

NOKIA



CONTAINERlab

Unleashing the power of
containerization for network
testing and development

Bastien Claeys
AUSNOG 2023

© 2023 Nokia | Public use

Network labs

A right, not a privilege

1

Change
management

2

Prototyping and
validation

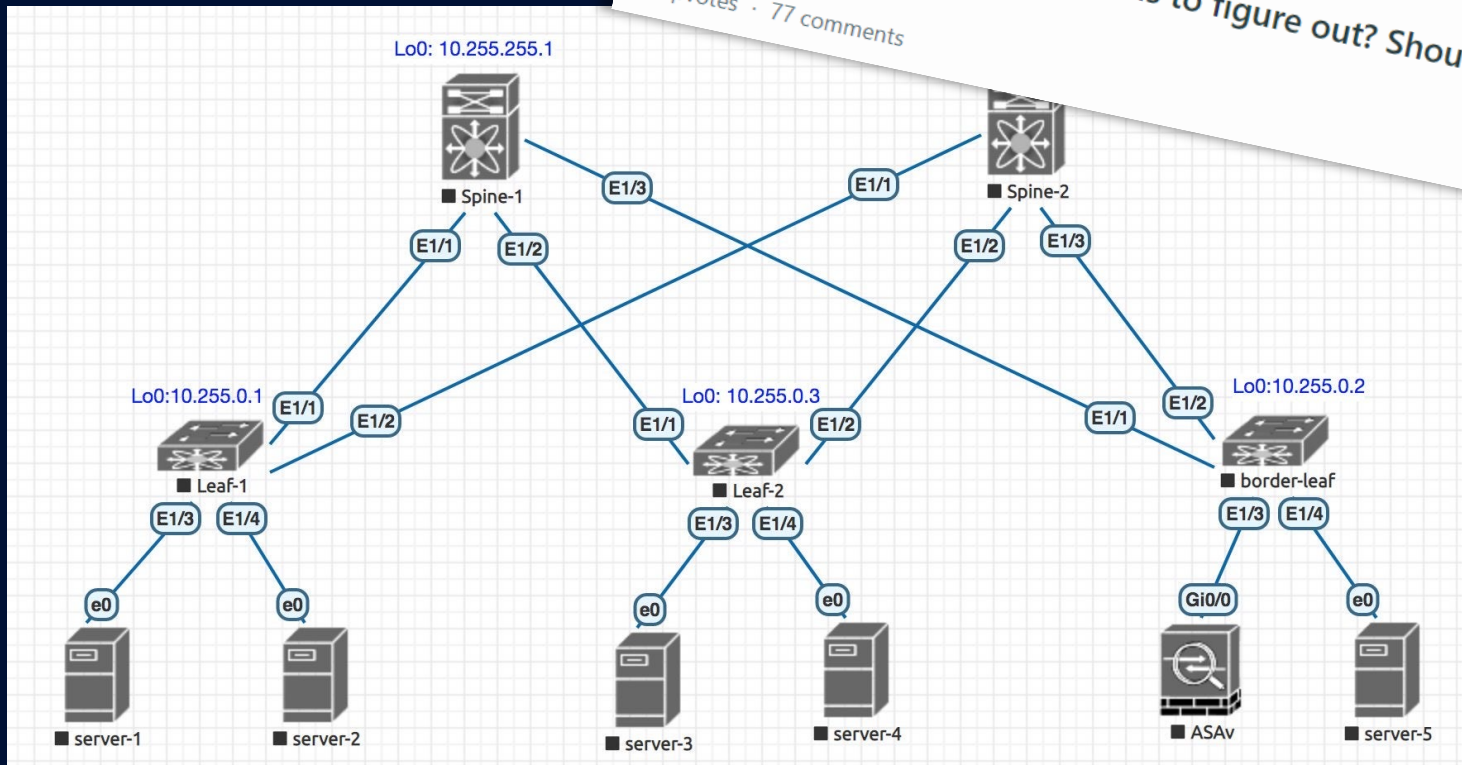
3

Learning

Network labs

How do we typically run labs today?

r/networking · 7 months ago
Do all lab simulators take months to figure out? Should I just buy physical equipment??
84 upvotes · 77 comments



Pic from https://www.reddit.com/r/networking/comments/g5fb23/eveng_lab_strage_packet_loss/

Containerlab

Bringing declarativeness to network labs

IT

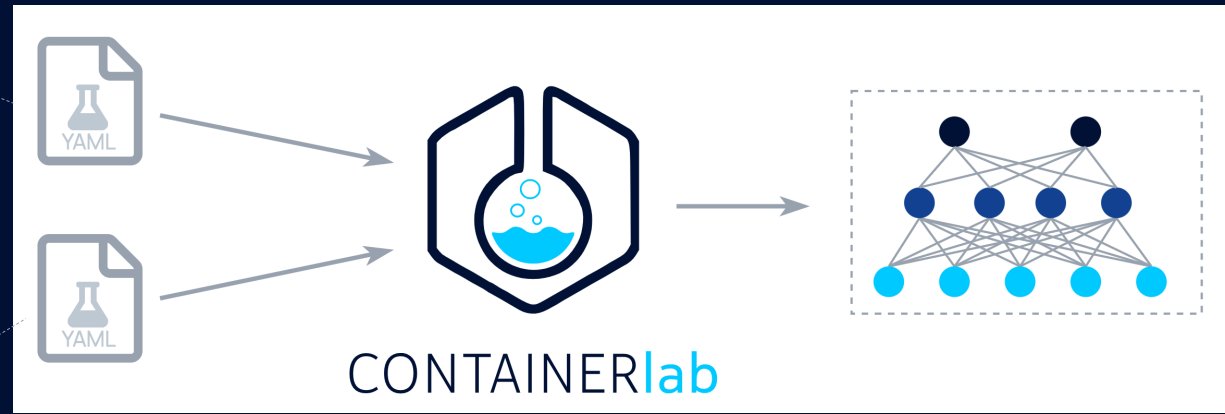


Network Labs

```
name: mylab

topology:
  nodes:
    [ ] : [ ]
    [ ] : [ ]

  links:
    - [ ] : [ ]
```



Why Containerlab

If we have lab emulation tools already?

Network emulation SW



- + Purpose built & proven
- + Free versions available
- + U
I
- VM-centric
weak containers support
- Heavy and semi-open
- U
I

Containerlab




CONTAINERlab

- + First class support for containerized NOSes
- + Transparent datapath
- + Git friendly & better image sharing and handling
- + Repeatable lab builds and CI friendly
- + Small footprint, open, free and fast
- No UI
- Fewer Network OSes supported

Supported NOS

NOKIA

 srl
vr-sros




 cvx

ipinfusion

ipinfusion_ocnos

JUNIPER
NETWORKS

 crpd
vr-vmx
vr-vqfx




vr-pan



checkpoint_cloudguard

ARISTA

 ceos
vr-veos



vr-ftosv



vr-xrv9k
vr-csr
vr-n9kv

ixia


 keysight_ixia-c



 sonic-vs
 frr

MikroTik

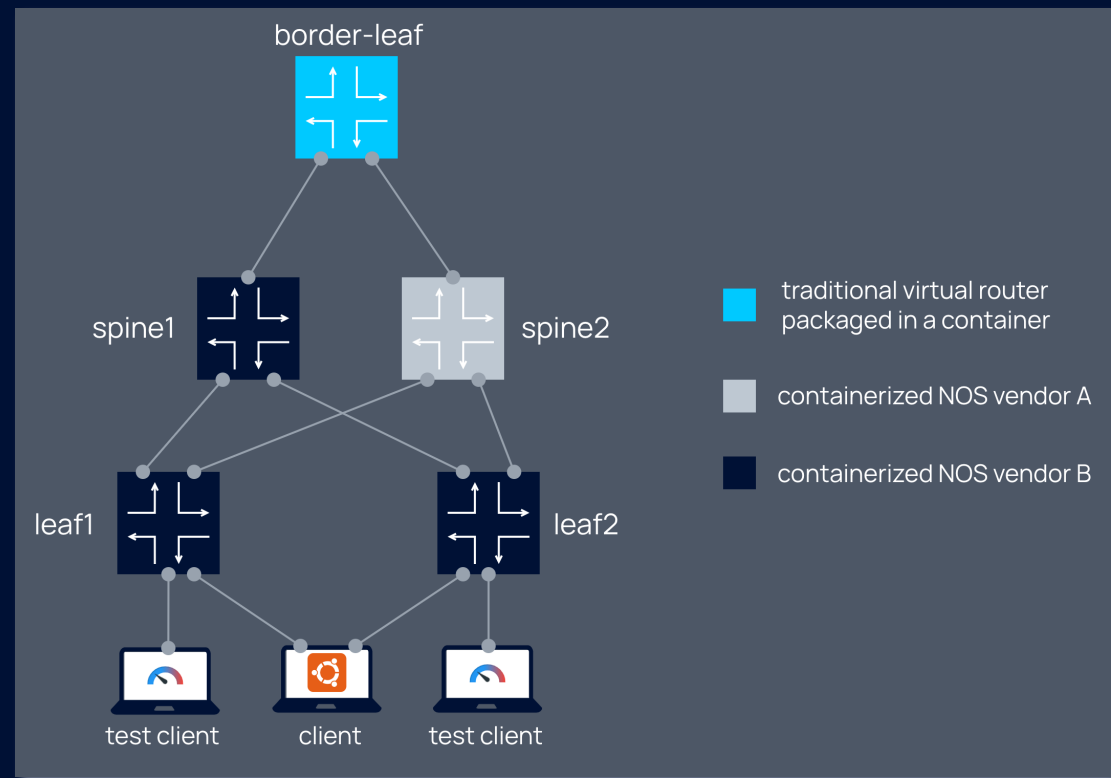
vr-ros

 Container-based NOS
vr-x VM-based NOS

Container-based and VM-based NOS

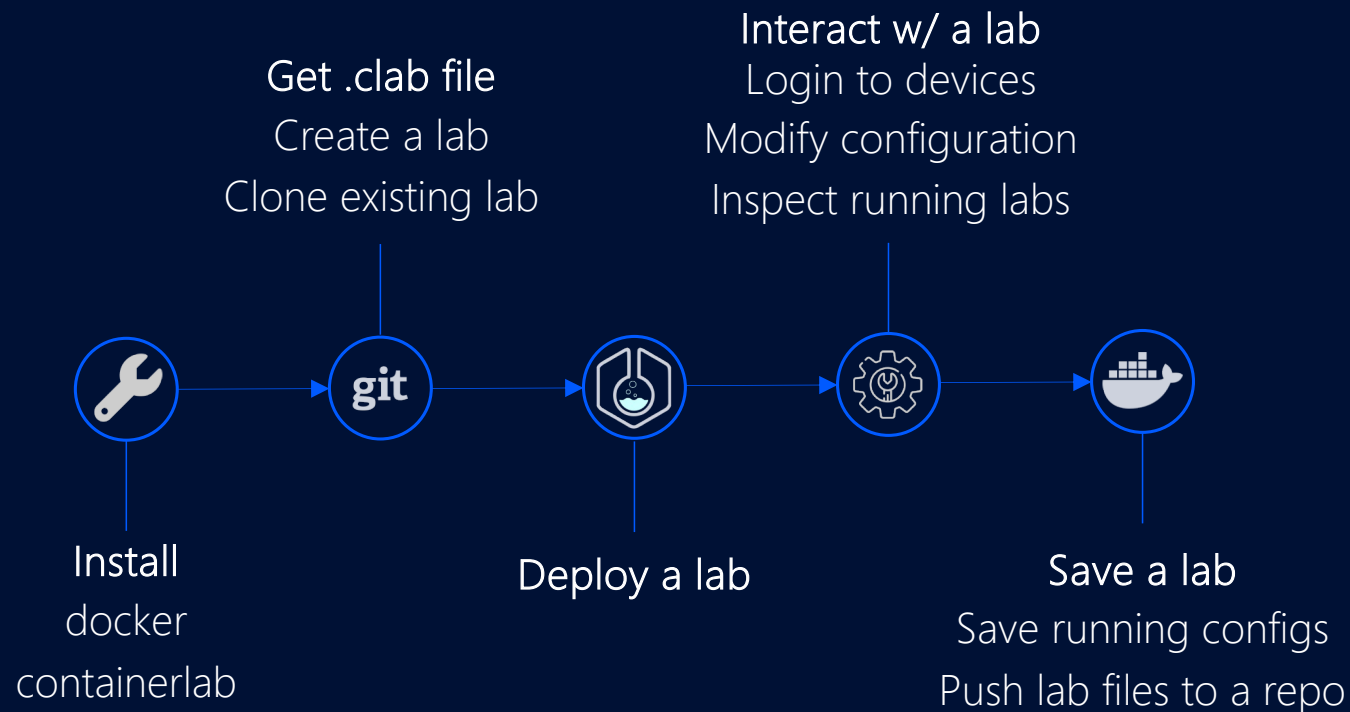
A seamless integration

- Containerlab manages labs consisting of containers
- VM-based nodes are part of a topology alongside containerized nodes
- Virtual machine is wrapped in a container image making it indistinguishable from containerized NOSes



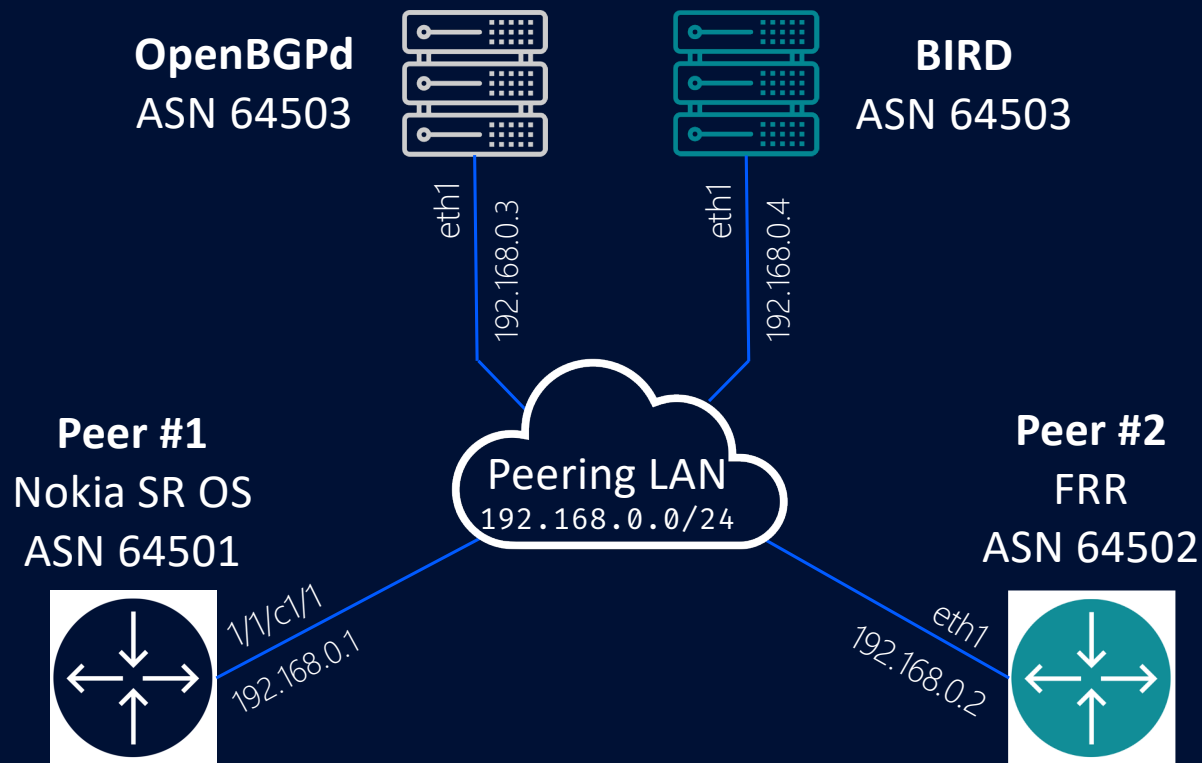
Up for some adventure ?

Basic operations on Containerlab



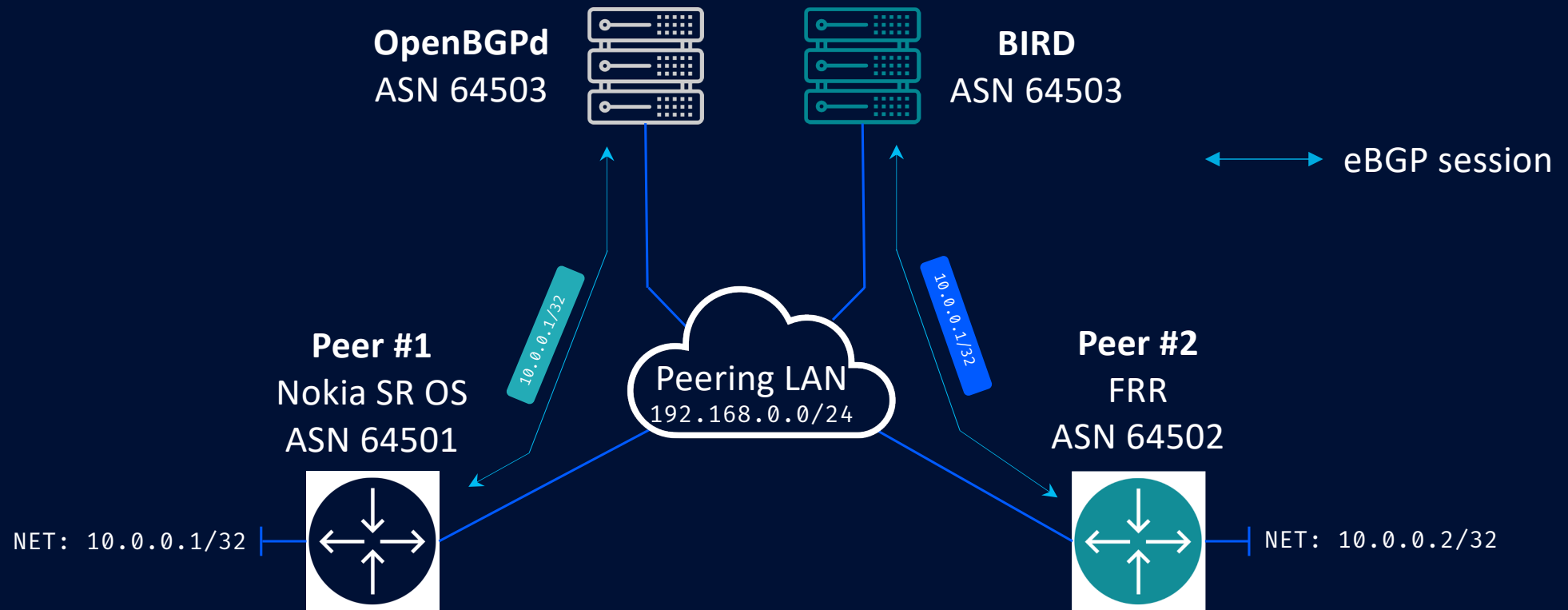
Learn by doing

Basic IXP topology with Route Servers



Learn by doing

Basic IXP topology with Route Servers



Installation

Just a single command

```
~ ➔ bash -c "$(curl -sL https://get.containerlab.dev)"  
Downloading https://github.com/srl-labs/containerlab/releases/download/v0.44.3/containerlab_0.44.3_linux_amd64.rpm  
Preparing to install containerlab 0.44.3 from package
```

```
containerlab
```

```
version: 0.44.3  
commit: cbfa6cbc  
date: 2023-08-22T12:42:06Z  
source: https://github.com/srl-labs/containerlab  
rel. notes: https://containerlab.dev/rn/0.44/#0443
```



Refer to docs for other
installation options:
<https://containerlab.dev/install/>

Building an IXP lab

Adding Nokia SR OS node

topology definition

```
name: ixp ixp.clab.yml  
  
topology:  
  nodes:  
    peer1:  
      kind: vr-nokia_sros  
      image: sros:23.3.R1  
      license: license.key
```

logical view

peer1
(Nokia SR OS)



Building an IXP lab

Adding FRR node

topology definition

```
name: ixp ixp.clab.yml
topology:
  nodes:
    peer1: {...}

    peer2:
      kind: linux
      image: quay.io/frrouting/frr:8.4.1
```

logical view



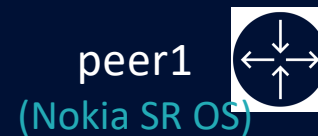
Building an IXP lab

Adding Route Servers

topology definition

```
name: ixp ixp.clab.yml
topology:
  nodes:
    peer1: {...}
    peer2: {...}
  rs1:
    kind: linux
    image: quay.io/openbgpd/openbgpd:7.9
```

logical view



Building an IXP lab

Adding Route Servers

topology definition

```
name: ixp ixp.clab.yml
topology:
  nodes:
    peer1: {...}
    peer2: {...}
    rs1: {...}

    rs2:
      kind: linux
      image: ghcr.io/srl-labs/bird:2.0.11
```

logical view



Building an IXP lab

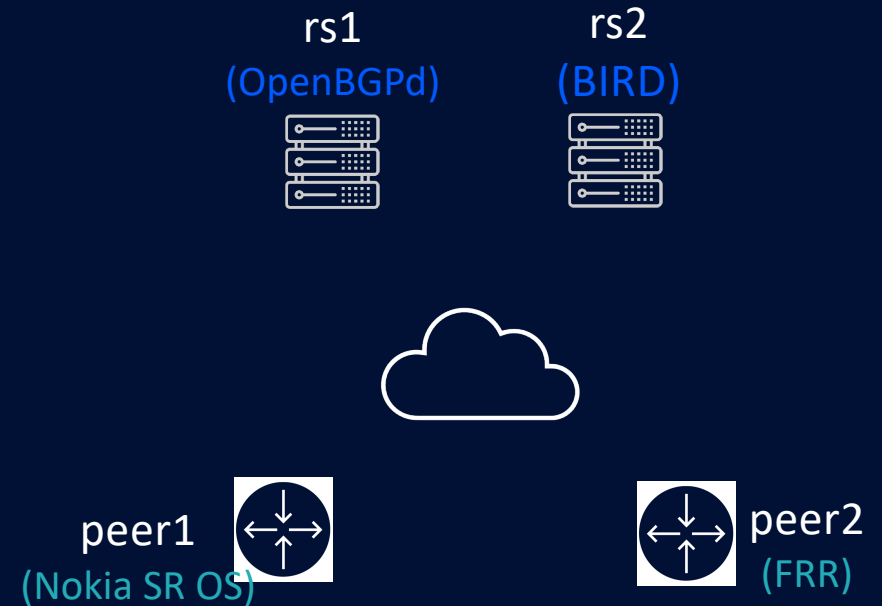
Adding Peering LAN

topology definition

```
name: ixp
topology:
  nodes:
    peer1: {}
    peer2: {}
    rs1: {}
    rs2: {}
  ixp-net:
    kind: bridge
```

ixp.clab.yml

logical view



Building an IXP lab

Adding links

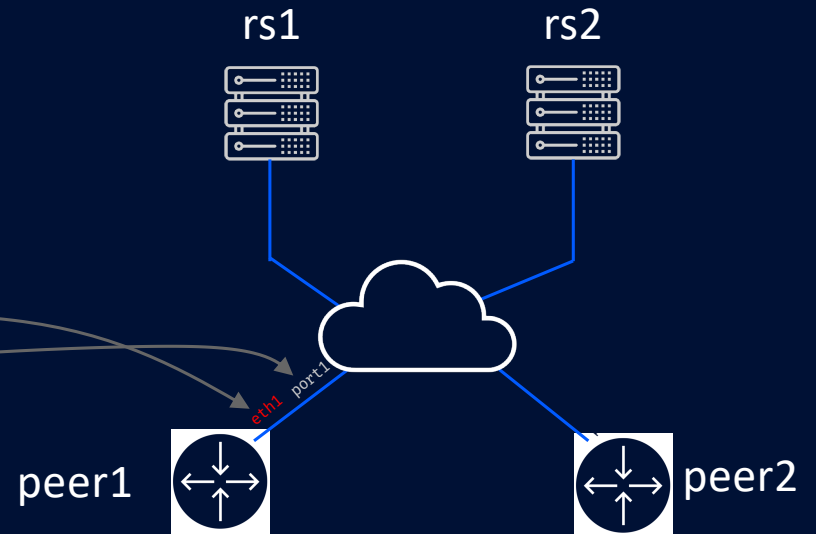
topology definition

```
name: ixp ixp.clab.yml

topology:
  nodes:
    peer1: [...]
    peer2: [...]
    rs1: [...]
    rs2: [...]
    ixp-net: [...]

  links:
    - endpoints: ["peer1:eth1", "ixp-net:port1"]
    - endpoints: ["peer2:eth1", "ixp-net:port2"]
    - endpoints: ["rs1:eth1", "ixp-net:port3"]
    - endpoints: ["rs2:eth1", "ixp-net:port4"]
```

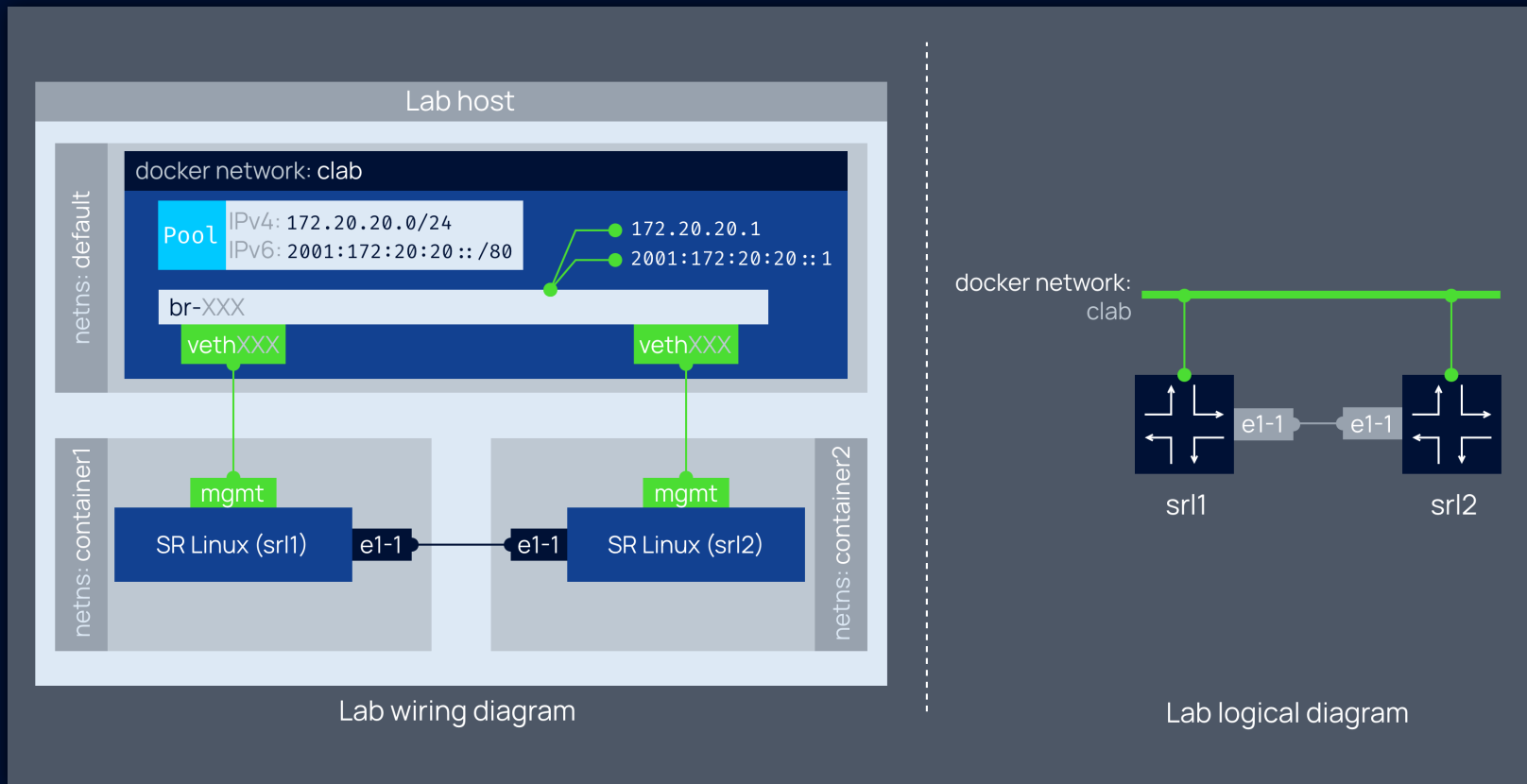
logical view



* Management links are not shown

Containerlab networking

...is based on container networking



Building an IXP lab Deployment

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

Building an IXP lab

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>
```

Building an IXP lab

Adding startup configurations

topology definition

```
name: ixp ixp.clab.yml  
  
topology:  
  nodes:  
    peer1:  
      kind: vr-nokia_sros  
      image: sros:23.3.R1  
      license: license.key  
      startup-config: sros.partial.cfg
```

logical view



Building an IXP lab

Adding startup configurations

topology definition

```
name: ixp
topology:
  nodes:
    peer1: {...}

    peer2:
      binds:
        - frr.conf:/etc/frr/frr.conf
        - daemons:/etc/frr/daemons
```

ixp.clab.yml

logical view

peer1
(Nokia SR OS)



peer2
(FRR)



NOKIA

Building an IXP lab

Adding startup configurations

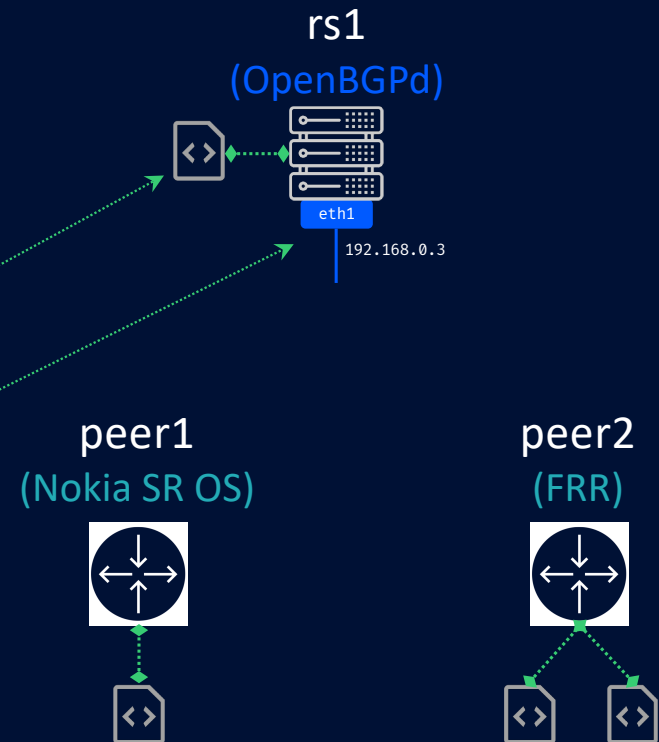
topology definition

```
name: ixp
topology:
  nodes:
    peer1: {}
    peer2: {}

    rs1:
      binds:
        - openbgpd.conf:/etc/bgpd/bgpd.conf
      exec:
        - ip address add dev eth1 192.168.0.3/24
```

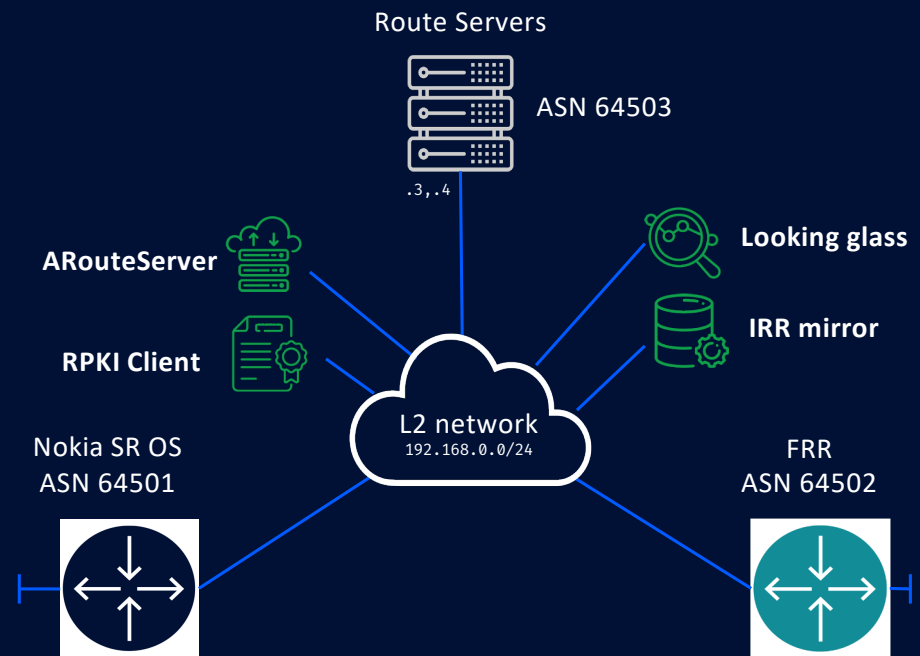
ixp.clab.yml

logical view



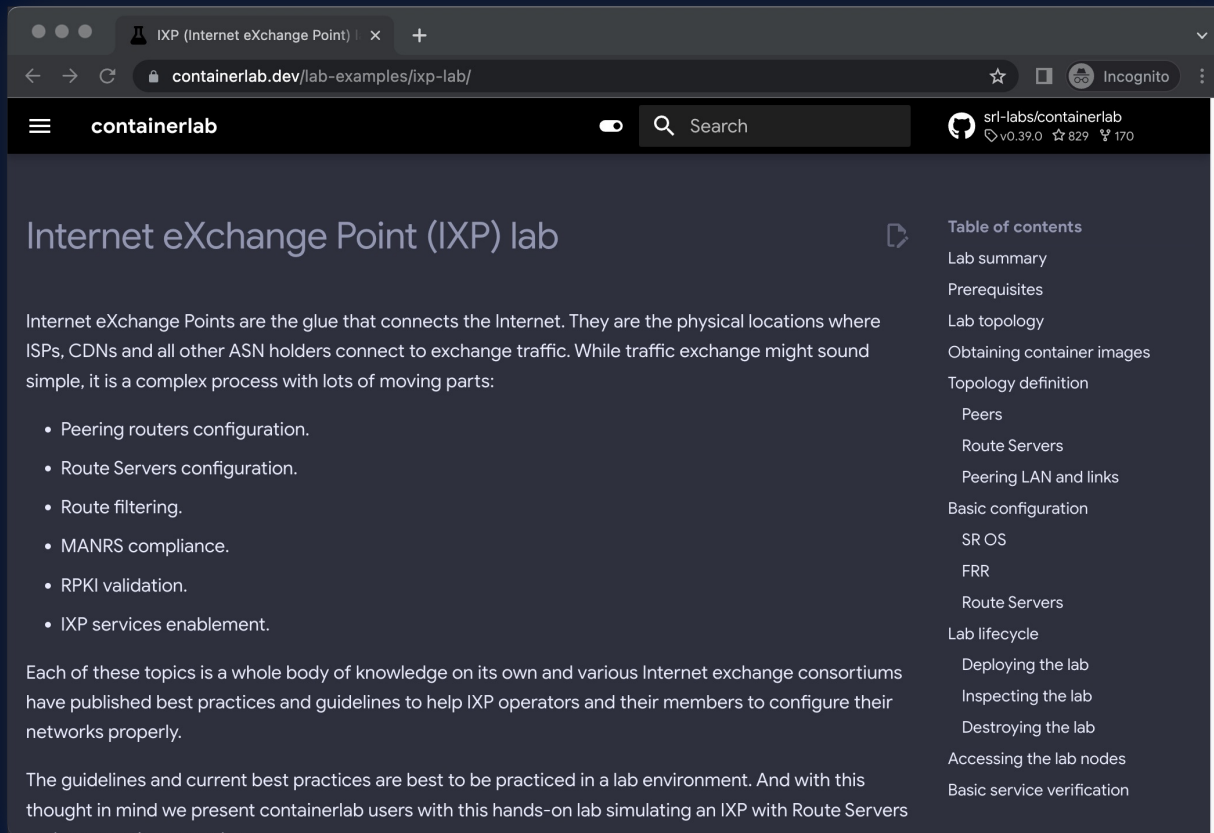
What's next ?

- ~All IXP use cases
 - ACL filtering
 - MAC filtering
 - BUM filtering
 - RPKI validation
 - MANRS conformance
 - Route Servers config (ARouteServer)
 - Looking glass integration
 - Vendor try-out



Lab

A to Z explanation



The screenshot shows a web browser window with the URL `containerlab.dev/lab-examples/ixp-lab/`. The page title is "Internet eXchange Point (IXP) lab". The main content area contains a paragraph explaining that IXP is the glue connecting the Internet and lists several topics: Peering routers configuration, Route Servers configuration, Route filtering, MANRS compliance, RPKI validation, and IXP services enablement. A table of contents is visible on the right side of the page, listing sections such as Lab summary, Prerequisites, Lab topology, and various configuration steps.

Internet eXchange Point (IXP) lab

Internet eXchange Points are the glue that connects the Internet. They are the physical locations where ISPs, CDNs and all other ASN holders connect to exchange traffic. While traffic exchange might sound simple, it is a complex process with lots of moving parts:

- Peering routers configuration.
- Route Servers configuration.
- Route filtering.
- MANRS compliance.
- RPKI validation.
- IXP services enablement.

Each of these topics is a whole body of knowledge on its own and various Internet exchange consortiums have published best practices and guidelines to help IXP operators and their members to configure their networks properly.

The guidelines and current best practices are best to be practiced in a lab environment. And with this thought in mind we present containerlab users with this hands-on lab simulating an IXP with Route Servers

Table of contents

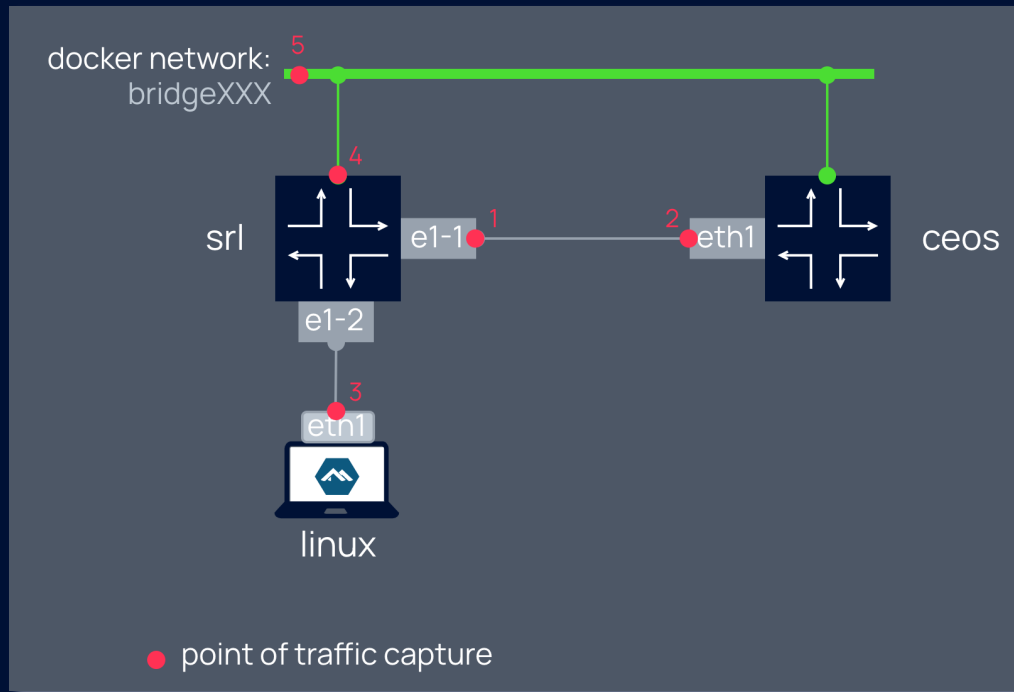
- Lab summary
- Prerequisites
- Lab topology
- Obtaining container images
- Topology definition
- Peers
- Route Servers
- Peering LAN and links
- Basic configuration
- SR OS
- FRR
- Route Servers
- Lab lifecycle
- Deploying the lab
- Inspecting the lab
- Destroying the lab
- Accessing the lab nodes
- Basic service verification



IXP-Lab

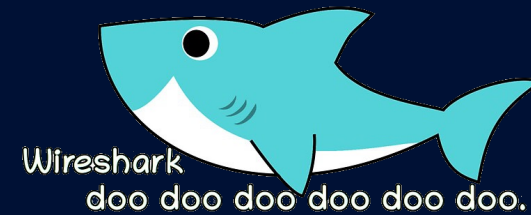
Traffic capture

Pcapng or it didn't happen



Command to capture at point #1

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



Link impairment

Ain't no network reliable enough

```
> clab tools netem set -n leaf1 -i e1-49 --delay 10ms --jitter 1ms
```

Interface	Delay	Jitter	Packet Loss	Rate (kbit)
e1-49	10ms	1ms	0.00%	0

- Delay & jitter
- Packet loss
- Rate limiting

```
> clab tools netem set -n leaf1 -i e1-50 --loss 50
```

Interface	Delay	Jitter	Packet Loss	Rate (kbit)
e1-50	0s	0s	50.00%	0

```
> clab tools netem set -n leaf1 -i mgmt0 --rate 500
```

Interface	Delay	Jitter	Packet Loss	Rate (kbit)
mgmt0	0s	0s	0.00%	500

```
> clab tools netem show -n leaf1
```

Interface	Delay	Jitter	Packet Loss	Rate (kbit)
lo	N/A	N/A	N/A	N/A
mgmt0	0s	0s	0.00%	500
e1-1	N/A	N/A	N/A	N/A
e1-49	10ms	1ms	0.00%	0
e1-50	0s	0s	50.00%	0
gway-2800	N/A	N/A	N/A	N/A
monit_in	N/A	N/A	N/A	N/A
mgmt0-0	N/A	N/A	N/A	N/A
gway-2801	N/A	N/A	N/A	N/A
e1-49-0	N/A	N/A	N/A	N/A
e1-50-0	N/A	N/A	N/A	N/A

Containerlab

Try it, join the community



[Discord server](#)



CONTAINERlab

<https://containerlab.dev>



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



NOKIA