CONTAINERlab

Free and opensource networking lab environment for the modern age

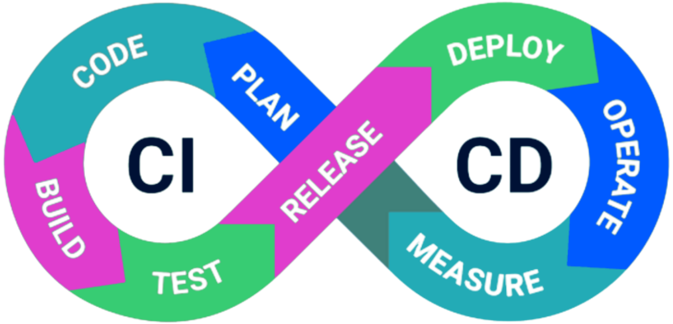
Roman Dodin @ntdvps

RONOG 8

Virtual labs

A right, not a privilege

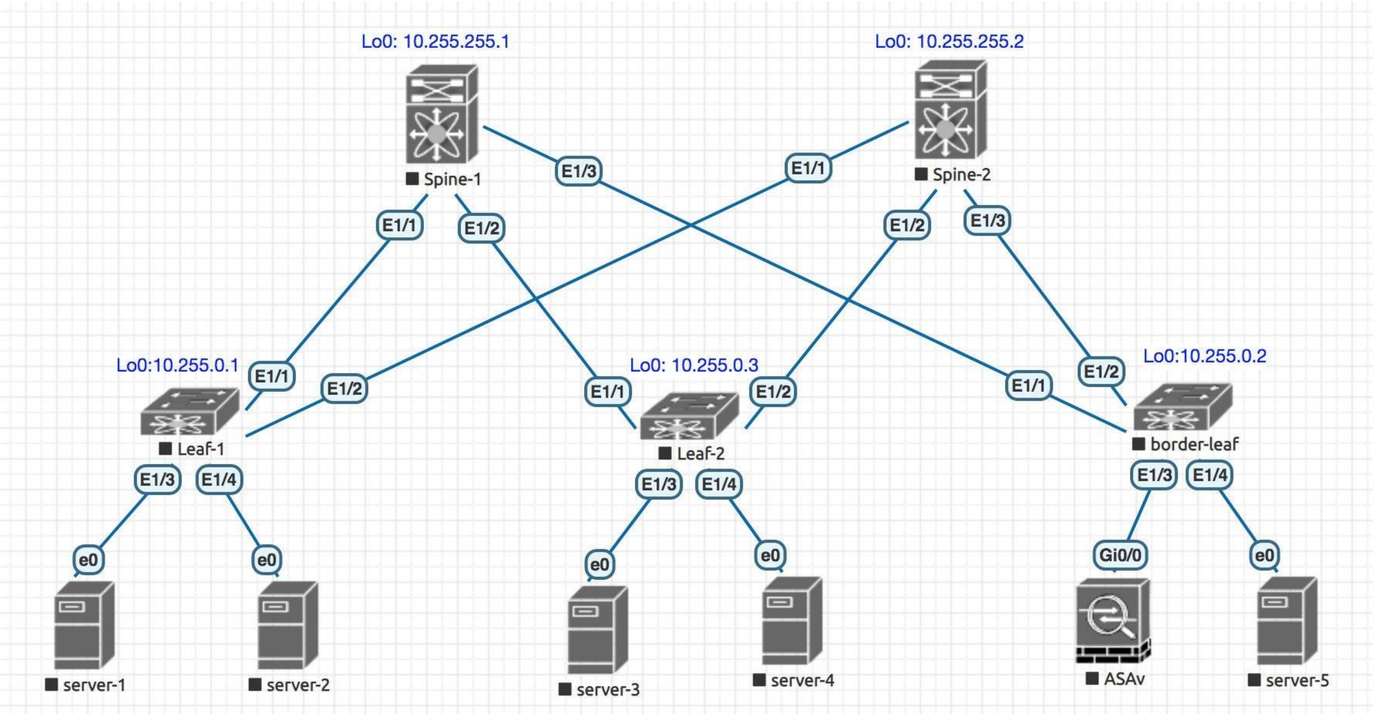
Education
and Learning



Change management
and validation

© 2023 Nokia

How do we (typically) run labs?

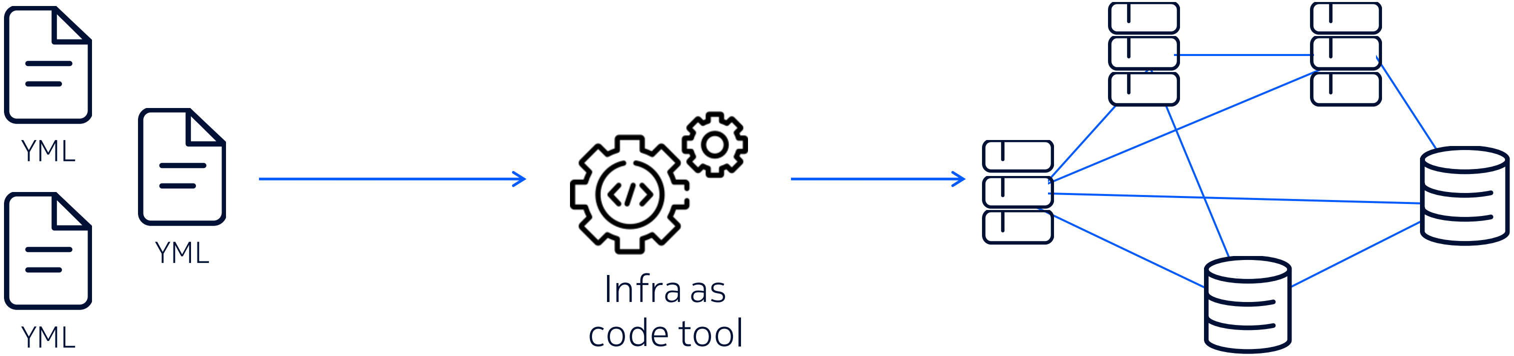


Pic from reddit

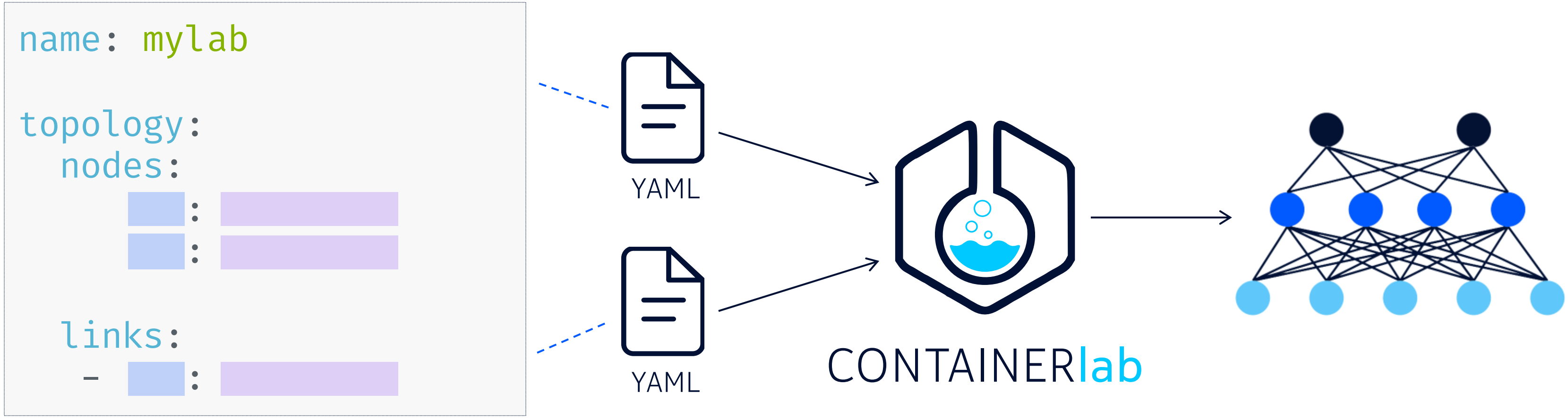
© 2023 Nokia

How do application teams (typically) deploy the services?

Declarative way



Bringing declarativeness to network labs

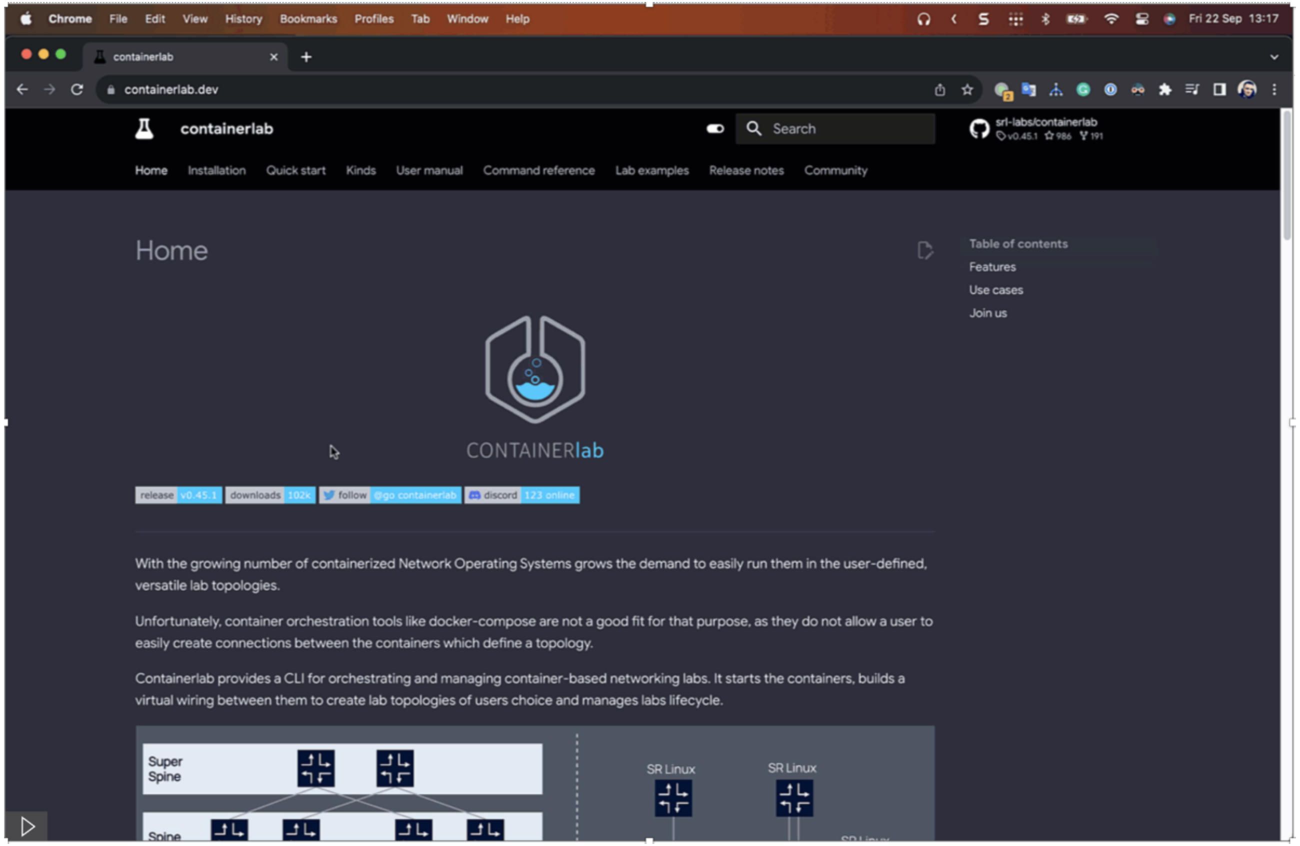


```
name: mylab

topology:
  nodes:
     ▨: ▨
     ▨: ▨

  links:
   - ▨: ▨
```

containerlab.dev

© 2023 Nokia

Installation



Home

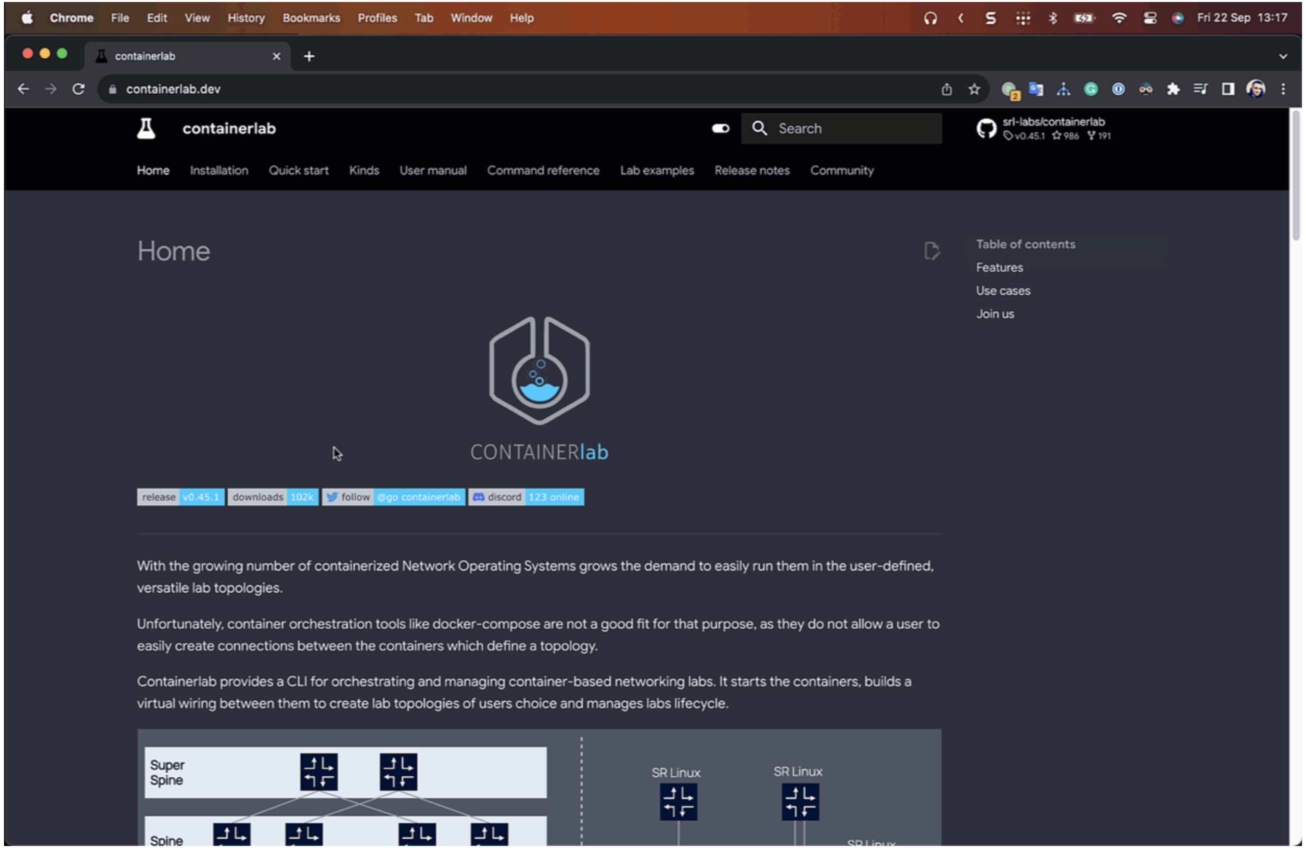
With the growing number of containerized Network Operating Systems grows the demand to easily run them in the user-defined, versatile lab topologies.

Unfortunately, container orchestration tools like docker-compose are not a good fit for that purpose, as they do not allow a user to easily create connections between the containers which define a topology.

Containerlab 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.

Other installation options:
https://containerlab.dev/install/

Installation



Home

With the growing number of containerized Network Operating Systems grows the demand to easily run them in the user-defined, versatile lab topologies.

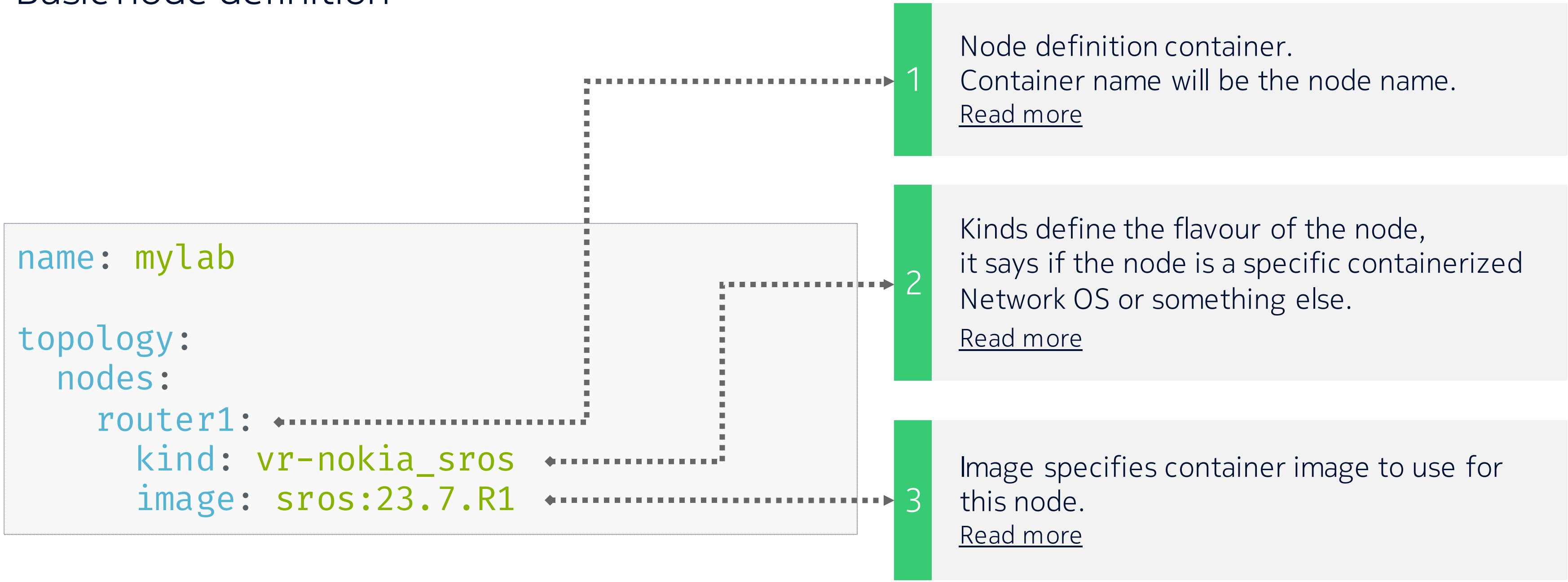
Unfortunately, container orchestration tools like docker-compose are not a good fit for that purpose, as they do not allow a user to easily create connections between the containers which define a topology.

Containerlab 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.

Other installation options:
https://containerlab.dev/install/

Topology file

Basic node definition



```yaml
name: mylab

topology:
  nodes:
    router1:
      kind: vr-nokia_sros
      image: sros:23.7.R1
```

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

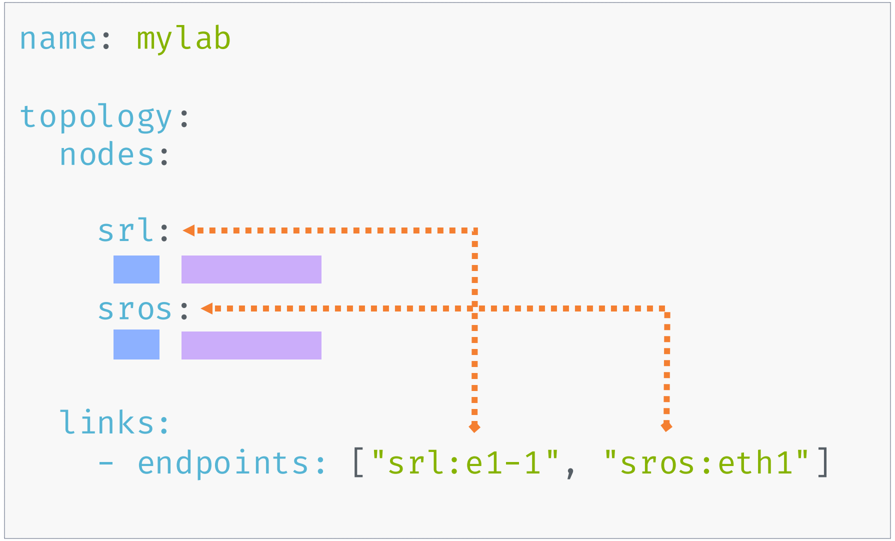
topology definition file

© 2023 Nokia

Topology file

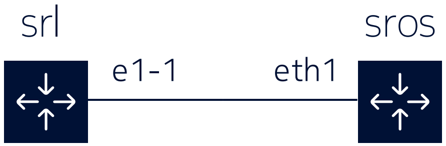
Links definition

Topology definition



```
name: mylab

topology:
  nodes:

    srl:

    sros:


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

Logical view



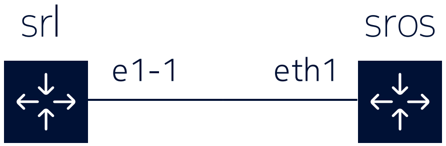
Topology file

Bringing nodes and links together

Topology definition

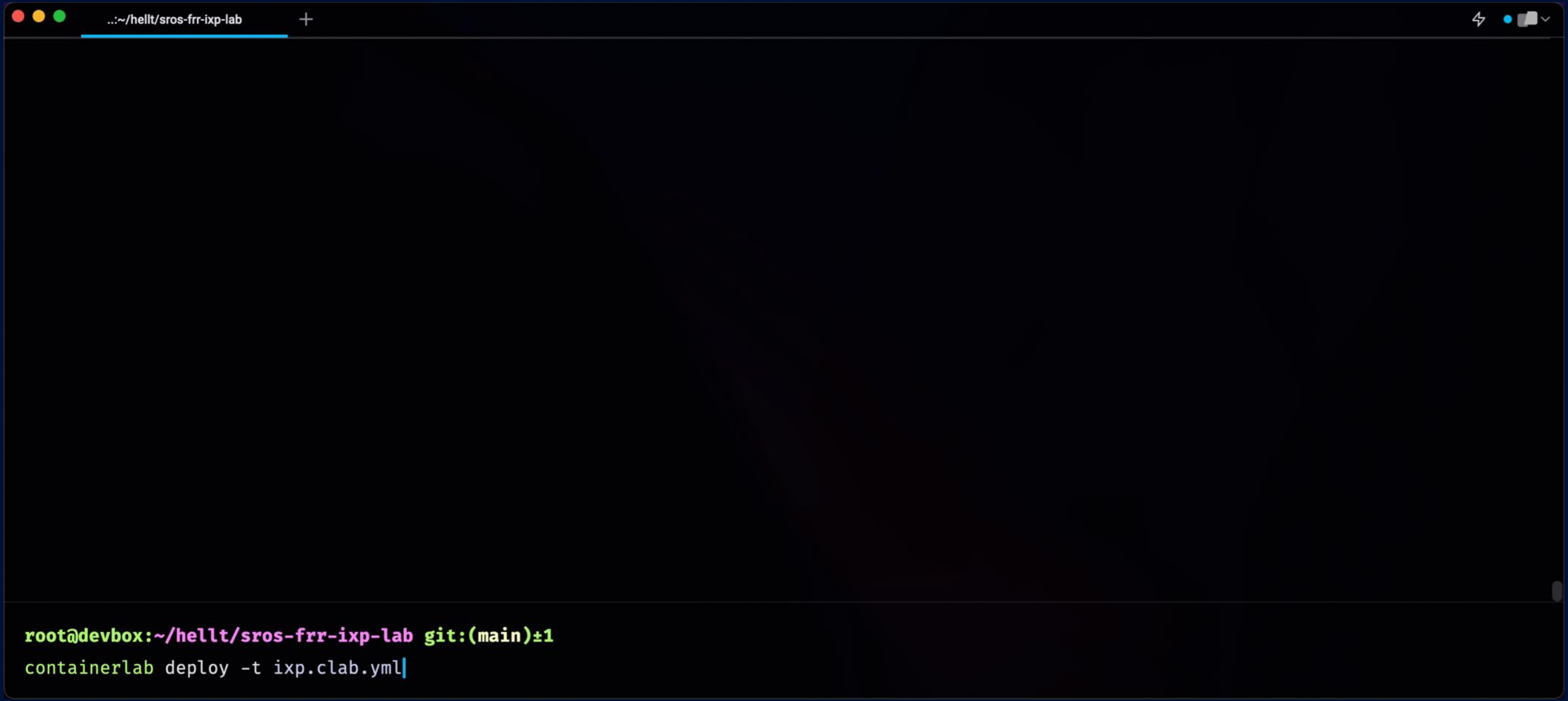
```yaml
name: mylab

topology:
  nodes:

    srl:
      kind: nokia_srlinux
      image: ghcr.io/nokia/srlinux:23.7.1

    sros:
      kind: vr-nokia_sros
      image: sros:23.7.R1
      license: license.txt

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

Logical view



© 2023 Nokia

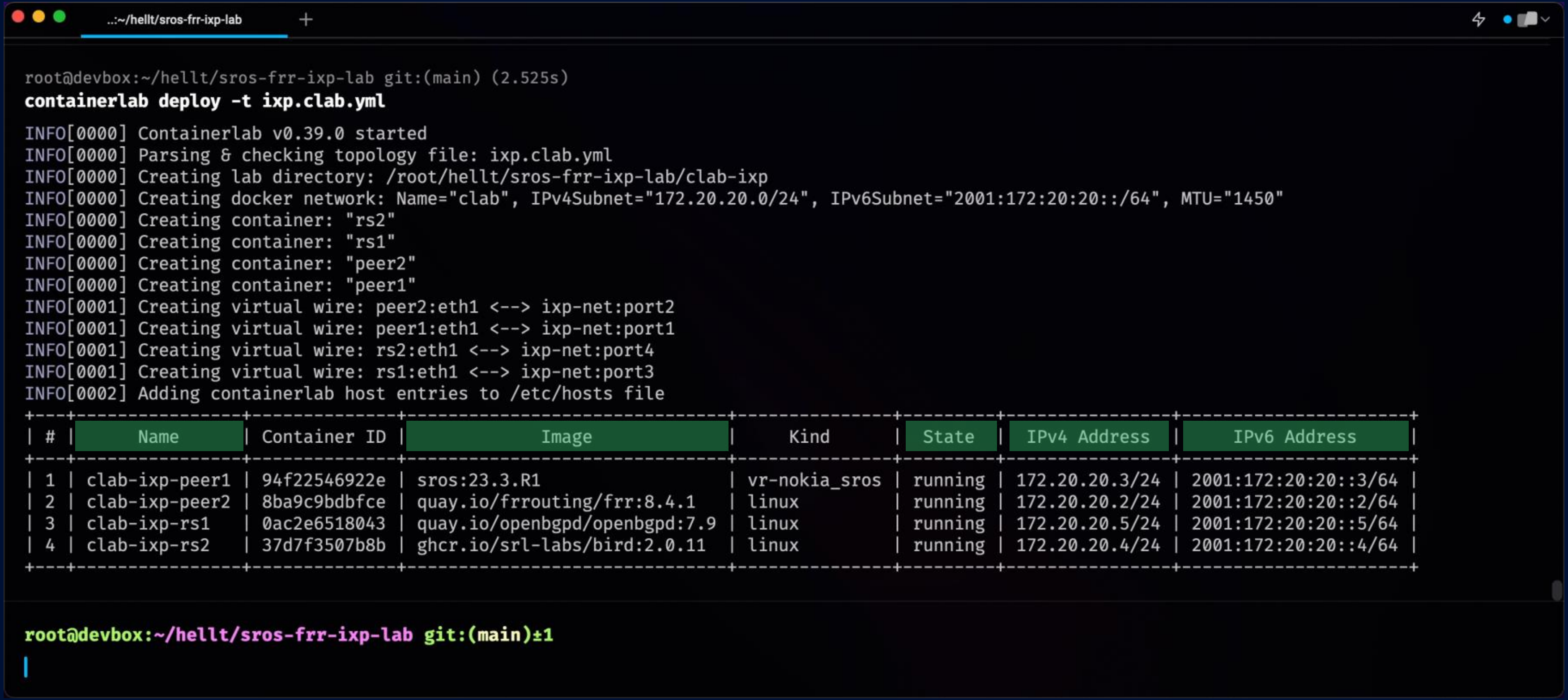
Deploying a lab



```
root@devbox:~/hellt/sros-frr-ixp-lab git:(main)±1
containerlab deploy -t ixp.clab.yml
```

© 2023 Nokia

Deployment summary



```
root@devbox:~/hellt/sros-frr-ixp-lab git:(main) (2.525s)
containerlab deploy -t ixp.clab.yml
INFO[0000] Containerlab v0.39.0 started
INFO[0000] Parsing & checking topology file: ixp.clab.yml
INFO[0000] Creating lab directory: /root/hellt/sros-frr-ixp-lab/clab-ixp
INFO[0000] Creating docker network: Name="clab", IPv4Subnet="172.20.20.0/24", IPv6Subnet="2001:172:20:20::/64", MTU="1450"
INFO[0000] Creating container: "rs2"
INFO[0000] Creating container: "rs1"
INFO[0000] Creating container: "peer2"
INFO[0000] Creating container: "peer1"
INFO[0001] Creating virtual wire: peer2:eth1 <--> ixp-net:port2
INFO[0001] Creating virtual wire: peer1:eth1 <--> ixp-net:port1
INFO[0001] Creating virtual wire: rs2:eth1 <--> ixp-net:port4
INFO[0001] Creating virtual wire: rs1:eth1 <--> ixp-net:port3
INFO[0002] Adding containerlab host entries to /etc/hosts file
+---+----------------+--------------+-------------------------------+---------------+---------+----------------+----------------------+
| # |      Name      | Container ID |             Image             |     Kind      |  State  |  IPv4 Address  |     IPv6 Address     |
+---+----------------+--------------+-------------------------------+---------------+---------+----------------+----------------------+
| 1 | clab-ixp-peer1 | 94f22546922e | sros:23.3.R1                  | vr-nokia_sros | running | 172.20.20.3/24 | 2001:172:20:20::3/64 |
| 2 | clab-ixp-peer2 | 8ba9c9bdbfce | quay.io/frrouting/frr:8.4.1   | linux         | running | 172.20.20.2/24 | 2001:172:20:20::2/64 |
| 3 | clab-ixp-rs1   | 0ac2e6518043 | quay.io/openbgpd/openbgpd:7.9 | linux         | running | 172.20.20.5/24 | 2001:172:20:20::5/64 |
| 4 | clab-ixp-rs2   | 37d7f3507b8b | ghcr.io/srl-labs/bird:2.0.11  | linux         | running | 172.20.20.4/24 | 2001:172:20:20::4/64 |
+---+----------------+--------------+-------------------------------+---------------+---------+----------------+----------------------+

root@devbox:~/hellt/sros-frr-ixp-lab git:(main)±1
```

Connecting to the nodes

SSH

```
ssh admin@clab-ixp-peer1

admin@clab-ixp-peer1's password:

[/]
A:admin@peer1#
```

Docker exec

```
docker exec -it clab-ixp-rs2 birdc

BIRD 2.0.11 ready.
bird>
```

Containerlab node types

Containerized Network OSes

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

A 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

and others…

© 2023 Nokia

Containerlab node types

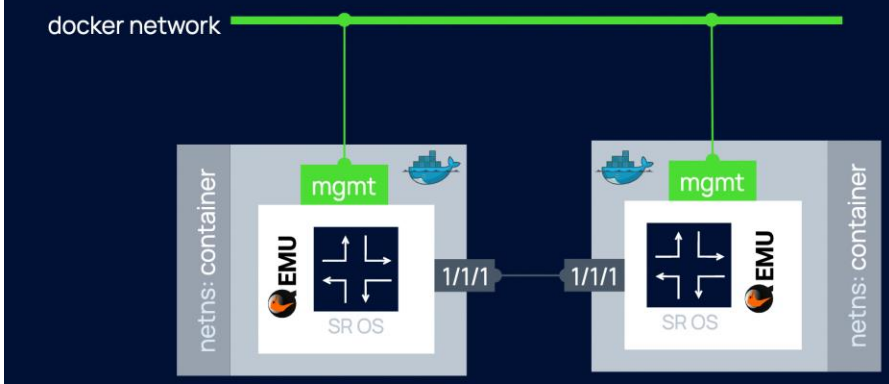
Regular container images

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

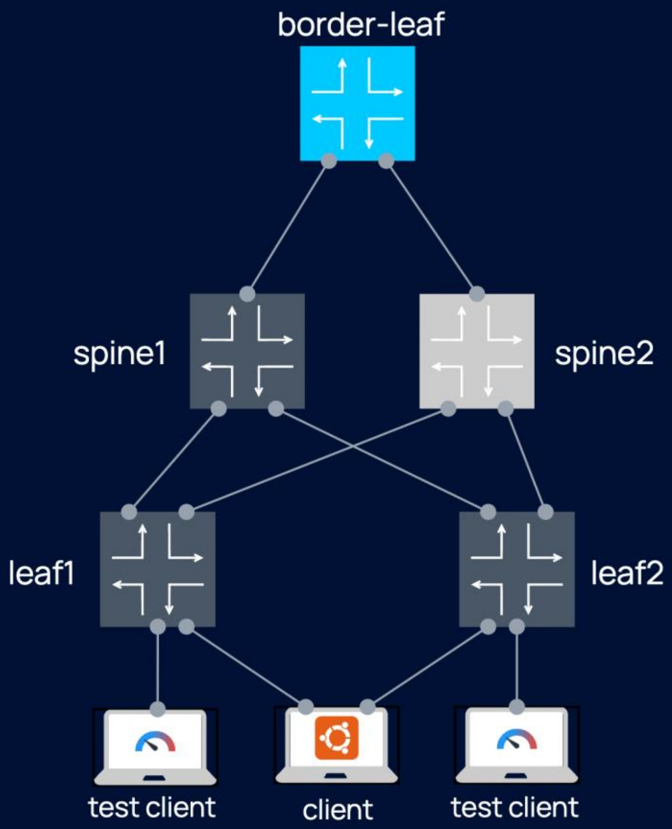
Containerlab node types

Virtual machines in container packaging

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



© 2023 Nokia



border-leaf

spine1

spine2

leaf1

leaf2

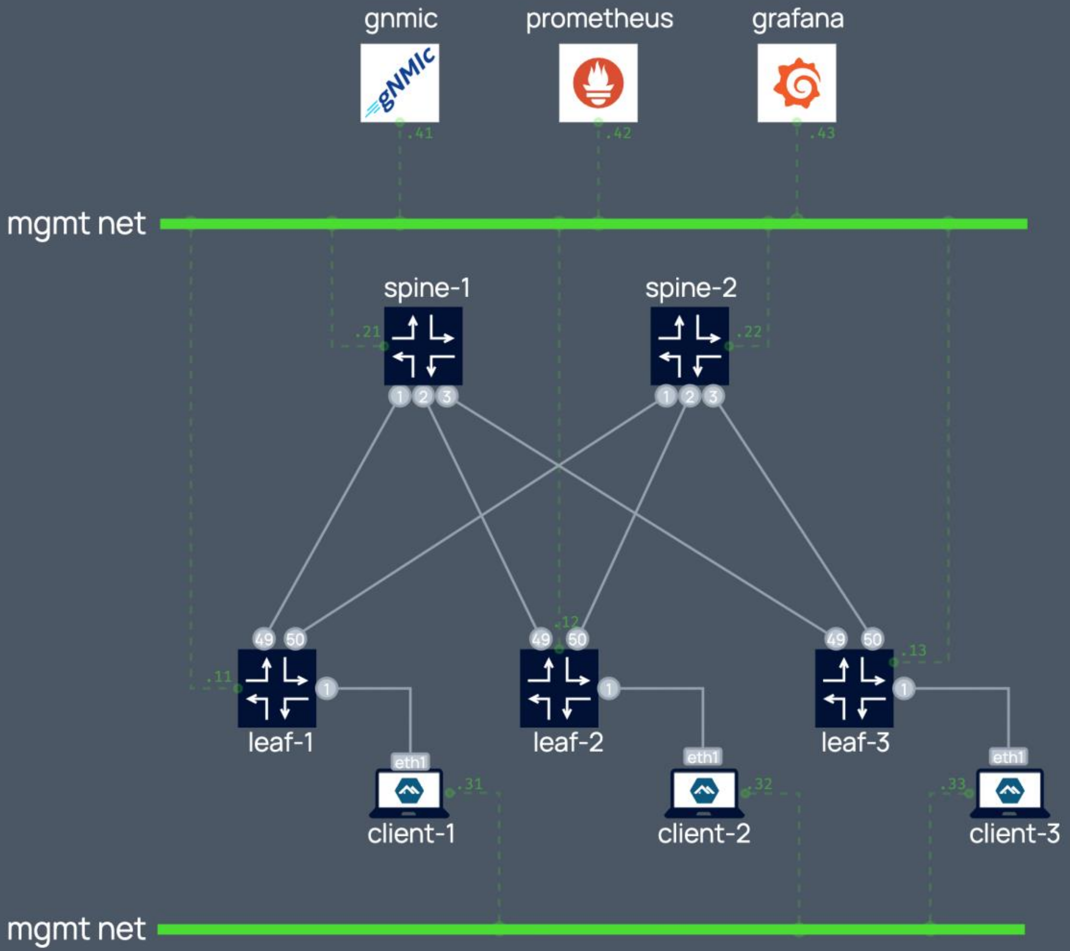
test client

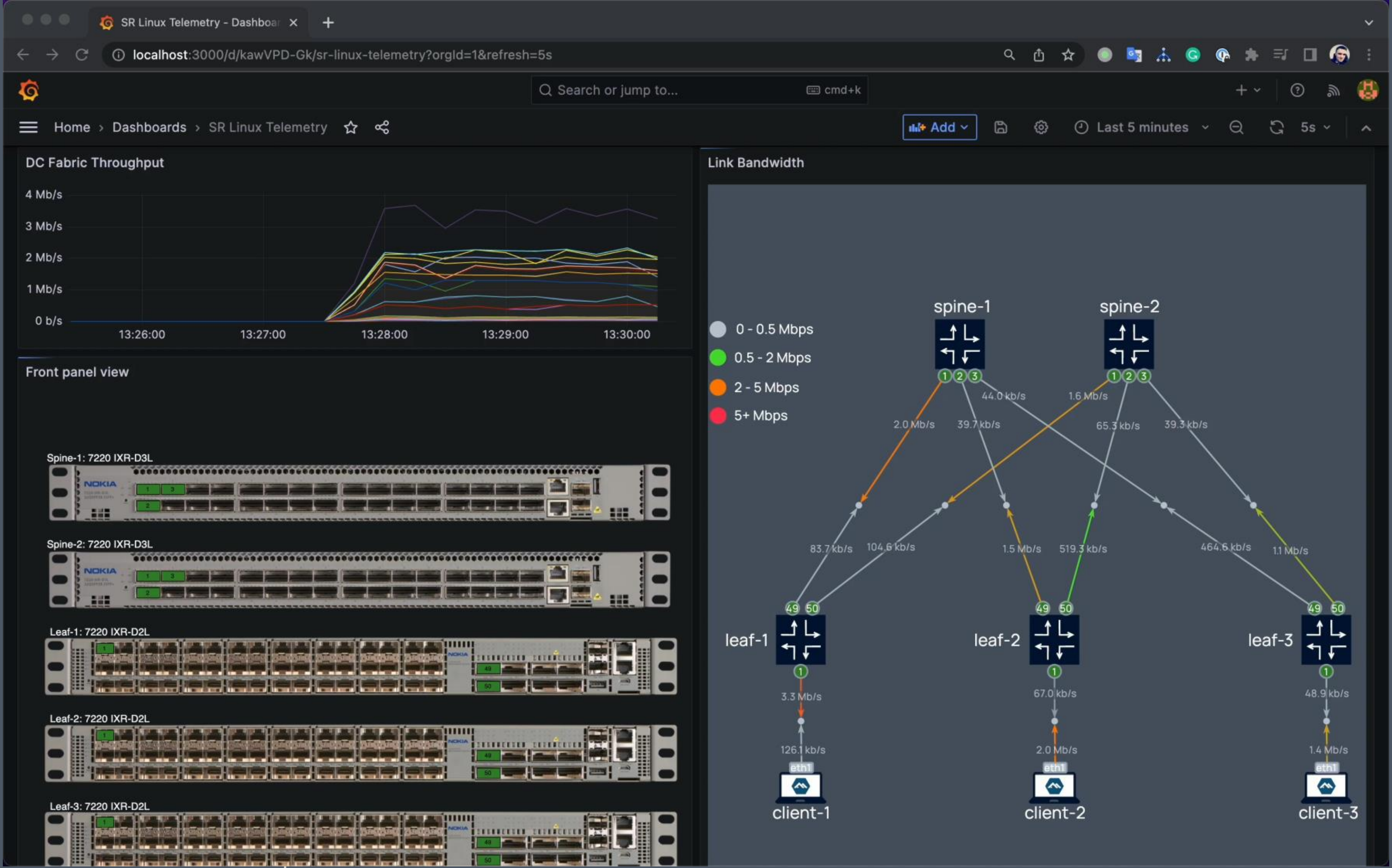
client

test client

- 🟦 traditional virtual router packaged in a container
- ⬜ containerized NOS vendor A
- ⬛ containerized NOS vendor B

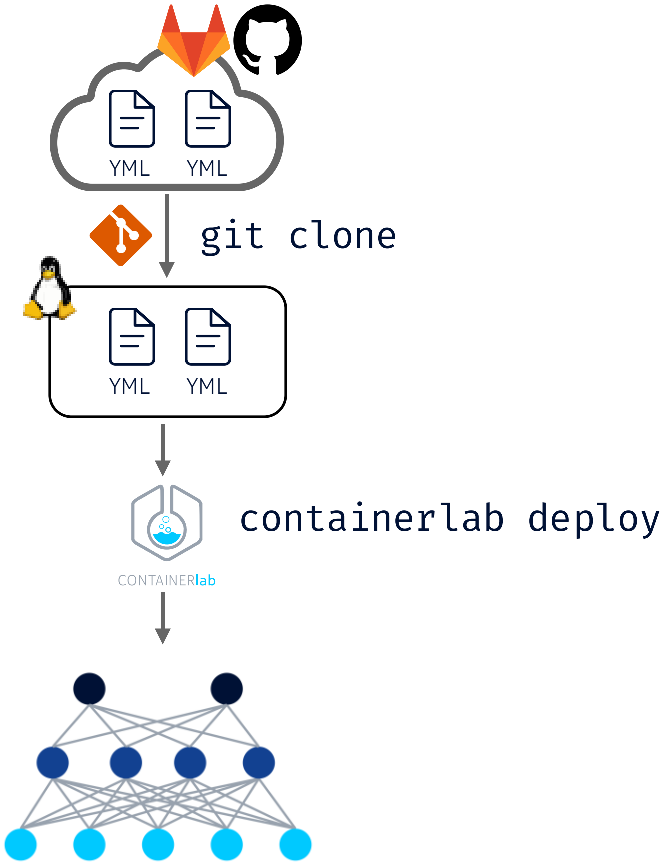
© 2023 Nokia


gnmic
prometheus
grafana
gNMIc
.41
.42
.43
mgmt net
spine-1
spine-2
.21
.22
1
2
3
1
2
3
49
50
49
50
49
50
.11
.12
.13
1
1
1
leaf-1
leaf-2
leaf-3
eth1
eth1
eth1
.31
.32
.33
client-1
client-2
client-3
mgmt net


SR Linux Telemetry - Dashboar
localhost:3000/d/kawVPD-Gk/sr-linux-telemetry?orgId=1&refresh=5s
Search or jump to...
cmd+k
Home › Dashboards › SR Linux Telemetry
Add
Last 5 minutes
5s
DC Fabric Throughput
4 Mb/s
3 Mb/s
2 Mb/s
1 Mb/s
0 b/s
13:26:00
13:27:00
13:28:00
13:29:00
13:30:00
Front panel view
Spine-1: 7220 IXR-D3L
Spine-2: 7220 IXR-D3L
Leaf-1: 7220 IXR-D2L
Leaf-2: 7220 IXR-D2L
Leaf-3: 7220 IXR-D2L
Link Bandwidth
● 0 - 0.5 Mbps
● 0.5 - 2 Mbps
● 2 - 5 Mbps
● 5+ Mbps
spine-1
spine-2
1 2 3
1 2 3
44.0 kb/s
1.6 Mb/s
2.0 Mb/s
39.7 kb/s
65.3 kb/s
39.3 kb/s
83.7 kb/s
104.6 kb/s
1.5 Mb/s
519.3 kb/s
464.6 kb/s
1.1 Mb/s
49 50
leaf-1
leaf-2
leaf-3
1
3.3 Mb/s
67.0 kb/s
48.9 kb/s
126.1 kb/s
2.0 Mb/s
1.4 Mb/s
eth1
client-1
client-2
client-3

Lab As Code

Telemetry Lab



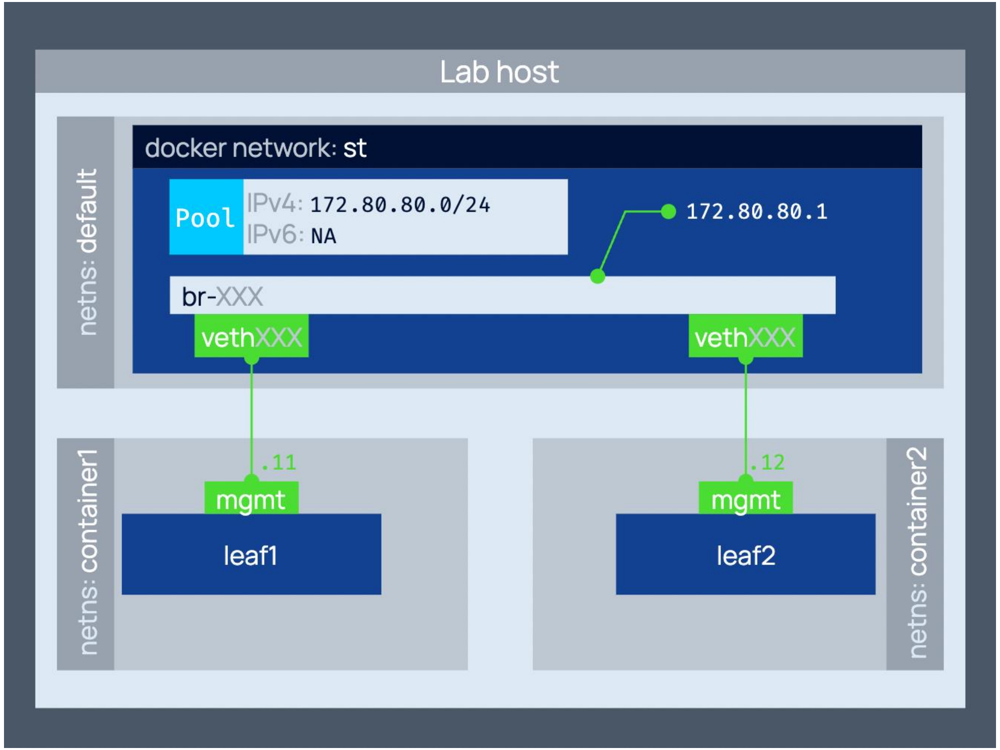
```
git clone
```

```
containerlab deploy
```

Management network

Staticly assigned IP addresses

```
mgmt:
  network: st
  ipv4-subnet: 172.80.80.0/24

topology:
  nodes:
    leaf1:
      mgmt-ipv4: 172.80.80.11

    leaf2:
      mgmt-ipv4: 172.80.80.12
```



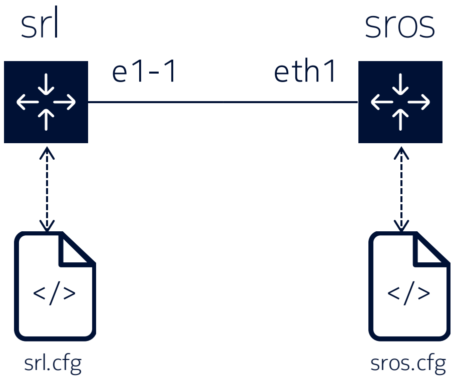
© 2023 Nokia

Startup configuration

Topology definition

```
topology:
  nodes:

    srl:
      kind: nokia_srlinux
      image: ghcr.io/nokia/srlinux:23.7.1
      startup-config: srl.cfg


    sros:
      kind: vr-nokia_sros
      image: sros:23.7.R1
      startup-config: sros.cfg
```

Logical view



© 2023 Nokia

Executing commands

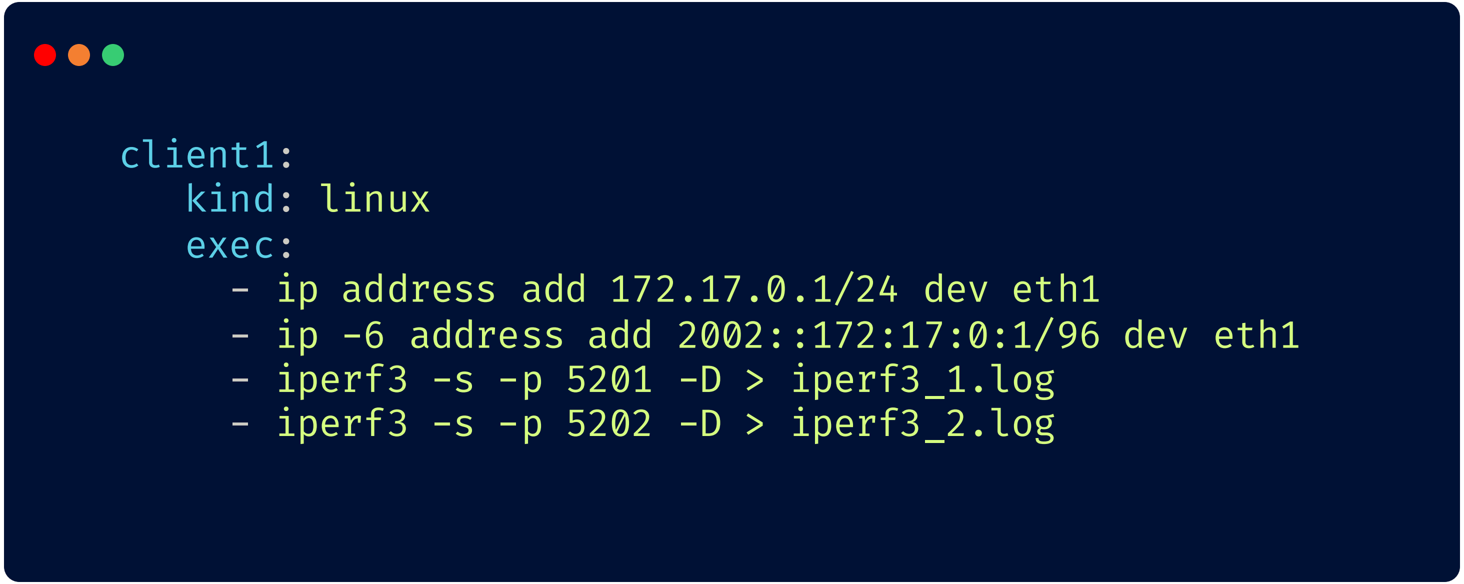
- Exec is a list of commands executed inside the container once it reaches running state
  - configure network params (ip, mac)
  - run the provisioning or workload scripts

```yaml
nodes:
  server:
    kind: linux
    image: alpine:3
    exec:
      - ip address add 172.17.0.1/24 dev eth1
```

© 2023 Nokia

Executing commands

How we used it in the lab?



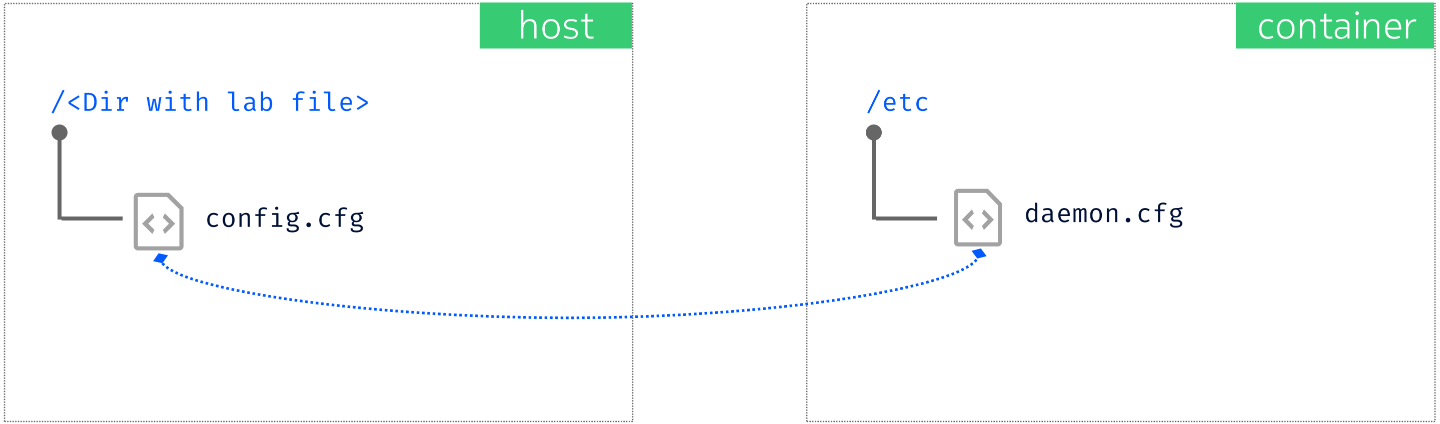
```
client1:
   kind: linux
   exec:
     - ip address add 172.17.0.1/24 dev eth1
     - ip -6 address add 2002::172:17:0:1/96 dev eth1
     - iperf3 -s -p 5201 -D > iperf3_1.log
     - iperf3 -s -p 5202 -D > iperf3_2.log
```

© 2023 Nokia

File binding

```
server:
  kind: linux
  binds:
    - config.cfg:/etc/daemon.cfg
```

- Bind mount files from host to a container
  - Providing configuration files
  - Providing executable scripts
  - Access to container’s files

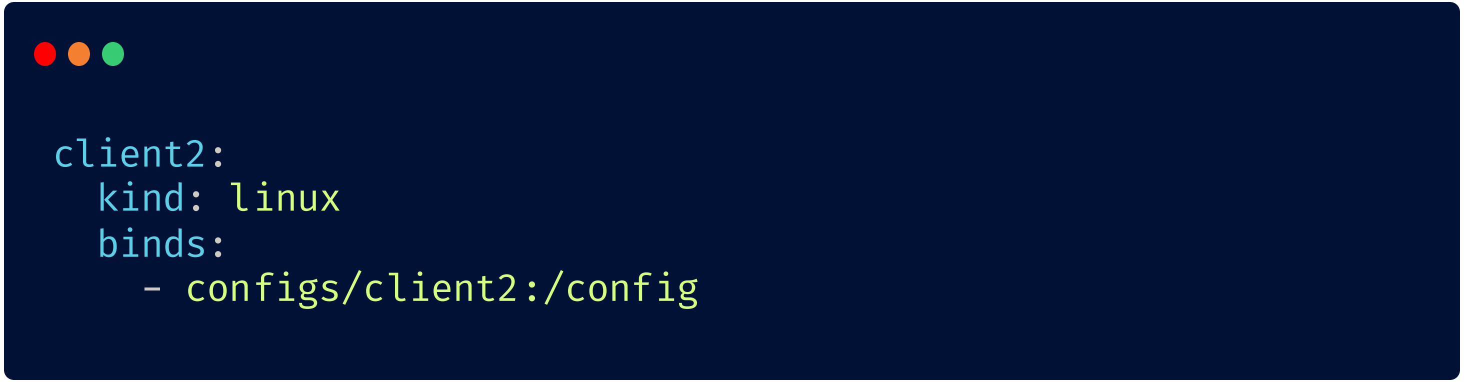


© 2023 Nokia

File binding

How we used it in the lab?

- Share a config folder with shell scripts from the host to the node
  - Provide iperf.sh script that manages traffic flow



```
client2:
  kind: linux
  binds:
    - configs/client2:/config
```

© 2023 Nokia

Entrypoint and command

- Entrypoint is the “command” that starts in a container
- Command (aka CMD) is a list of arguments passed to the entrypoint

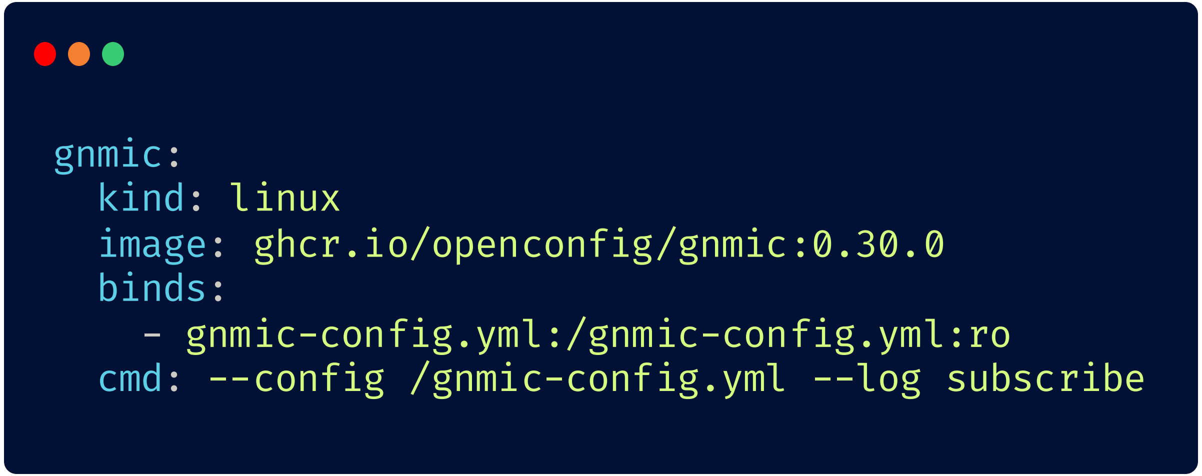
```
server:
  kind: linux
  image: alpine:3
  entrypoint: sleep
  cmd: ”10”
```

SReXperts - Confidential

Entrypoint and command

How we used it in the lab?

- Provide configuration options to the processes running in a container



```
gnmic:
  kind: linux
  image: ghcr.io/openconfig/gnmic:0.30.0
  binds:
    - gnmic-config.yml:/gnmic-config.yml:ro
  cmd: --config /gnmic-config.yml --log subscribe
```

© 2023 Nokia

Environment variables

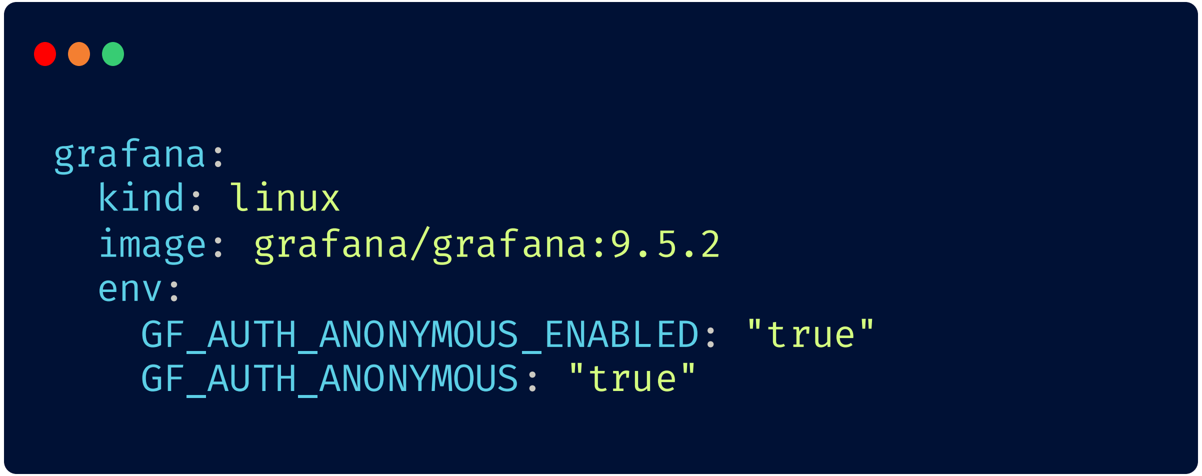
- Configure processes via env vars

```
server:
  kind: linux
  image: alpine:3
  env:
    MYVAR:SOMEVALUE
```

Environment variables

How we used it in the lab?

- Provide configuration options for some nodes



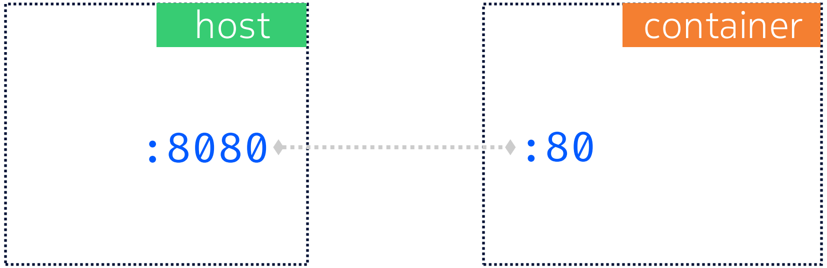
```yaml
grafana:
  kind: linux
  image: grafana/grafana:9.5.2
  env:
    GF_AUTH_ANONYMOUS_ENABLED: "true"
    GF_AUTH_ANONYMOUS: "true"
```

© 2023 Nokia

Exposing ports

- Make services inside a container available to containerlab host

```yaml
server:
  kind: linux
  image: nginx
  ports:
    - 8080:80
```

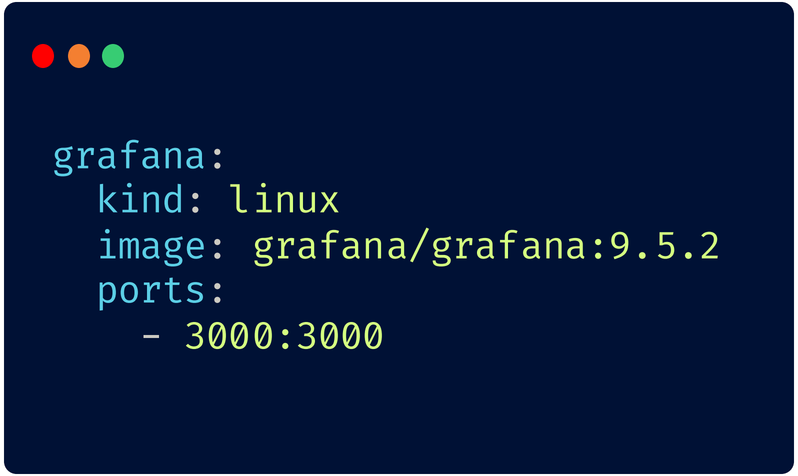


© 2023 Nokia

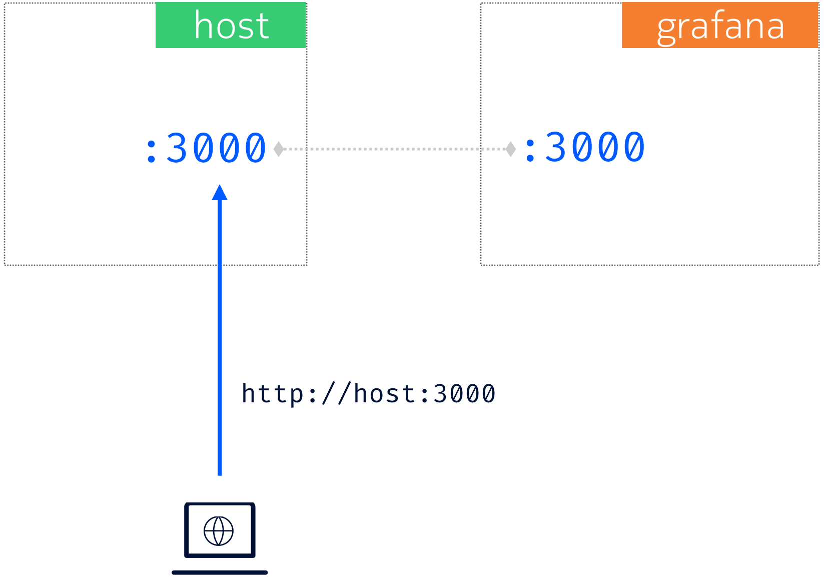
Exposing ports

How we used it in the lab?

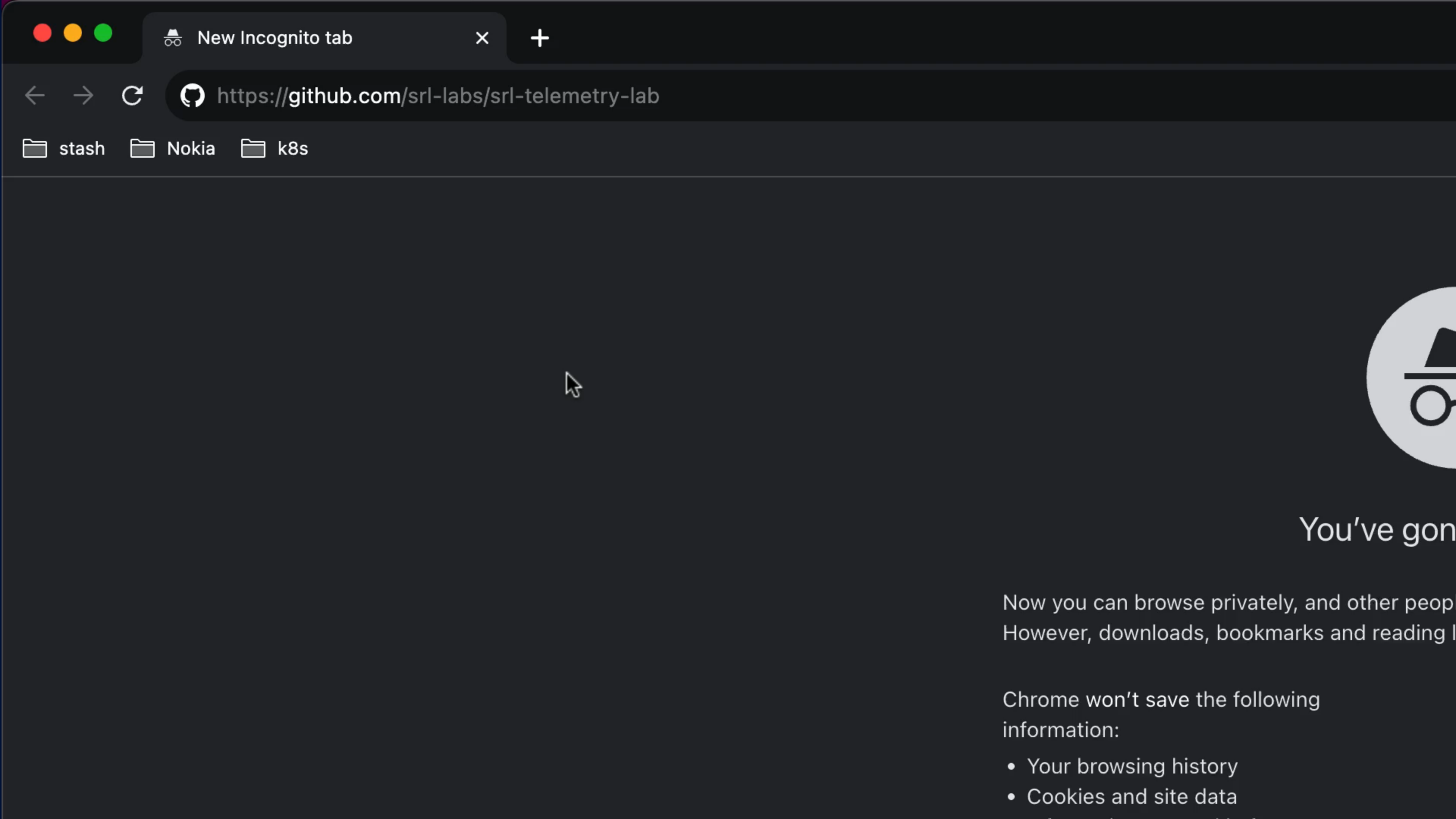
- Expose Grafana Web UI to allow remote access



```
grafana:
  kind: linux
  image: grafana/grafana:9.5.2
  ports:
    - 3000:3000
```



Let’s deploy the lab!

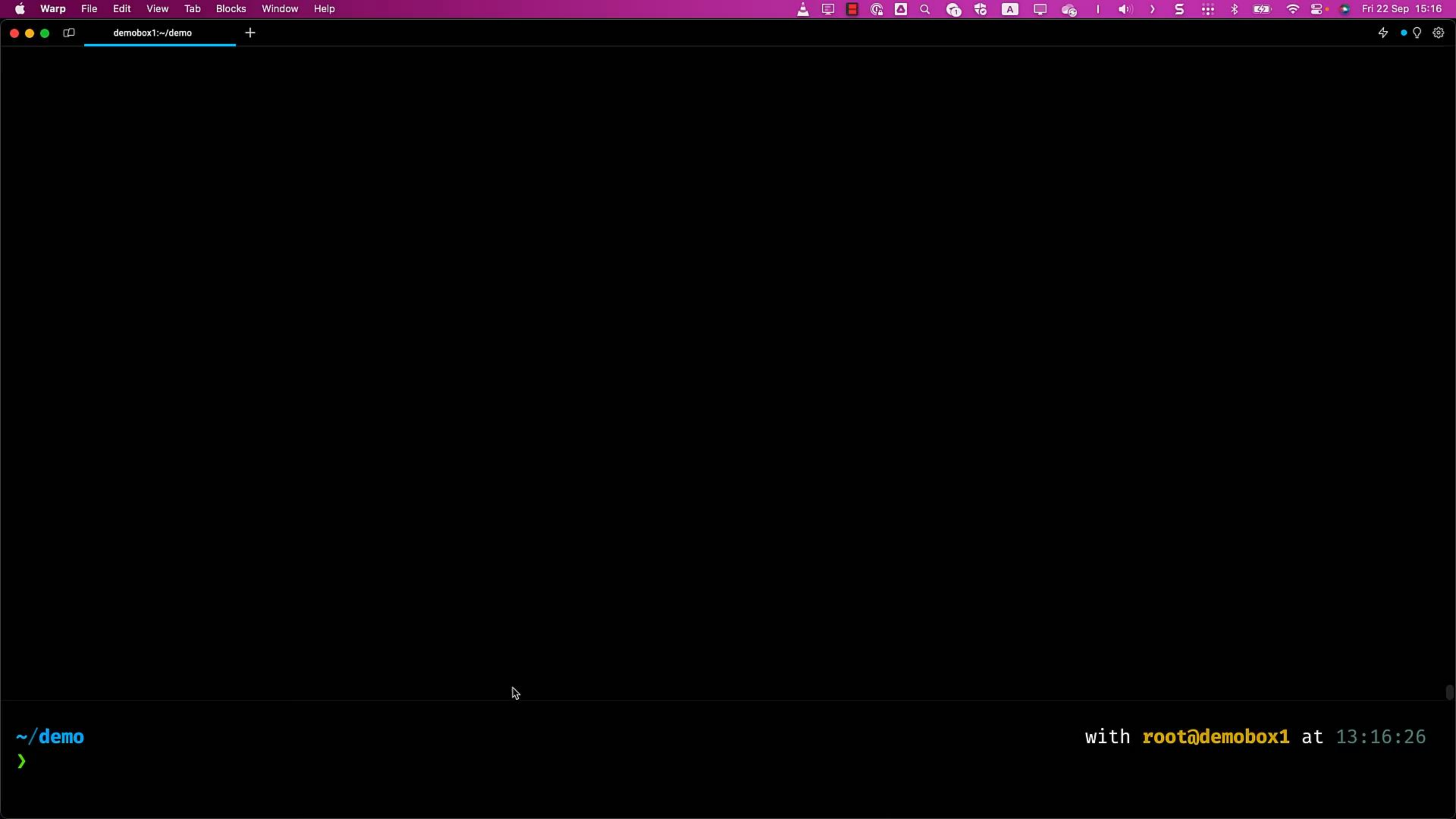


You've gon

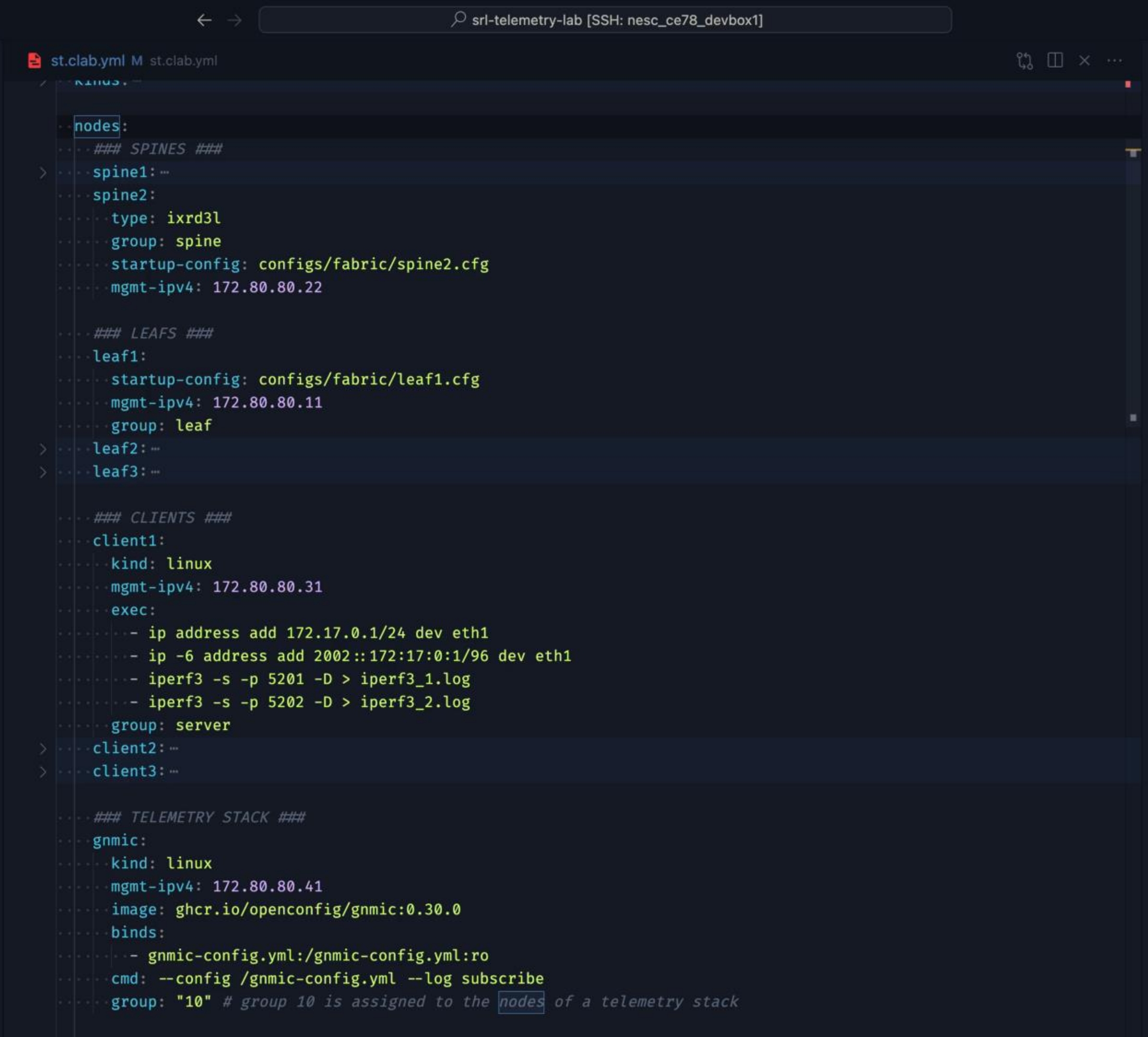
Now you can browse privately, and other peop
However, downloads, bookmarks and reading l

Chrome won't save the following information:

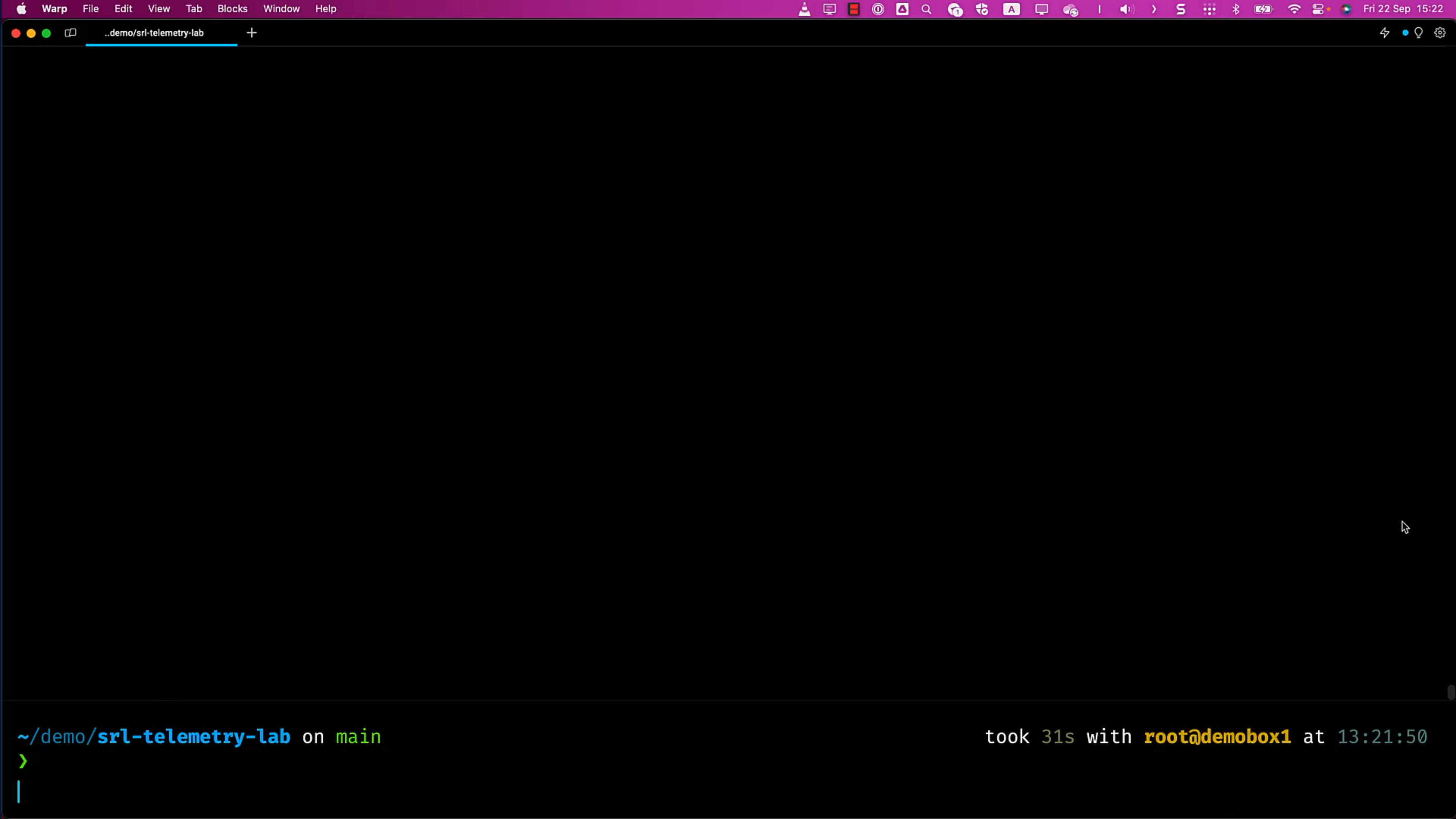
- Your browsing history
- Cookies and site data



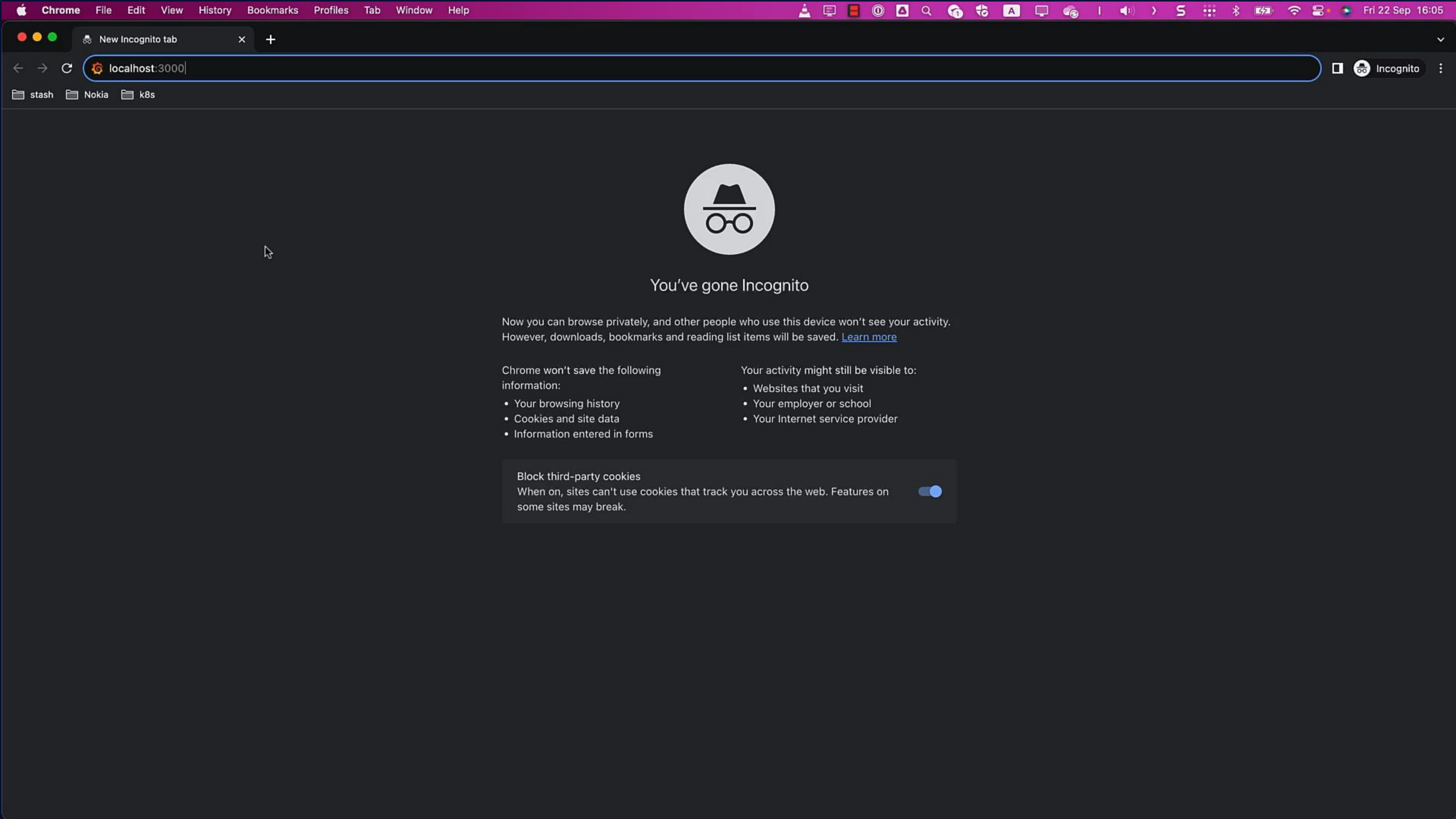
```
~/demo                                              with root@demobox1 at 13:16:26
❯
```


srl-telemetry-lab [SSH: nesc_ce78_devbox1]
st.clab.yml M st.clab.yml

```
  kinds: ⋯

  nodes:
    ### SPINES ###
    spine1: ⋯
    spine2:
      type: ixrd3l
      group: spine
      startup-config: configs/fabric/spine2.cfg
      mgmt-ipv4: 172.80.80.22

    ### LEAFS ###
    leaf1:
      startup-config: configs/fabric/leaf1.cfg
      mgmt-ipv4: 172.80.80.11
      group: leaf
    leaf2: ⋯
    leaf3: ⋯

    ### CLIENTS ###
    client1:
      kind: linux
      mgmt-ipv4: 172.80.80.31
      exec:
        - ip address add 172.17.0.1/24 dev eth1
        - ip -6 address add 2002::172:17:0:1/96 dev eth1
        - iperf3 -s -p 5201 -D > iperf3_1.log
        - iperf3 -s -p 5202 -D > iperf3_2.log
      group: server
    client2: ⋯
    client3: ⋯

    ### TELEMETRY STACK ###
    gnmic:
      kind: linux
      mgmt-ipv4: 172.80.80.41
      image: ghcr.io/openconfig/gnmic:0.30.0
      binds:
        - gnmic-config.yml:/gnmic-config.yml:ro
      cmd: --config /gnmic-config.yml --log subscribe
      group: "10" # group 10 is assigned to the nodes of a telemetry stack
```



```
~/demo/srl-telemetry-lab on main                                         took 31s with root@demobox1 at 13:21:50
❯
```



You've gone Incognito

Now you can browse privately, and other people who use this device won't see your activity. However, downloads, bookmarks and reading list items will be saved. Learn more

Chrome won't save the following information:

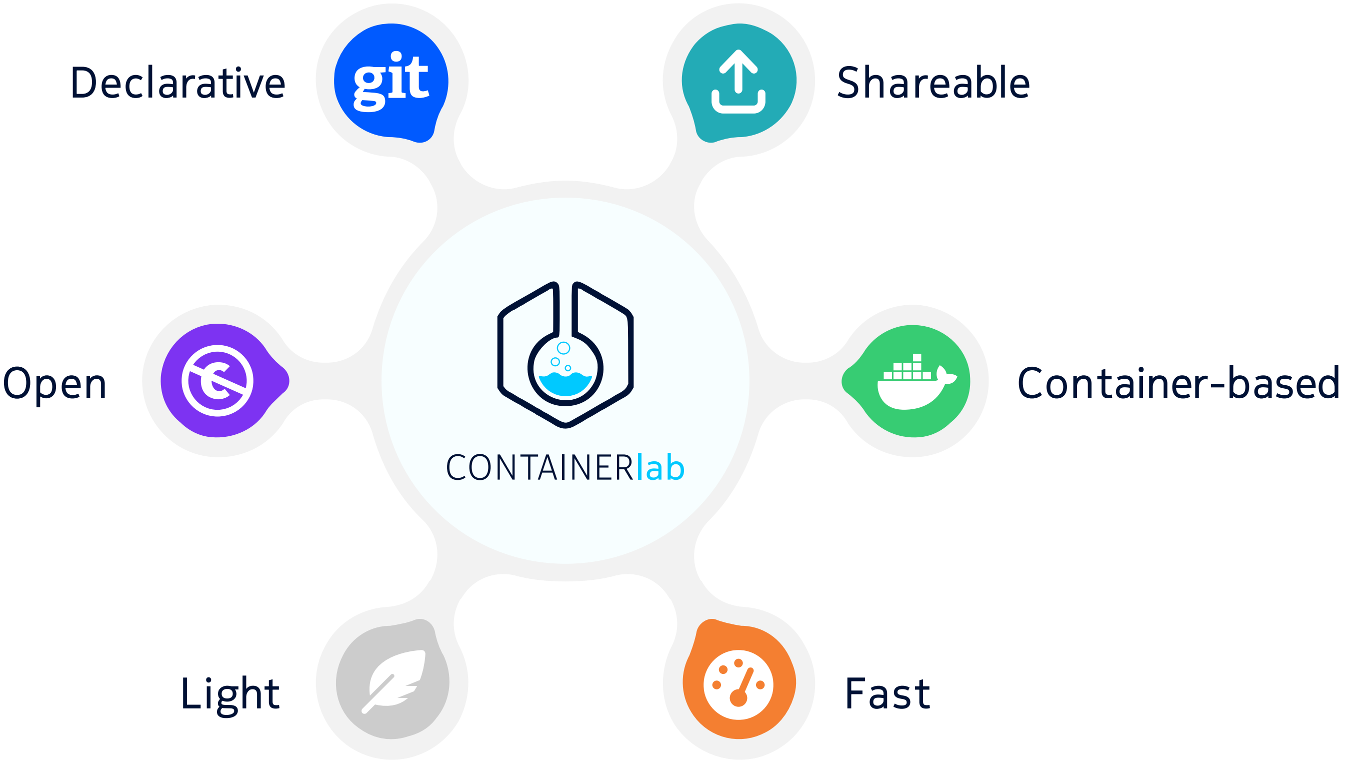
- Your browsing history
- Cookies and site data
- Information entered in forms

Your activity might still be visible to:

- Websites that you visit
- Your employer or school
- Your Internet service provider

Block third-party cookies
When on, sites can't use cookies that track you across the web. Features on some sites may break. ●

Containerlab



Declarative

Shareable

Open

Container-based

Light

Fast

CONTAINERlab

https://containerlab.dev