Axians TechUpdate 2025

Containerlab

Florian Schwarz

Virtual Labs

A better alternative for physical labs !

1. Cost-Effectiveness
2. Flexibility and Scalability
3. Safe Testing Environment
4. Rapid Prototyping
5. Standardization and Repeatability
6. Remote Access
7. Integration with Automation Tools
8. Support for Diverse Technologies
9. Improved Learning Outcomes

ContainerLab

First Class support for containerized Network OSes

1. The ever-growing number of containerized Network OSes requires a tool that enables building labs topologies in a container-native way

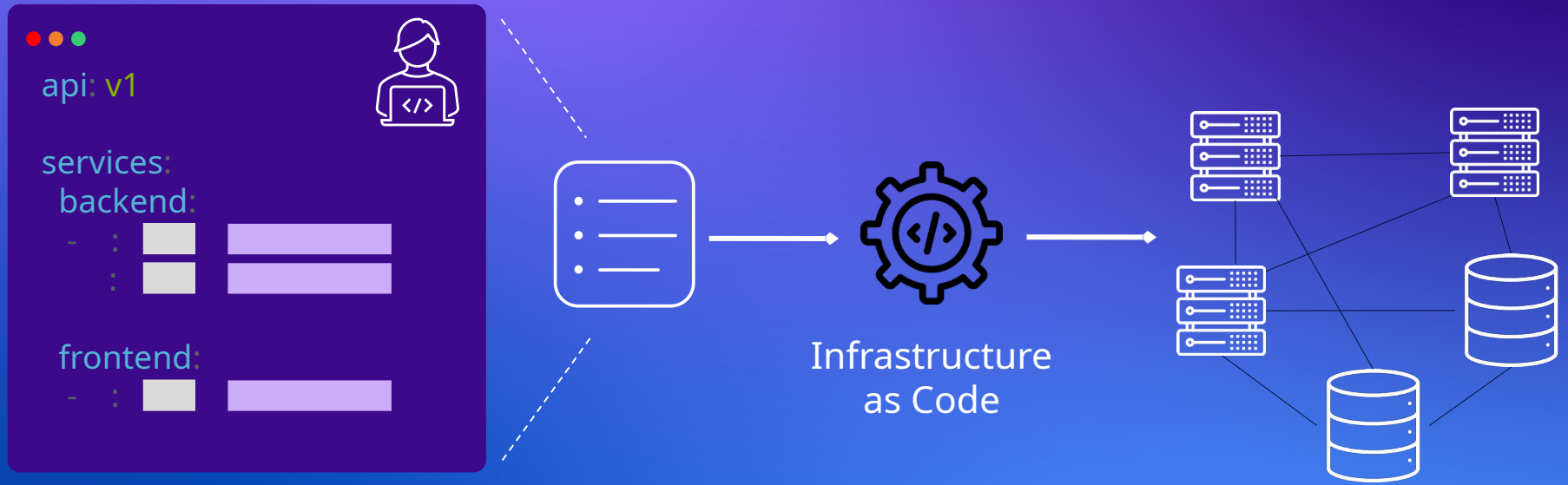
2. ContainerLab provides a simple framework to create and manage container-based networking labs with arbitrary meshed topologies.

3. With Multivendor support, integration with VM based routers, scaled topology generation and lab catalog, ContainerLab delivers a unique Lab-as-Code experience

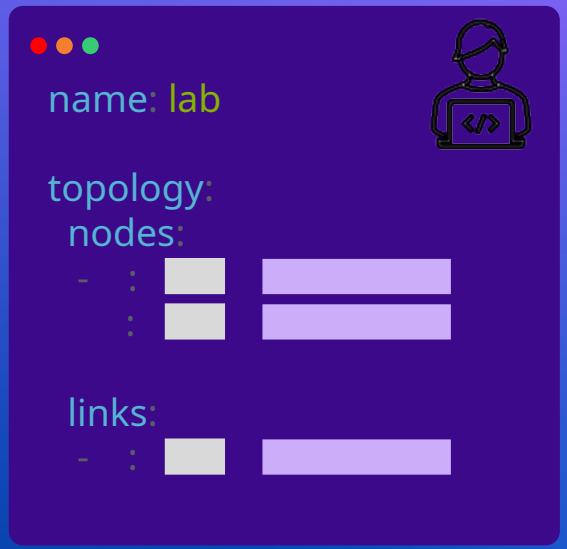
4. And it runs anywhere where docker is installed

How they deploy things over there?

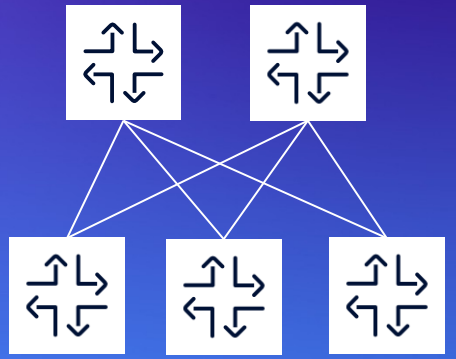
Declarative, infrastructure as code approach



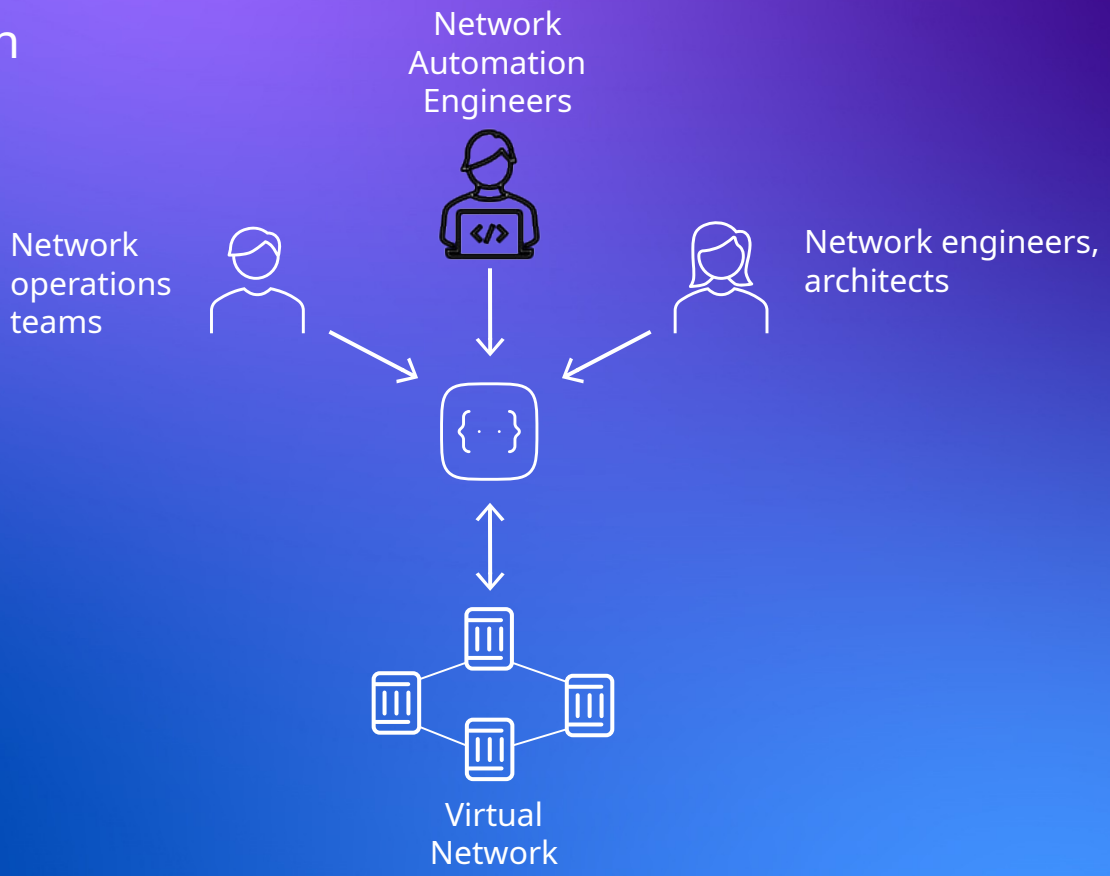
Lab as code with Containerlab



Infrastructure
as Code

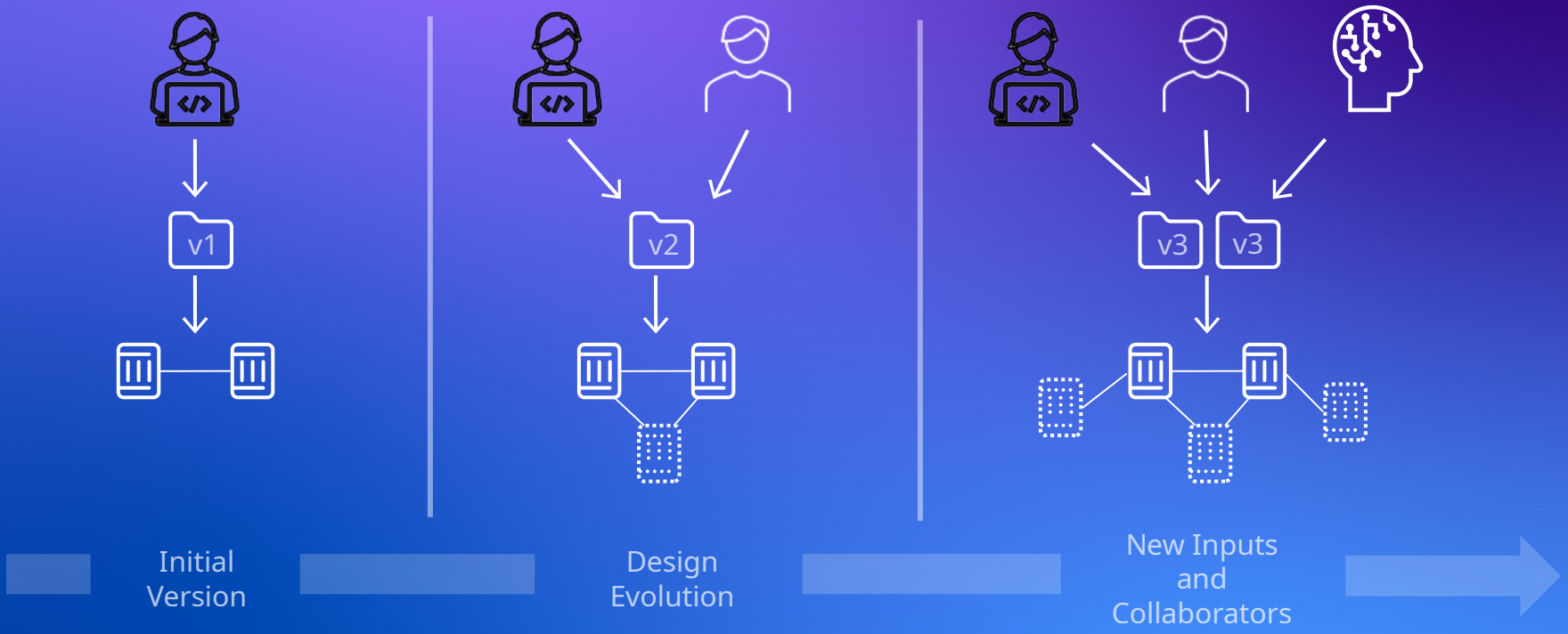


Why Lab as Code?
#1 - Collaboration

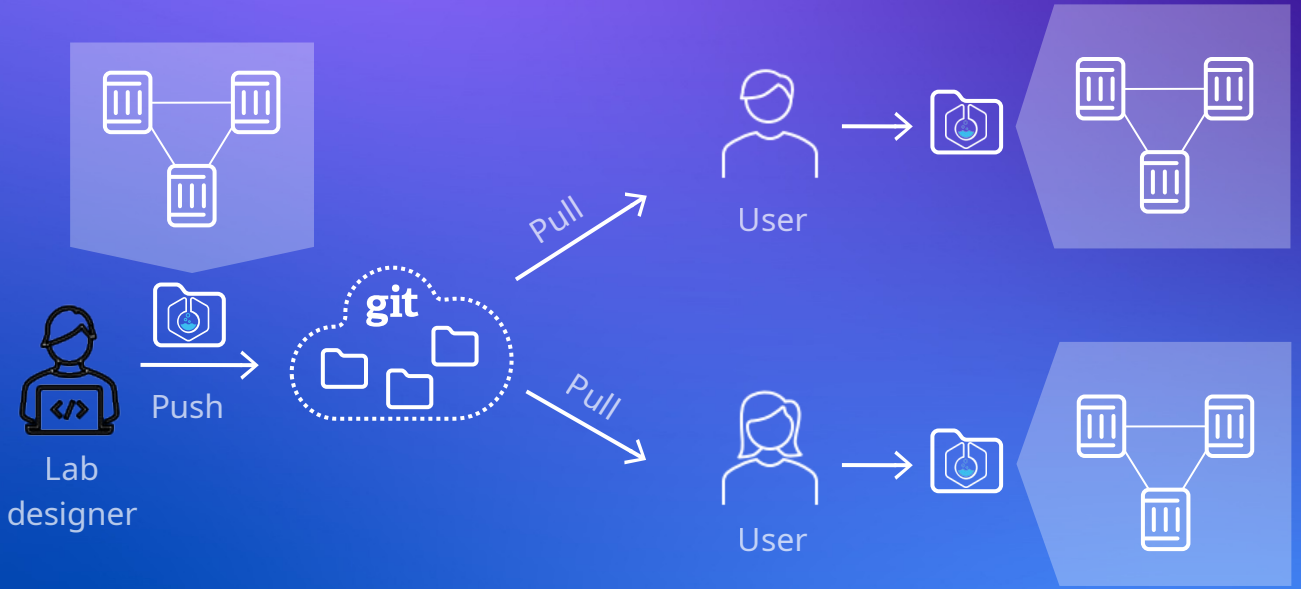


Why Lab as Code?

#2 - Versioning



Why Lab as Code?
#3 - Sharing



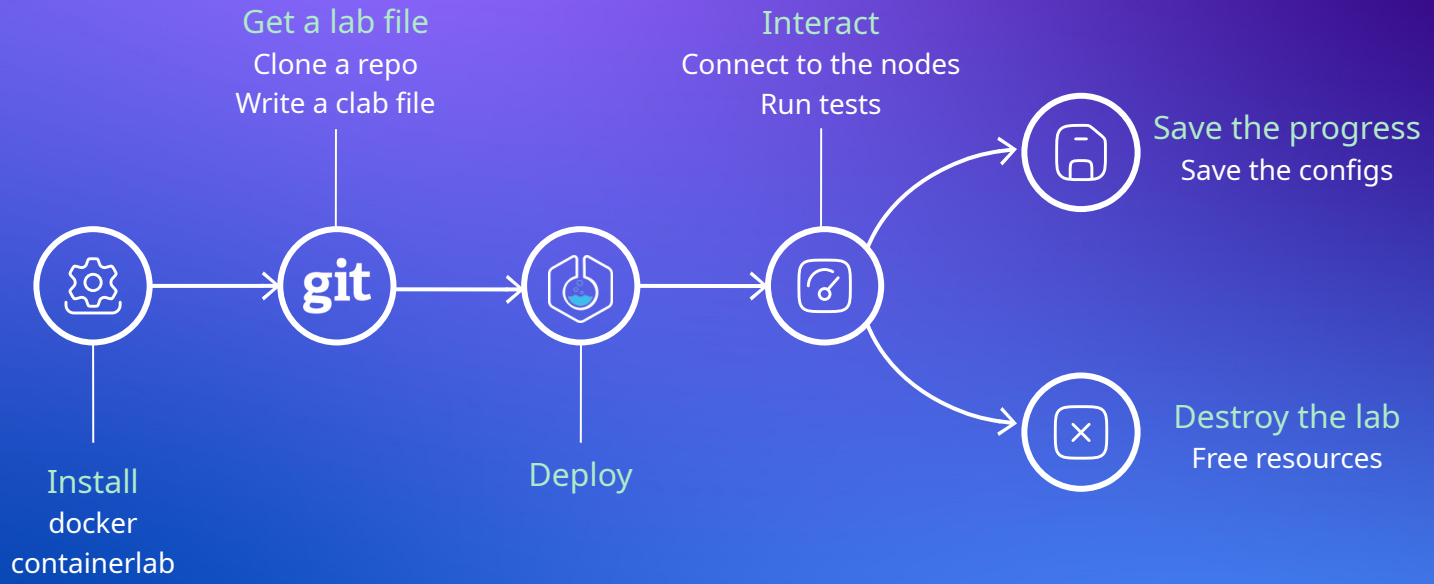
Containerlab
4-year checkpoint

450K Installations	140+ Contributors
32 Supported NOSes	24 Releases in 12mo

They use Containerlab

Containerlab

The Workflow



Installation

In a few seconds

Installation options: https://containerlab.dev/install/

Containerlab node types

Containerized Network OSes

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

SR Linux

cRPD

cEOS

XRd

cVX

IXIA-c

and others...

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

Containerlab node types

Regular container images

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

gNMIc

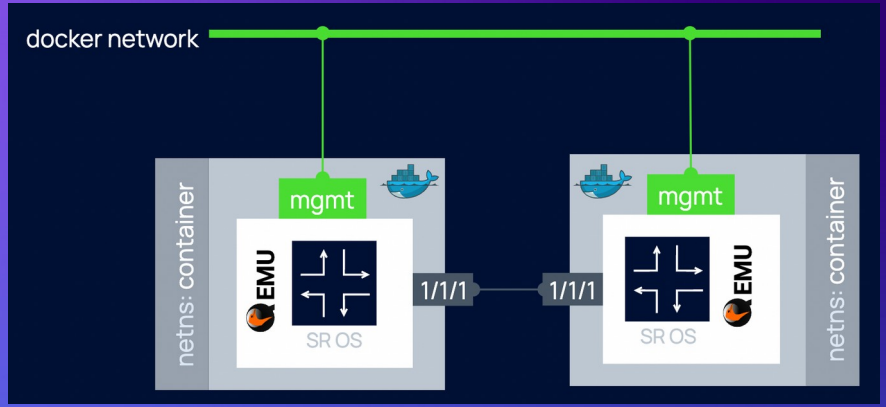
Get / Set / Subscribe / Collect

Where are my Good old VM-base Network OSes?

Containerlab node types

Virtual machines in a container package

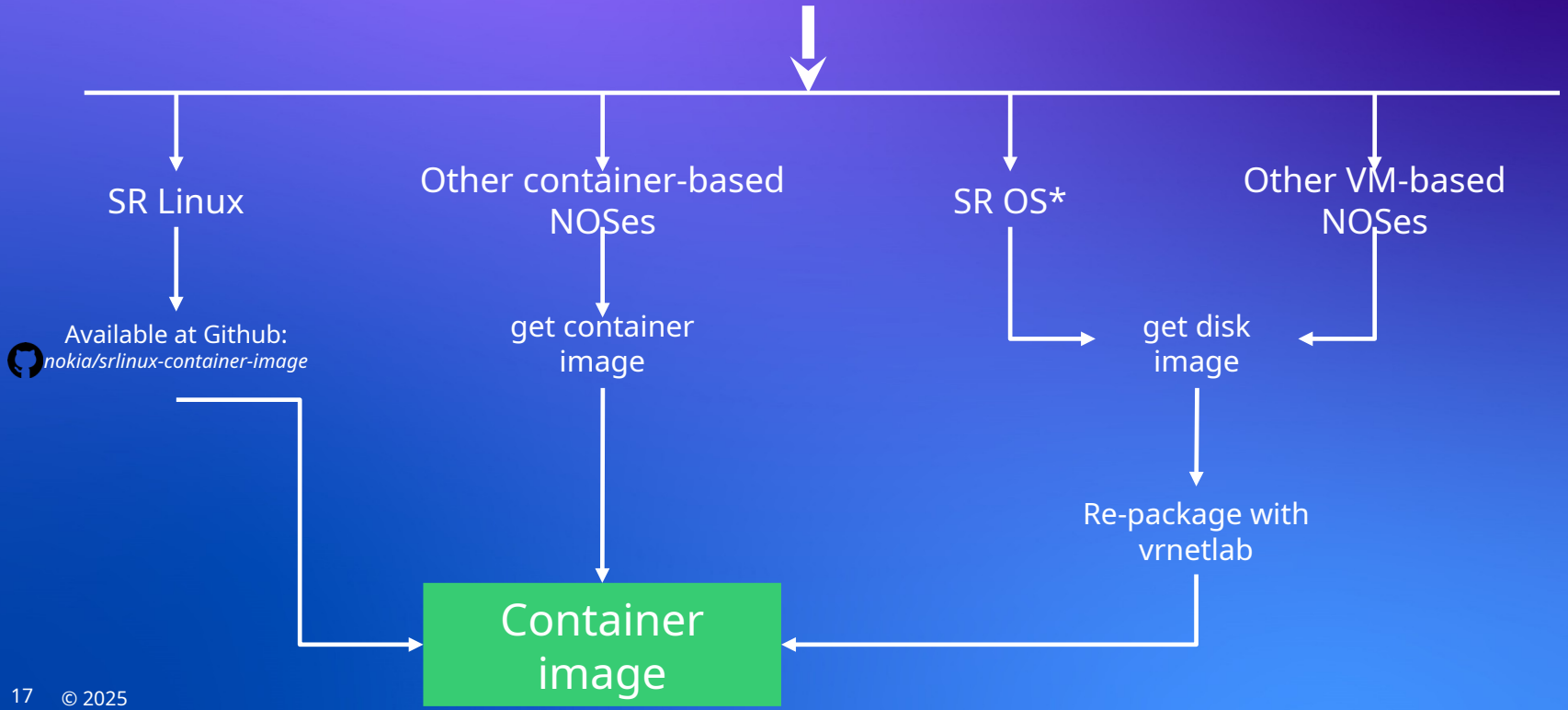
- Traditional Network OS packaged as a VM
- Integrated with containerlab via vrnetlab open-source project
- Onboard existing VM-based NOSes



Juniper vMX, vswitch, EVO
Arista vEOS
Cisco XRv9k, c8000v, NX-OS
Fortinet Fortigate
IPInfusion OcNOS
Palo Alto PAN-OS

* using srl-labs/vrnetlab project

Container Images
Where do I get one?



Topology file
Writing your own topology

```
name: my-lab

topology:
  nodes:
    router1:
      kind: nokia_sros
      image: sros:24.3.R1
      startup-config: r1.cfg
```

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
4. Path to a startup configuration file (optional).
   Read more

topology definition file

Topology File
Logical view

topology definition

demo1.clab.yml

```
name: demo1

topology:
  nodes:

    srl:
      kind: nokia_srlinux
      image: ghcr.io/nokia/srlinux:24.3.2

    sros:
      kind: nokia_sros
      image: sros:24.3.R2-1
      license: license-sros24.txt

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

logical view

Containerlab CLI: Deploy

Deploy and access the nodes

$ containerlab deploy –t demo1.clab.yml

```
+---+-----------------+-------------+-----------------------------+---------+---------+---------------+---------------------+
| # |      Name       | Container ID |          Image             |  Kind   |  State  | IPv4 Address  |    IPv6 Address     |
+---+-----------------+-------------+-----------------------------+---------+---------+---------------+---------------------+
| 1 | clab-demo1-srl  | b241e0db0a79 | ghcr.io/nokia/srlinux:22.6.2 | srl     | running | 172.20.20.3/24 | 2001:172:20:20::3/64 |
| 2 | clab-demo1-sros | 62610c5026f3 | sros:21.10.R2                | vr-sros | running | 172.20.20.2/24 | 2001:172:20:20::2/64 |
+---+-----------------+-------------+-----------------------------+---------+---------+---------------+---------------------+
```

SR Linux

ssh admin@clab-demo1-srl

Welcome to the srlinux CLI.
Type 'help' (and press <ENTER>) if you need any help using this.
--{ running }--[ ]—
A:srl#

SR OS

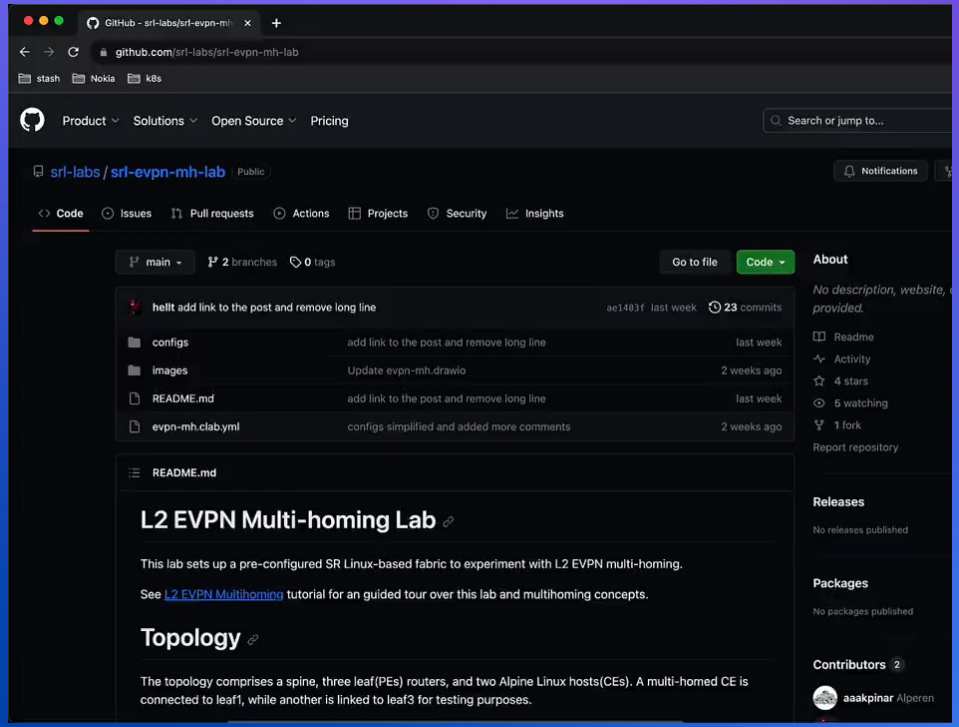
ssh admin@172.20.20.2

admin@172.20.20.2's password:

SR OS Software
Copyright (c) Nokia 2021. All Rights Reserved.

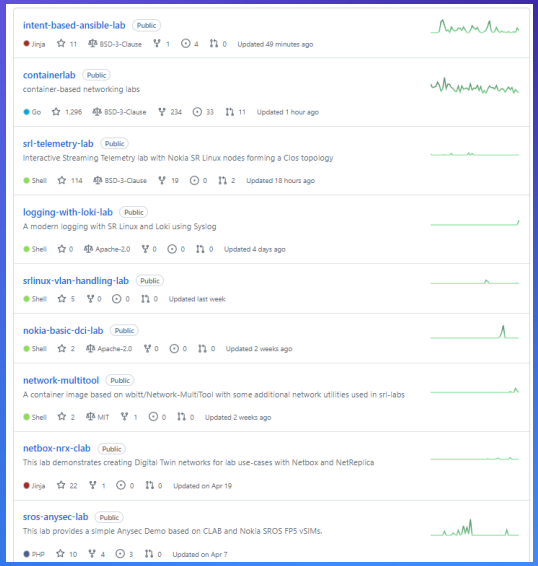
Containerlab CLI: Deploy Remote

Deploy from github



Eliminate the step of git clone:

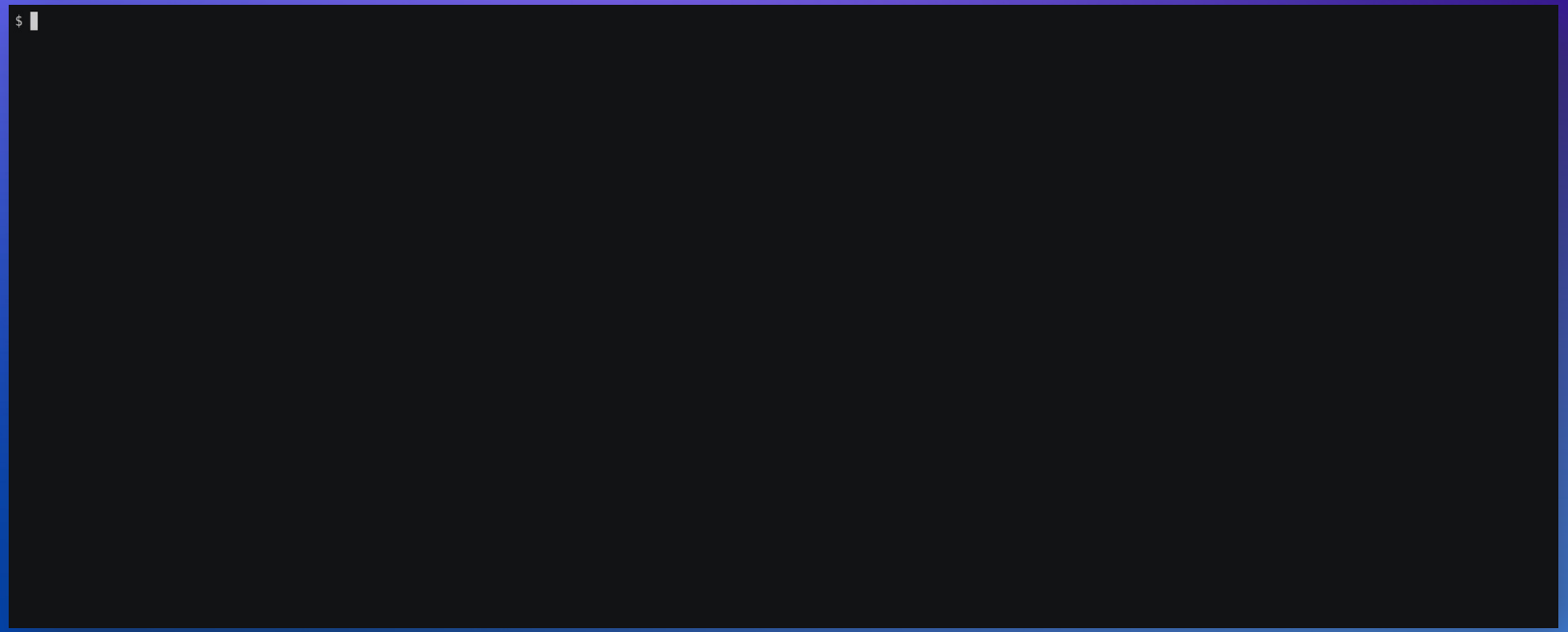
```
$ clab deploy –t https://github.com/srl-labs/srl-evpn-mh-lab
```



Containerlab CLI

Inspect

containerlab **inspect** --topo <path to clab file> or containerlab **inspect** --all



Containerlab CLI
Save

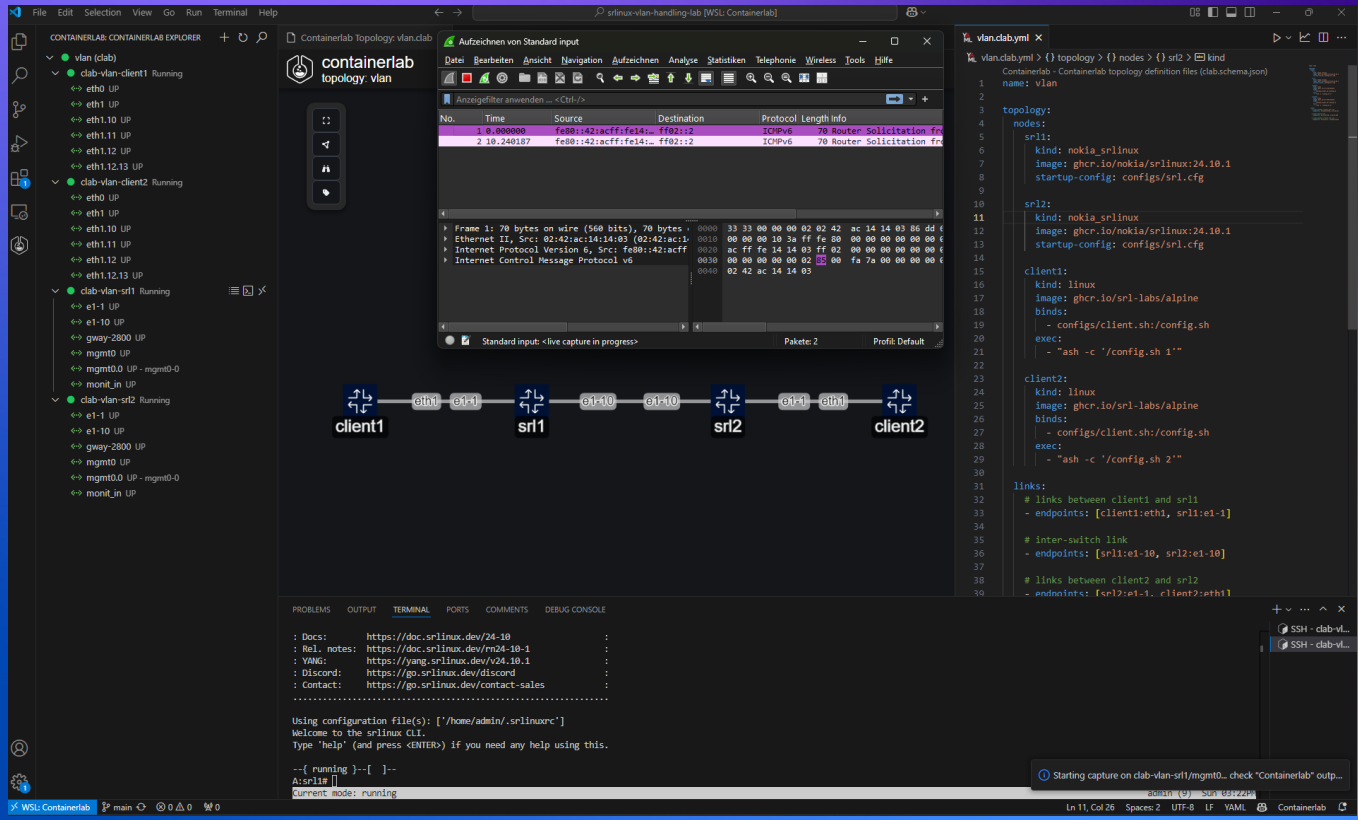
```
# executed in a directory where demo1.clab.yml is present
❯ containerlab save

INFO[0000] Parsing & checking topology file: demo1.clab.yml
INFO[0000] saved sros running configuration to startup configuration file
INFO[0001] saved SR Linux configuration from srl node.
```

Perform lab-scoped configuration save in a single sweep

Containerlab CLI
Destroy

Command	Effect
containerlab **destroy** [--topo <path to clab file>]	Removes containers for a given topology. Leave lab directory intact
containerlab **destroy** [--topo <path to clab file>] --cleanup	Removes containers and lab directory for a given topology.
containerlab **destroy** --all	Remove containers for all running labs. Keep lab directories.
containerlab **destroy** –-all --cleanup	Remove containers and lab directories for all running labs.

What’s else?
Vscode Extension



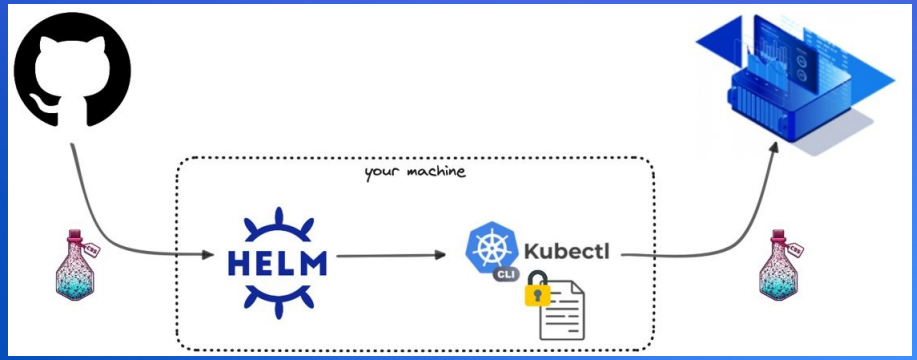
But can it scale?

What’s else?
Clabernetes: Scale-out your virtual lab

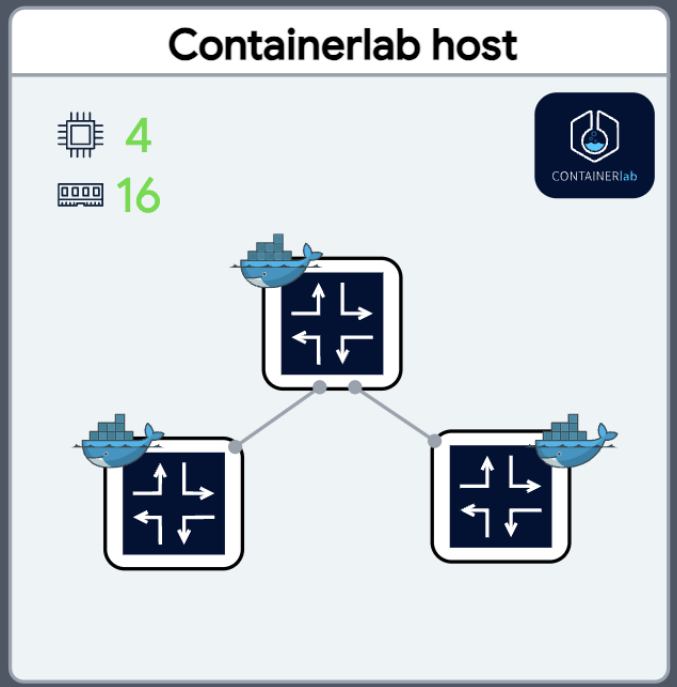
Clabernetes deploys Containerlab topologies into a k8s cluster.
Makes it possible to distribute a virtual lab to multiple servers.

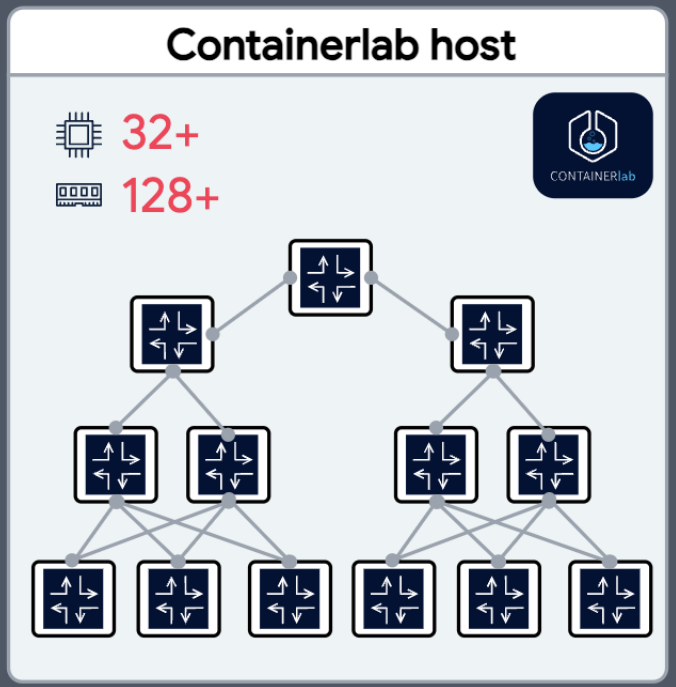
$ kubectl get pods --namespace c9s-vlan -o wide

```
NAME                READY   STATUS    RESTARTS   AGE   IP           NODE          NOMINATED NODE   READINESS GATES
client1-5c4698f68c-v4z2n   1/1     Running   0          19m   10.244.1.15   c9s-worker    <none>           <none>
client2-6dfc49bc8f-hpkd4   1/1     Running   0          19m   10.244.2.15   c9s-worker2   <none>           <none>
srl1-78bdc85795-l9bl4      1/1     Running   0          19m   10.244.1.14   c9s-worker    <none>           <none>
srl2-7fffcdb79-vxfn9       1/1     Running   0          19m   10.244.2.16   c9s-worker2   <none>           <none>
```

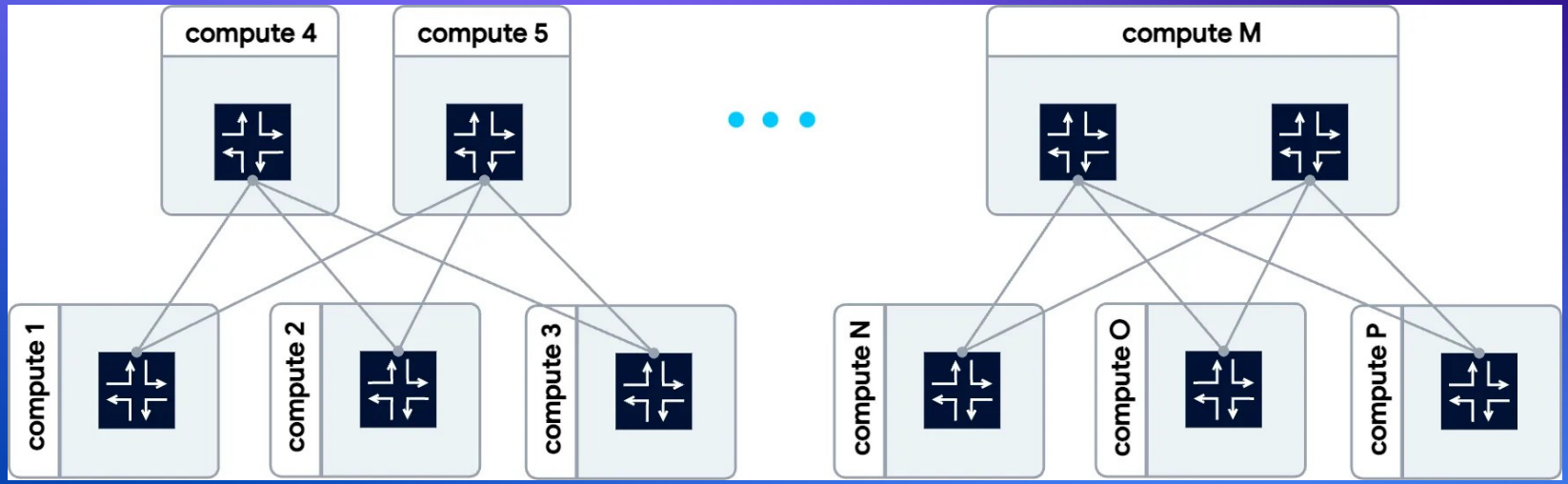


Vertical scaling is costly

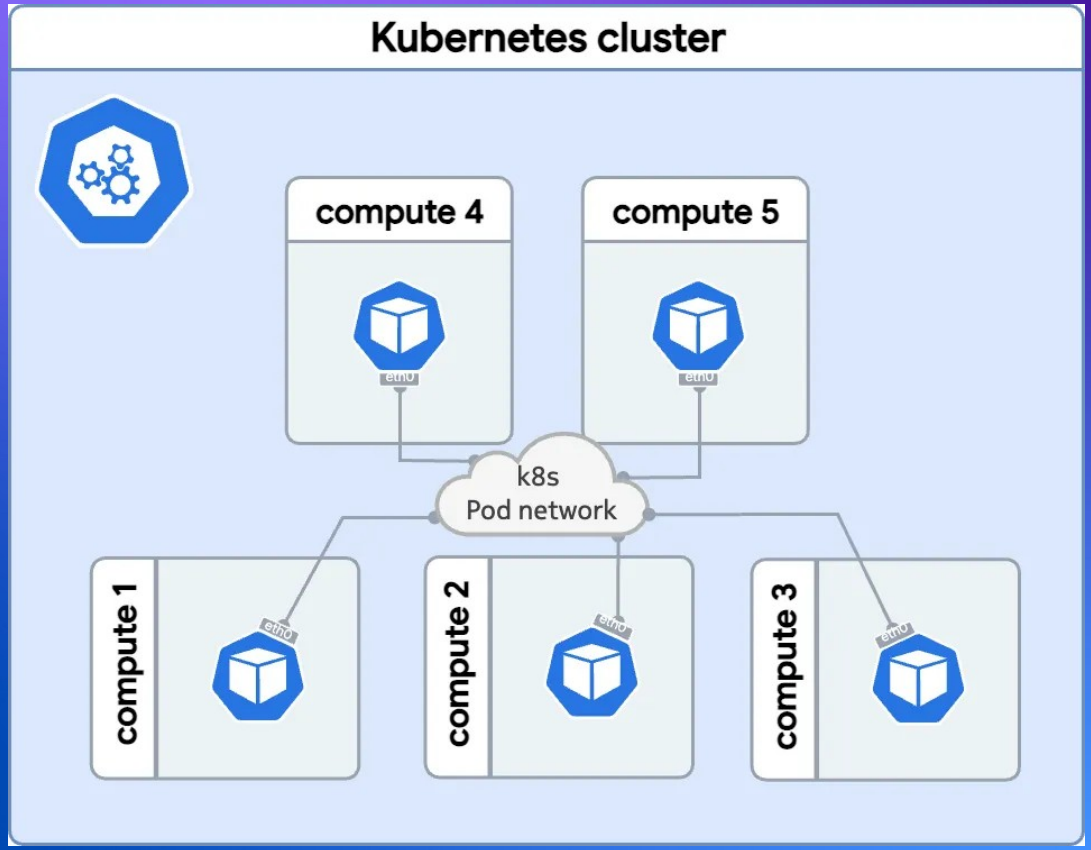




Horizontal scaling is cheaper

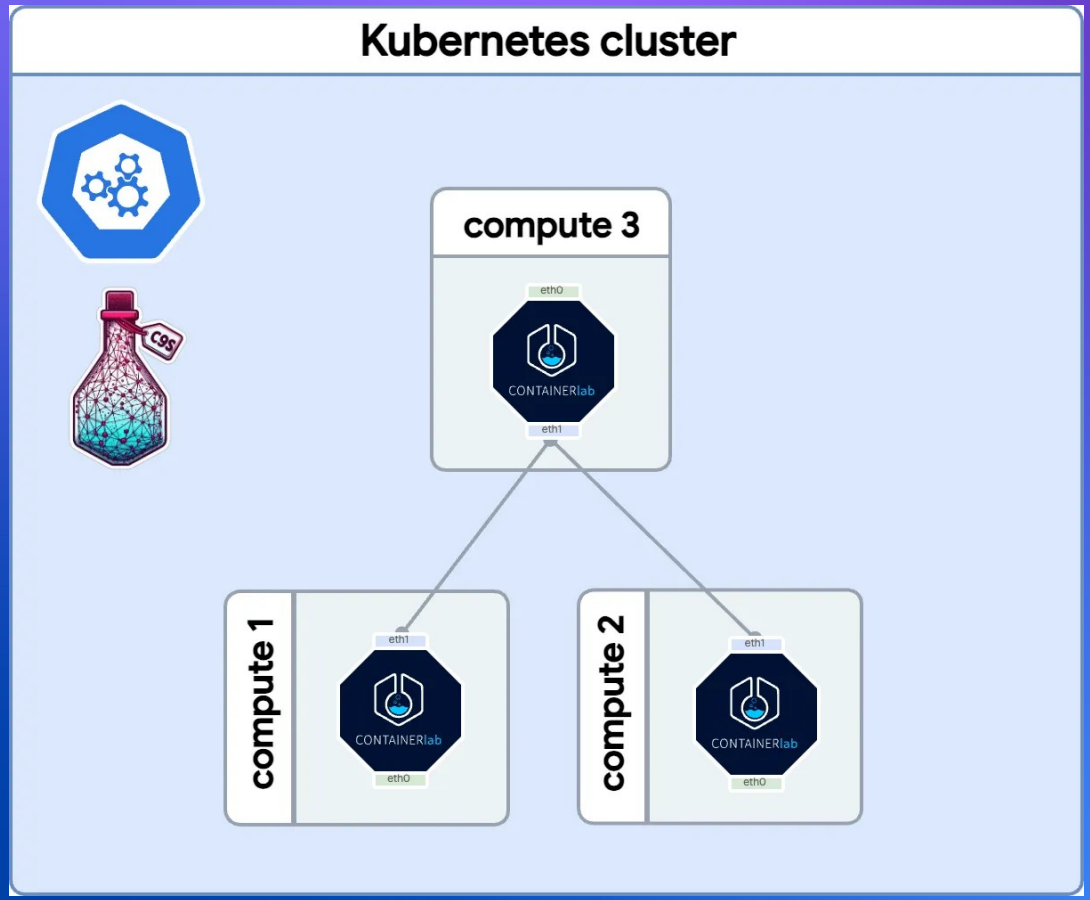


If only we had a system to schedule workload on top of workers…



Enter Clabernetes

containerlab.dev/manual/clabernetes/



- Same topology structure
- Same supported NOSes
- With horizontal scale

Clabernetes Goals

Containerlab, just in K8s

No K8s PhD required

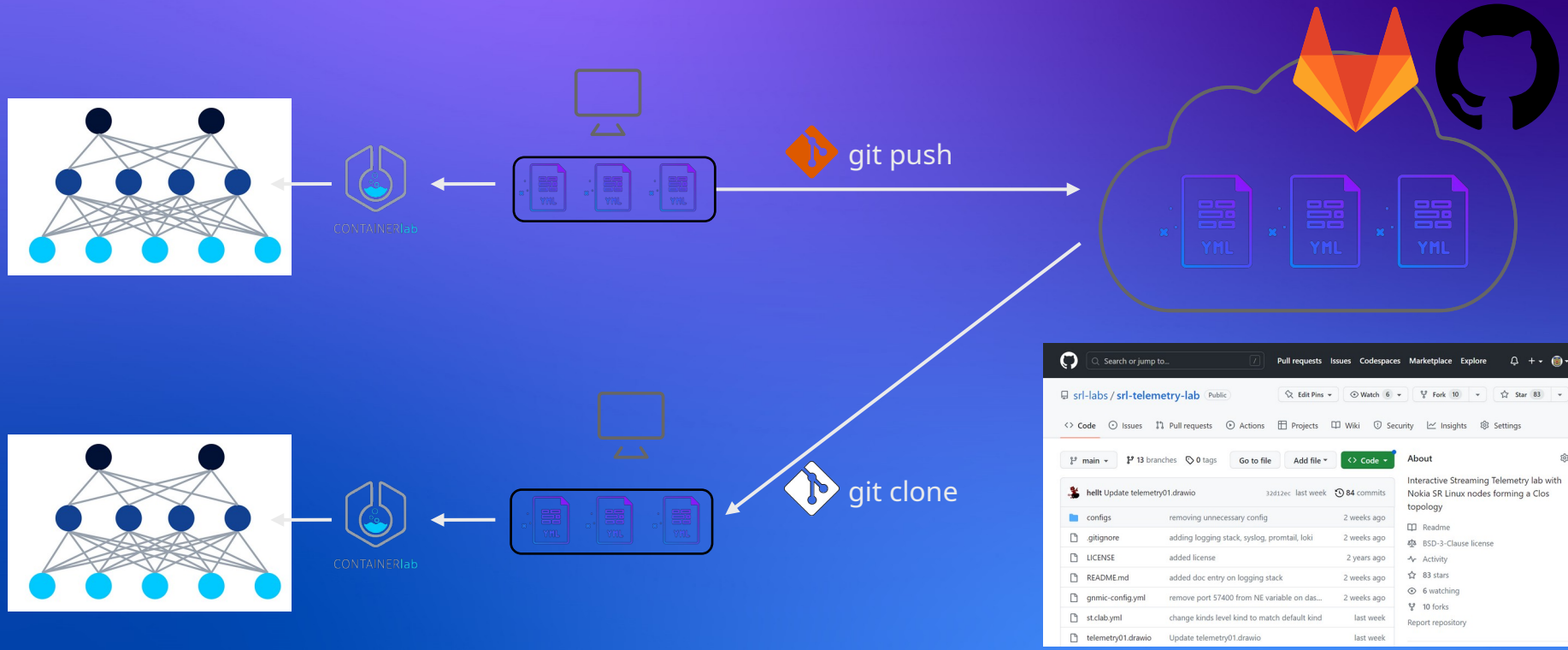
Open

Any K8s Cluster

Light ... other than K8s part

!? Fun, Profit!?

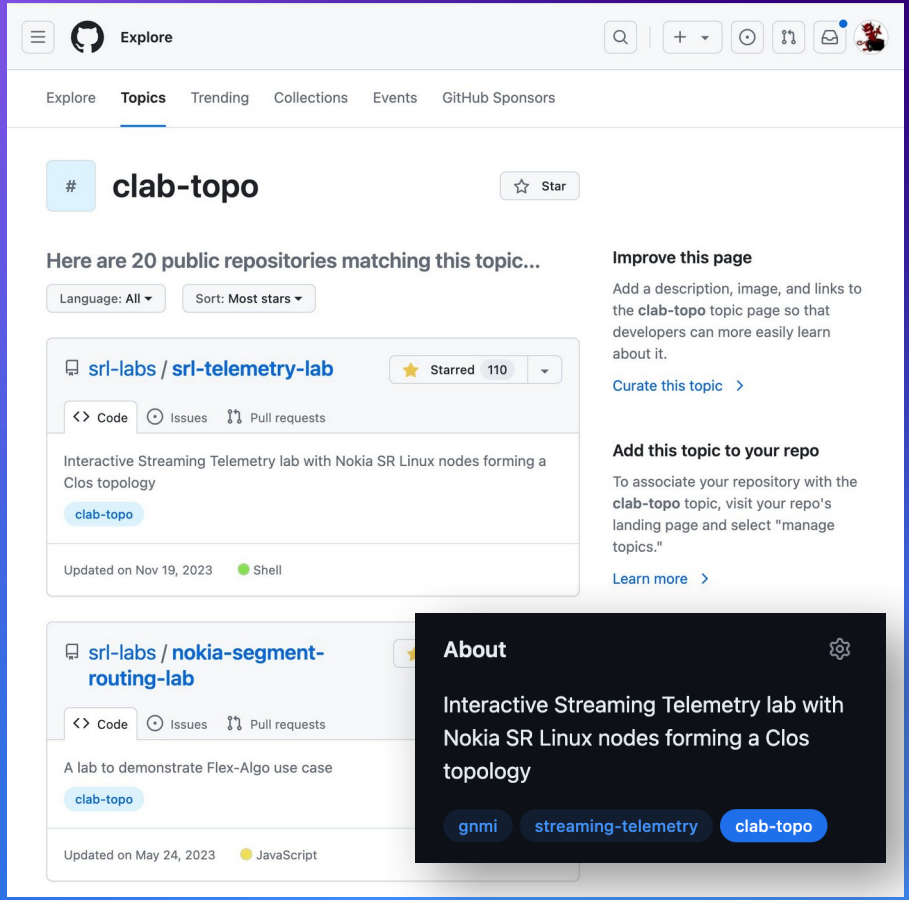
Share your labs
Lab As Code



Distributed collection of runnable labs

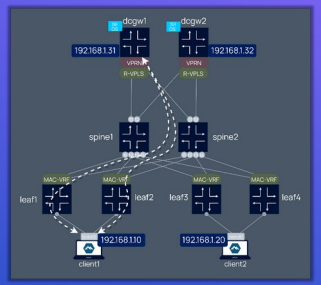
Add clab-topo topic to your repo

- Make your labs easily discoverable in a distributed way
- Each repo represents a runnable topology
- Publish your lab and others will see it!



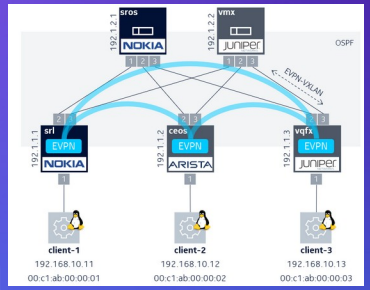
Other labs built for the community

Nokia EVPN Interop



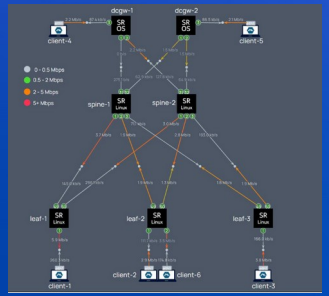
srl-labs/nokia-evpn-lab

Multivendor EVPN



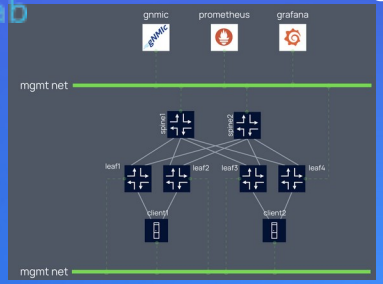
srl-labs/multivendor-evpn-lab

SR Linux & SROS Telemetry



srl-labs/srl-sros-telemetry-lab

SR Linux Oper-Group

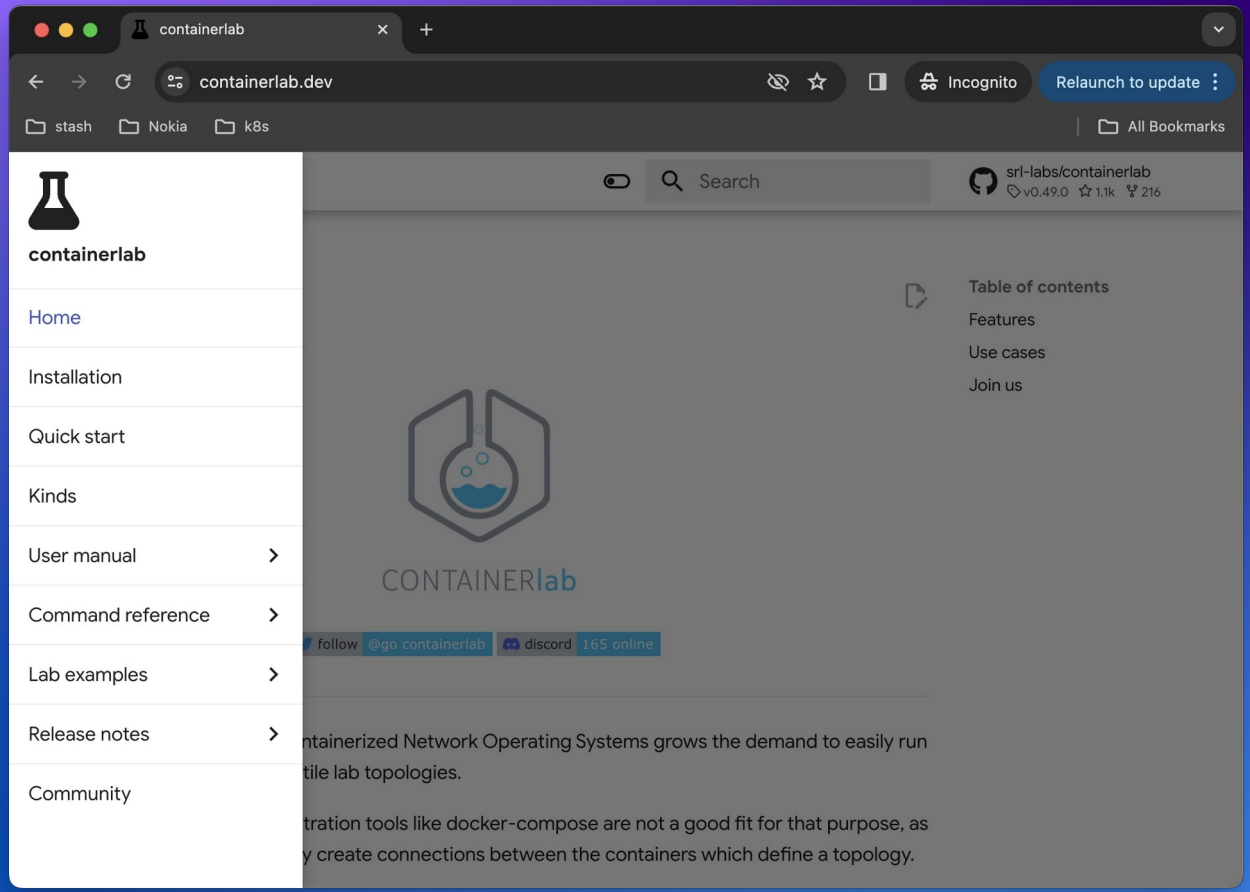


srl-labs/opergroup-lab

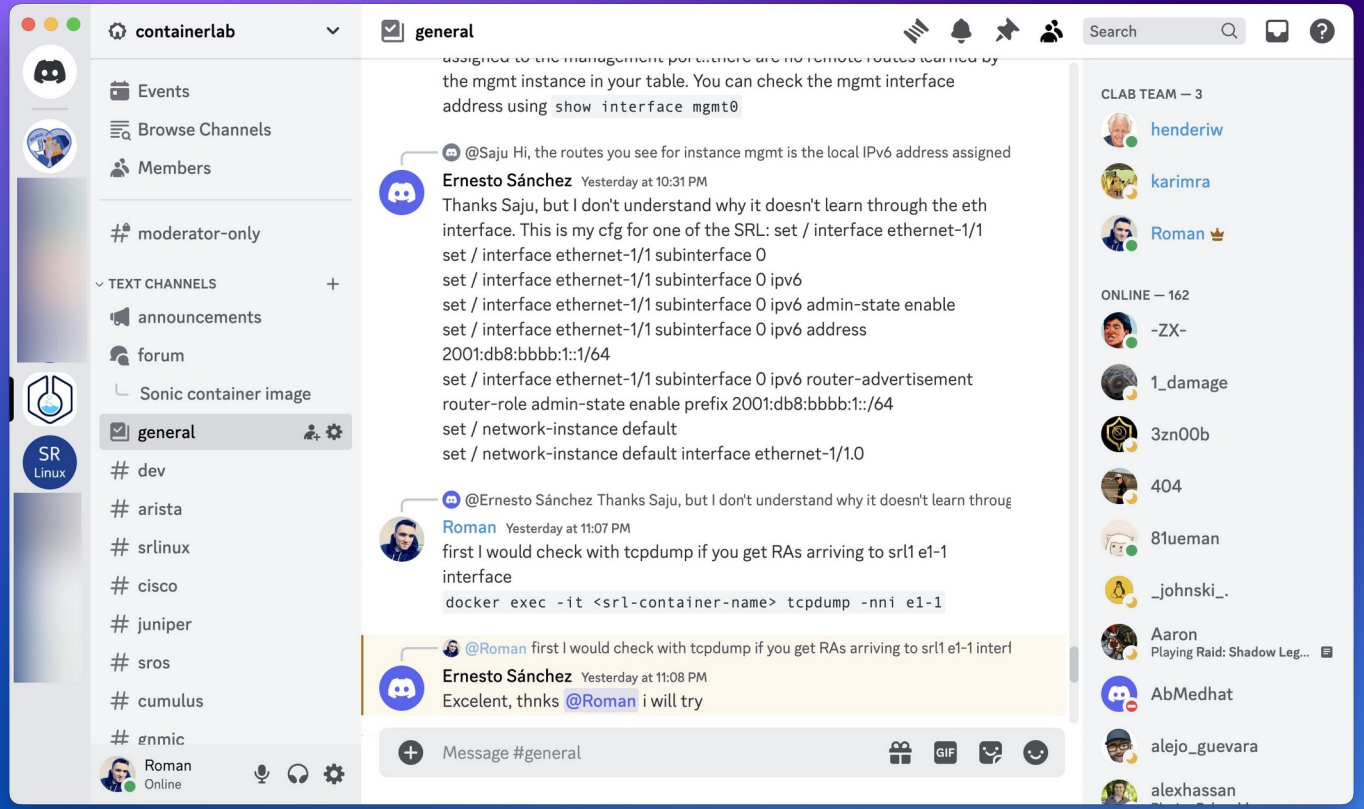
CONTAINERlab

Get in touch!

Containerlab docs



Containerlab Community



Now let’s show some Demos live


NOKIA