Orchestrating Change: Automating GÉANT Network Migration

Simone Spinelli - <simone.spinelli@geant.org>
NETWORK ARCHITECT

FOSDEM 2024 - Brussel
2-3 Feb 2024
What is GÉANT

- Runs a membership association for Europe’s National Research & Education Networks (NRENs)
- Coordinates and participates in EC-funded projects
- Operates a pan-European e-infrastructure
- Manages a portfolio of services for research & education

In Europe: 38 NRENs + NORDUnet, supporting 10,000 institutions and 50 million academic users
The global R&E network

- June 2023

Legend:
- Dark shading: connected to regional network
- Light shading: eligible to connect to regional network

- Multiples of 10Gbps
- 10Gbps
- Multiples of 100Gbps
- 1-10 Gbps
- <1Gbps
GÉANT and Network Automation

• Started in 2019 as a “nice to have”
• First Saltstack, then Ansible, then Ansible + Jenkins
• Now stable with:
  • Workflow Orchestrator: high level coordination + Service Database
  • Ansible: Low-level automation and routers interaction
  • LSO - Lightweight Service Orchestrator: API around ansible, does nothing.
• Not much internal buy-in until Jun 2023: IP/MPLS layer is re-procured and NOKIA wins the tender:
  • No more Junos,
  • copy-paste is not possible,
  • New NOS, new constraints, new concepts.
  • ...and do it quickly.
Migrating to another vendor

- **Current platform: Juniper MX (960/480/204)**
  - 10+ years of experience
  - Very good integration with Ansible
- **New platform is NOKIA 7750SR-S – SR OS, not SR Linux:**
  - No previous experience with SR OS: but it is really close to JunOS – with Caveats
  - Relatively small community: but Netconf works as expected
  - Brings 400/800G to the edge: a big step towards Terabit network
  - Introduces ZR+ optics
- **Sites organized by TIERS:**
  - One ring at the time
  - Going from the core to the edge and then back
- **Migration will be in 2 phases:**
  - PHASE1: Nokia nodes deployed as LSR (starts Q1 2024)
  - PHASE2: Nokia nodes promoted to PE (starts Q3 2024)
- **Totally automated:**
  - Operators run workflows
  - No CLI needed in sunny days
Automation & Orchestration

• Necessary together, but not the same thing
• Modelling configuration constructs that actually represent your services is complex.
• Understanding what is the lifecycle process end to end is also complex
• Generating the right configuration is “just” complicated.

Automation:
• Atomic: compile a template, do something
• Context specific: (Nokia/Juniper/.../Linux)
• Its really about the final artifact

Orchestration:
• Involves multiple systems
• Agnostic to the local context
• Represent the business logic
GAP – GÉANT AUTOMATION PLATFORM
WorkflowOrchestrator

• Originally created by SURF (Dutch NREN)
• Now under The Commons Conservancy foundation with partners SURF, Esnet, GÉANT

• Winning points:
  • Generic, all-purpose orchestrator, written in python
  • Pydantic as data-model engine: strong validation
  • Takes care of the high-level mechanics, but has “plugins” to take care of special tasks
  • SURF and ESnet runs it in production: this gave us confidence.
  • From the community for the community 😊

• And specially, for us:
  • No more change documents but workflows
  • We now have a service database, not (only) YAML files.

https://workfloworchestrator.org/
**WorkflowOrchestrator: main concepts**

- **Product**: A blueprint for a service
- **Product block**: A blueprint for a component of a service
- **Resource**: An attribute
- **Subscription**: An instance of a service
- **Workflow**: A set of steps that manages the lifecycle of a service. Typically:
  - Creation
  - Modification
  - Deletion

```python
class Router(RouterProvisioning, lifecycle=[SubscriptionLifecycle.ACTIVE]):
    '''A router that is currently active.'''
    router: RouterBlock
...

class RouterBlock(RouterBlockProvisioning, lifecycle=[SubscriptionLifecycle.ACTIVE]):
    '''A router that's currently deployed in the network.'''
    router_fqdn: str
    router_ts_port: PortNumber
    router_access_via_ts: bool
    router_lo_ipv4_address: ipaddress.IPv4Address
    router_lo_ipv6_address: ipaddress.IPv6Address
    router_lo_iso_address: str
    router_role: RouterRole
    router_site: SiteBlock
    vendor: RouterVendor
```
LSO – Lightweight Service Orchestrator

- We wanted to reuse our Ansible/Jinja
  - Ansible Tower/AWX seemed too complex
  - No appetite for Cisco NSO (While others use it happily)

- No traditional inventory:
  - Vars are shipped as extra_vars or included in the plays/roles
  - Targets are "on the fly"

```python
extra_vars = {
  "wfo_trunk_json": json.loads(json.dumps(subscription)),
  "dry_run": True,
  "verb": "deploy",
  "config_object": "trunk_interface",
  "commit_comment": f"GSO_PROCESS_ID: {process_id} - TT_NUMBER: {tt_number} - Deploy config for "
              f"{subscription.iptrunk.geant_s_sid}"",
}
execute_playbook(
  playbook_name="iptrunks.yaml",
  callback_route=callback_route,
  inventory="{subscription.iptrunk.iptrunk_sides[0].iptrunk_side_node.router_fqdn}\n              {subscription.iptrunk.iptrunk_sides[1].iptrunk_side_node.router_fqdn}\n"
   extra_vars=extra_vars,
)
Modelling: our approach

- Decompose node configuration in functional parts [PRODUCTS]
- Decompose and model these parts in smaller functional parts [PRODUCT BLOCKS]
Modelling: current models

- Mapping between Products/ProductBlocks and Ansible roles
- Separate roles for migration
- Hierarchy is important:
  - to avoid data replication
  - to describe the network
  - It is important. Really

- For us has been challenging:
  - Which attributes for which product
  - What is the authoritative source of truth
  - “Perfection is not when there is nothing more to add, but when there is nothing more to remove.”
Example of a workflow

**IPAM&DNS**
- IPv4 and IPv6 Loopback addresses are allocated
- DNS is configured

**Resources**
- Device is created in Netbox, with all the interfaces
- This is based on a template

**Config deployment**
- Ansible can target the router via OOB
- Deploys base config
- First dry, then for real
Keeping complexity manageable

A software solution for a networking problem
• Strict DTAP has been more challenging than expected
• Exception will break our models
• Hard to stop thinking in terms of lines of config

Ansible:
• For PHASE1 (P routers and Core Links) we have Ansible roles per Product
• For PHASE2 (Full PE routers and customer migration) we want to have:
  • Ansible roles per product block
  • Check playbooks integrated in the same role

Team:
• People from many different organizations are helping us inside the project
• Many people but not so many FTEs: lots of overhead.
You want to know more?

• About the GÉANT network: https://network.geant.org/

• About WorkflowOrchestrator: https://workfloworchestrator.org/orchestrator-core/
  • Beginner workshop: https://workfloworchestrator.org/orchestrator-core/workshops/beginner/overview/
  • Intermediate workshop: https://workfloworchestrator.org/orchestrator-core/workshops/advanced/overview/

• About GAP:
  • https://docs.gap.geant.org/
  • https://gitlab.software.geant.org/goat/gap
  • LSO: https://workfloworchestrator.org/lso/

Or find me around if you want to chat!
Thank You, Happy FOSDEM!

Any questions?

www.geant.org