

NOKIA



CONTAINERlab

Building multivendor IXP lab with Containerlab

Roman Dodin  @ntdvps

38th EURO-IX FORUM

Network labs

A right, not a privilege

1

Change
management

2

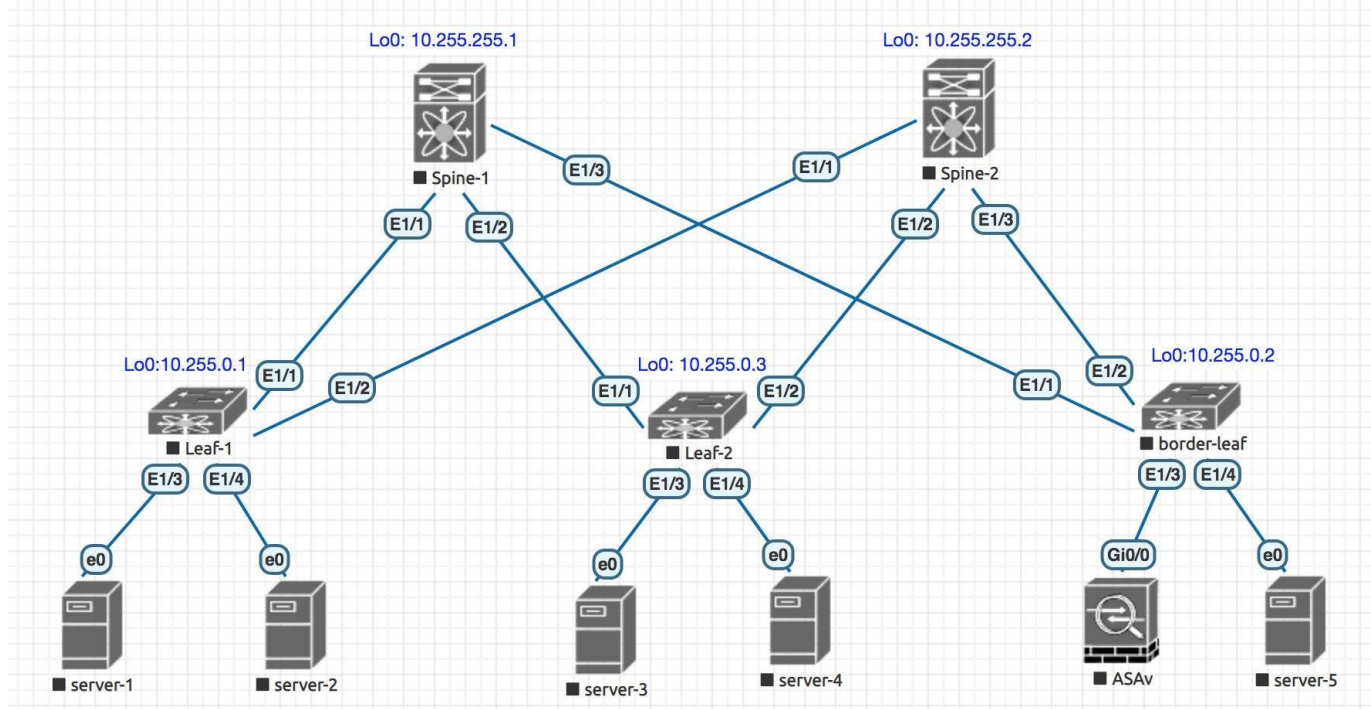
Prototyping and
validation

3

Learning

Network labs

How do we typically run labs today?

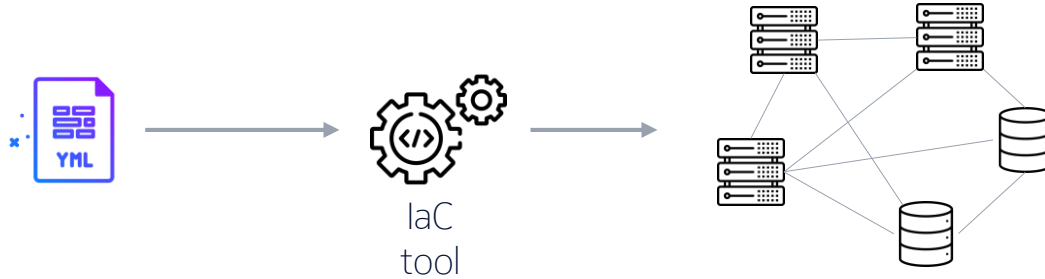


Pic from https://www.reddit.com/r/networking/comments/g5fb23/eveng_lab_strage_packet_loss/

Containerlab

Bringing declarativeness to network labs

IT

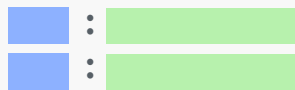


Network Labs

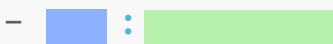
```
name: mylab
```

```
topology:
```

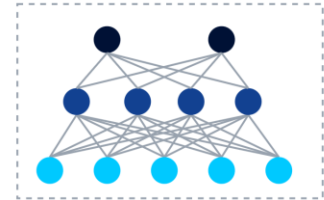
```
  nodes:
```



```
  links:
```



CONTAINERlab



Why containerlab

If we have lab emulation tools already?

Network emulation SW



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

Containerlab

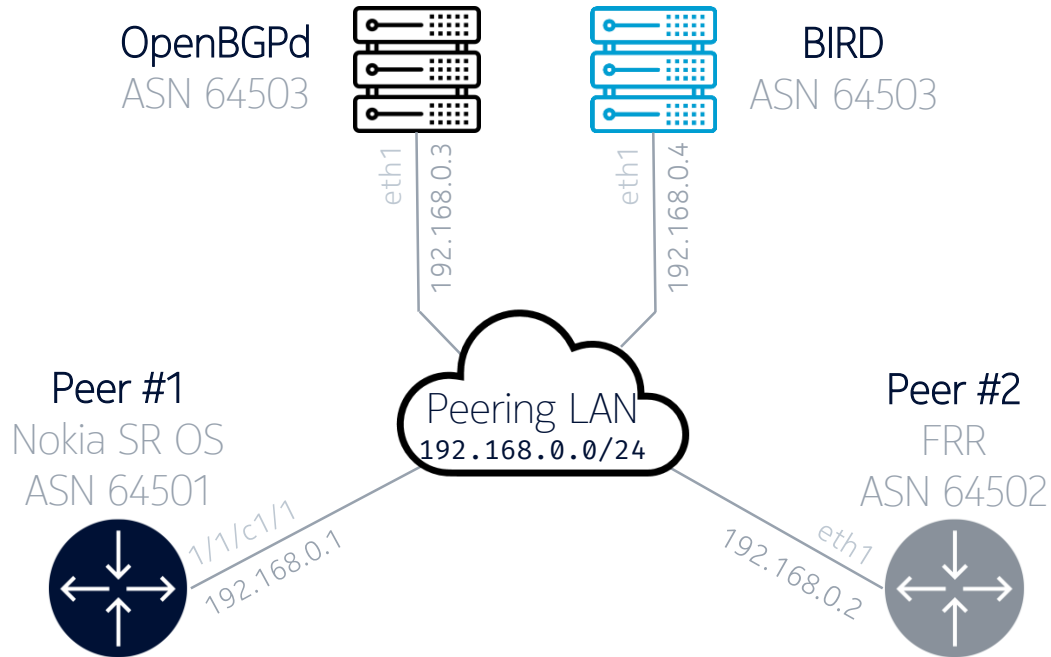


CONTAINERlab

- + 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
- Fewer Network OSes supported
- No UI

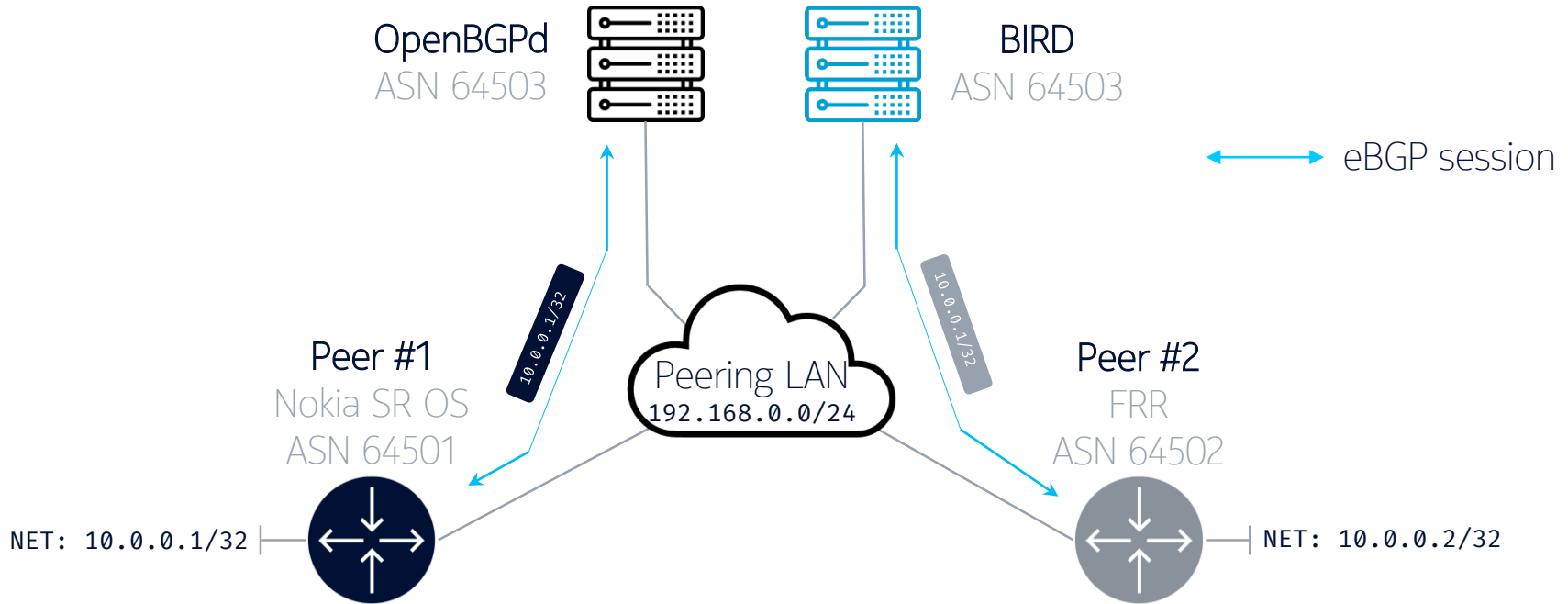
Learn by doing

Basic IXP topology with Route Servers



Learn by doing

Basic IXP topology with Route Servers



Installation

Just a single command

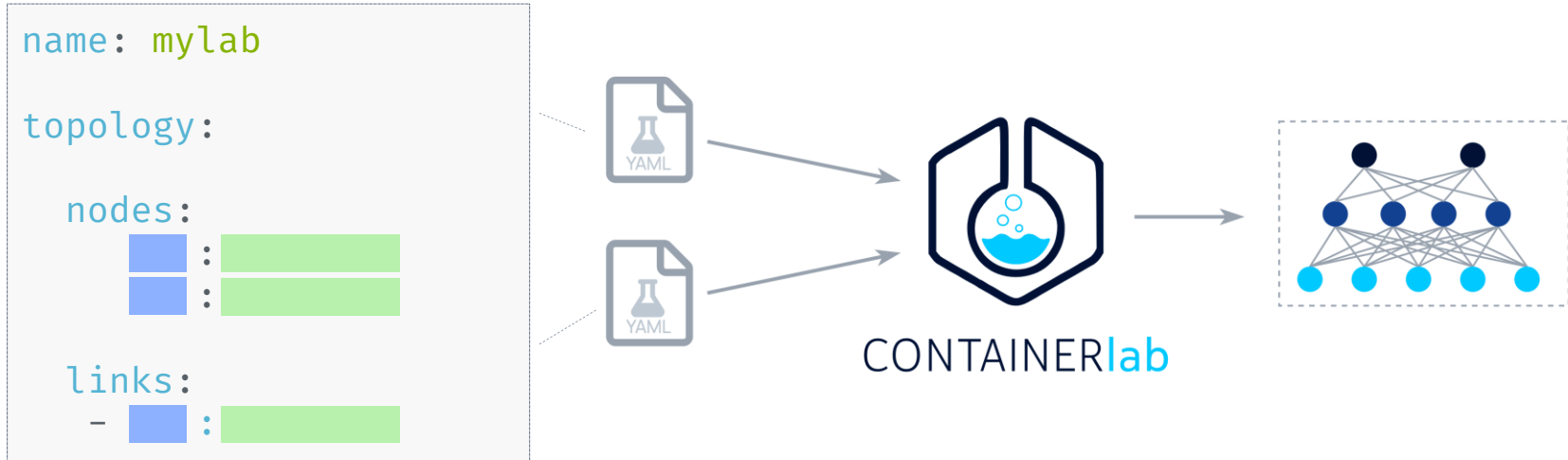
```
root@devbox:~  
bash -c "$(curl -sL https://get.containerlab.dev)"
```



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

Topology file

Declarative way to define a lab



<https://containerlab.dev>

Building an IXP lab

Adding Nokia SR OS node

topology definition

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

ixp.clab.yml

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

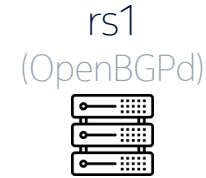
topology definition

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

    rs1:
      kind: linux
      image: quay.io/openbgpd/openbgpd:7.9
```

ixp.clab.yml

logical view



Building an IXP lab

Adding Route Servers

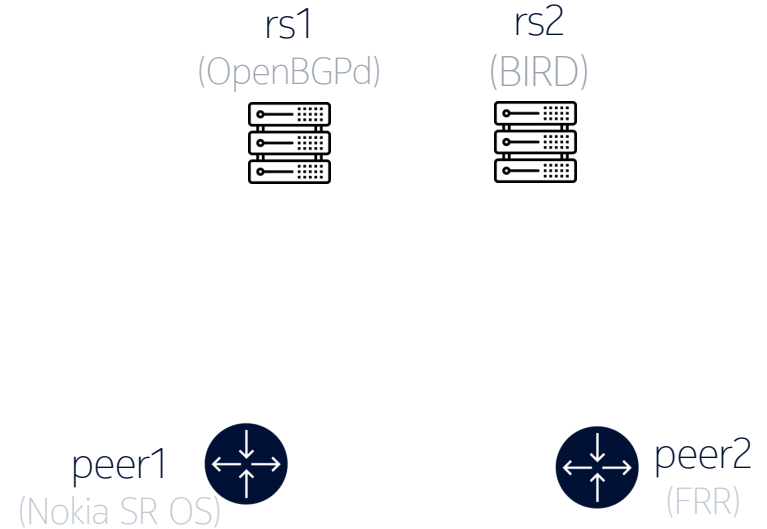
topology definition

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

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

ixp.clab.yml

logical view



Building an IXP lab

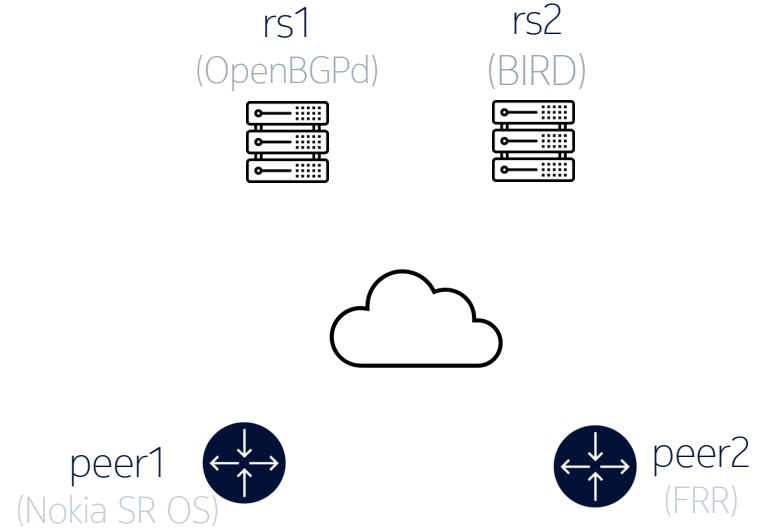
Adding Peering LAN

topology definition

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

ixp.clab.yml

logical view



Building an IXP lab

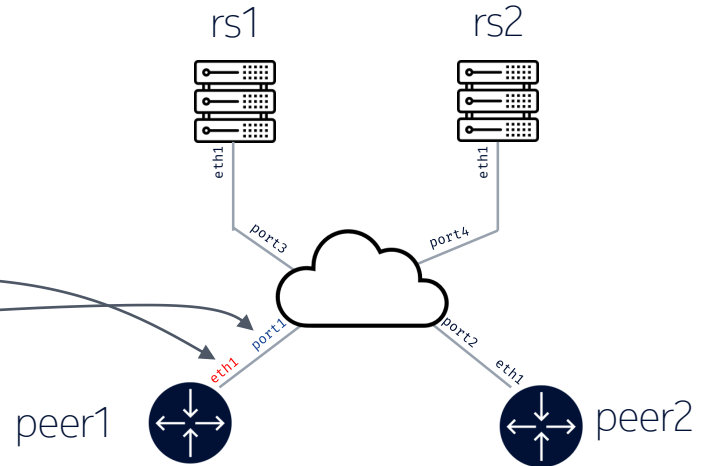
Adding links

topology definition

```
name: ixp
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"]
```

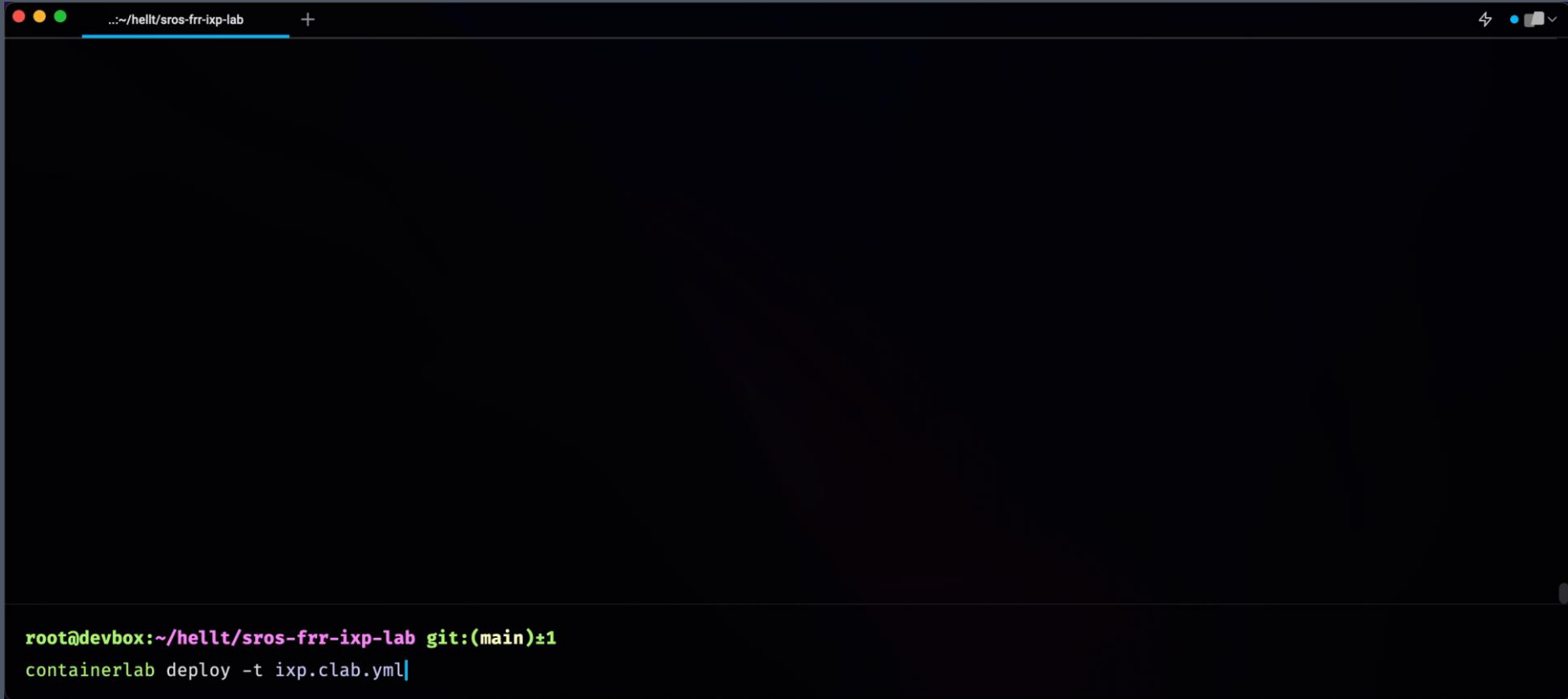
ixp.clab.yml

logical view



* Management links are not shown

Deploying the lab

A terminal window with a dark background. The title bar shows the path ~/hell/sros-frr-ixp-lab and a plus sign. The terminal content shows a prompt and a command.

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



```
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
|
```

Containerlab

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>
```

```
root@devbox:~/hell/sros-frr-ixp-lab git:(main) (11.93s)
docker exec -it clab-ixp-rs2 birdc
BIRD 2.0.11 ready.

root@devbox:~/hell/sros-frr-ixp-lab git:(main)±1
docker exec -it clab-ixp-peer2 |--detach
```

Building an IXP lab

Adding startup configurations

topology definition

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

ixp.clab.yml

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
```

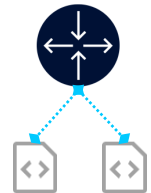
ixp.clab.yml

logical view

peer1
(Nokia SR OS)



peer2
(FRR)



Building an IXP lab

Adding startup configurations

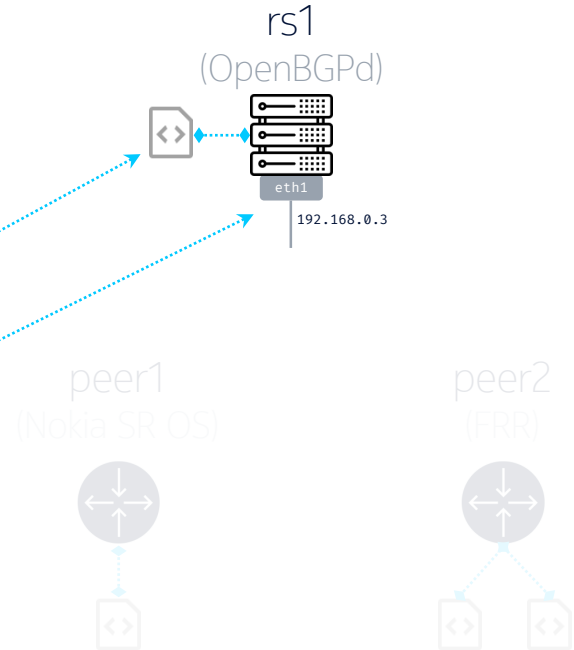
topology definition

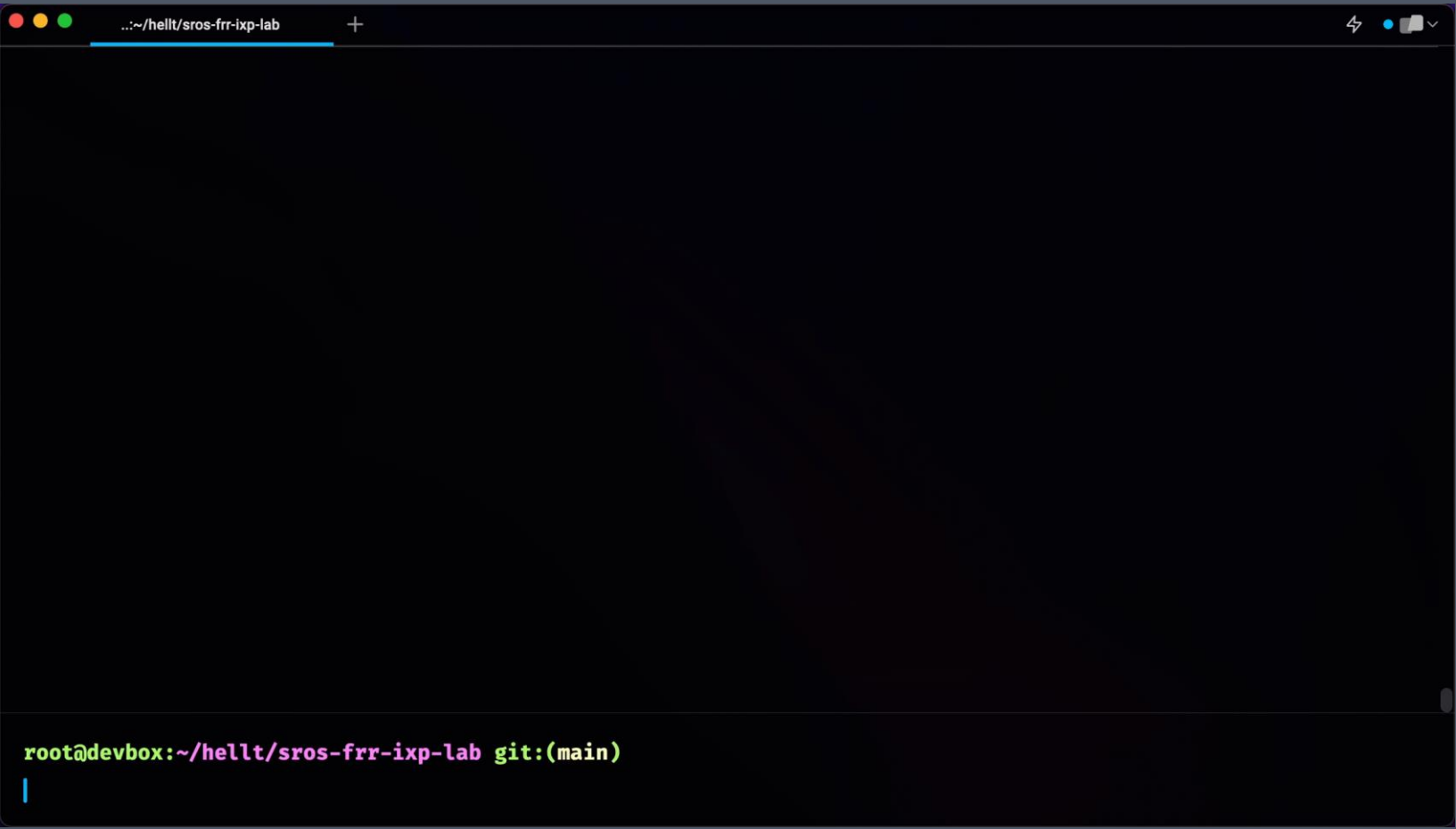
```
name: ixp
topology:
  nodes:
    peer1: {...}
    peer2: {...}

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

ixp.clab.yml

logical view



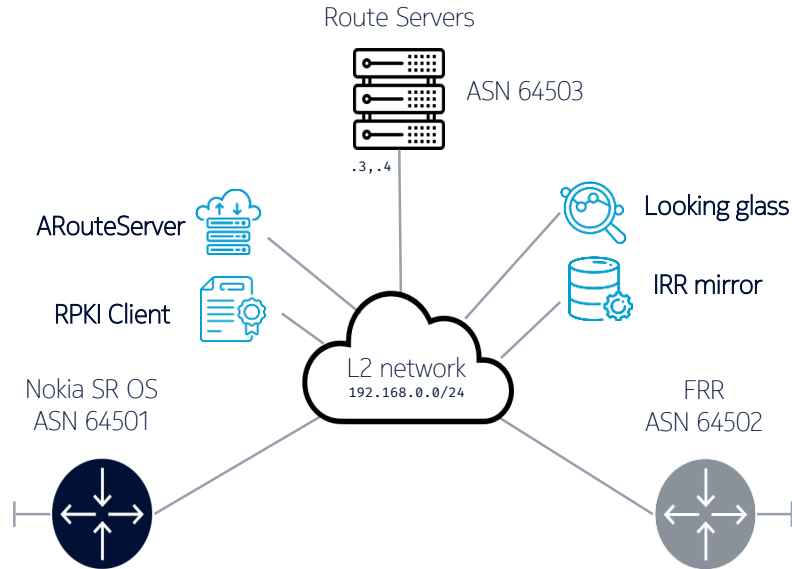


```
root@devbox:~/hellt/sros-frr-ixp-lab git:(main)
```

```
|
```

What next?

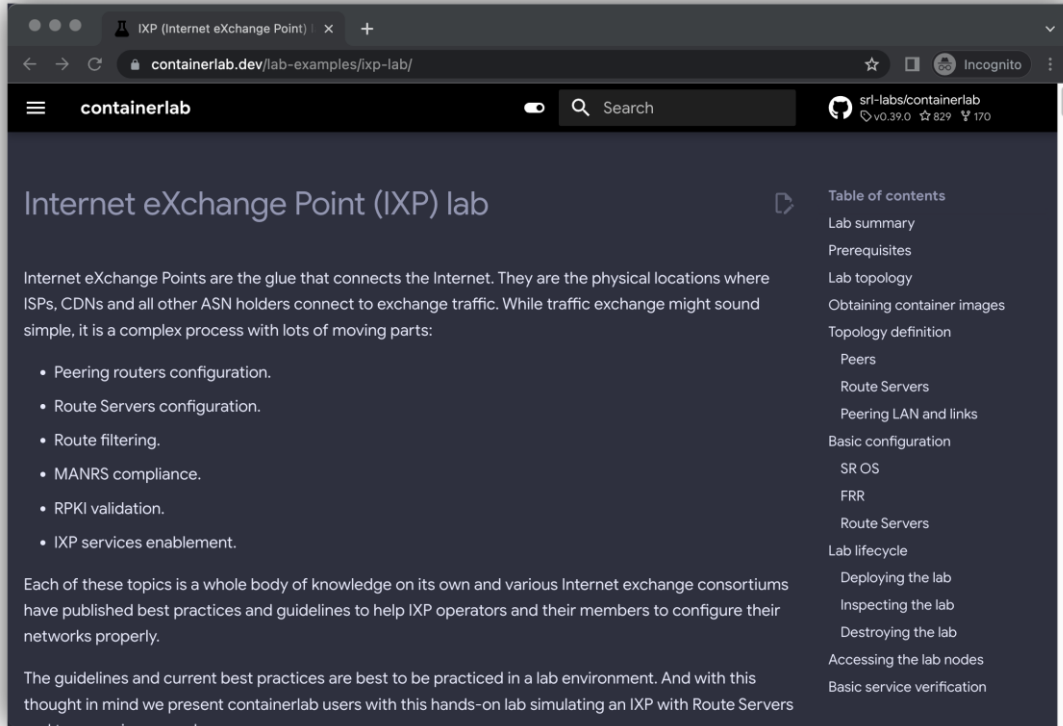
- ~All IXP use cases
 - ACL filtering
 - MAC filtering
 - BUM filtering
 - RPKI validation
 - MANRS conformance
 - Route Servers config (ARouteServer)
 - Looking glass integration
 - Vendor try-out



[hellt/sros-frr-ixp-lab](https://github.com/hellt/sros-frr-ixp-lab)

Lab

A to Z explanation



Internet eXchange Point (IXP) lab

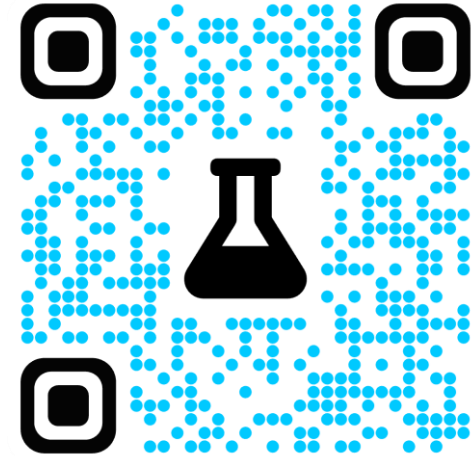
Internet eXchange Points are the glue that connects the Internet. They are the physical locations where ISPs, CDNs and all other ASN holders connect to exchange traffic. While traffic exchange might sound simple, it is a complex process with lots of moving parts:

- 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

Supported systems


NOKIA

 srl
vr-sros

JUNIPER
NETWORKS

 crpd
vr-vmx
vr-vqfx


ARISTA

 ceos
vr-veos


CISCO

vr-xrv9k
vr-csr
vr-n9kv


SONIC

 sonic-vs
 frr


NVIDIA

 cvx


paloalto
NETWORKS

vr-pan


DELL

vr-ftosv

ixia

 keysight_ixia-c

MikroTik

vr-ros

ipinfusion

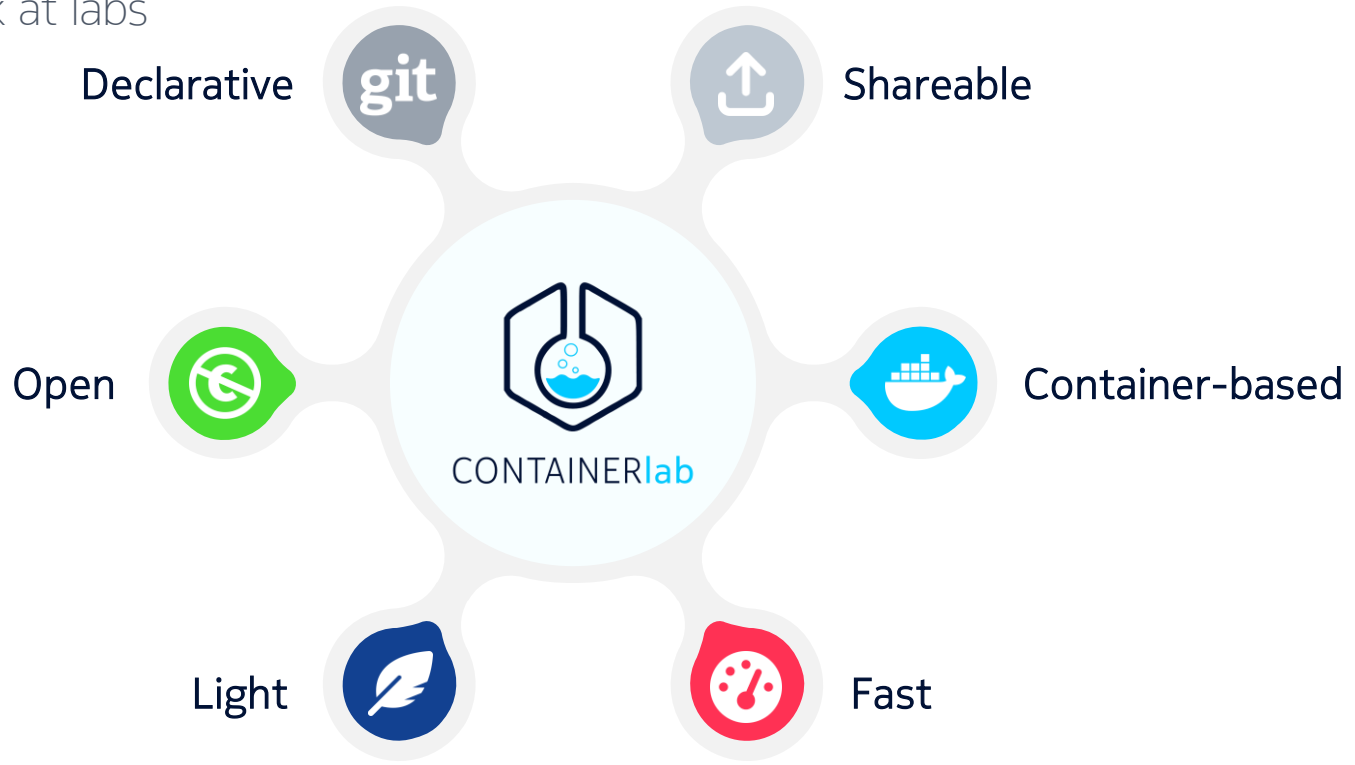
ipinfusion_ocnos


CHECK POINT

checkpoint_cloudguard

Containerlab

A different look at labs



Containerlab

Get in touch



CONTAINERlab

<https://containerlab.dev>



[Discord server](#)



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



I am here today!



CONTAINERlab

<https://containerlab.dev>