Operationalizing SONiC in Modern Data Centers: Deployment, Telemetry, and Automation

Amer Fakhar

Reda Laichi

Saju Salahudeen

© 2026 Nokia

Workshop Objectives

- SONiC Architecture and Deployment
  - Key architecture components, lab environment
- Operating SONiC Data Center Fabric
  - Configuration, Validation and Troubleshooting
- SONiC Visibility and Telemetry
  - gNMI and Dashboards
- Automating SONiC Operations
  - Configuration, Maintenance

© 2026 Nokia

Agenda

- Introduction to SONiC and Architecture
- Introduction to Containerlab as lab emulation tool
- Hands-on section and operation use cases
  - Building SONiC image for Containerlab
  - Configuring SONiC EVPN, validation and troubleshooting
  - SONiC Telemetry using gNMI and Dashboards
  - SONiC Automation
  - Maintenance Operation
- Summary

© 2026 Nokia

SONiC Introductionn

What is SONiC ?

What is SONiC ?

- Software for Open Networking in the Cloud
- Open-source network operating system, based on Linux
- Built by Microsoft for Azure's data centers
- Now maintained by the Linux Foundation (contributed to the Open Compute Project first)
- Runs on switches from many hardware vendors: the network software is separate from the hardware

© 2026 Nokia

SONiC: Accelerating Growth in the AI Era

Analyst projections and industry validation highlight SONiC's strong trajectory

AI-DRIVEN DEMAND IS FUELING SONiC ADOPTION

$5B+
Projected SONiC-based switching revenue **by 2026**, according to 650 Group

25%+
Enterprise adoption growth year-over-year **through 2027**

Expanding beyond data center into **edge and AI inference** workloads

OPEN SOURCE MOMENTUM STRENGTHENS SONiC'S POSITION

92%
of organizations view open source networking **as critical**

40-50%
reduction in **total cost of ownership** with SONiC

Recognized as the **most widely deployed open source network OS** in hyperscale environments

NEW SONiC Deployment Story

SONiC Powering Real-World Networks at Scale

Trusted by global leaders across industries to build open, scalable, and future-ready infrastructure

DELL Technologies

SONiC Powering Dell IT Global Enterprise Network

23	10	~1000,000	3,400+
Data Centers	Manufacturing Sites	Users Worldwide	SONiC Switches Deployed

Key Outcomes

- **Unified and standardized network OS** replacing multiple proprietary systems
- **Leaf-spine architecture** with EVPN/VLAN and high-speed fabrics (up to 800G)
- **AI-ready infrastructure** and containerized microservices architecture
- **Zero major incidents** during phased global rollout
- **Reduced OPEX and operational complexity** through automation and tandardization

SONiC Adopted Across Diverse Industries and Use Cases

Sakura Internet

TOP500 GPU Supercomputer, 800-GPU Cloud Infrastructure

Rakuten Mobile

Multi-data center PoV with >50% cost savings and vendor flexibility

Orange

Telco Use Case with Open Switch Deployment

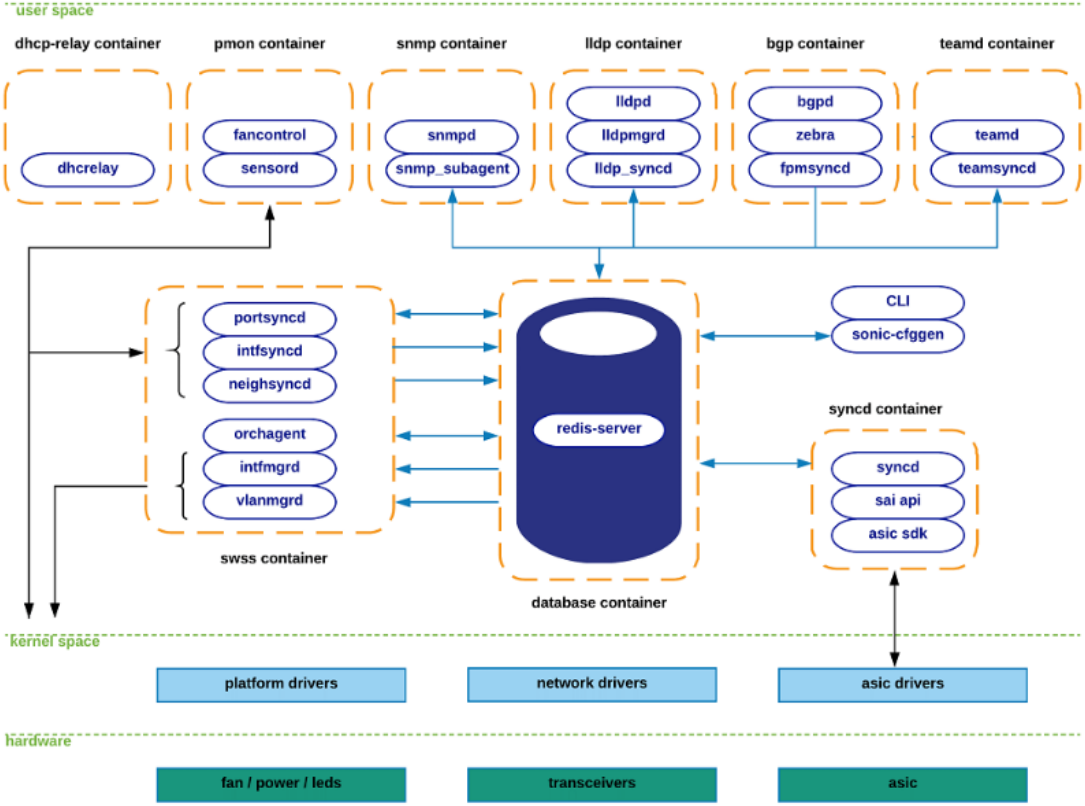
Mitsui Knowledge Industry (MKI)

Tokyo-1 AI Supercomputing Infrastructure

... and many more

A growing ecosystem of innovators building the future with SONiC

SONiC Architecture

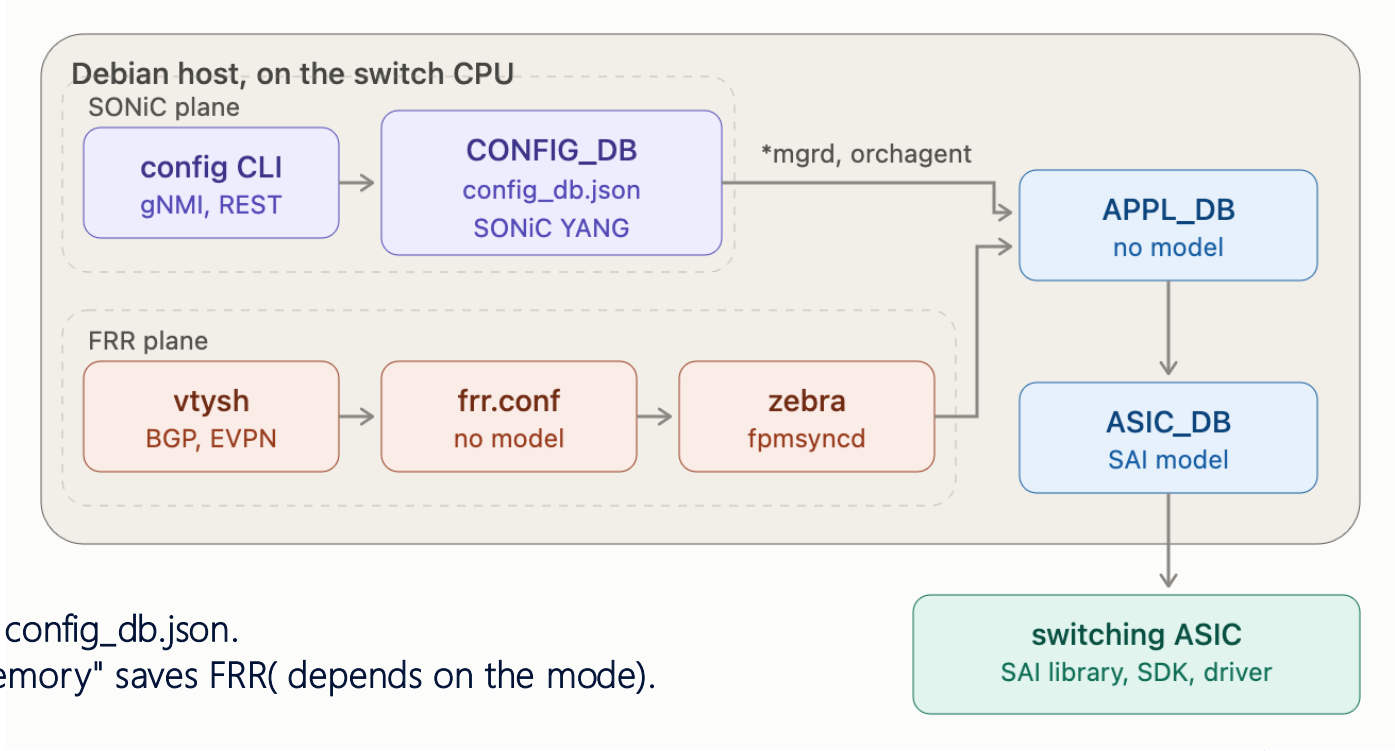


Each function runs as a container

Redis – Database
SAI – Switch Abstracted Interface

© 2026 Nokia
Reda Laichi

SONiC Architecture



Two separate saves:
- config save writes config_db.json.
- vtysh -c "write memory" saves FRR(depends on the mode).

© 2026 Nokia

Using Containerlab



CONTAINERlab

© 2026 Nokia

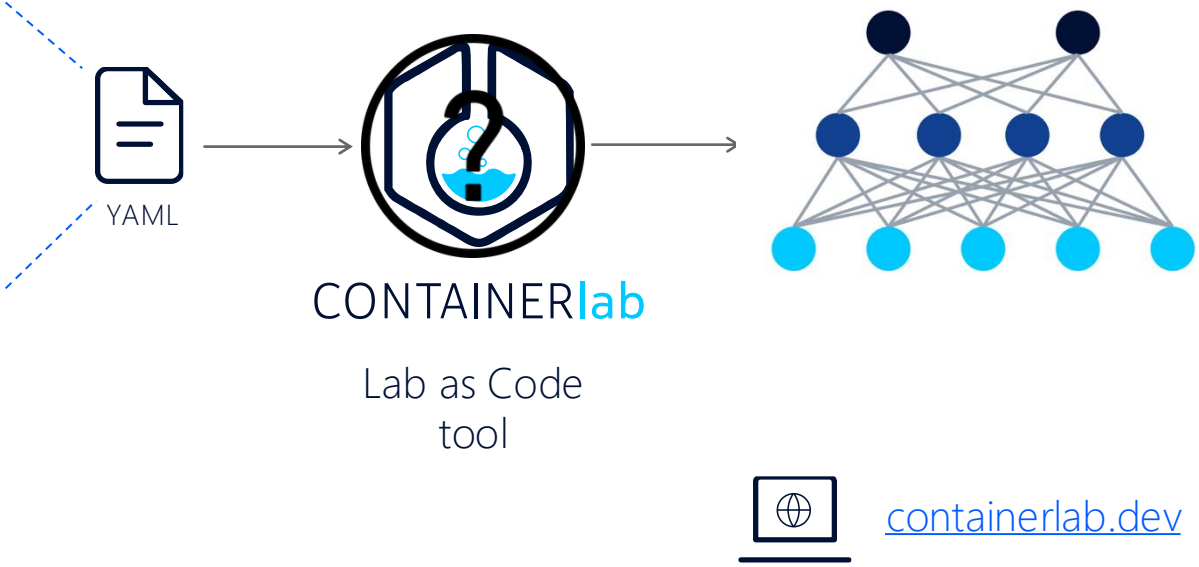
Lab as Code

Declarative labs as code with containers

```
name: my-lab

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

  links:
   - [redacted]: [redacted]
```



containerlab.dev

© 2026 Nokia

Introduction to Containerlab

Sample topology definition

demo1.clab.yml

```
name: demo1

topology:
  nodes:

    node1:
      kind: nokia_srlinux
      image: ghcr.io/nokia/srlinux:24.3.1

    node2:
      kind: nokia_srlinux
      image: ghcr.io/nokia/srlinux:23.10.4


  links:
    - endpoints: [“node1:e1-1", "node2:e1-1"]
```

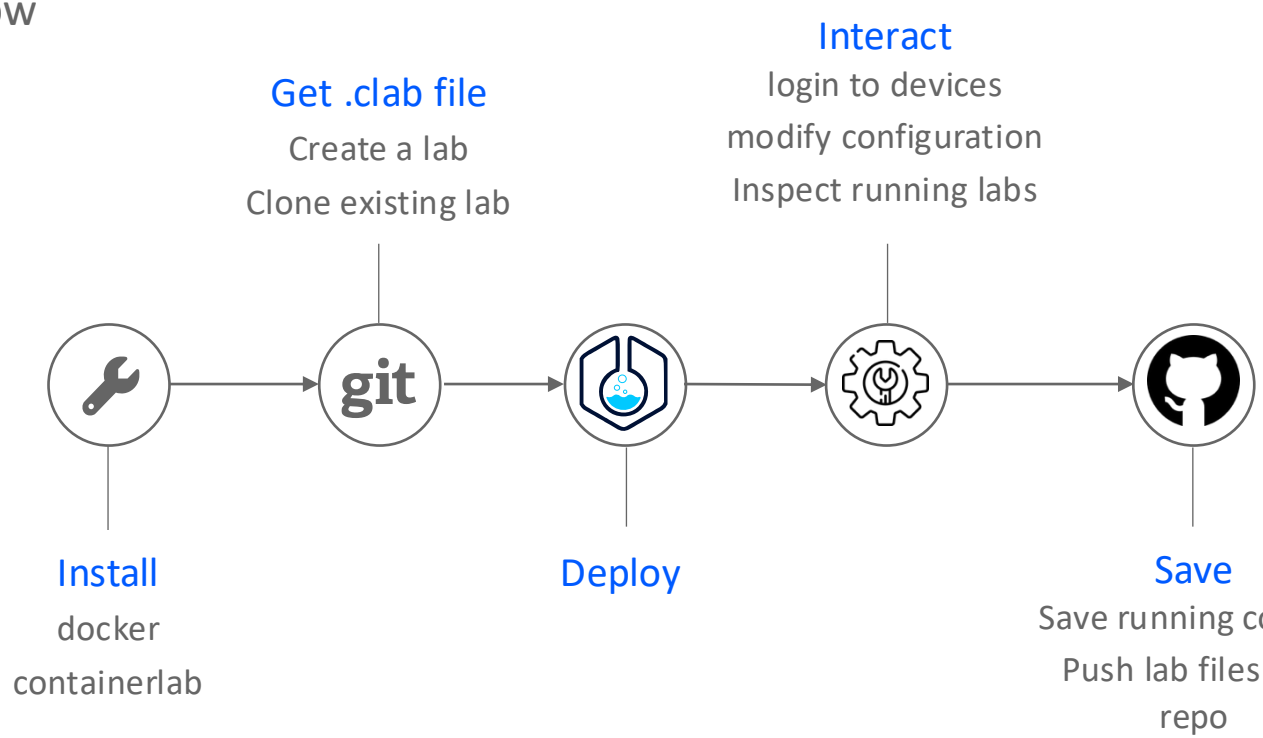
Logical view



© 2026 Nokia

Containerlab

The Workflow

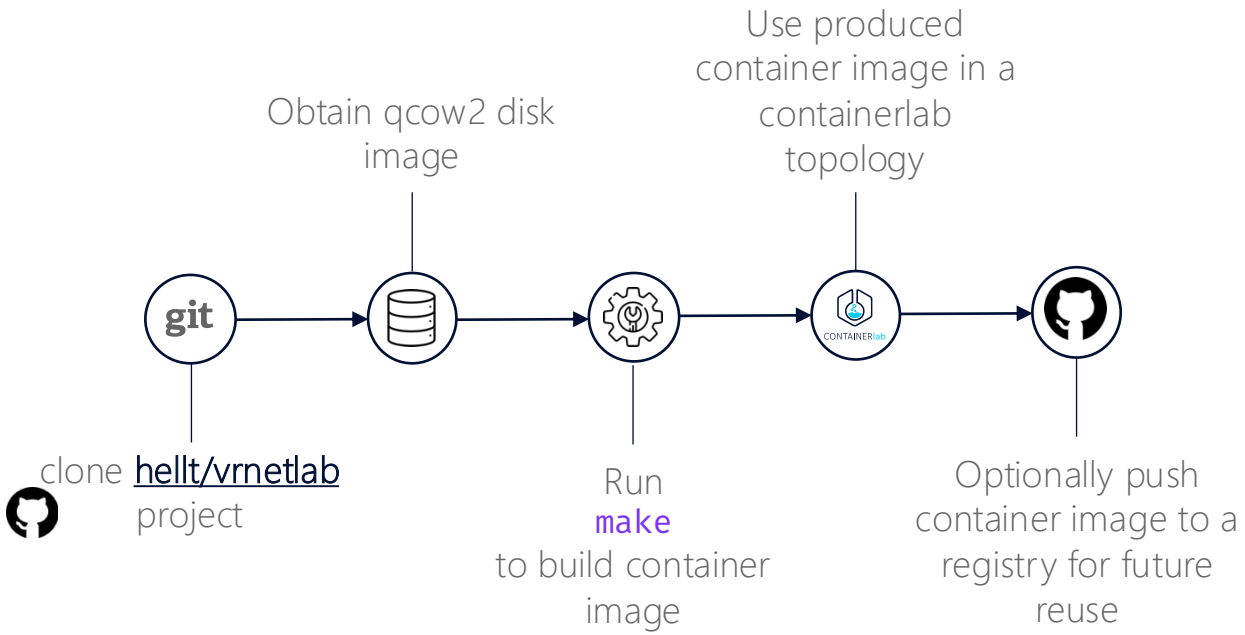


© 2026 Nokia

Reda Laichi

Bringing VM-based nodes to Containerlab

The workflow



© 2026 Nokia

Your personal hands-on experience

And how to get the most out of it



© 2026 Nokia

Workshop repository

Repo

github.com/srlinuxamericas/ac6-sonic

© 2026 Nokia

Hands-On: Create your Lab



- Login to your VM
- Install containerlab
- Build SONiC image
- Clone the Git repo – *git clone https://github.com/srlinuxamericas/ac6-sonic.git*
- Go to the directory
  - *cd ac6-grpc*
- Deploy lab
  - *sudo clab deploy -t ac6-sonic.clab.yml*

© 2026 Nokia

Building SONiC Image for Containerlab

© 2024 Nokia

Cloning `hellt/vrnetlab`



01-deploy

```
[~/]
└──> cd ~ && git clone https://github.com/hellt/vrnetlab.git &&
cd ~/vrnetlab
```


© 2024 Nokia

Building SONiC image

01-deploy

```
[~/vrnetlab]
└──> cp ~/images/sonic-vs-202605.qcow2 ~/vrnetlab/sonic/
```

```
[~/vrnetlab]
└──> cd ~/vrnetlab/sonic && make
```

Estimated time to complete: ~3-4mins

© 2024 Nokia

Verifying SONiC image in Docker repo

01-deploy

```
[~/vrnetlab]
└──> docker images
REPOSITORY                TAG          IMAGE ID       CREATED          SIZE
vrnetlab/sonic_sonic-vs   202605       d1a9e9c8a9c0   2 minutes ago    8.73GB
```


© 2024 Nokia

Deploying the lab

01-deploy

```
[~/]
└──> cd ~/ac6-sonic && sudo clab dep -c
```


© 2024 Nokia

Monitoring the boot process

01-deploy

```
[~/ac6-sonic]
└──> containerlab logs --name ac6-sonic -n leaf11 --tail 100 --follow
--timestamps
```

Boot log

- The first thing to check when something “does not work”
- Gives insight in what is provisioned by Containerlab in terms of the base configuration

© 2024 Nokia

Connecting to the nodes

01-deploy

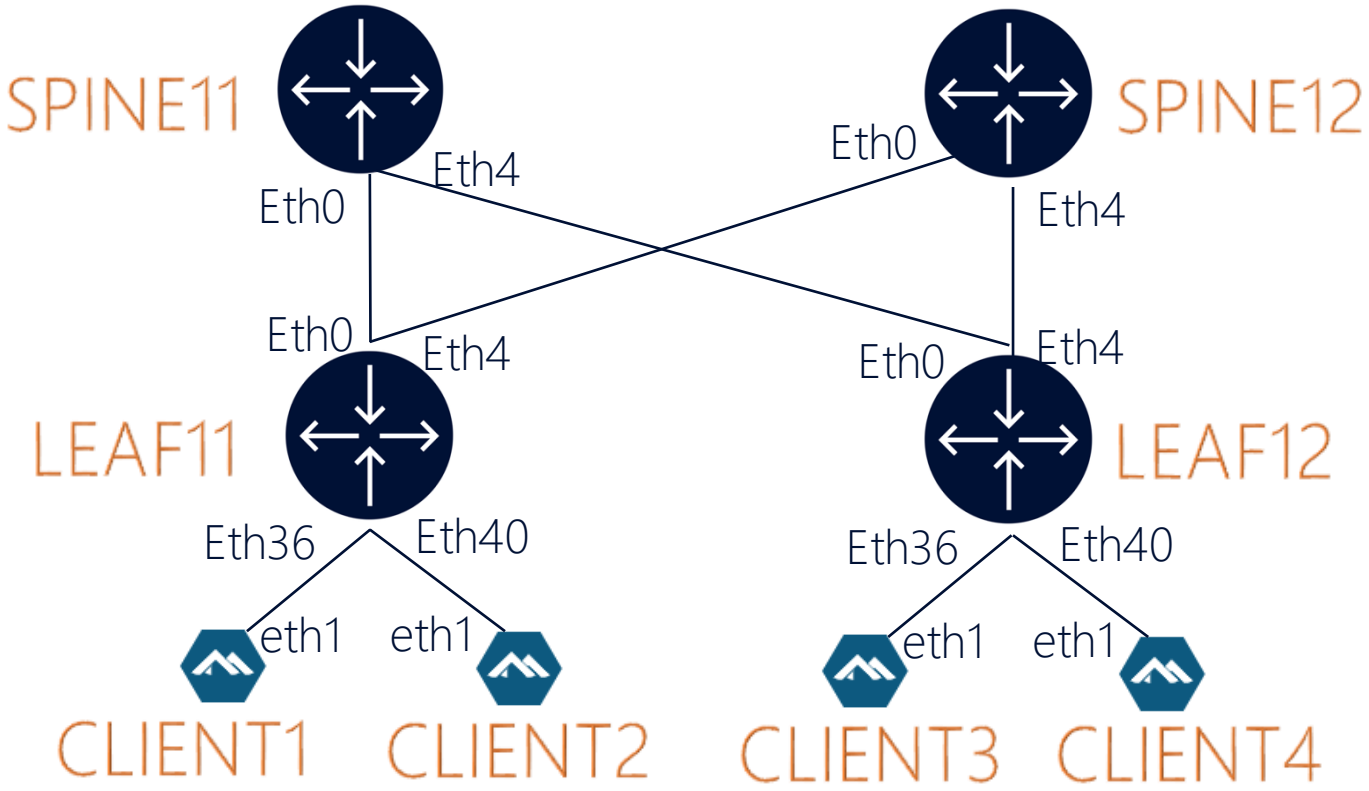
```
+---+----------------+---------------+-----------------+----------------------+
| # |      Name      |     Kind      |  IPv4 Address   |     IPv6 Address     |
+---+----------------+---------------+-----------------+----------------------+
| 1 | leaf11         | sonic-vs      | 172.20.20.3/24  | 2001:172:20:20::3/64 |
| 2 | spine12        | sonic-vs      | 172.20.20.2/24  | 2001:172:20:20::2/64 |
+---+----------------+---------------+-----------------+----------------------+
```





© 2024

Lab Topology



© 2026 Nokia

Reda Laichi

Building the Data center

© 2026 Nokia

Lab Use cases

Participants will be assigned a dedicated lab setup to perform hands-on:

- Step 1: Configure BGP for Underlay
- Step 2: Configure BGP for Overlay
- Step 3: Configure & Deploy Layer 2 EVPN-VxLAN
- Step 4: Configure & Deploy Layer 3 EVPN-VxLAN

SONiC FRR Configuration Modes

Mode	Who generates the FRR config	Source of truth	Configure with	Survives restart
separated (default)	bgpcfgd, from templates	CONFIG_DB (basic BGP only)	config CLI, gNMI, REST	Only what the templates and CONFIG_DB cover
Unified – (workshop)	frrcfgd, from CONFIG_DB	CONFIG_DB	config CLI, gNMI, REST	Yes, once it's in CONFIG_DB*
split	bgpcfgd for BGP, a file for the rest	BGP: CONFIG_DB. Others: frr.conf	BGP: config CLI, gNMI, REST. Others: vtysh	BGP: yes. Others: only if saved to frr.conf
split-unified	frrcfgd for BGP, a file for the rest	BGP: CONFIG_DB. Others: frr.conf	BGP: config CLI, gNMI, REST. Others: vtysh	BGP: yes. Others: only if saved to frr.conf

```
"DEVICE_METADATA": {
    "localhost": {
        "docker_routing_config_mode": "unified"
    }
}
```

sonic-db-cli CONFIG_DB hset "DEVICE_METADATA|localhost" docker_routing_config_mode **unified**

© 2026 Nokia

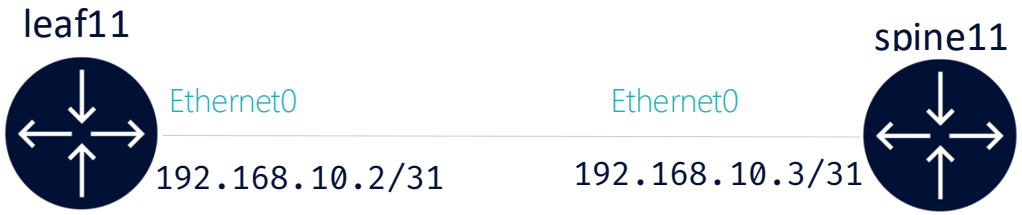
SONiC CLI

- SONiC CLI:
  - Used for Interfaces, VLANs, VXLAN and VRF to VXLAN mappings
  - Saved in config_db.json
  - CLI prompt: *admin@leaf1:~$*
- FRR (Free Range Routing) CLI:
  - Used for BGP, EVPN configuration
  - Saved in /etc/frr/frr.conf
  - CLI prompt: *leaf1#*

© 2026 Nokia

Configure basic L3 connectivity

02-sonic-bgp-evpn




```
sudo config interface ip add Ethernet0 192.168.10.2/31
```

```
sudo config interface ip add Ethernet0 192.168.10.3/31
```


© 2024

Interface IP configuration

02-sonic-bgp-evpn

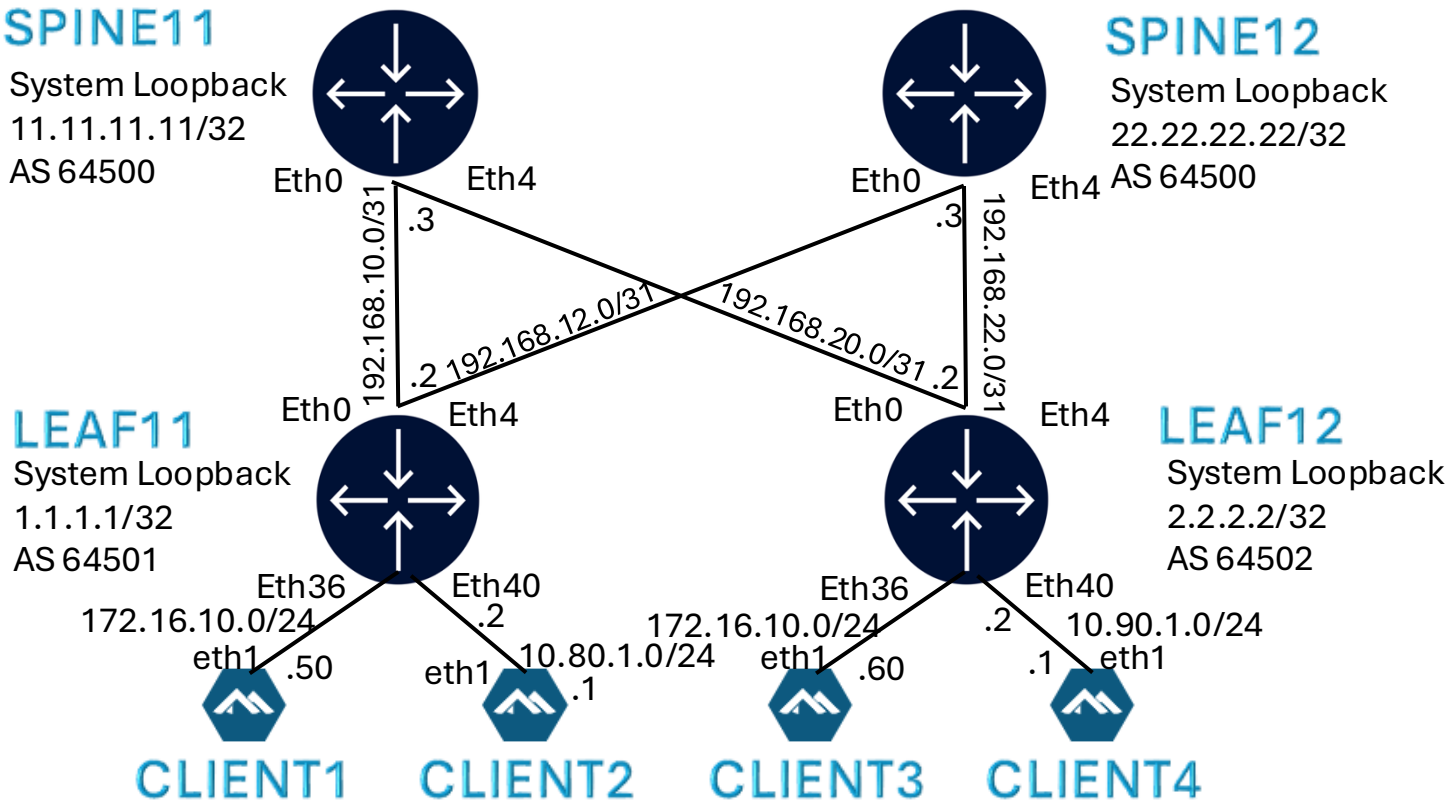
```
[~/ac6-sonic/02-sonic-bgp-evpn]
└──> ./deploy_fabric.sh
```

Script Actions:

- Configure 'Unified' mode on all 4 switches
- Configure interface IP address on all 4 switches
- Load custom yang definition for BGP

© 2024 Nokia

Lab – IPv4 Information



Reda Laichi

CLI Introduction



02-sonic-bgp-evpn

SONiC CLI

Default CLI: Yes

Prompt: admin@leaf1:~$

File: config_db.json

Save: sudo config save -y

FRR CLI

Command to enter : vtysh

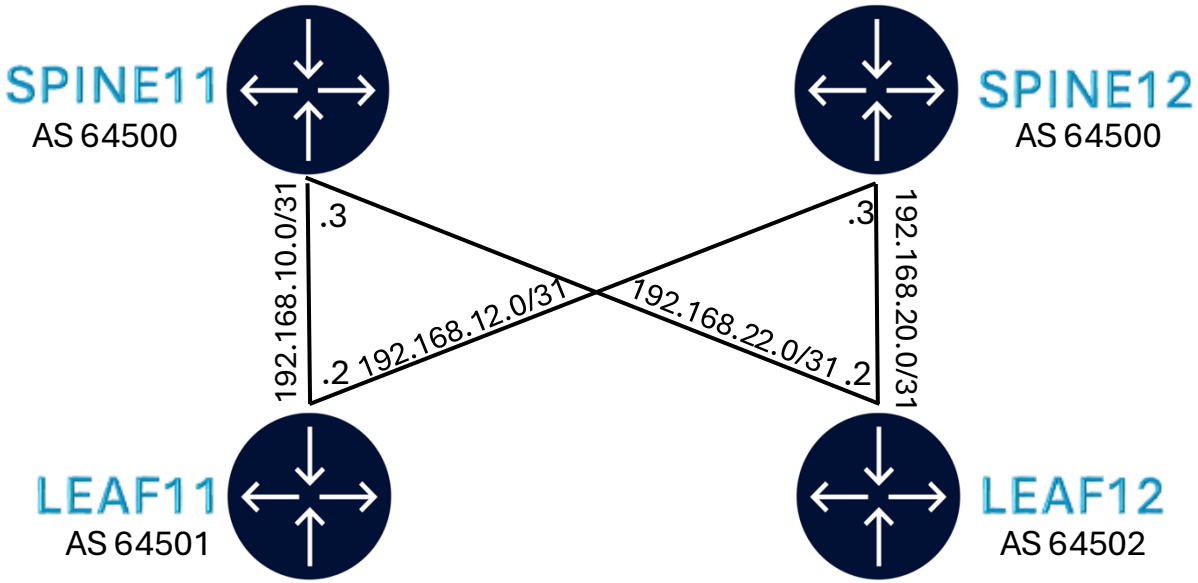
Prompt: leaf11#

File: /etc/frr/frr.conf

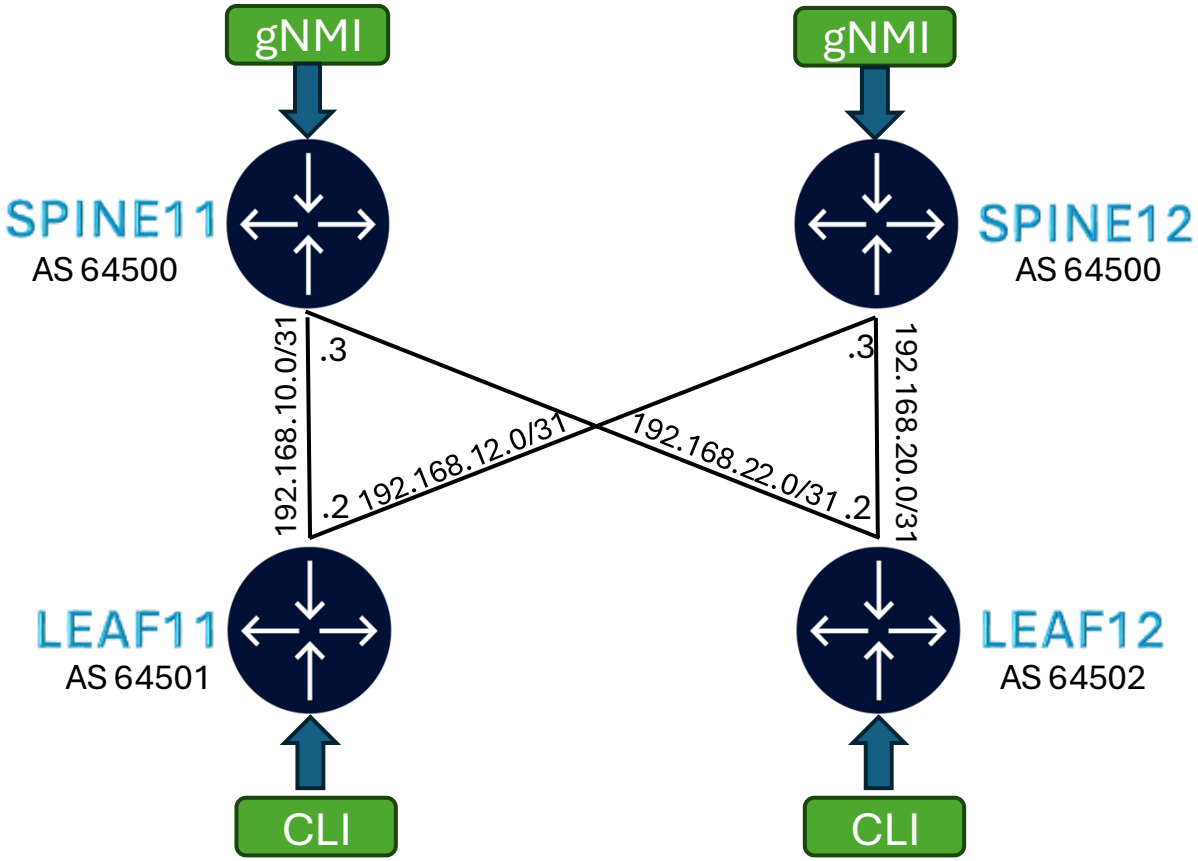
Save: write memory

© 2024 Nokia

Lab – BGP Underlay



BGP Underlay – Configuration Overview



Reda Laichi

BGP Underlay – Configuration Steps



02-sonic-bgp-evpn

- Configure BGP instance
- Configure Underlay with interface IPs as BGP Neighbors and IPv4 Address Family
  - Leaf11 and Leaf12 will be configured using FRR CLI (Enter using '**vtysh**')
  - Spine11 and Spine12 will be configured using gNMI (Run from VM)
- Verify BGP Neighbors are UP

© 2024

BGP Underlay – CLI Configuration Example



02-sonic-bgp-evpn

Underlay configuration on leaf11

```
router bgp 64501
 bgp router-id 1.1.1.1
 no bgp ebgp-requires-policy
 neighbor 192.168.10.3 remote-as 64500
 neighbor 192.168.12.3 remote-as 64500
 !
 address-family ipv4 unicast
  network 1.1.1.1/32
 exit-address-family
 !
exit
```


© 2024

BGP Underlay – gNMI Configuration Example

02-sonic-bgp-evpn

Underlay configuration on spine11

```
gnmic -a spine11:8080 --insecure \
  -e json_ietf set \
  --update-path 'sonic-db:/CONFIG_DB/localhost/BGP_GLOBALS/default' \
  --update-value '{"vrf_name":"default","local_asn":"64500","ebgp_requires_policy":
"false","default_ipv4_unicast": "true","load_balance_mp_relax": "true","router_id":
"11.11.11.11"}'

  --update-path 'sonic-db:/CONFIG_DB/localhost/BGP_GLOBALS_AF/default|ipv4_unicast' \
  --update-value '{"max_ebgp_paths": "64"}' \
  --update-path 'sonic-db:/CONFIG_DB/localhost/BGP_GLOBALS_AF_NETWORK' \
  --update-value '{"default|ipv4_unicast|11.11.11.11/32": {}}'

  --update-path 'sonic-db:/CONFIG_DB/localhost/BGP_NEIGHBOR/default|192.168.10.2' \
  --update-value '{"asn": "64501"}' \
  --update-path 'sonic-
db:/CONFIG_DB/localhost/BGP_NEIGHBOR_AF/default|192.168.10.2|ipv4_unicast' \
  --update-value '{"admin_status": "up"}' \
```

BGP Instance

Address Family

Neighbor

© 2024

BGP Underlay – Verification



02-sonic-bgp-evpn

Underlay BGP status on spine11

```
spine11# show bgp summary

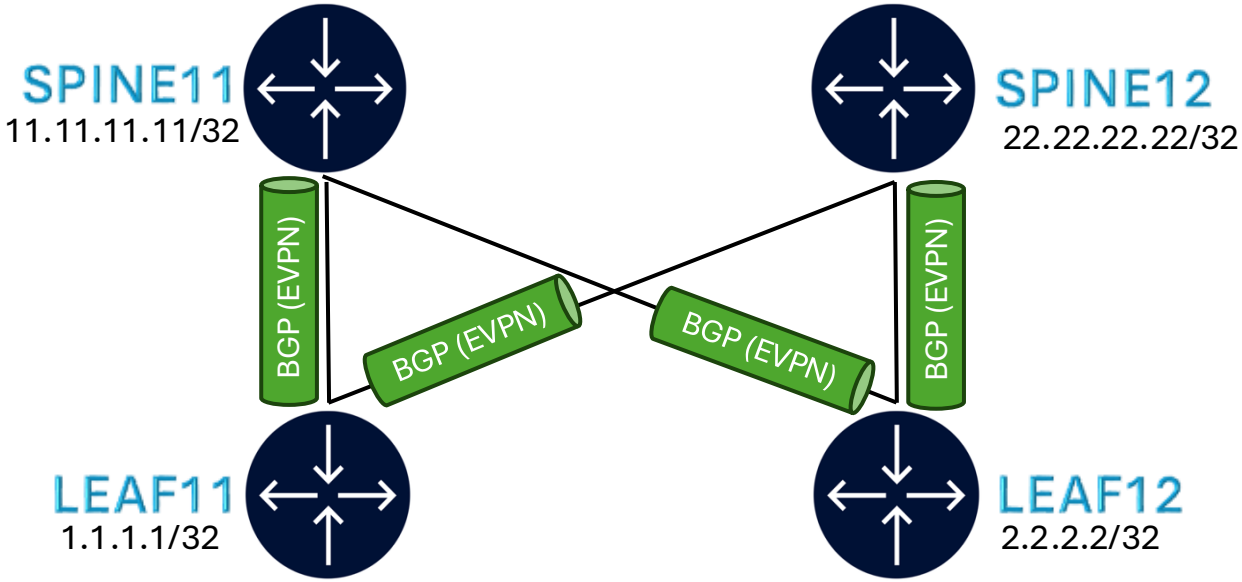
IPv4 Unicast Summary:
BGP router identifier 11.11.11.11, local AS number 64500 VRF default vrf-id 0
BGP table version 8
RIB entries 17, using 2176 bytes of memory
Peers 2, using 47 KiB of memory

Neighbor        V         AS   MsgRcvd   MsgSent   TblVer  InQ OutQ  Up/Down State/PfxRcd   PfxSnt Desc
192.168.10.2    4      64501        13        11        8    0    0 00:00:12            1        8 N/A
192.168.20.2    4      64502        13        11        8    0    0 00:00:12            1        8 N/A

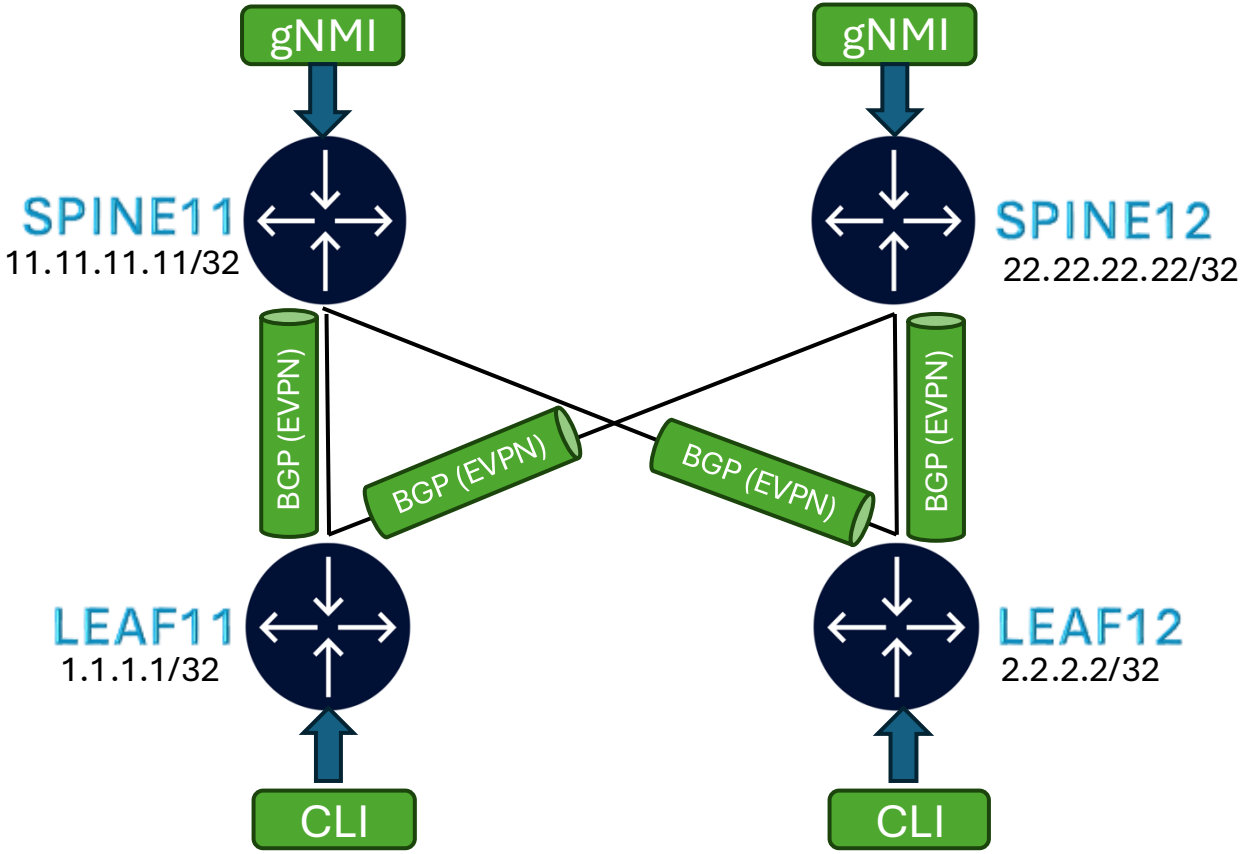
Total number of neighbors 2
```


© 2024

Lab – BGP Overlay



Lab – BGP Overlay – Configuration Overview



BGP Overlay – Configuration Steps



02-sonic-bgp-evpn

- Configure BGP instance
- Configure Overlay with loopback IPs as BGP Neighbors and L2VPN Family
  - Leaf11 and Leaf12 will be configured using FRR CLI (Enter using '**vtysh**')
  - Spine11 and Spine12 will be configured using gNMI (Run from VM)
- Verify BGP Neighbors are UP

© 2024

BGP Overlay – CLI Configuration Example



02-sonic-bgp-evpn

```
router bgp 64501
 neighbor 11.11.11.11 remote-as 64500
 neighbor 11.11.11.11 update-source Loopback0
 neighbor 11.11.11.11 ebgp-multihop 2
 neighbor 22.22.22.22 remote-as 64500
 neighbor 22.22.22.22 update-source Loopback0
 neighbor 22.22.22.22 ebgp-multihop 2
 !
 address-family l2vpn evpn
  neighbor 11.11.11.11 activate
  neighbor 22.22.22.22 activate
 exit-address-family
address-family ipv4 unicast
  no neighbor 11.11.11.11 activate
  no neighbor 22.22.22.22 activate
 !
exit
!
```

Overlay configuration on leaf11

© 2024

BGP Overlay – gNMI Configuration Example

02-sonic-bgp-evpn

Overlay configuration on spine11

```
gnmic -a spine11:8080 --insecure \
  -e json_ietf set \
  --update-path 'sonic-db:/CONFIG_DB/localhost/BGP_NEIGHBOR/default|1.1.1.1' \
  --update-value '{"asn": "64501","ebgp_multihop_ttl": "2","local_addr": "11.11.11.11"}' \
  --update-path 'sonic-db:/CONFIG_DB/localhost/BGP_NEIGHBOR/default|2.2.2.2' \
  --update-value '{"asn": "64502","ebgp_multihop_ttl": "2","local_addr": "11.11.11.11"}'
```

Neighbor

Address Family

```
  --update-path 'sonic-db:/CONFIG_DB/localhost/BGP_NEIGHBOR_AF/default|1.1.1.1|ipv4_unicast' \
  --update-value '{"admin_status": "down"}' \
  --update-path 'sonic-db:/CONFIG_DB/localhost/BGP_NEIGHBOR_AF/default|1.1.1.1|l2vpn_evpn' \
  --update-value '{"admin_status": "up"}' \
  --update-path 'sonic-db:/CONFIG_DB/localhost/BGP_NEIGHBOR_AF/default|2.2.2.2|ipv4_unicast' \
  --update-value '{"admin_status": "down"}' \
  --update-path 'sonic-db:/CONFIG_DB/localhost/BGP_NEIGHBOR_AF/default|2.2.2.2|l2vpn_evpn' \
  --update-value '{"admin_status": "up"}'
```


© 2024

BGP Overlay – Verification

02-sonic-bgp-evpn

Overlay BGP status on leaf11

```
leaf11(config)# do show bgp l2vpn evpn summary

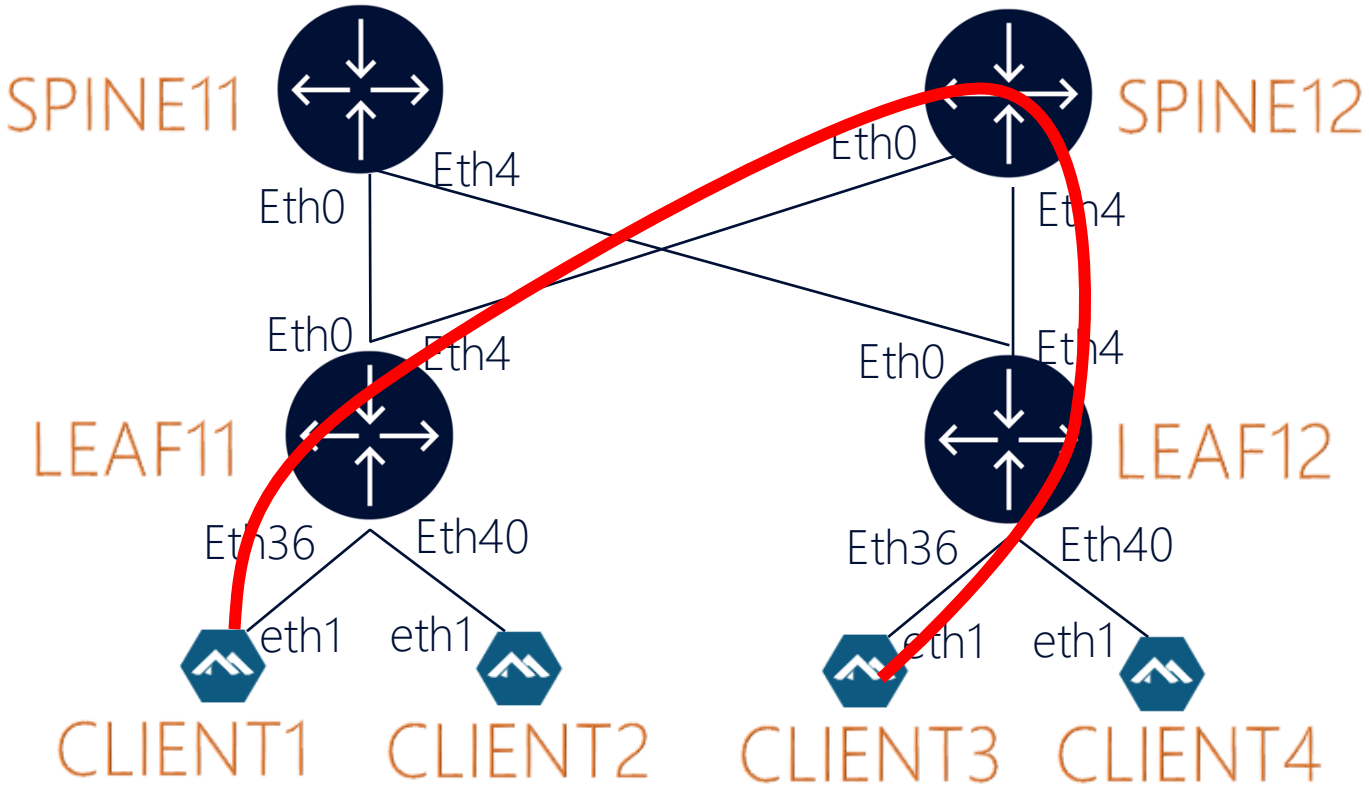
BGP router identifier 1.1.1.1, local AS number 64501 VRF default vrf-id 0
BGP table version 0
RIB entries 0, using 0 bytes of memory
Peers 2, using 47 KiB of memory

Neighbor        V         AS   MsgRcvd   MsgSent   TblVer  InQ OutQ  Up/Down State/PfxRcd   PfxSnt Desc
11.11.11.11     4      64500         3         4        0    0    0 00:00:27            0        0 N/A
22.22.22.22     4      64500         3         4        0    0    0 00:00:24            0        0 N/A

Total number of neighbors 2
```


© 2024

Layer 2 EVPN-VXLAN - Objective



© 2026 Nokia

Reda Laichi

Layer 2 EVPN-VXLAN – Configuration Steps



02-sonic-bgp-evpn

- Configuration only on leafs
- Create VXLAN tunnels (in **SONiC CLI**)
- Configure VLAN to VXLAN mapping (in **SONiC CLI**)
- Configure L2 EVPN (in **FRR CLI**)

© 2026 Nokia

Reda Laichi

Layer 2 EVPN – Configuration Example

02-sonic-bgp-evpn

Layer 2 EVPN configuration on leaf11

```
sudo config vxlan add vtep 1.1.1.1
sudo config vxlan evpn_nvo add nvo vtep

sudo config vlan add 110
sudo config interface startup Ethernet36
sudo config vlan member add -u 110 Ethernet36
sudo config vxlan map add vtep 110 10110
sudo config save -y


router bgp 64501
 address-family l2vpn evpn
  advertise-all-vni
```

VxLAN Tunnel

VLAN to VxLAN mapping

L2VPN (FRR CLI)

© 2026 Nokia

Reda Laichi

Layer 2 EVPN – Verification

02-sonic-bgp-evpn

Ping from Client3 to Client1

```
client3# ping -c 3 172.16.10.50

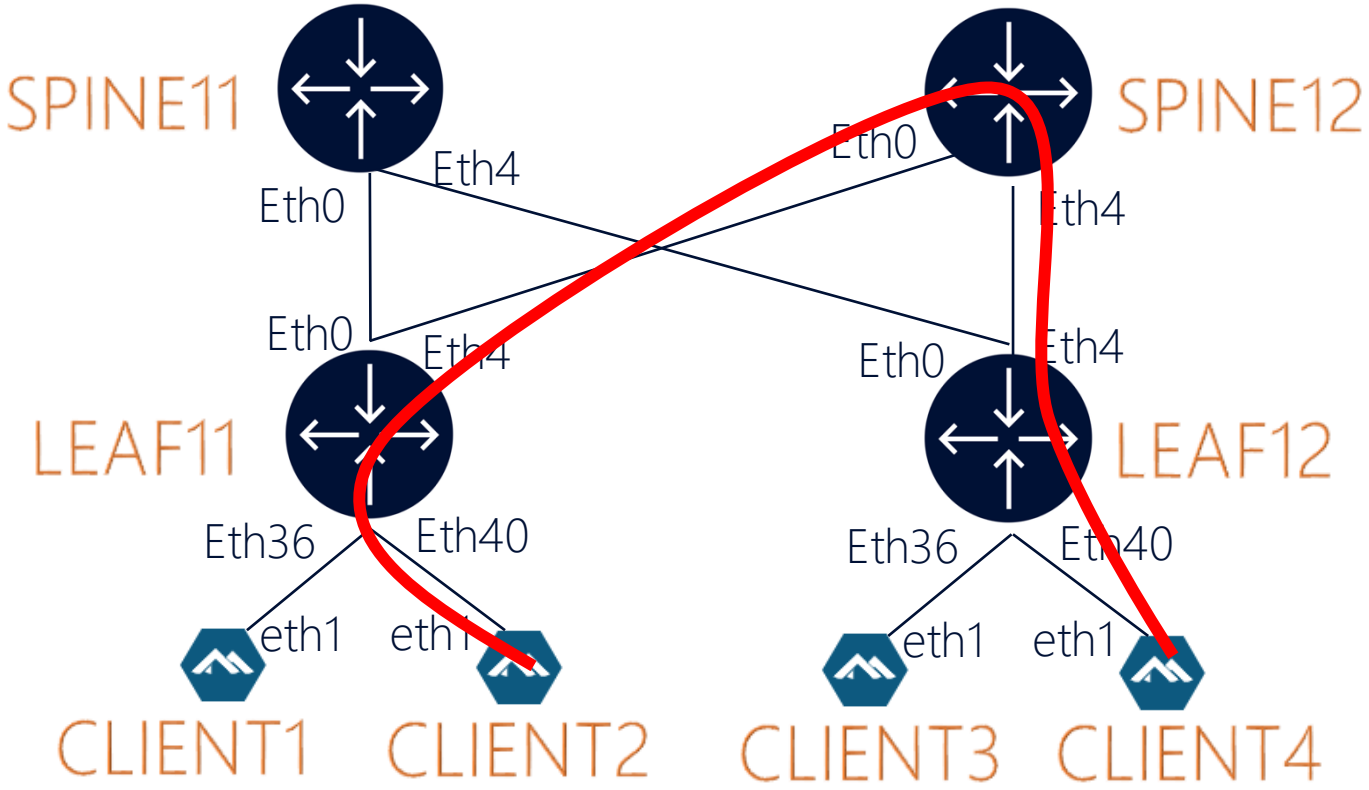
PING 172.16.10.50 (172.16.10.50) 56(84) bytes of data.
64 bytes from 172.16.10.50: icmp_seq=1 ttl=64 time=12.7 ms
64 bytes from 172.16.10.50: icmp_seq=2 ttl=64 time=7.13 ms
64 bytes from 172.16.10.50: icmp_seq=3 ttl=64 time=5.20 ms

--- 172.16.10.50 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 5.198/8.357/12.741/3.198 ms
```


© 2026 Nokia

Reda Laichi

Layer 3 EVPN-VXLAN - Objective



© 2026 Nokia

Reda Laichi

Layer 3 EVPN-VXLAN – Configuration Steps



02-sonic-bgp-evpn

- Configuration only on leafs
- Clients 2 & 4 are configured with IP during startup
- Configure MAC-VRF (VLAN) on leaf switches
- Configure VRF (in **FRR CLI**)
- Configure client facing interfaces on leaf switches (in **SONiC CLI**)
- Configure VLAN to VXLAN mapping (in **SONiC CLI**)

© 2026 Nokia

Reda Laichi

Layer 3 EVPN – Configuration Example

02-sonic-bgp-evpn

Layer 3 EVPN configuration on leaf11

VLAN

```
sudo config vlan add 200
sudo config interface startup Ethernet40
sudo config vlan member add -u 200 Ethernet40
```

Interface Config

VxLAN

```
sudo config vrf add Vrf_Type5
sudo config interface vrf bind Vlan240 Vrf_Type5
sudo config interface ip add Vlan240 10.90.1.254/24
sudo config vlan add 999
sudo config interface vrf bind Vlan999 Vrf_Type5
sudo config vxlan map add vtep 240 10240
sudo config vxlan map add vtep 999 10999
sudo config vrf add_vrf_vni_map Vrf_Type5 10999
```

VRF

```
vrf Vrf_Type5
 ip router-id 1.1.1.1
 vni 10999
 ip nht resolve-via-default
 ipv6 nht resolve-via-default
exit-vrf
router bgp 64501 vrf Vrf_Type5
 no bgp ebgp-requires-policy
 address-family ipv4 unicast
  network 10.80.1.0/24
 exit-address-family

 address-family l2vpn evpn
  advertise ipv4 unicast
  rd 1.1.1.1:10999
  route-target import 64999:10999
  route-target export 64999:10999
 exit-address-family
exit
```


© 2026 Nokia

Layer 3 EVPN – Verification

02-sonic-bgp-evpn

BGP Routes on Leaf11

```
leaf11# show ip route vrf Vrf_Type5 bgp

leaf11# show ip route vrf Vrf_Type5 bgp
B>* 10.90.1.0/24 [20/0] via 2.2.2.2, Vlan999 onlink, weight 1, 00:33:36
                        via 2.2.2.2, Vlan999 onlink (dup), weight 1, 00:33:36
```

2 possible next-hops to reach client4 IP:

- via spine11
- via spine12

© 2026 Nokia

Reda Laichi

Layer 3 EVPN – Verification

02-sonic-bgp-evpn

Ping from Client2 to Client4

```
client2# ping -c 3 10.90.1.1

PING 10.90.1.1 (10.90.1.1) 56(84) bytes of data.
64 bytes from 10.90.1.1: icmp_seq=1 ttl=62 time=4.18 ms
64 bytes from 10.90.1.1: icmp_seq=2 ttl=62 time=2.28 ms
64 bytes from 10.90.1.1: icmp_seq=3 ttl=62 time=2.07 ms

--- 10.90.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 2.066/2.842/4.184/0.952 ms
```

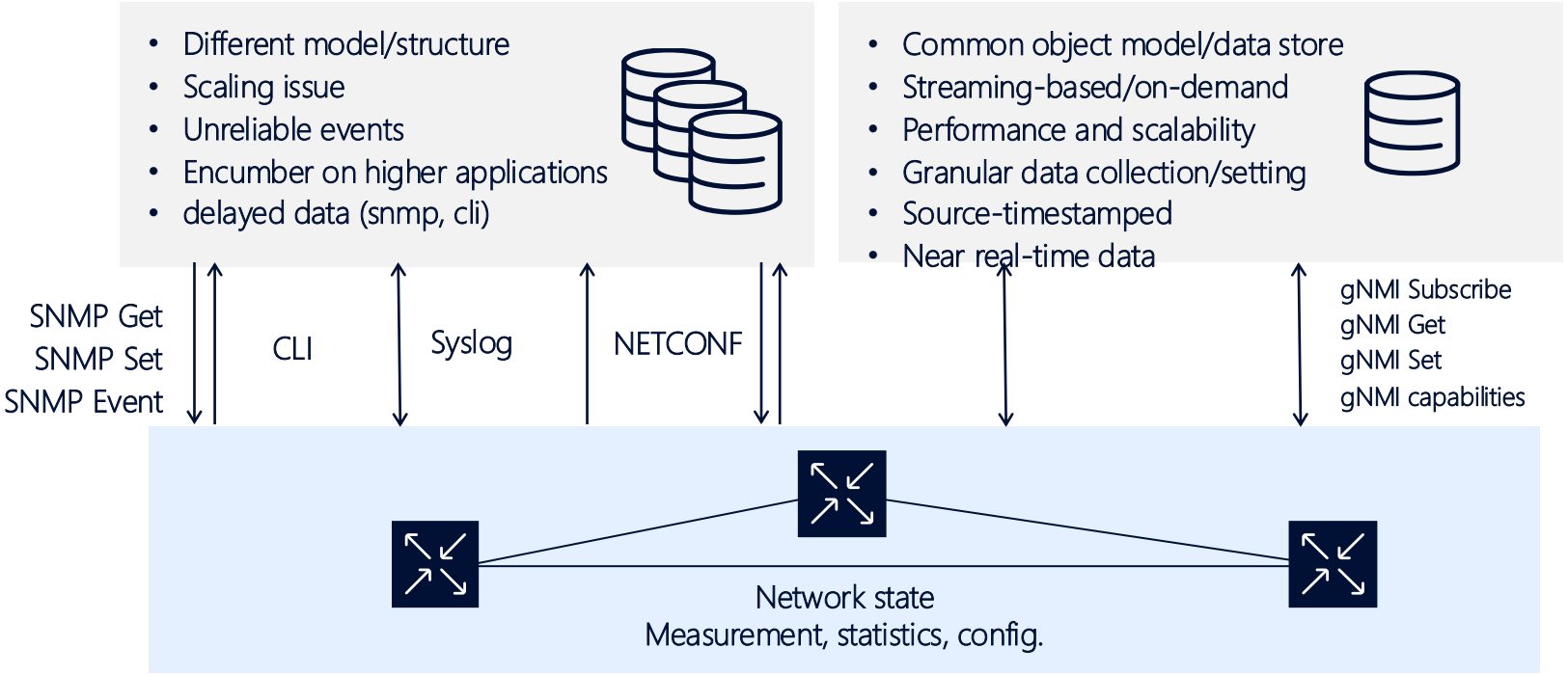

© 2026 Nokia

Reda Laichi

gNMI

© 2026 Nokia

gNMI Overview



© 2026 Nokia

gNMI Use Cases

- Configuration: Initial deployment, underlay/overlay configuration, update configuration
- State retrieval: Troubleshooting, pre/post-configuration checks, closed-loop automation
- Streaming Telemetry: Statistics collection, performance monitoring, Alarm notification

© 2026 Nokia

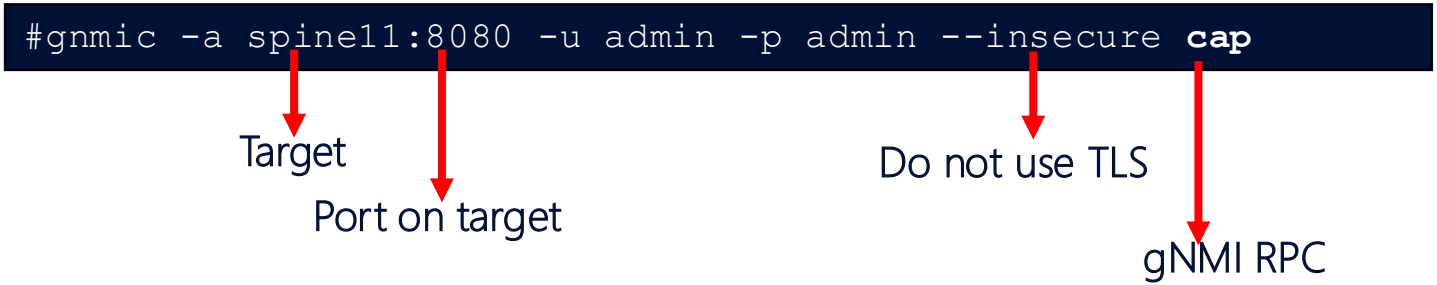
gNMI Client

- gNMIc – gNMI CLI client that provides support for gNMI services
- Developed by Nokia and donated to Openconfig
- https://gnmic.openconfig.net/

© 2026 Nokia

gNMI Capabilities

Used to verify supported gNMI models on the device



© 2026 Nokia

gNMI Get



03-sonic-gnmi

Used to get configuration or state

```
#gnmic -a spine11:8080 --insecure   -e json_ietf get   --path
'sonic-db:/CONFIG_DB/localhost/BGP_NEIGHBOR/default|1.1.1.1'
```


© 2026 Nokia

gNMI Set – Update path

Specify path and value separately

```
#gnmic -a spine11:8080 -u admin -p admin --insecure \
  -e json_ietf set \
  --update-path 'sonic-
db:/CONFIG_DB/localhost/ROUTE_MAP/TEST_GNMI_SET|10' \
  --update-value
'{"route_operation":"permit","match_protocol":"ospf","
set_asn":"65501"}' \
```

Leaf path to be updated

Leaf value to be updated

© 2026 Nokia

gNMI – Config delete



03-sonic-gnmi

Deleting configuration using gNMI Set

```
#gnmic -a spine11:8080 --insecure   -e json_ietf set   --delete 'sonic-
db:/CONFIG_DB/localhost/ROUTE_MAP/TEST_GNMI_SET|10'
```

Delete one route policy entry

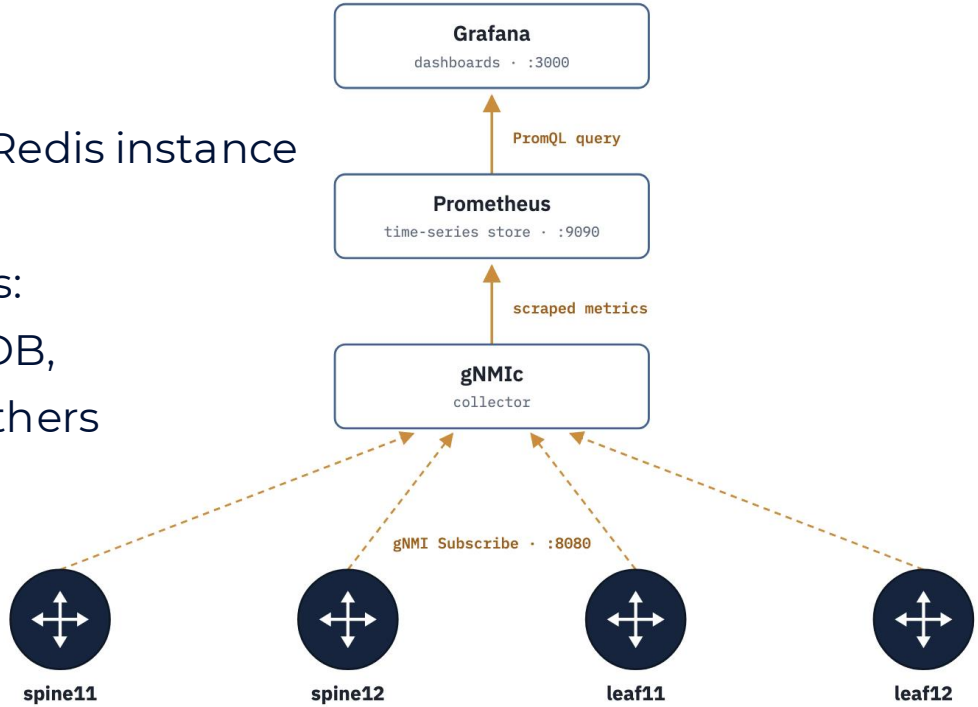
© 2026 Nokia

SONiC Telemetry

© 2026 Nokia

Telemetry – Subscription based

- SONiC keeps everything in one Redis instance
- In one process, one container
- Organized into logical databases:
- CONFIG_DB, APPL_DB, STATE_DB,
- COUNTERS_DB, ASIC_DB and others



© 2026 Nokia

gNMI Streaming Telemetry



04-sonic-telemetry

- Using gNMI Subscribe RPC
- Options:
  - Stream every X interval
  - Stream on change only

© 2026 Nokia

gNMI – Subscribe Sample



04-sonic-telemetry

Stream counters every X interval

```
#gnmic -a spine11:8080   -u admin -p admin   --insecure
subscribe   --target COUNTERS_DB   --mode stream   --stream-
mode sample   --sample-interval 5s --path
"COUNTERS/Ethernet0/SAI_PORT_STAT_IF_IN_UCAST_PKTS"
```


© 2026 Nokia

gNMI – Subscribe On-Change



04-sonic-telemetry

Stream counters on-change

```
#gnmic -a spine11:8080   -u admin -p admin   --insecure
subscribe   --target COUNTERS_DB   --mode stream   --stream-
mode on-change   --path
"COUNTERS/Ethernet0/SAI_PORT_STAT_IF_IN_UCAST_PKTS"
```


© 2026 Nokia

Grafana Dashboard

Using the browser :

<id>.workxshop.net:3000

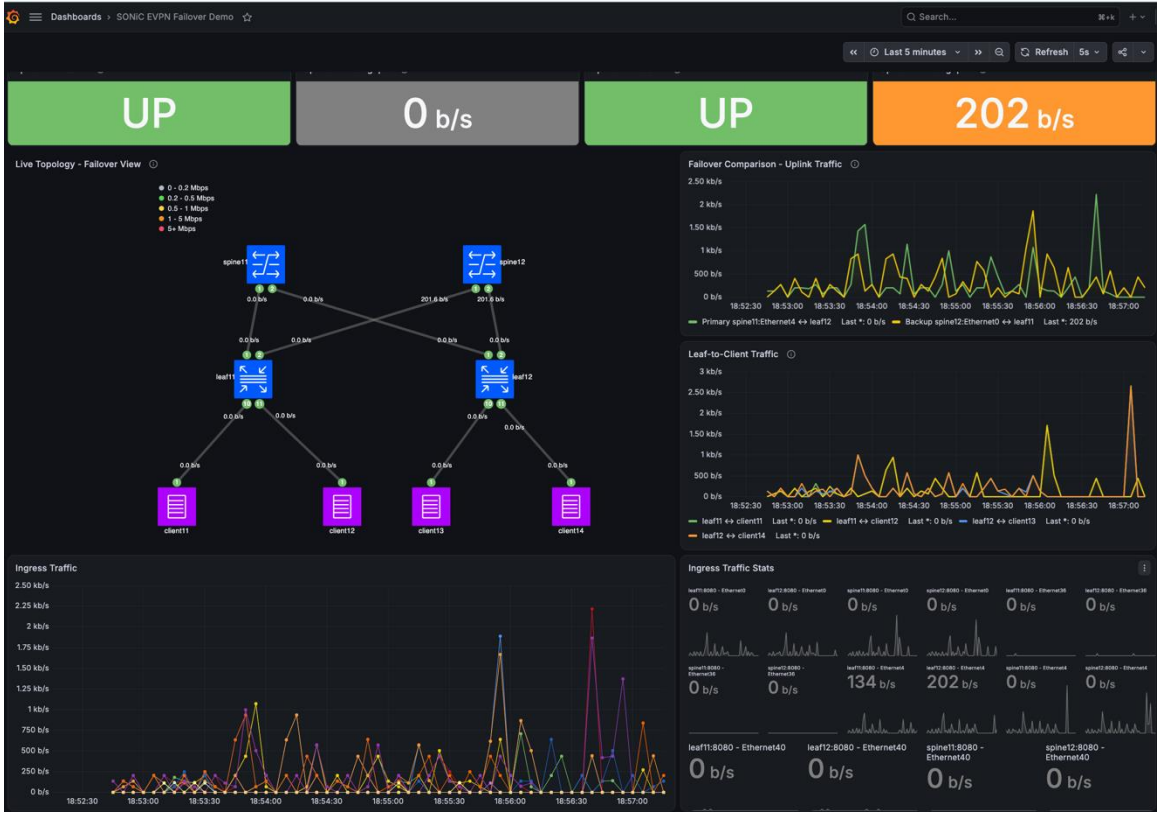
User:admin

Password: admin

(skip changing the password step)

On left menu, select Dashboard

On Dashboard page, select ‘SONiC EVPN Failover Demo’

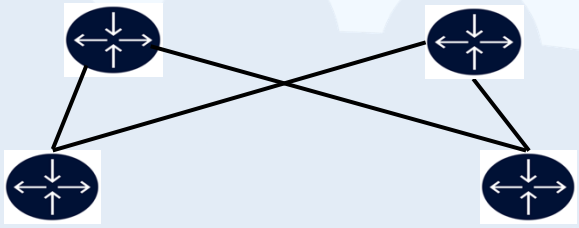


© 2026 Nokia

SONiC Automation

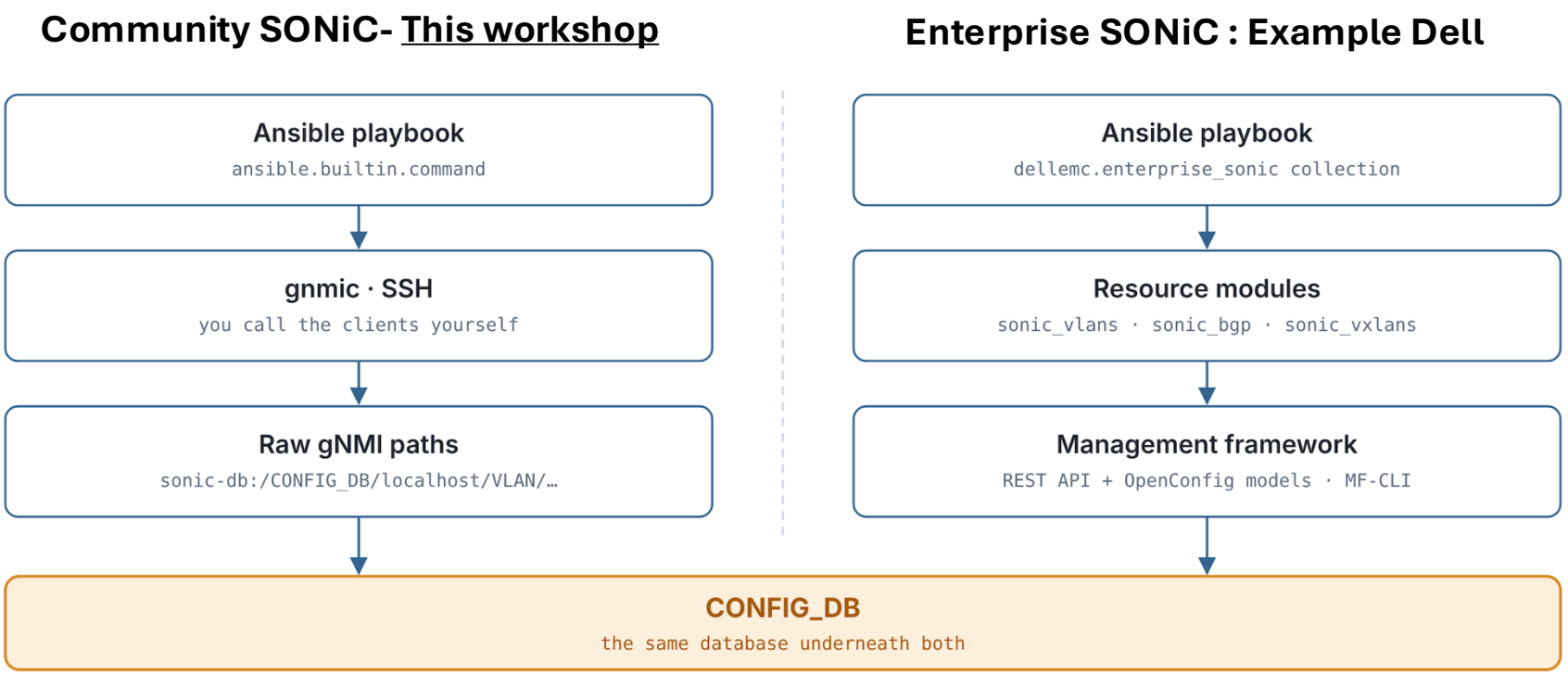
© 2026 Nokia

SONiC Network Automation

Intent what it should be	Git configs, versions	Source of truth Inventory, IPAM	CI/CD test/deploy	ITSM incidents, jira
Tools How it gets there	Ansible-Nornir config mgmt	gnmic-Python API Clients. scripts	DB · Grafana show, config, vtysh	Controller Fabric manager
Interfaces How tools reach it	gNMI Get, Set, Subscribe	REST/RESTCONF openConfig, Yang	SSH-CLI show, config, vtysh	SNMP Legacy polling
Network				

© 2026 Nokia

Automation: Community vs Enterprise SONiC



- open: no vendor tooling
- raw paths, repetitive check

- tied to a vendor and a release
- check mode built-in

© 2026 Nokia

Automation: Inventory Use Case

05-sonic-automation

- Using gNMI
- Using SSH/CLI
- Retrieve physical and Logical inventory

```
SONiC Fabric Inventory

=====================================================================
 leaf11
=====================================================================

INTERFACES
Port         Address              Admin  Oper   BGP Neighbor
-------------------------------------------------------------------
Ethernet0    192.168.10.2/31      up     up     192.168.10.3
Ethernet4    192.168.12.2/31      up     up     192.168.12.3
Loopback0    1.1.1.1/32           up     up     -
eth0         10.0.0.15/24         up     up     -

BGP SESSIONS
Neighbour          Family     AS       State         PfxRcd   PfxSnt
-------------------------------------------------------------------
192.168.10.3       ipv4       64500    Established   2        4
192.168.12.3       ipv4       64500    Established   2        4
11.11.11.11        evpn       64500    Established   3        11
22.22.22.22        evpn       64500    Established   3        11

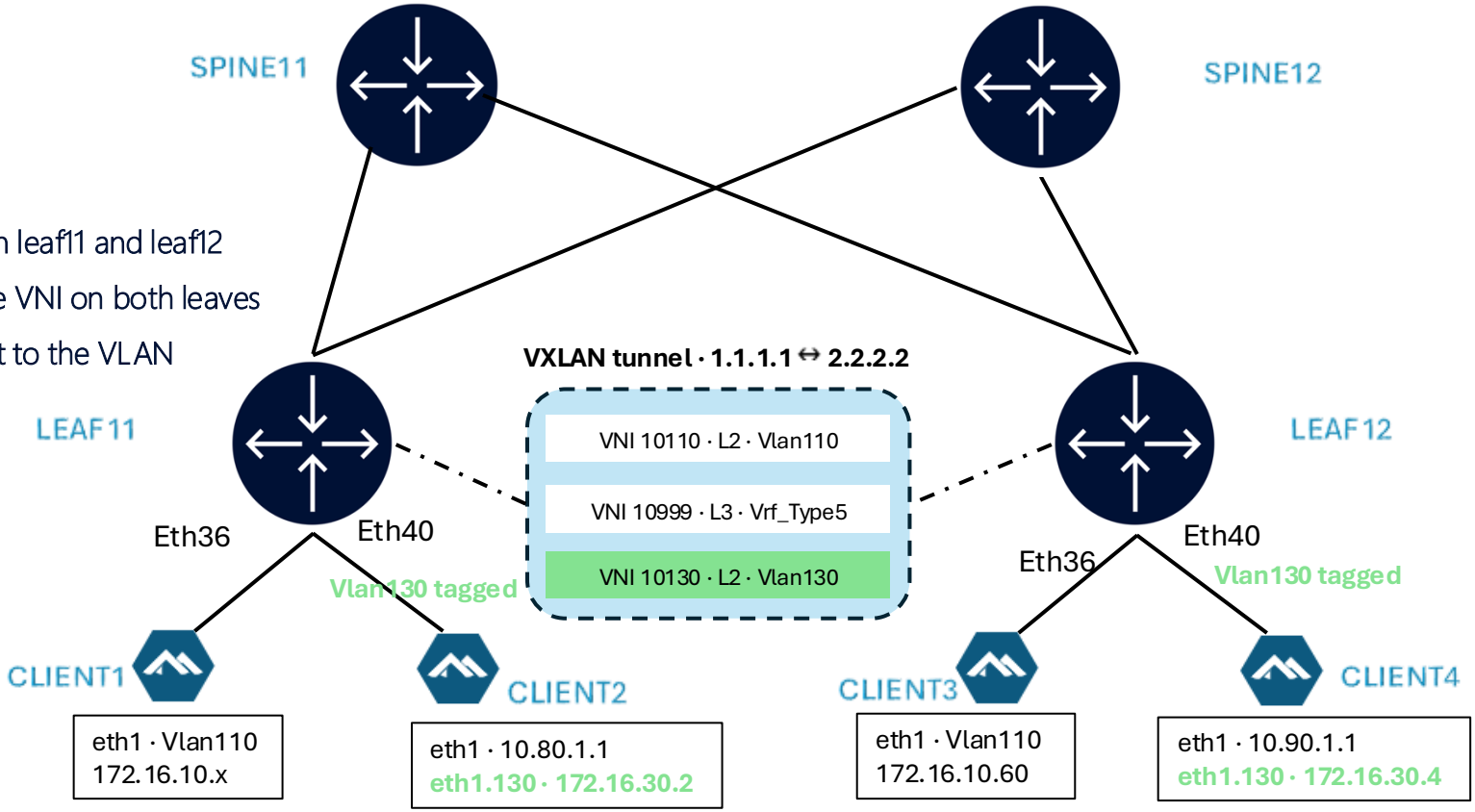
VLANS
Vlan         ID       VNI
-----------------------------
Vlan110      110      10110
Vlan200      200      10200
Vlan999      999      10999
```


© 2026 Nokia

Reda Laichi

Automation: Multi-Tenant Use Case

- Using gNMI
- Create VLAN (130) on leaf11 and leaf12
- Map the VLAN to the VNI on both leaves
- Add the tenant's port to the VLAN



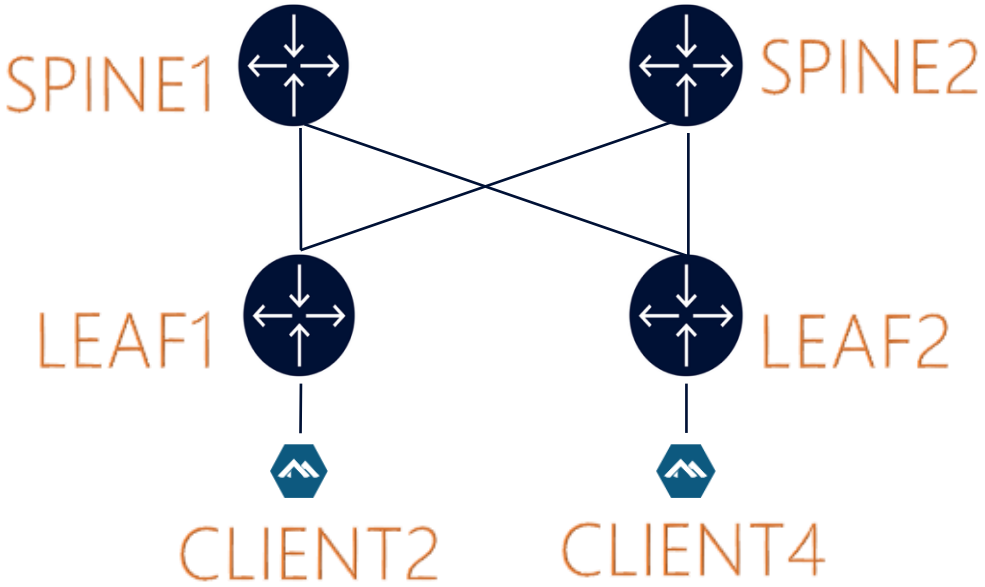
© 2026 Nokia

SONiC Maintenance

© 2026 Nokia

Maintance Operation - Overview

- During this activity, we will simulate a maintenance operation on active Spine
- To perform maintenance, Spine traffic needs to be drained out and taken out of service
- We will use the generally known GSHUT community to achieve this



© 2026 Nokia

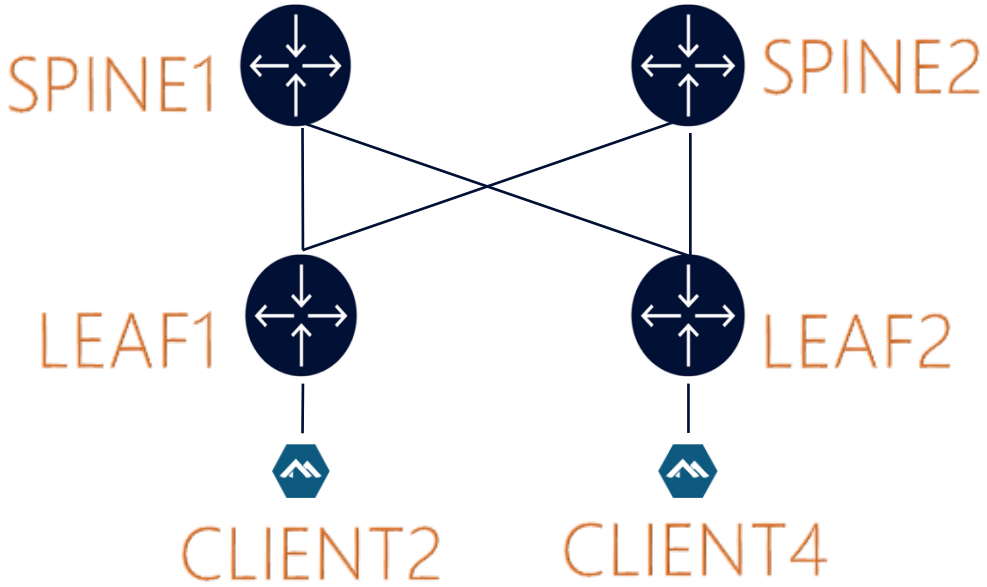
BGP Graceful Shutdown Community

- Graceful Shutdown (or GSHUT) is a well known BGP community (65535:0) standardized in IETF RFC 8326
- Upon receiving this GSHUT community, based on a pre-defined route policy that matches with this GSHUT community, the local-pref value will be set to 0.
- As a result, the remote sender will now prefer a higher preference and drain all traffic from Spine

© 2026 Nokia

Maintance Operation – Configuration Overview

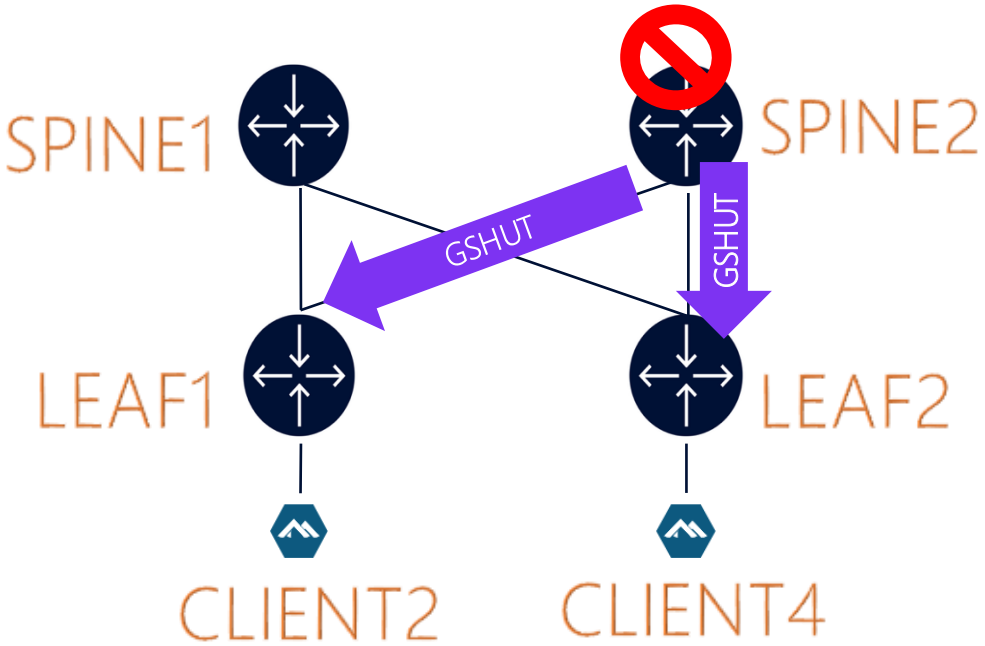
- Configure Route policy on Spines to push GSHUT community
- Configure Route policy on Leafs to change local-preference when GSHUT community is received



© 2026 Nokia

Maintance Operation - Implementation

- Assuming Spine 2 is active and needs to be taken out
- Attach route policy to all BGP neighbors on Spine 2
- Spine 2 sends GSHUT community to all neighbors
- All Leafs receiving GSHUT will set local-pref to 0 for Spine 2 routes
- Traffic moves to Spine 1



© 2026 Nokia

Maintenance – Pre-Configuration Steps



06-sonic-mw

- Configure Route policies on leafs to set local-preference if GSHUT is received
- Configure Route policies on spines to send GSHUT community (Policy is only applied before the maintenence window)
- Start traffic between clients
- Monitor Grafana to find the active Spine
- Depending on which Spine is active, apply route policy on active Spine BGP neighbors
- Monitor Grafana to confirm traffic drain
- At the end of the maintenance window, remove the route policy

© 2026 Nokia

Route Policy on Leafs



06-sonic-mw

```
bgp community-list standard CL_GSHUT seq 5
permit graceful-shutdown
route-map RM_EVPN_GSHUT_IN permit 10
match community CL_GSHUT
set local-preference 0
exit
!
route-map RM_EVPN_GSHUT_IN permit 20
exit
!
```

Apply policy to both Underlay and Overlay

```
router bgp 64501
address-family ipv4 unicast
neighbor 192.168.10.3 route-map RM_EVPN_GSHUT_IN in
neighbor 192.168.12.3 route-map RM_EVPN_GSHUT_IN in

address-family l2vpn evpn
neighbor 11.11.11.11 route-map RM_EVPN_GSHUT_IN in
neighbor 22.22.22.22 route-map RM_EVPN_GSHUT_IN in
```


© 2026 Nokia

Route Policy on Spines



06-sonic-mw

```
gnmic -a spine11:8080 -u admin -p admin --insecure \
  -e json_ietf set \
  --update-path 'sonic-db:/CONFIG_DB/localhost/ROUTE_MAP_SET' \
  --update-value '{"RM_SPINE_GSHUT_OUT": {}}' \
  --update-path 'sonic-db:/CONFIG_DB/localhost/ROUTE_MAP/RM_SPINE_GSHUT_OUT|10' \
  --update-value '{"route_operation": "permit","set_community_inline": ["graceful-
shutdown","additive"]}'
```

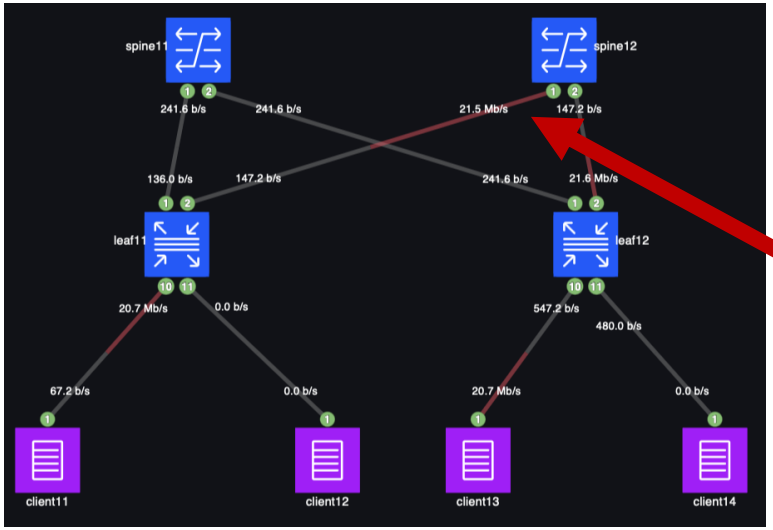
Policy is only applied before Maintenance starts

© 2026 Nokia

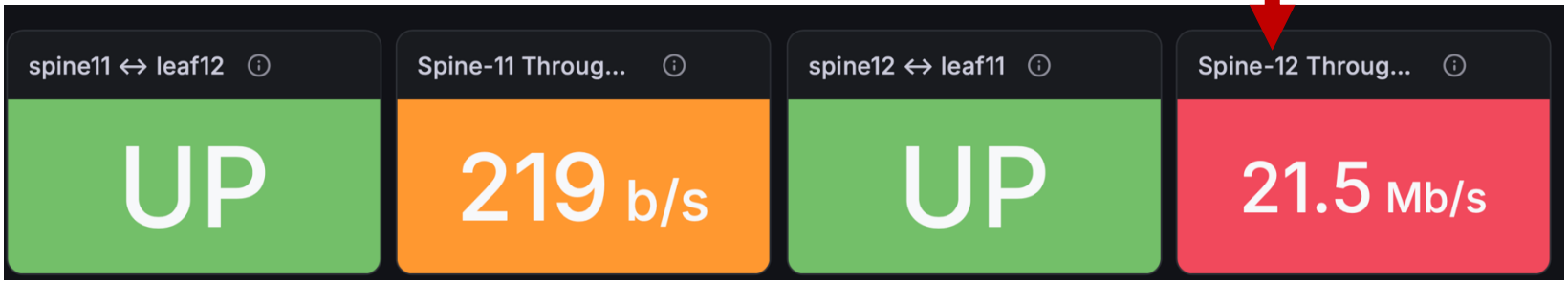
Before Maintenance

06-sonic-mw

Start traffic from client3 to client1



Find Active Spine



© 2026 Nokia

Reda Laichi

Maintenance Start – Apply policy on Active Spine Neighbors

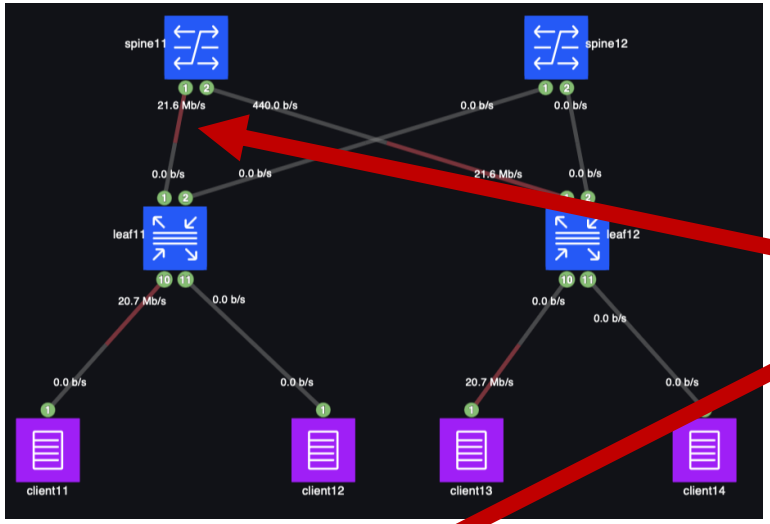
```
gnmic -a spine12:8080 --insecure \
  -e json_ietf set \
  --update-path 'sonic-
db:/CONFIG_DB/localhost/BGP_NEIGHBOR_AF/default|192.168.12.2|ipv4_unicast' \
  --update-value '{"admin_status":"up","route_map_out":["RM_SPINE_GSHUT_OUT"]}' \
  --update-path 'sonic-
db:/CONFIG_DB/localhost/BGP_NEIGHBOR_AF/default|192.168.22.2|ipv4_unicast' \
  --update-value '{"admin_status":"up","route_map_out":["RM_SPINE_GSHUT_OUT"]}' \
  --update-path 'sonic-
db:/CONFIG_DB/localhost/BGP_NEIGHBOR_AF/default|1.1.1.1|l2vpn_evpn' \
  --update-value '{"admin_status":"up","route_map_out":["RM_SPINE_GSHUT_OUT"]}' \
  --update-path 'sonic-
db:/CONFIG_DB/localhost/BGP_NEIGHBOR_AF/default|2.2.2.2|l2vpn_evpn' \
  --update-value '{"admin_status":"up","route_map_out":["RM_SPINE_GSHUT_OUT"]}'
```

Policy is applied to both Underlay and Overlay neighbors

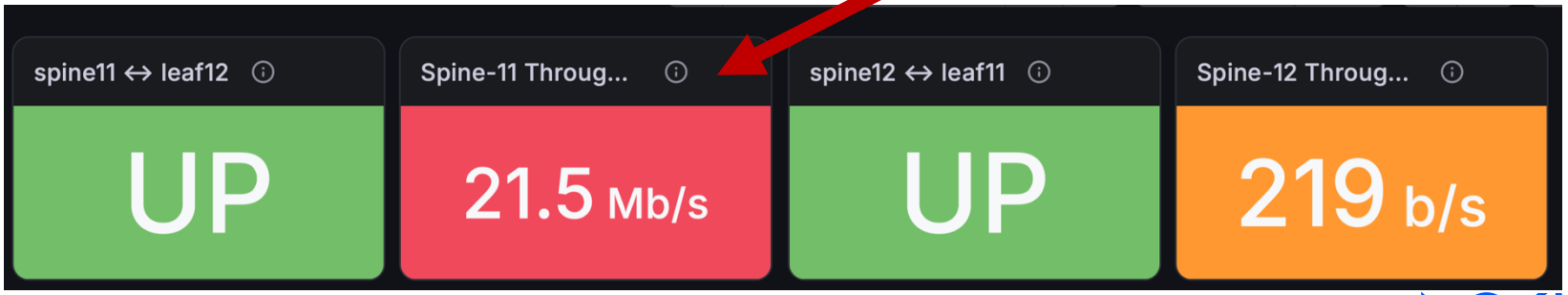
© 2026 Nokia

Check traffic drain

06-sonic-mw



Active Spine has switched



© 2026 Nokia

Reda Laichi

Post Maintenance – Remove policy on Spines

```
gnmic -a spine12:8080 --insecure   -e json_ietf set   --delete 'sonic-
db:/CONFIG_DB/localhost/BGP_NEIGHBOR_AF/default|192.168.12.2|ipv4_unicast/route_map_out'
--delete 'sonic-
db:/CONFIG_DB/localhost/BGP_NEIGHBOR_AF/default|192.168.22.2|ipv4_unicast/route_map_out’

gnmic -a spine12:8080 --insecure   -e json_ietf set   --delete 'sonic-
db:/CONFIG_DB/localhost/BGP_NEIGHBOR_AF/default|1.1.1.1|l2vpn_evpn/route_map_out'   --
delete 'sonic-
db:/CONFIG_DB/localhost/BGP_NEIGHBOR_AF/default|2.2.2.2|l2vpn_evpn/route_map_out’
```

Policy is removed from both Underlay and Overlay neighbors

© 2026 Nokia

SONiC PCAP

© 2026 Nokia

New Containerlab features

Inspect Interfaces

07-sonic-pcap

```
#containerlab inspect interfaces --name ac6-sonic --node leaf11
```

```
11:36:31 INFO Parsing & checking topology file=ac6-sonic.clab.yml
```

Container Name	Name	Alias	MAC	Index	MTU	Type	State
leaf11	eth0	N/A	aa:c1:ab:5a:d0:40	2	1500	veth	up
	eth1	N/A	aa:c1:ab:98:27:c1	2331	9500	veth	up
	eth10	N/A	aa:c1:ab:fd:38:e3	2339	9500	veth	up
	eth11	N/A	aa:c1:ab:3e:2b:5f	2345	9500	veth	up
	eth2	N/A	aa:c1:ab:05:bf:a6	2344	9500	veth	up

© 2026 Nokia

Reda Laichi

New Containerlab features

07-sonic-pcap

Packet Capture

Starting Containerlab 0.83, it is now possible to do a packet capture on one or more interfaces managed in Containerlab.

Start a ping from spine11 to leaf11.

```
# containerlab capture --name ac6-sonic --target leaf11:eth1 --duration
10s -w sonic.pcap
```


© 2026 Nokia

PCAP in SONiC

tcpdump

07-sonic-pcap

SONiC runs on a Debian kernel.

Standard tcpdump can be used to capture packets on a SONiC interface.

Login to leaf11 while the previous ping is running:

```
# tcpdump -v -i Ethernet0
```

```
tcpdump: listening on Ethernet0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
18:06:48.838619 IP (tos 0x0, ttl 64, id 7486, offset 0, flags [DF], proto ICMP (1), length 84)
     192.168.10.3 > 192.168.10.2: ICMP echo request, id 4, seq 153, length 64
18:06:48.838682 IP (tos 0x0, ttl 64, id 31471, offset 0, flags [none], proto ICMP (1), length 84)
     192.168.10.2 > 192.168.10.3: ICMP echo reply, id 4, seq 153, length 64
```


© 2026 Nokia

Reda Laichi

PCAP in SONiC

Remote tcpdump

07-sonic-pcap

Start a remote pcap from VM to a SONiC switch and write to a file on the VM.

```
# ssh admin@leaf11 "sudo timeout 10 tcpdump -i Ethernet0 -U -w -" > sonic-tcpdump.pcap
```


© 2026 Nokia

Conclusion

© 2026 Nokia

What did we learn today?

- Deploying SONiC for a Data Center Fabric
- Hands on practice with:
  - Deploying a SONiC virtual lab using Containerlab
  - Configuring BGP and EVPN on SONiC using CLI and gNMI
  - Using gNMI for configuration and streaming Telemetry
  - SONiC Observability using Grafana Dashboard
  - SONiC Maintenance Operation

© 2026 Nokia

NOKIA

Join Nokia for Happy Hour at AutoCon 6

We invite you to connect with fellow network automators for an evening of food & drinks. This is your opportunity to share ideas, exchange insights, and unwind in a relaxed setting.

Join us on the stunning Fiesta Lawn at The Westin La Paloma Resort, set against the breathtaking backdrop of the Santa Catalina Mountains.

This event is by invite only and space is limited.

Please ensure you bring your AutoCon 6 badge and Nokia Happy Hour confirmation email. This event is only for AutoCon 6 attendees.

November 17, 2026
7:00 to 9:00 pm

Fiesta Lawn
The Westin La Paloma Resort
3800 E. Sunrise Drive
Tucson, AZ 85718

Register today!

© 2026 Nokia

Reda Laichi

NOKIA


NOKIA