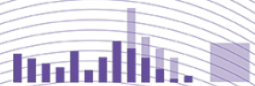


CSNOG 2024  
2024-01-23

QRATOR  
LABS

# Simulace sítě

Alexander Zubkov  
green@qrator.net

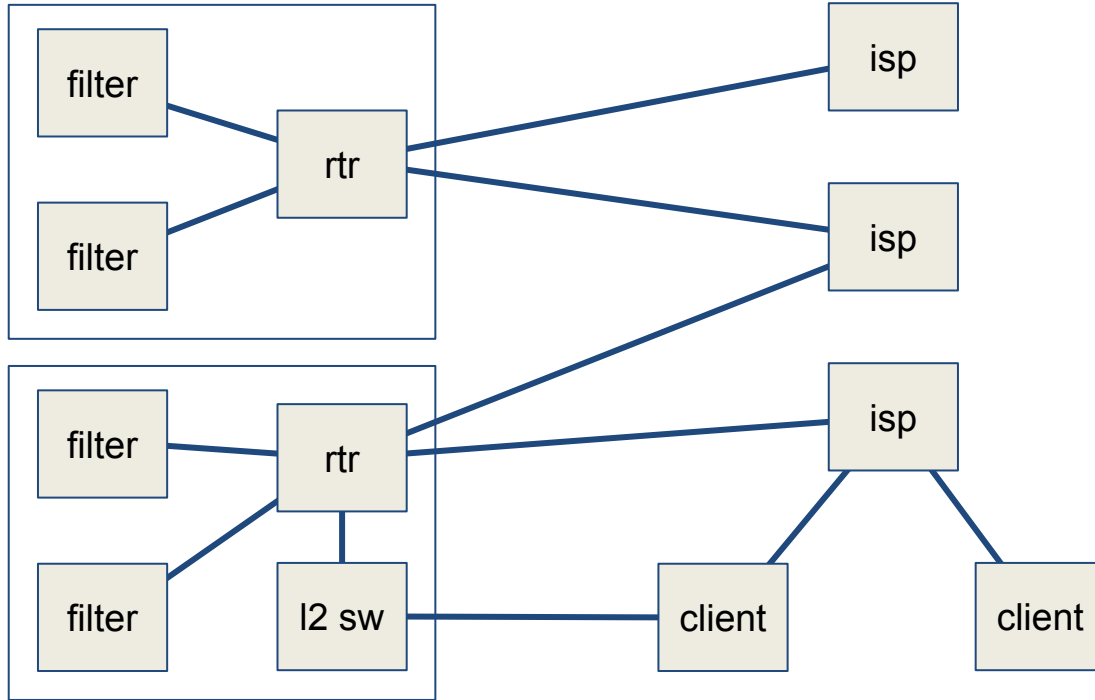


- DDoS mitigation service
- anycast sites around the world
  - filtering nodes
  - network equipment

- NOC: port, VLAN, IP, BGP, metrics, ...
- Linux
  - servers
  - Mellanox, switchdev, [mlxtoolkit](#)
- automated configuration (IaaS)
  - calculated/configs
  - scripts/ansible

- **soft**
  - bird (features, bugs, syntax, ...)
  - mlxtoolkit
  - scripts, ...
- **templates**
- **runtime behaviour**
- **interoperation with clients**

- single instance
  - missing scale, interoperation
  - real hw
- static test stand
  - hard to repeat or update
- network simulation
  - how to run model
  - how to replicate network



- [Open-Source Network Simulators](#)
  - GNS3, Mininet, ...
- [Batfish](#)
- [Hadal](#)
- [Digital Twin Network \(IETF draft\)](#)
- bird: birdlab, netlab (custom)
- frr: [topotest](#) (custom) based on [munet](#)

- lightweight
  - many Linux nodes
  - containers, not VM
- first attempt
  - custom scripts with netns
  - only one “image”
- [containerlab](#)



- <https://containerlab.dev/>
- docker
- images
  - NOS images library
  - VMs are supported
- YAML configuration

```
name: test
topology:
  nodes:
    node1:
      kind: linux
      image: emumlx
    node2:
      kind: nokia_srlinux
      image: ghcr.io/nokia/srlinux
  links:
    - endpoints: ["node1:port1", "node2:eth1"]
```

```

~# cd ~/clab-test
~/clab-test# clab deploy
INFO[0000] Parsing & checking topology file: qrator.clab.yml
INFO[0001] Creating lab directory: /root/clab-test/clab-qrator
INFO[0001] Creating container: "lumen"
INFO[0001] Creating container: "mellanox08"
INFO[0001] Creating container: "tier0"
INFO[0057] Creating virtual wire: lumen:port1 <--> mellanox08:port56
INFO[0081] Creating virtual wire: lumen:port2 <--> tier0:port2
INFO[0107] Adding containerlab host entries to /etc/hosts file
+-----+-----+-----+-----+-----+-----+-----+-----+
| # | Name | Container ID | Image | Kind | State | IPv4 Address | IPv$
+-----+-----+-----+-----+-----+-----+-----+-----+
| 18 | clab-qrator-lumen | bb371325200d | emuisp | linux | running | 172.20.20.22/24 | 2001:17$
| 22 | clab-qrator-mellanox08 | 124b6ee275c0 | emumlx | linux | running | 172.20.20.12/24 | 2001:17$
| 31 | clab-qrator-tier0 | 7ba518467a94 | emuisp | linux | running | 172.20.20.32/24 | 2001:17$
~/clab-test# clab destroy
INFO[0000] Parsing & checking topology file: qrator.clab.yml
INFO[0000] Destroying lab: qrator
INFO[0031] Removed container: clab-qrator-tier0
INFO[0074] Removed container: clab-qrator-lumen
INFO[0076] Removed container: clab-qrator-mellanox08
INFO[0081] Removing containerlab host entries from /etc/hosts file

```

- base image
  - wait for interfaces
- “Mellanox” image
  - almost equivalent
  - other ACL semantics
- same image for ISP, switches

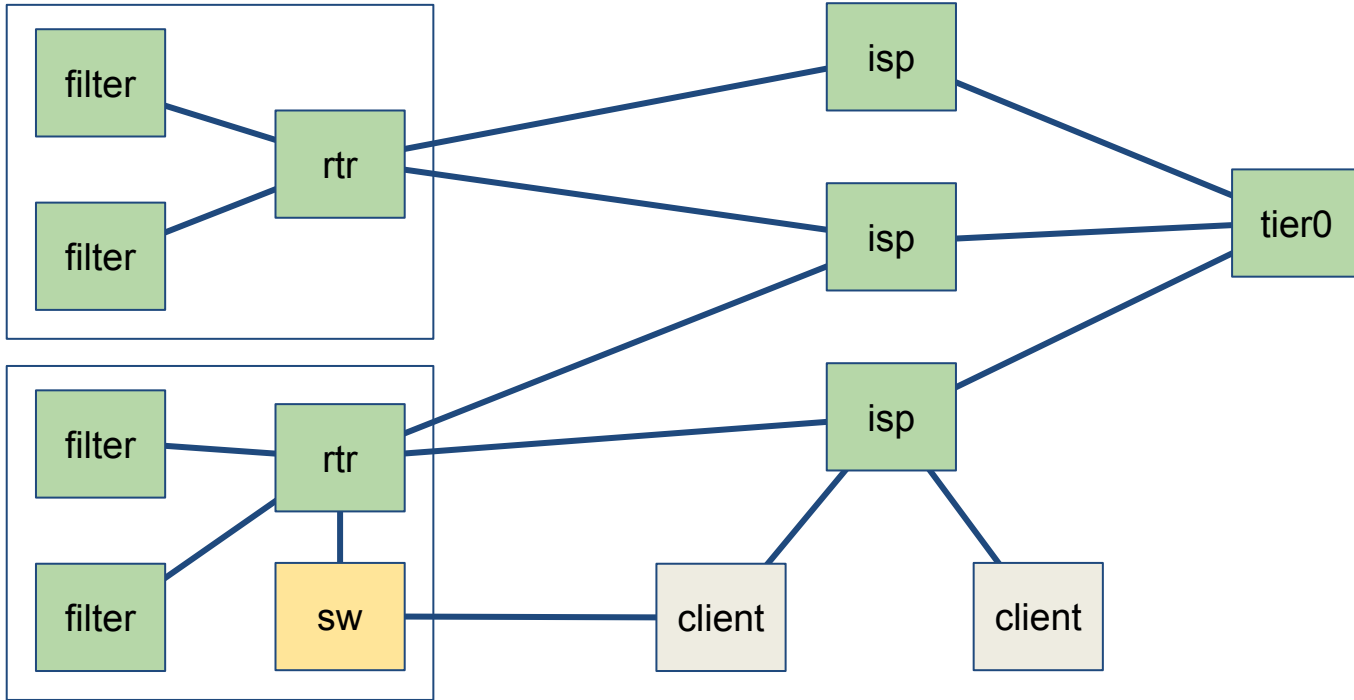
```
# cat Dockerfile
```

```
...  
CMD ["/sbin/waitlink"]
```

```
# cat assets/waitlink
```

```
...  
while true; do  
    NUM="$(ip -o -br link show type veth | grep -v '^eth0[@ ]' | wc -l)"  
    [[ "$NUM" -eq "${CLAB_INTFS:-0}" ]] && break  
    sleep 1  
done  
  
...  
if [[ $# -eq 0 ]]; then  
    exec /sbin/init  
else  
    exec "$@"  
fi
```

- same source
  - automated configuration (IaaS)
- collect (ids, whois, ...)
- blind spots
  - manual (switch, client templates)
  - discover (parse running conf, lldp)



- IaaS is good
- some programming skills are needed
  - but it's worth the effort
- step by step
  - no need to automate everything
- describe test cases along with new features
  - future you will thank you



- Questions?
- Contact me
  - Alexander Zubkov
  - [green@qrator.net](mailto:green@qrator.net)