narrowIN

Building Digital Twins

with Containerlab – using ntop tools and Wireshark for traffic analysis.

Mischa Diehm
CTO
narrowin.ch

Who?

Mischa Diehm

- Founder of narrowin
- Network design and development
- Computer and network infrastructure

narrowin

- Networking and security
- Micro-/Endpoint segmentation
- Lightweight Network Explorer

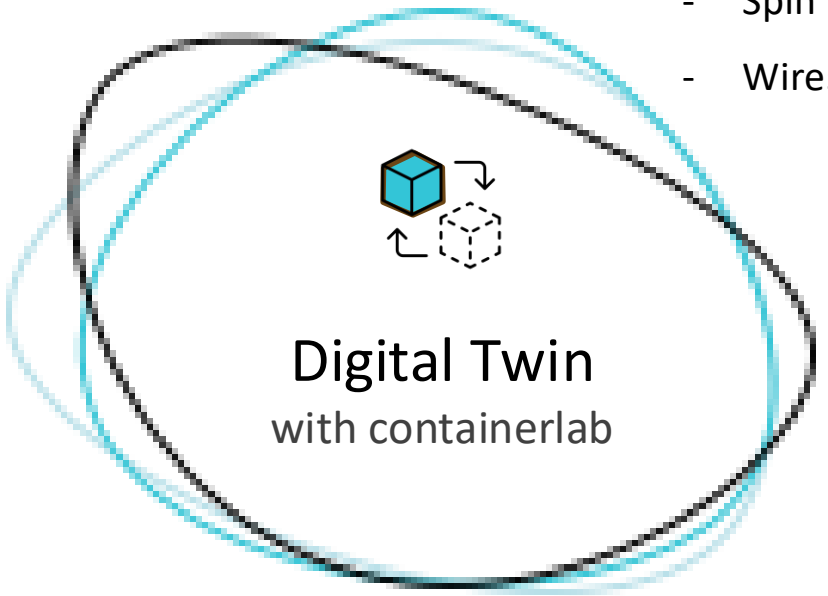
https://demo.narrowin.ch

University of Basel SWISSGAS eniwa EnBW Kantonsspital Baselland ganz nah

What can I use a Digital Twin of my Network for?

Network Development

- Design
- Implementation
- Testing
- Validation



Education

- Spin up parts of your prod network on your laptop
- Wireshark on all links

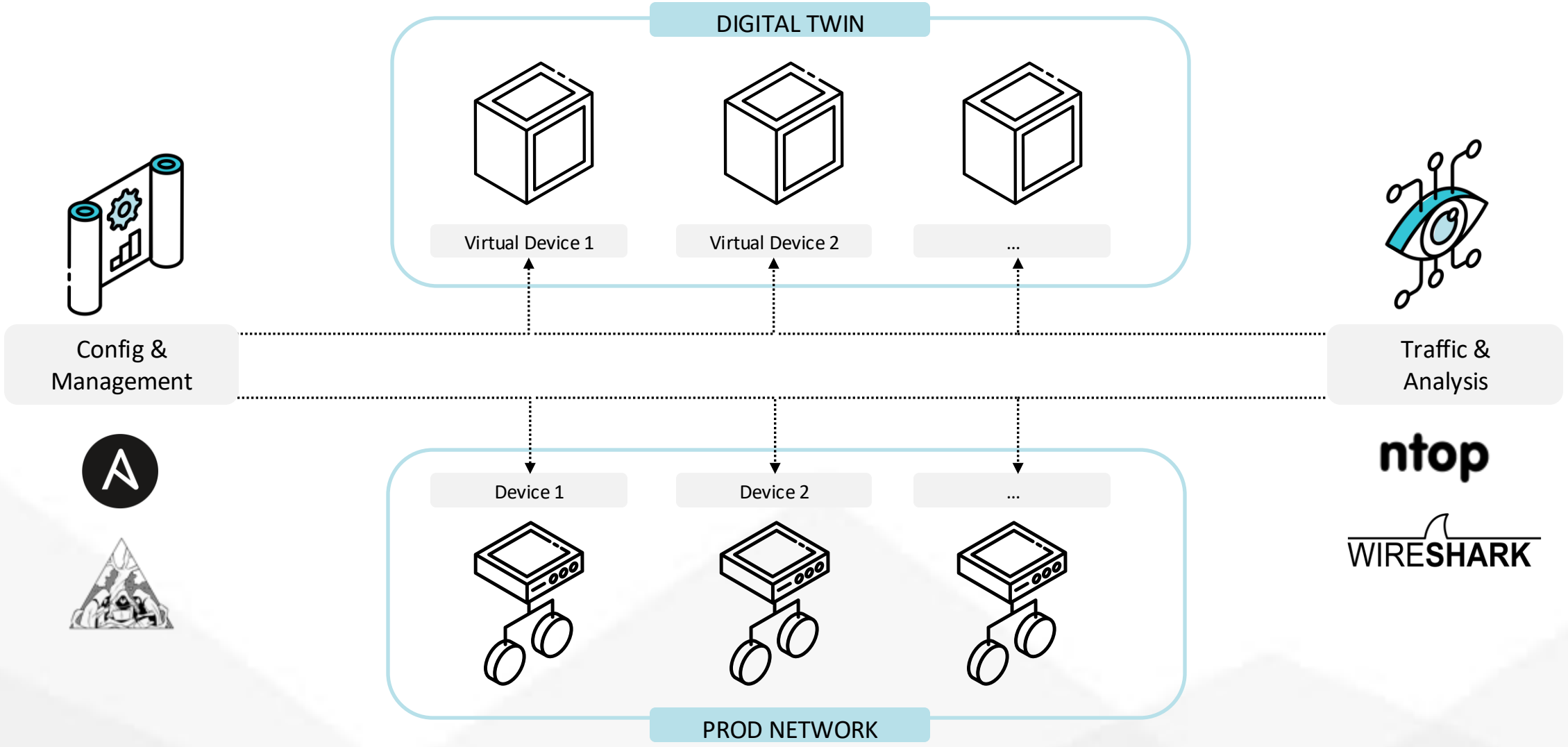
Testing

- New tools for your production network
  - Ntop, Netbox, librenms, …
- Run and test your full ci/cd pipelines
- Test and validate security systems
  - IDS detection, alarming, FW-Rules
- Drive automation

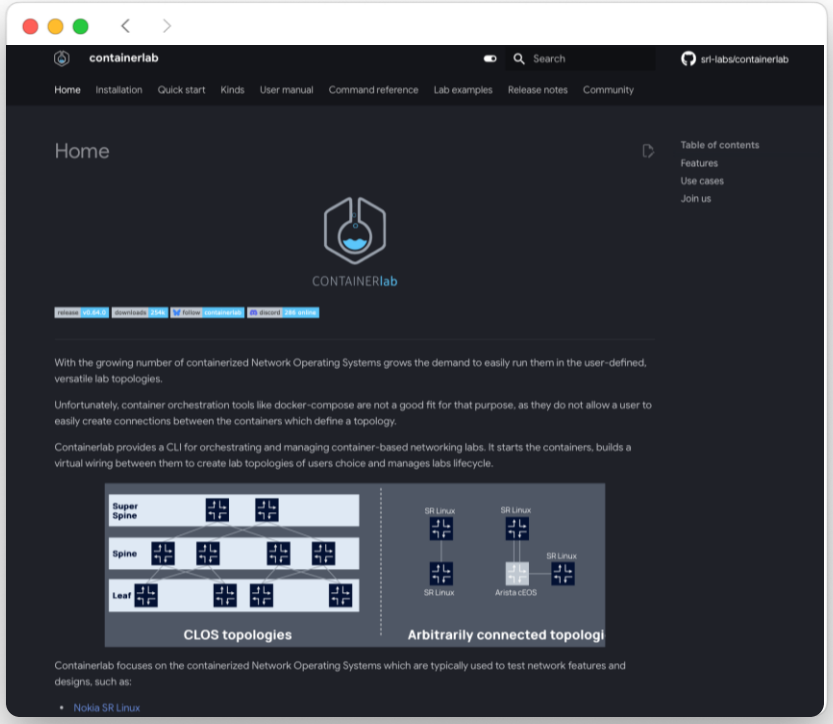
Operations

- Run a full production clone – if needed - in Multi-node labs
- Combine containerlab with your real HW-labs

Running in containerlabs



Introducing Containerlab



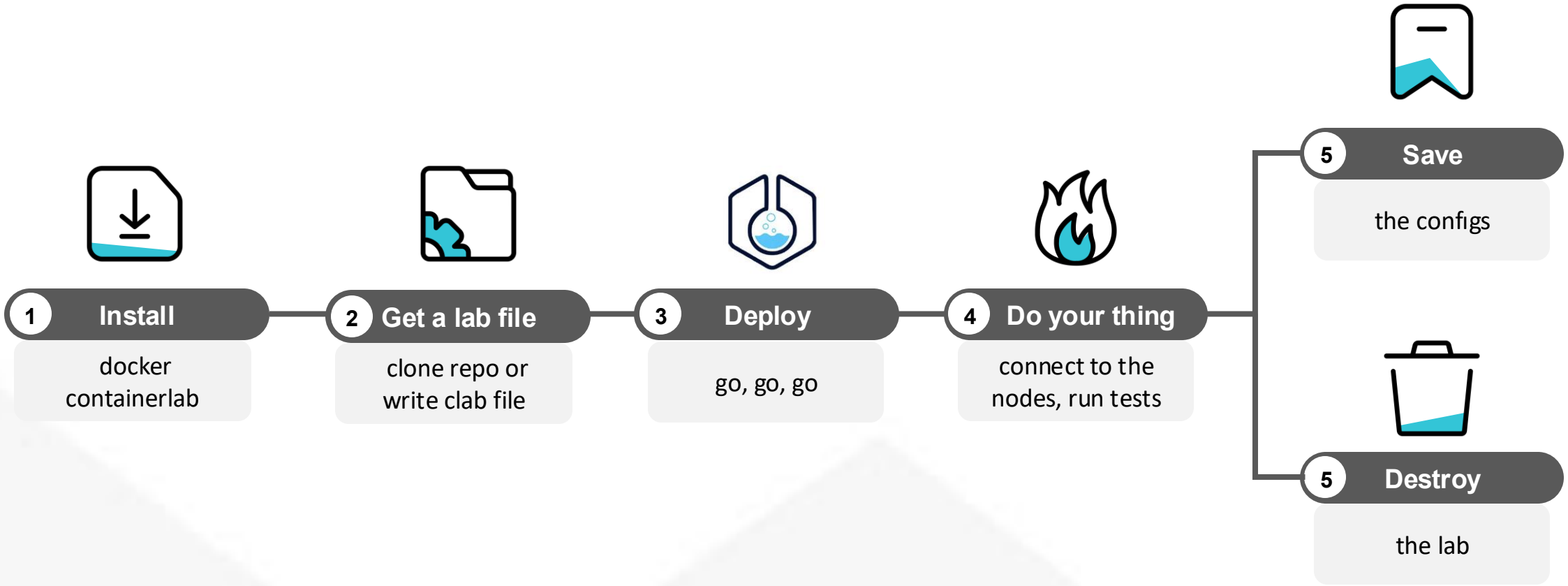
https://containerlab.dev

«Containerlab provides a CLI and GUI for orchestrating and managing container-based networking labs.

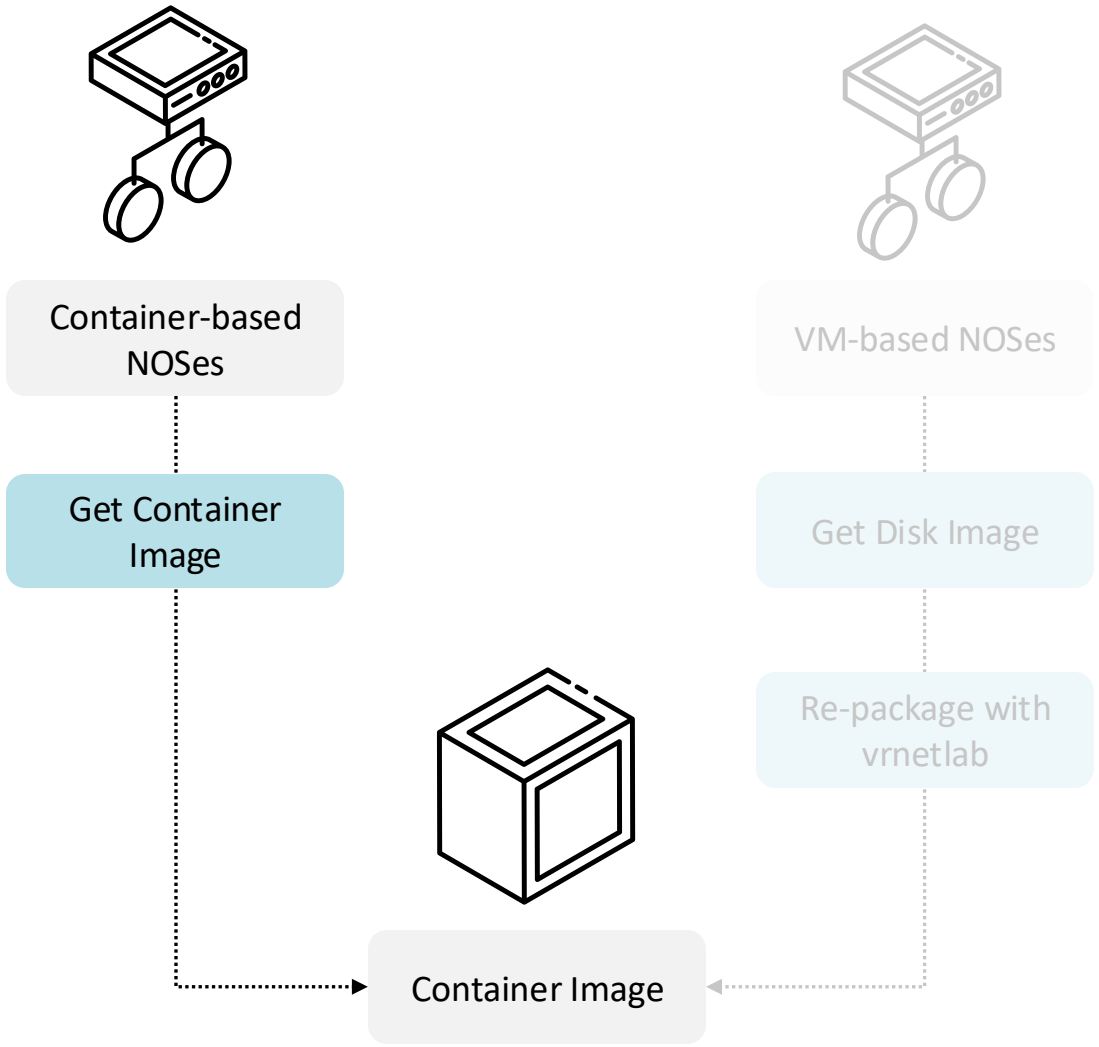
It starts the containers, builds a virtual wiring between them to create lab topologies of users' choice and manages labs lifecycle.»

- ✓ Covers many vendors
- ✓ Declarative by nature
  - Easy topology definition
- ✓ Scales really well

Containerlab workflow



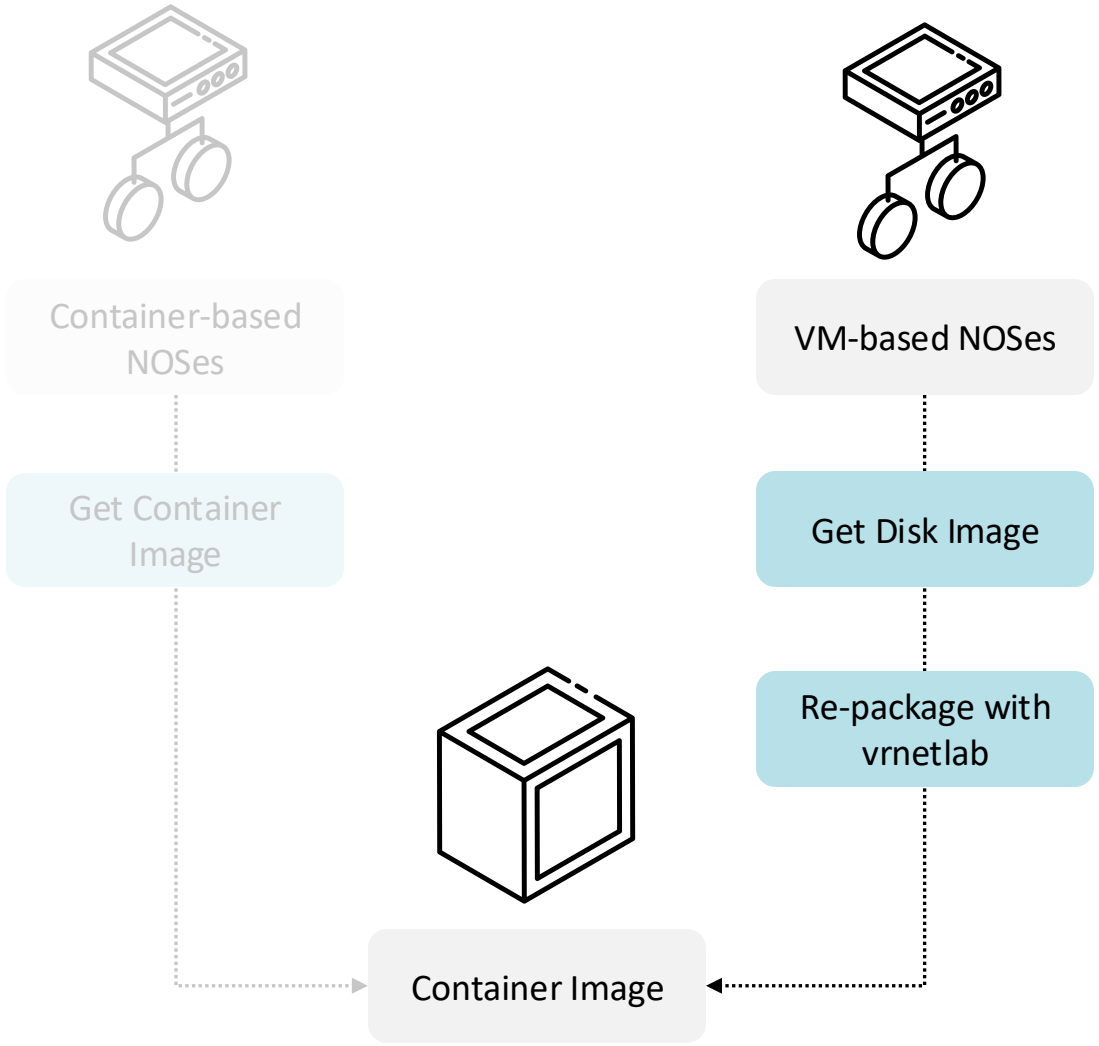
Where do I get a container Image?
Containerized NOSes



- Sourced by the vendor
- Fast and easy to use

The trend is to move away from VM packaging towards containers. Still, many NOS are VM-based.

Where do I get a container Image? Containerizing VM-based NOSes



- Use vrnetlab to containerize
- Launch topologies with VM-based NOS within the same topology definition file, alongside containerized NOS.
- > 30 NOS kinds supported

https://github.com/hellt/vrnetlab

https://containerlab.dev/manual/vrnetlab/#supported-vm-products

Important: Containerlab uses original vrnetlab project fork hellt/vrnetlab. Container built with upstream vrnetlab project will not be compatible with Containerlab.

Containerlab basics: Topology file definition

```yaml
topology:
  kinds:
    mikrotik_ros:
      image: ghcr.io/narrowin/vrnetlab_mikrotik_routeros:7.18
    linux:
      image: ghcr.io/network-unit-testing-system/nuts-testclient:0.0.2
      env:
        ADMIN_PASSWORD: admin
  nodes:
    # SWITHCES
    sw-acc1:
      kind: mikrotik_ros
      mgmt-ipv4: 10.10.1.11
      startup-config: startup-configs/sw-acc1.rsc
      env:
        CLAB_MGMT_PASSTHROUGH: "true"
    # ENDPOINTS / CLIENTS
    linux1:
      kind: linux
      mgmt-ipv4: 10.10.1.101
      exec:
        - ip address add 10.1.1.1/24 dev eth1
    linux2:
      kind: linux
      mgmt-ipv4: 10.10.1.102
      exec:
        - ip address add 10.1.1.2/24 dev eth1
    ntap1:
```

`clab deploy`	deploy the topology (start the lab).
`clab destroy`	shut down the lab.
`ssh clab-mylab-mkt1`	connect to the node.

Containerlab creates static entries in the `/etc/hosts` file and sets up `/etc/ssh_config.d/` to allow you to use SSH.

Live Demo / Screencast

Pray to the demo gods

Debugging with Wireshark

Command Line

Executing the capture script

```
# ~/bin/clab_pcap.sh cs.foo clab-s3n-sw-acc2 ether2
... execs:
 ssh cs.foo 'sudo ip netns exec clab-s3n-sw-acc2 tshark -l  -i ether2 -w -’ |  /usr/bin/wireshark -k -i -
```

GUI

- Edgeshark general stand-alone virtual network/communication diagnosis tool
- Can comfortably capture live container network traffic in Wireshark, using the csharg external capture plugin for Wireshark

Ntop tooling

NTOPNG

- Containerizes setups: https://github.com/ntop/docker-ntop
- adapt: docker-compose.yml

```
services:
  ntopng:
-   image: ntop/ntopng:stable
+   image: ntopng-dev
...
-   command: [ '-i', 'tcp://*:5556c', '-F', 'clickhouse', '--disable-login' ]
+   command: ['--license-mgr', '/etc/ntopng-lm-client.conf', '-w', '127.0.0.1:3333','-i',  'ntap:5678:secret', '-i', 'tcp://*:5556c', '-F', 'clickhouse', '--disable-login', '0']
...
```

NTAP

- Change Dockerfile.ntap.dev

```
-ENTRYPOINT ["/run.sh"]
+ENTRYPOINT ["/bin/bash"]
```

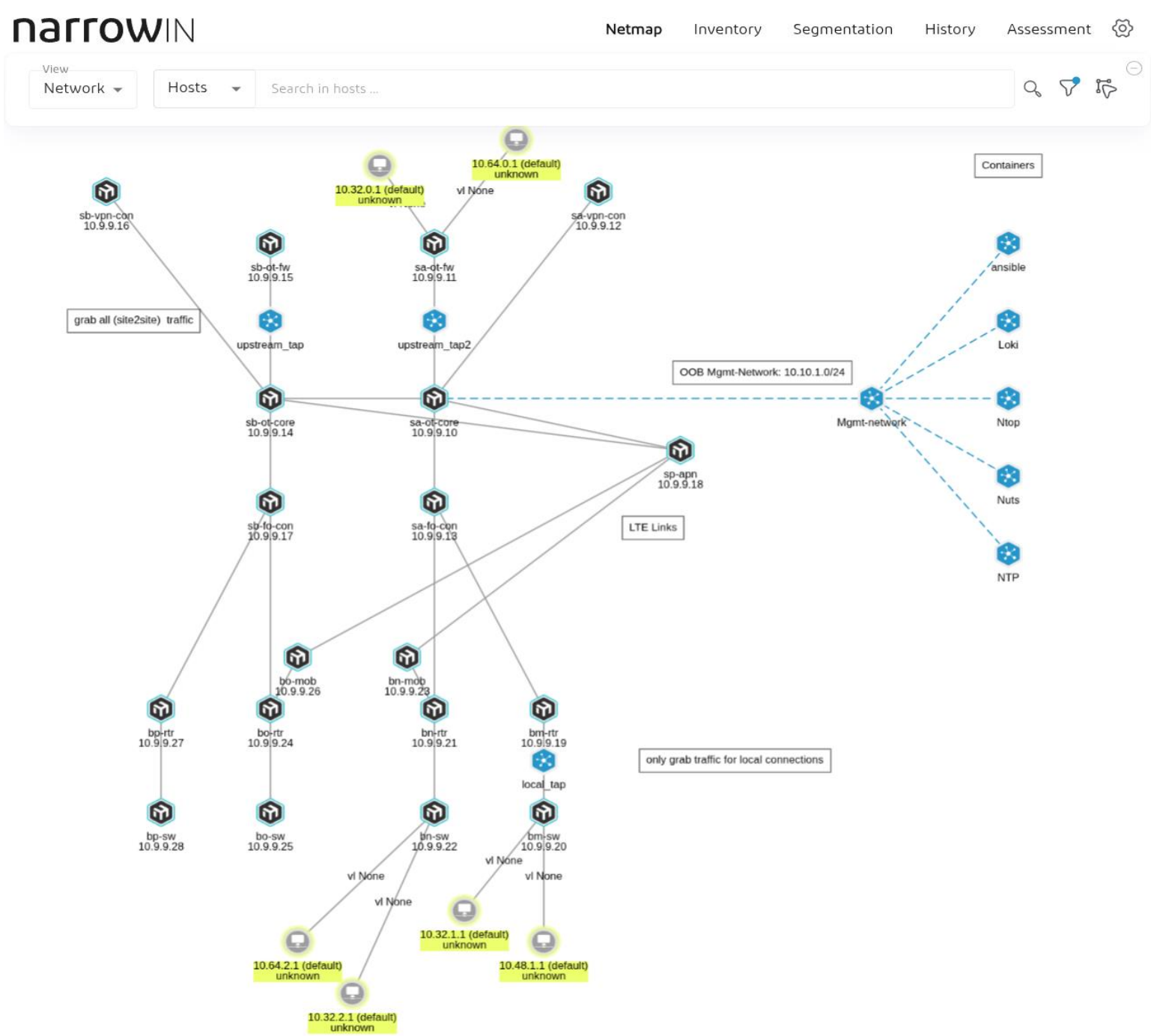
- Tweak clab node definition

```
ntap1:
  kind: linux
  image: ntap-clab
  mgmt-ipv4: 10.10.1.103
  binds:
    - ../setup-bridge-eth1-eth2.sh:/setup-bridge.sh
  cmd: ntap_remote -i eth2 -c 10.10.1.1:5678 -k secret
  exec:
    - bash /setup-bridge.sh
```

Transform Real Network Into Digital Twin

- Map your production network topolgy to containerlab
  - Use a software like the narrowin LNE that can generate contrainlab topos for you
  - Write/wait for tooling that taps into e.g. your SoT like netbox and does the limbo
- Use your production running configs in containerlab
  - Interface name mappings
    - Can be done - if supported - with interface aliases in containerlab
    - Renaming of interfaces inside the NOS itself
  - HW related features possibly NOT available in virtualized NOS
    - MLAG (multi-chassis link aggregation)
    - Mirror/span ports
    - Switch stacks
- Virtual wiring leads to link states always being up (watch out when testing fail-over scenarios)
- Some NOS features might work differently on virtual NOS than on real HW (e.g. logging in CHR)

OT-WAN Lab with network services



Some useful remarks for your labs:

- Dynamic inventory automatically created for anisble and nornir
  - Labels will be translated into group membership (hopefully soon also for nornir)
- Run your labs without any local dependencies
  - Local with devpod
  - Remote with github codespaces
- Share access to your labs with sshx a secure web-based, collaborative terminal
- External connectivity: https://containerlab.dev/lab-examples/ext-bridge/
- Containerlab API: https://github.com/srl-labs/clab-api-serve

Live Demo / Screencast

Pray to the demo gods

Lab examples for inspiration

- https://containerlab.dev/lab-examples/lab-examples/ – huge number of very advanced labs
- https://ccie-sp.gitbook.io/ccie-spv5.1-labs – all labs for Cisco CCIE Service Provider v5.1
- https://github.com/srl-labs/srl-telemetry-lab – The lab topology consists of a Clos topology, plus a Streaming Telemetry stack comprised of gnmic, prometheus and grafana applications.
- https://github.com/narrowin/ansible-mikrotik/ - Automating MikroTik Device Management with Ansible

- https://containerlab.dev/manual/topo-def-file/ - containerlab docs -> absolutely exceptional!
- https://www.youtube.com/@RomanDodin - great vidoes on many aspects of containerlab

Thanks – stay in touch

LINKEDIN

mischa.diehm@narrowin.ch