NO

CONTAINERIab

Unleashing the power of containerization for network testing and development

Songkhram Sundaranu – APAC IP Routing rPLM/Consulting Engineer BKNIX PF 2024 & ThaiNOG 6

© 2024 Nokia | Public use

Network labs A right, not a privilege





Education and Learning

Change management and validation





Cloud Native Shifting away from large - monolithic Functions



Cloud Native = Infrastructure Agnostic

Agility and Cost Savings

VMs vs. Containers

VM is loaded with the Application software and guest OS.



App 3

Bins/Libs

NO

How is this different from virtual machines

| Aspect | Virtual Machine | Docker Container |
|---------------------|-----------------|------------------|
| HW Interfaces | Emulated | Native |
| Operation System | Support Various | Linux mainly |
| Running Spaces | User | Kernel |
| Isolation Technique | Hypervisor | CGroups |
| System Overhead | 5~15% | 0 ~ 5% |
| Start Up Time | Minutes | Seconds |
| Storage Image | GB - TB | KB - MB |
| Cluster Scale | 100 + | 10000 + |



Declarative domination

A de-facto standard in infra/service provisioning



© 2024 Nokia 6

How they deploy things over there? Declarative, infrastructure as code approach





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



8 © 2024 Nokia Public use

Lab as code with Containerlab



Containerlab Bringing declarativeness to network labs



Why Containerlab If we have lab emulation tools already?

Network emulation SW



- Purpose built & proven
 Free versions available
- D UI



- Heavy and semi-open
- U





Supported NOSes





Why Lab as Code? #2 - Versioning



Why Lab as Code? #3 - Sharing







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





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



Learn by doing Basic IXP topology with Route Servers





19 © 2024 Nokia | Public use

Learn by doing Basic IXP topology with Route Servers



Installation Just a single command





21 © 2024 Nokia | Public use



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



Building an IXP lab Adding Nokia SR OS node

| topology definition | | logical view | | |
|--|--------------|------------------------|--|--|
| name: ixp topology: nodes: peer1: kind: vr-nokia_sros image: sros:23.3.R1 license: license.key | ixp.clab.yml | peer1 (Nokia SR OS) | | |
| 22 © 2024 Nokia Public use | | NOSIA | | |

Building an IXP lab Adding FRR node

| topology definition | | logical view | | |
|--|--------------|------------------------|----------------------|--|
| name ixp | ixp.clab.yml | | | |
| topology nodes peer1 {} | | | | |
| peer2 kind: linux image quay.io/frrouting/ | frr:8.4.1 | | | |
| | | peer1 (Nokia SR OS) | peer2 (FRR) | |
| 23 © 2024 Nokia Public use | | | NO <ia< td=""></ia<> | |

Building an IXP lab Adding Route Servers



Building an IXP lab Adding Route Servers



Building an IXP lab Adding Peering LAN



Building an IXP lab Adding links



Building an IXP lab Deployment

••• ..:~/hellt/sros-frr-ixp-lab 47 🔹 🚛 🗸 root@devbox:~/hellt/sros-frr-ixp-lab git:(main)±1 containerlab deploy -t ixp.clab.yml



28 © 2024 Nokia | Public use

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>

29 © 2024 Nokia | Public use



Building an IXP lab Adding startup configurations

| topology definition | logical view | | |
|---|------------------------|--|--|
| <pre>name: ixp topology: nodes peer1: kind: vr-nokia_sros image: sros:23.3.R1 license: license.key startup-config: sros.partial.cfg</pre> | peer1 (Nokia SR OS) | | |
| 30 © 2024 Nokia Public use | NO <ia< td=""></ia<> | | |

Building an IXP lab Adding startup configurations



Building an IXP lab Adding startup configurations



IXP Lab A to Z explanation

| ● ● ● ▲ IXP (Internet eXchange Point) ⊢ × + | | |
|--|------------|---|
| \leftarrow \rightarrow C \triangleq containerlab.dev/lab-examples/ixp-lab/ | | 🖈 🔲 👼 Incognito : |
| ≡ containerlab | 🖸 Q Search | Srl-labs/containerlab Sv0.39.0 ☎829 ¥170 |

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

| ab summary |
|----------------------------|
| rerequisites |
| ab topology |
| Obtaining container images |
| opology definition |
| Peers |
| Route Servers |
| Peering LAN and links |
| asic configuration |
| SR OS |
| FRR |
| Route Servers |
| ab lifecycle |
| Deploying the lab |
| Inspecting the lab |
| Destroying the lab |
| ccessing the lab nodes |
| asic service verification |





NO<IA

33 © 2024 Nokia | Public use



Other labs for the community

34 © 2024 Nokia Public use

Containerlab networking ...is based on container networking



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 -



VO<IA

Link impairment Ain't no network reliable enough

| > | clab tools | netem se | et -n leaf | 1 -i e1-490 | delay 10msjitter | 1m |
|---|------------|----------|------------|-------------|------------------|----|
| | Interface | Delay | Jitter | Packet Loss | Rate (kbit) | |
| + | e1-49 | 10ms | 1ms | 0.00% | | |

|) clab tools netem set -n leaf1 -i e1-50loss 50 | | | | | | |
|---|-------|--------|-------------|-------------|--|--|
| Interface | Delay | Jitter | Packet Loss | Rate (kbit) | | |
| e1-50 | 0s | 0s | 50.00% | 0 | | |

|) clab tools netem set -n leaf1 -i mgmt0rate 500 | | | | | | |
|--|-------|--------|-------------|-------------|--|--|
| Interface | Delay | Jitter | Packet Loss | Rate (kbit) | | |
| mgmt0 | 0s | 0s | 0.00% | 500 | | |

- Delay & jitter
- Packet loss
- Rate limiting

| > clab tools netem show -n leaf1 | | | | | | |
|----------------------------------|--------------|--------------|----------------|------------|--------|--|
| Interface | Delay | Jitter | Packet Loss | Rate | (kbit) | |
| lo mgmt0 | N/A 0s | N/A 0s | N/A 0.00% | N/A | 500 | |
| e1-1 | N/A 10ms | N/A 1ms | N/A | N/A | e i | |
| e1-50 | | 0s | 50.00% | N/A | 0 | |
| monit_in | N/A | N/A | N/A | N/A N/A | | |
| mgmt0-0 gway-2801 | N/A N/A | N/A N/A | N/A N/A | N/A N/A | | |
| e1-49-0 e1-50-0 | N/A N/A | N/A N/A | N/A N/A | N/A N/A | | |
| + | + | + | + | | + | |

NO<IA

37 © 2024 Nokia | Public use

Containerlab Try it, join the community



Discord server





srl-labs/containerlab

38 © 2024 Nokia | Public use



