Automating Nokia SR Linux Router using Ansible: Enhancing the Standard Ansible Collection

Uzma Saman Aejaz Chanderki

NetDevOps Engineer @Logicalis Connected GmbH

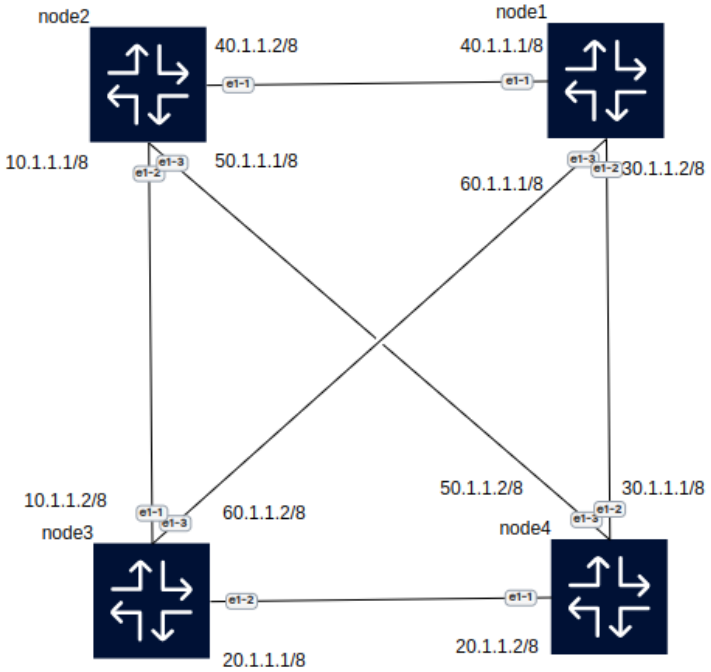
About Me

- 5 years experience across networking, hardware, IT support, and monitoring
- Certified: **CKA**, **AWS**, previously **CCNA/CCNP**
- 2 years at **Logicalis Connected GmbH**, worked with a wide range of tools and platforms
- Co-founded NetAuto group, Built open-source projects: **Terraform provider for GNS3**, **Ansible module for SR Linux**

Table of contents

1. Motivation

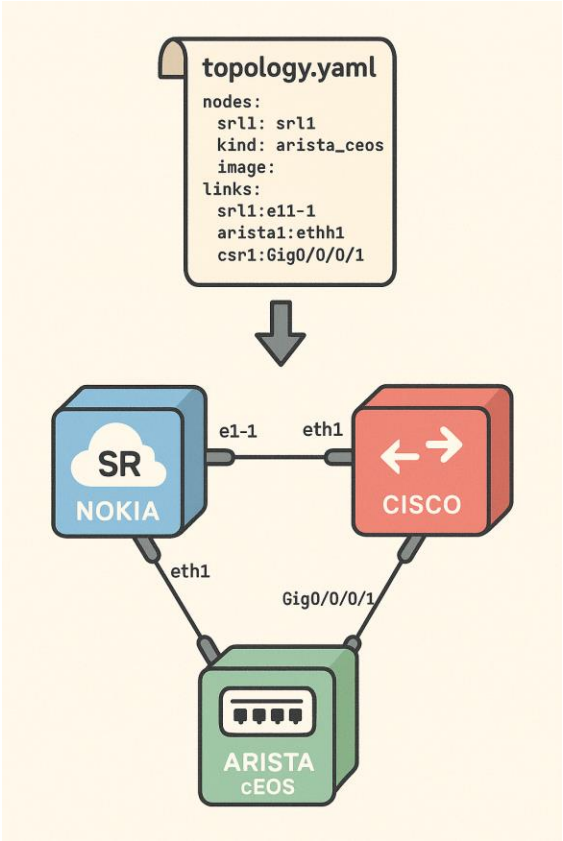
- **Introducing the extended Ansible collection for Nokia SR Linux Nokia's containerized network operating system.**
- I'll be demonstrating a 4-node SR Linux topology deployed with **Containerlab**, and configuring **static routing** so that all routers achieve full connectivity.



2. Why ContainerLab?

Containerlab is an open-source tool that lets you build and run network topologies in containers.

- **Deploy it and Manage**
- **Access nodes** via SSH
- **Destroy** when done (containerlab destroy -t clab.yml)



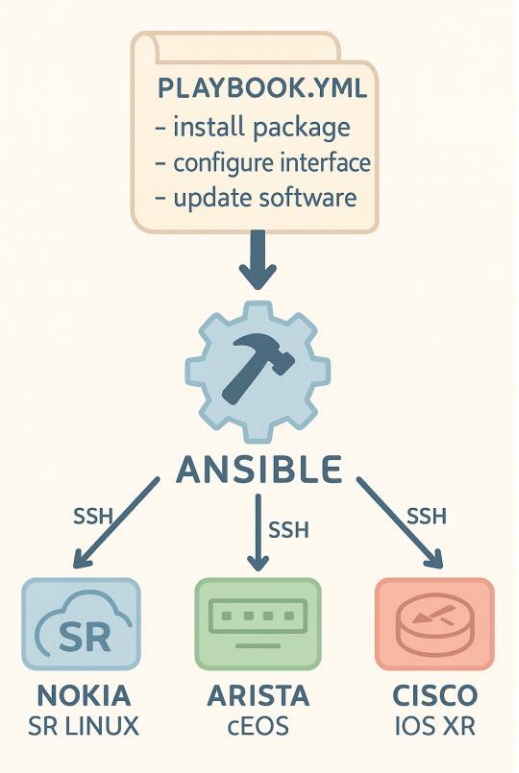
3. Why Ansible?

Ansible is an **automation tool**. **Beginner friendly? It's the *lingua franca.***
It lets you **describe what you want** on servers, routers, switches, cloud platforms, etc. in **simple YAML files**, and then it makes those changes for you.

- It's **agentless**
- It's **declarative**
- It's **idempotent**

3.1 How does it work ?

1. **Playbooks** → YAML files where you write tasks:
2. **Modules** → the little building blocks that do the actual work (cisco.ios.ios_l3_interface).
3. **Inventory** → list of devices you're managing.
4. **Ansible Engine** runs the playbook → connects to each host → runs modules → reports back "changed" or "ok".



4. Issue with the Existing Ansible Collection

- **Modules:** get, config, cli, validate
- **Low-level:** must use exact YANG paths (not intuitive)
- **Troubleshooting**: typos = cryptic errors
- **No abstraction:** only generic config, no ios_interface-style modules

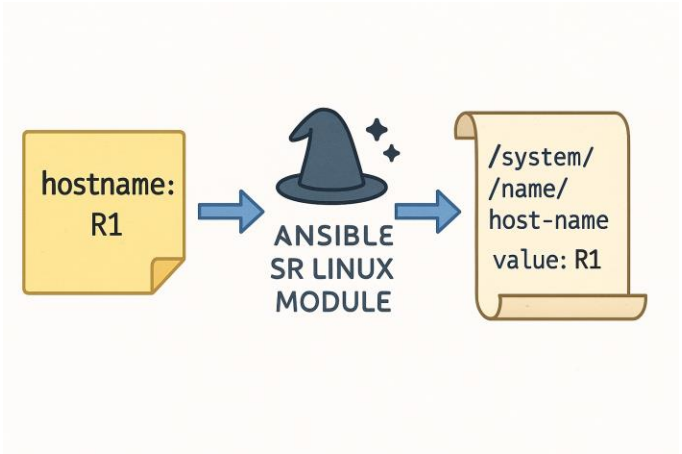
5. Extending the Nokia SR Linux ansible collection

The newly added modules to the existing collection:

- **`Hostname`** → Manage system hostname (`/system/name/host-name`)
- **`Interface (l2/l3)`** → Configure interfaces and subinterfaces (enable, IPs, VRF binding)
- **`static_routes`** → Define next-hop groups and static routes in a network instance
- **`ospf`** → Configure OSPF process, areas, and interface participation
- **`bgp`** → Configure BGP neighbors, ASN, policies
- **`network_instance`** → Create and manage VRFs (L3 forwarding contexts)

5.1 How High-Level Modules Work ?

- In SR Linux, **every configuration item lives at a YANG path** (e.g., /system/name/host-name).
- The **module understands these YANG paths** so you don't have to.
- Your **simple input** (hostname: R1) gets automatically translated into the correct low-level path.



DEMO

6. Conclusion

- **Was it worth it ? Yes learned more and more about the YANG model**
- **Learned how to make a ansible module.**
- **The configuration is more doable.**

Call for presentation at NetAutober Fest 2025

- **Save the Date:** 2nd October **NetAutober Fest 2025** 🎃
- Extension of the **Hacktoberfest tradition** (formerly hosted by Siticom).
- **Highlights of the event:** Engaging talks on **open-source contributions**.
- **Hackathon Challenge** with 3-4 real-world challenges.
- Hosted on the **NetAuto Bootcamp Code Space**.
- **Focus:** Open Source (with preference for **DevOps** and **Network Automation** topics).

THANK YOU AND QUESTION?