editor
- Full node access via web terminals
- Automatic lab creation & deletion via scheduled triggers
- Built-in user management (instructor & student roles)



Antimony project website



Interacting with Containerlab...

...through an API

Containerlab API Server: Containerlab-as-a-Service!

Run a **network CI/CD pipeline** against a Containerlab API server, like any other REST service!

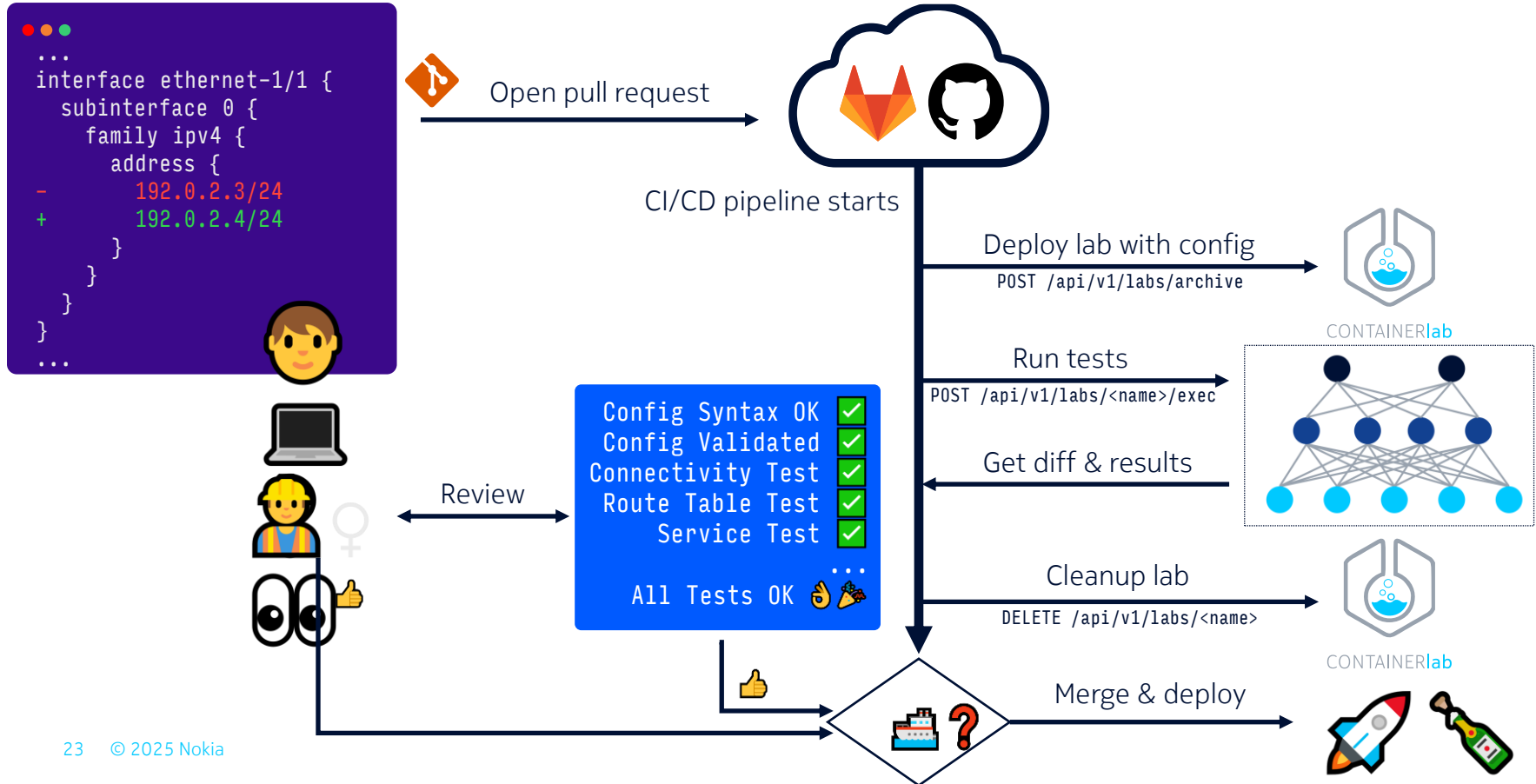
- Brand-new (released in 2025 May)
- Same capabilities as CLI tool
- RBAC and multitenancy-enabled
- Logfile streaming
- Tweak running labs:
 - Run commands in containers
 - Create VXLAN tunnels on the fly
 - Manage link impairments
- SSH Proxying for remote access

```
vista@kookaburra:~$ curl -H "Authorization: Bearer $CLAB_TOKEN" http://localhost:28080/api/v1/labs 2>/dev/null | jq
{
  "3-stage-nvd-without-eda": [
    {
      "name": "leaf1",
      "container_id": "f5d8ab78e967",
      "image": "ghcr.io/nokia/srlinux:24.10.2",
      "kind": "nokia_srlinux",
      "state": "running",
      "status": "Up About an hour",
      "ipv4_address": "172.21.21.11/24",
      "ipv6_address": "N/A",
      "lab_name": "3-stage-nvd-without-eda",
      "labPath": "../clab/nokia-validated-designs/3-stage-evpn-vxlan-clab-without-eda/3-stage-without-eda.clab.yaml",
      "absLabPath": "/home/vista/clab/nokia-validated-designs/3-stage-evpn-vxlan-clab-without-eda/3-stage-without-eda.clab.yaml",
      "group": "",
      "owner": "vista"
    },
    {
      "name": "leaf2",
      "container_id": "2714015f68b4",
      "image": "ghcr.io/nokia/srlinux:24.10.2",
      "kind": "nokia_srlinux",
      "state": "running",
      "status": "Up About an hour",
      "ipv4_address": "172.21.21.12/24",
      "ipv6_address": "N/A",
      "lab_name": "3-stage-nvd-without-eda",
      "labPath": "../clab/nokia-validated-designs/3-stage-evpn-vxlan-clab-without-eda/3-stage-without-eda.clab.yaml",
      "absLabPath": "/home/vista/clab/nokia-validated-designs/3-stage-evpn-vxlan-clab-without-eda/3-stage-without-eda.clab.yaml",
      "group": "",
      "owner": "vista"
    }
  ],
}
```

Containerlab API Server GitHub repo



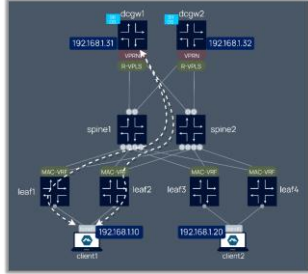
Containerlab API + CI/CD pipelines = (almost a) Digital Twin



Labs built for the community...



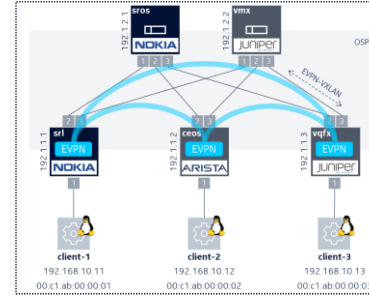
Nokia EVPN Interop



 [srl-labs/nokia-evpn-lab](https://github.com/srl-labs/nokia-evpn-lab)



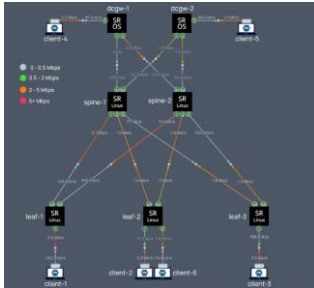
Multivendor EVPN



 [srl-labs/multivendor-evpn-lab](https://github.com/srl-labs/multivendor-evpn-lab)



SR Linux & SROS Telemetry

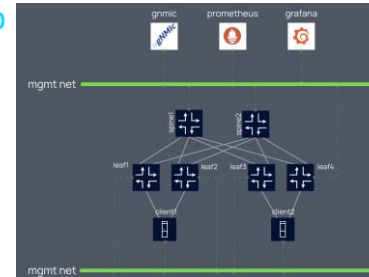


 [srl-labs/srl-sros-telemetry-lab](https://github.com/srl-labs/srl-sros-telemetry-lab)

CONTAINERlab



SR Linux Oper-Group

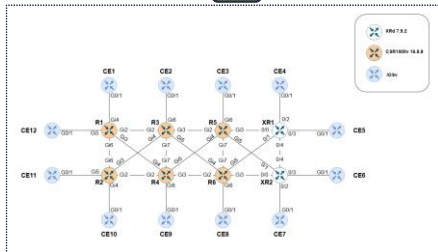


 [srl-labs/opergroup-lab](https://github.com/srl-labs/opergroup-lab)

And by the community!



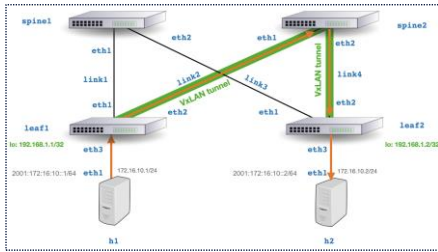
Andrew Ohanian's
full CCIE SPv5.1 labs



[andrewohanian/ccie-spv5.1-labs](https://github.com/andrewohanian/ccie-spv5.1-labs)



sFlow-RT telemetry lab



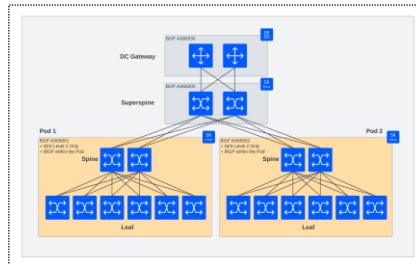
[sflow-rt/containerlab](https://github.com/sflow-rt/containerlab)



CONTAINERlab



NetBox + NRX + Containerlab



[srl-labs/netbox-nrx-clab](https://github.com/srl-labs/netbox-nrx-clab)



OSS routing interoper labs
w/ FRR, BIRD, Holo

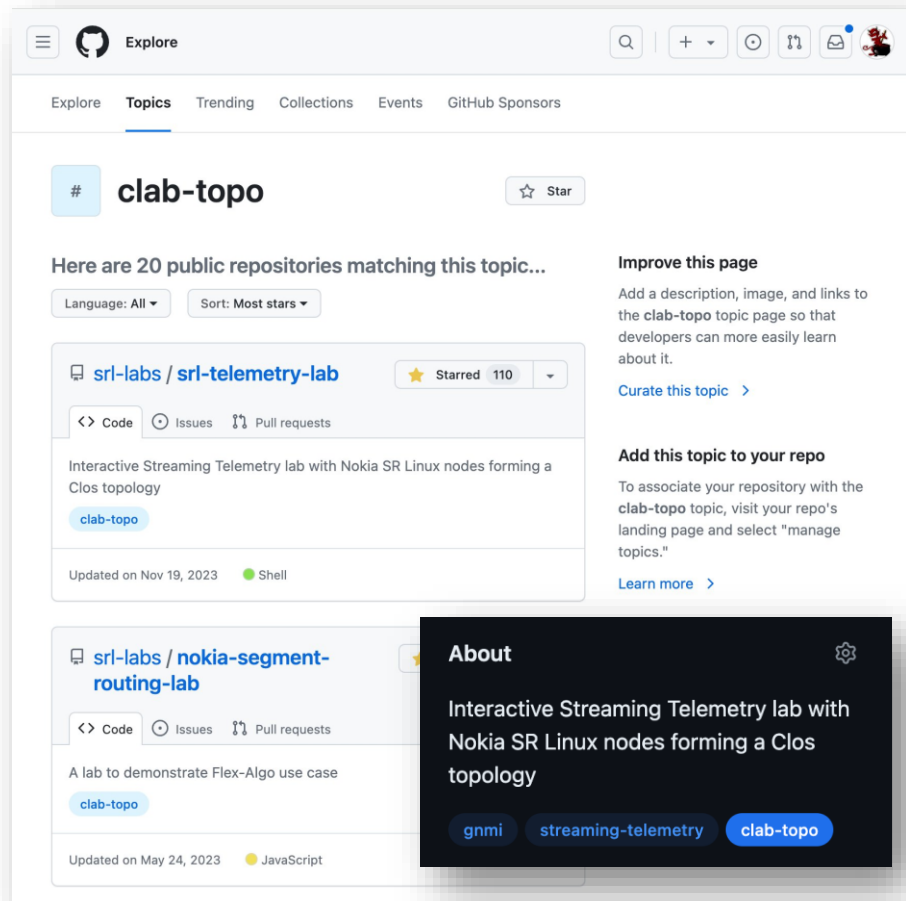


[holo-routing/containerlab-topologies](https://github.com/holo-routing/containerlab-topologies)

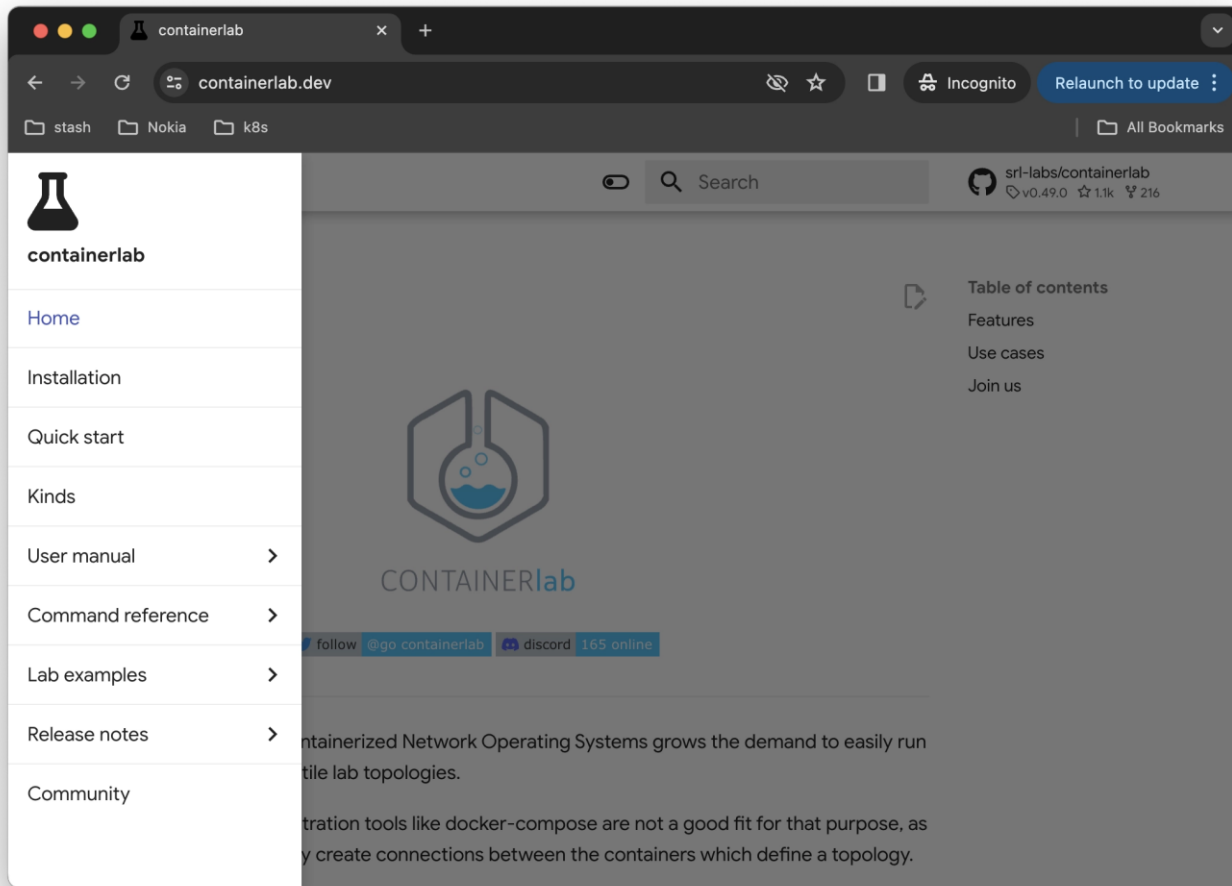
A community collection of Containerlabs

Add **clab-topo** topic to your repo today!

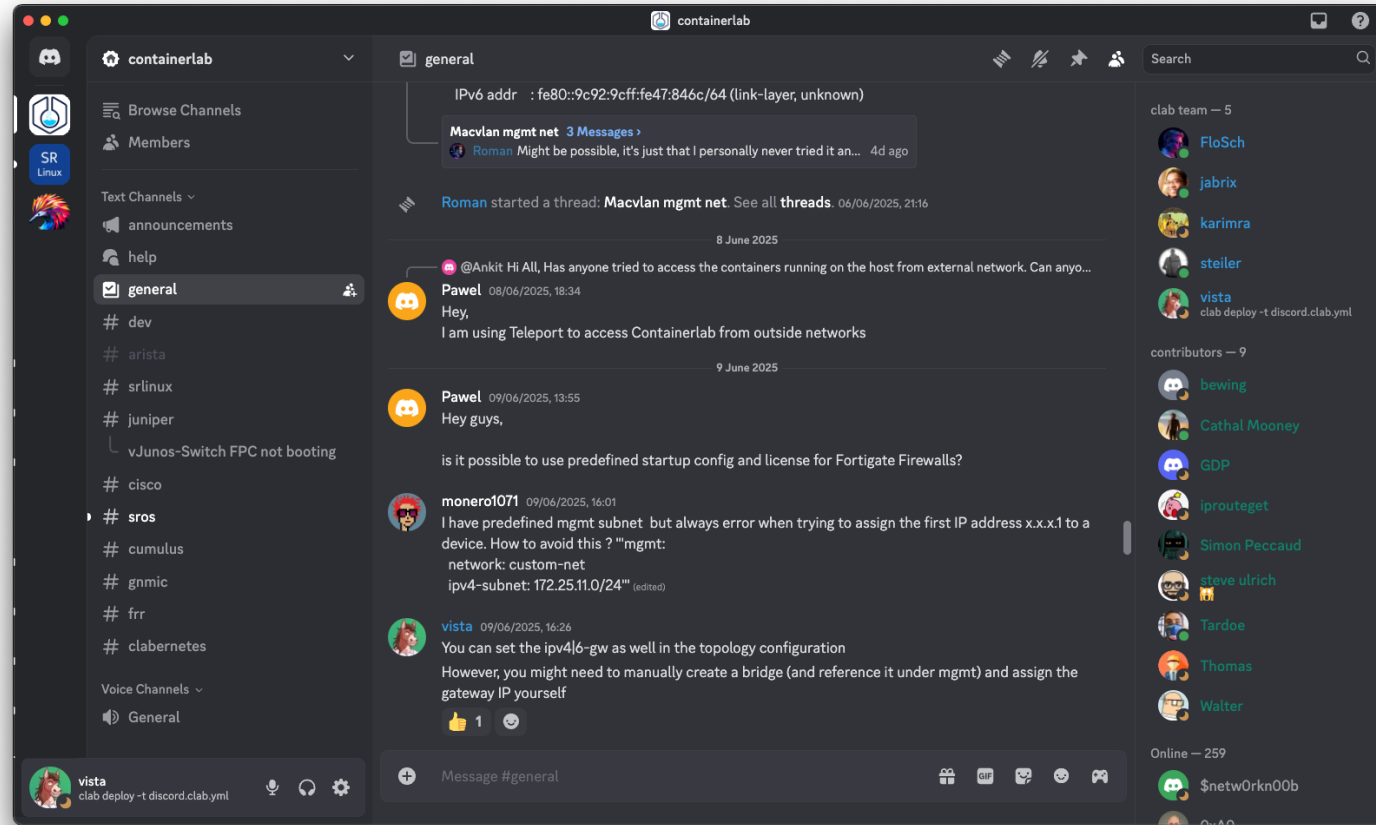
- Make your labs easily discoverable
- Each repo represents a runnable topology: with configs & topology included
- Publish your lab and others will see it: **show off your expertise!**
- Contribute back to the community: **It's a group effort!**



Containerlab Docs



Containerlab Discord



Questions?

Thank you for your attention!

NOKIA



Containerlab Docs



Containerlab Discord