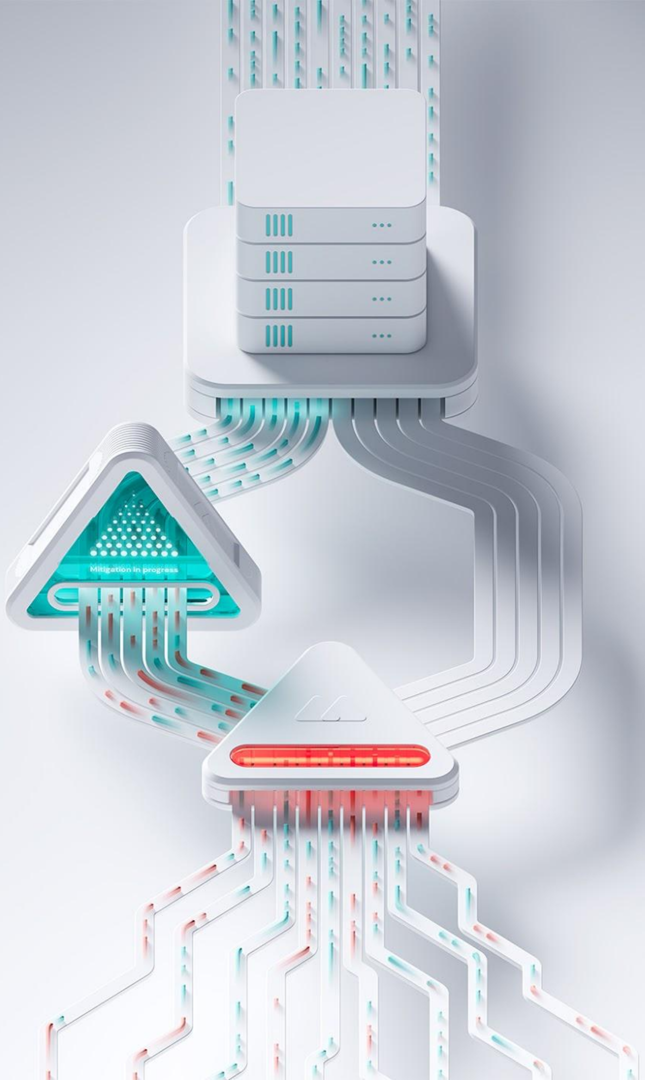
PROZETA



PROZETA

Blindspot

Next-Gen Network Management:

Our journey to implement
CI/CD and Open Source Technologies

David Cermak, CTO

Josef Miegl, NetOps Engineer

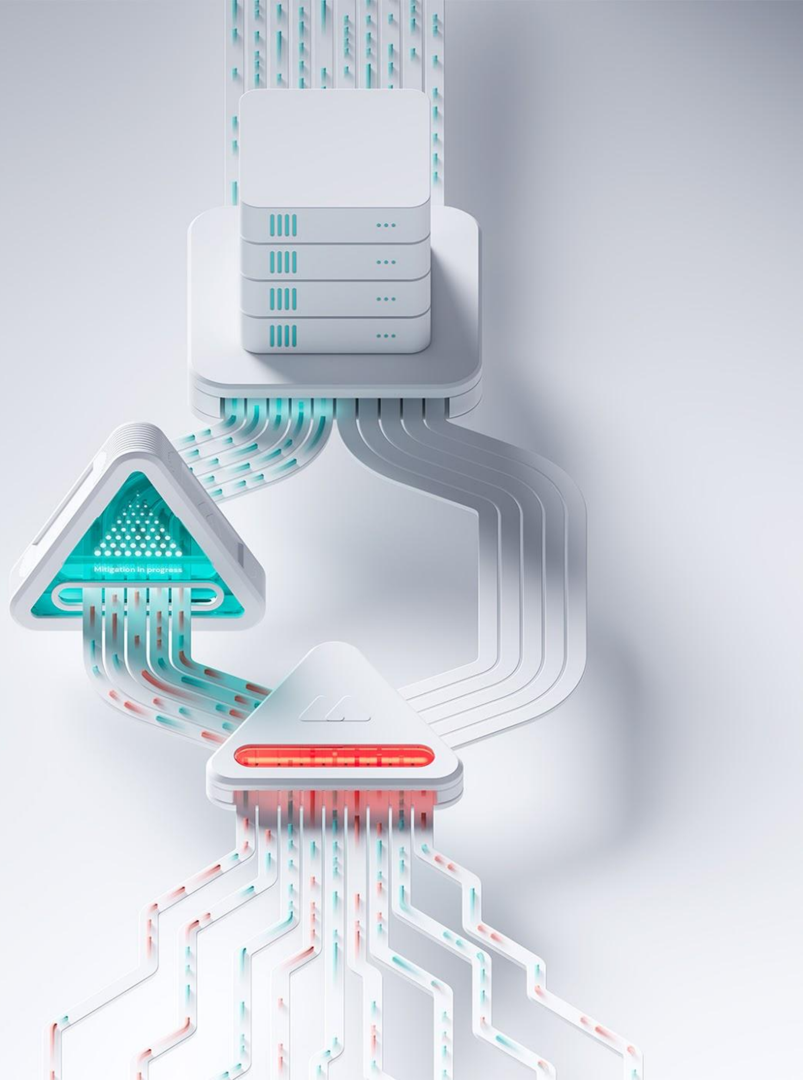
What will you hear today?

- We have managed to transform our network infrastructure and would like to share our experience with network automation.

- Agenda:

  Why us
  The bottlenecks of traditional network management
  Our plans for network automation
  How we achieved our goal
  Demo
  Lessons learned and future goals

- 10 minutes for Q&A at the end of the presentation



Our services

Blindspot

Cloud-based network and data center protection

- Protect customers from network threats (DDoS attacks & others)
- Demanding internet-facing infrastructure

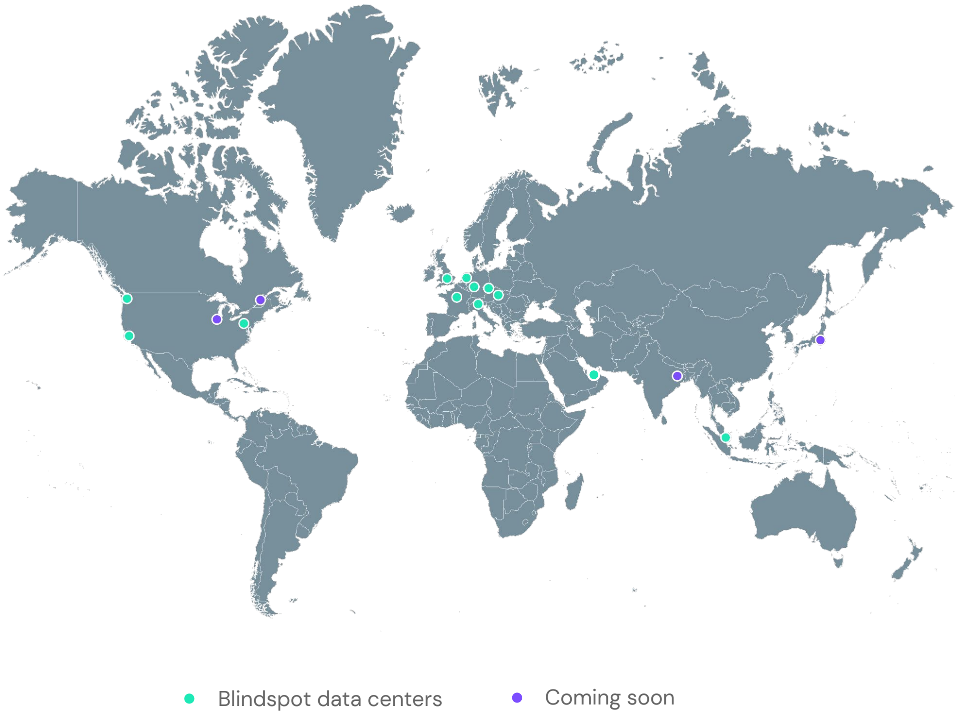
Tier·5®

Private Clouds & Computing Infrastructure

- IaaS, PaaS, HPC, Big Data, AI
- Hosted or on-premise
- Large internal network

Our network

- PoPs in 16 cities / 10 countries
- > 5 Tbit/s edge capacity
- 100 GbE backbone in Europe
- Routed (L3) network + VXLAN overlay



Why we need change – our starting point

- Manual processes
  “Hey, can you create a VLAN between A and B? I don't know what the available VLAN number is (obviously).”

- Network changes were the domain of network administrators only, although we had a larger operations team with devops/linux administrators.

- We needed a service-oriented infrastructure (just like we already had servers and applications).

- Configuration inconsistencies and bugs

- Limited scalability

PROZETA

Blindspot

Network Automation

The Holy Grail

PROZETA

Our Plan

Beyond the network automation

1

Service-oriented infrastructure

We want to configure services - not network devices and ports

Deploy server de.fra15.gb-rc3-11 as OpenStack hypervisor for service-id 45887 (customer ADIB)

2

Data Center Infrastructure Management (DCIM), IPAM

Asset management

Energy & environment

Security

...

3

Advanced Monitoring

Monitoring systems to fully understand the infrastructure as well as services that run on top of it

Better network flow monitoring

The Plan for Network Automation

Full automation (or semi-automation)

The network configuration should match the requirements of our services. Automatically.

Programmable, vendor-agnostic

We want to be free to choose the vendor and the technologies and workflows we want to use.

Anyone can manage it

We would like to enable all of our system engineers to modify the operation of our infrastructure, customize and deploy new services.

Continuous Integration / Continuous Delivery (CI/CD)

Based on well-tested templates, we should be able to deploy new services at any time.

Change management

We need full control over all changes in our network. Whether they are breaking or not. Customers should be informed when necessary.

Open source components

We need flexibility in how we automate our network. Any innovation should be possible. That's why we want open source components.

Options available

We need customizability & flexibility

Do it yourself

Creating such a solution from scratch seemed quite complex and with many potential dead ends.

Open source solutions

Many DCIM, asset management, IPAM tools. Netbox seems to be the strongest..

Network automation basically non-existent - it's a feature in the roadmap (Nautobot) or works instead just as a configuration backup.

Networking vendors

Not an option - vendor-independent solution required

Enterprise tools - Solarwinds, Netbrain, ..

Full feature set, difficult to implement service-oriented infrastructure, complex change management
We follow the KISS (Keep it Simple, Stupid) principle and want to use existing well-managed open source tools where possible.

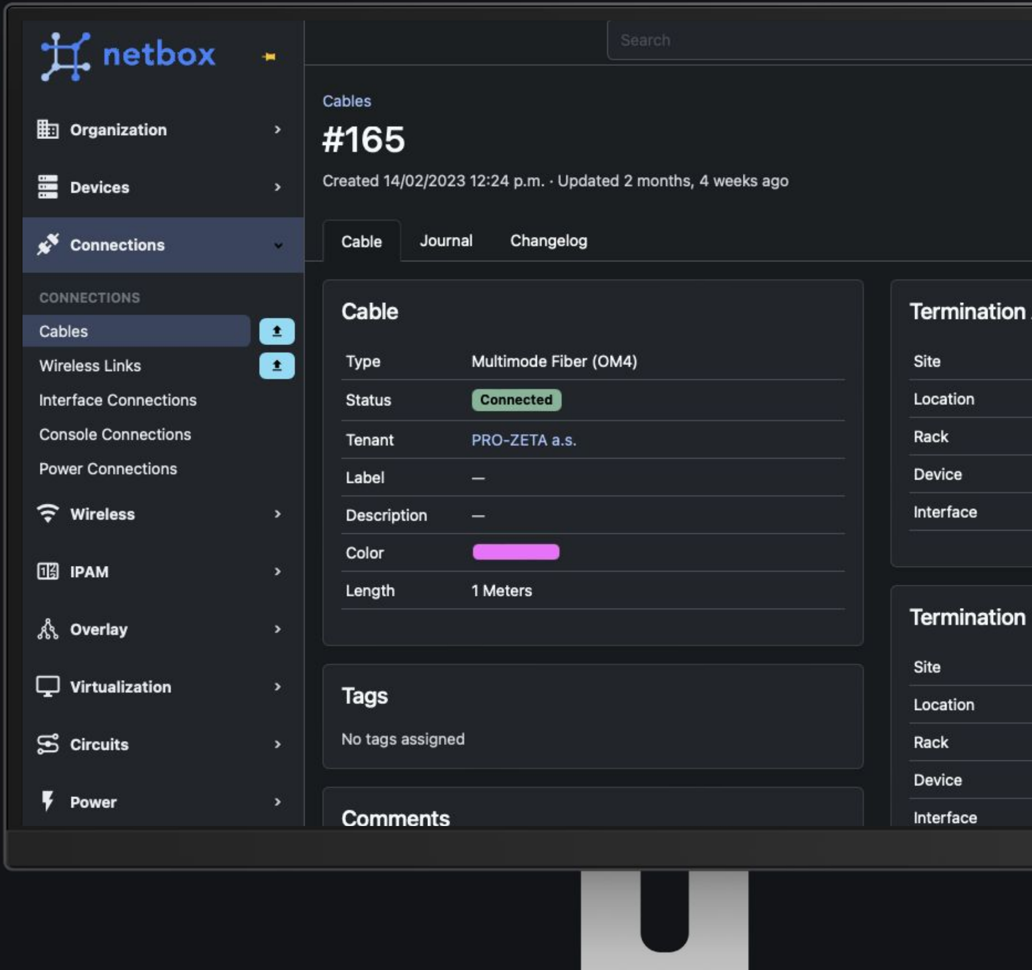
Netbox

Single Source of Truth

We have unified our resources and processes under one platform and gained clear visibility and control over our network and infrastructure.

Data Center Infrastructure Management (DCIM)

Sites, Locations, Racks
Network, Servers & other devices
Power management
Device configurations
Circuits & Connections
Virtualization
Tenants, Contracts, SLAs
… any many more



Cables

#165

Created 14/02/2023 12:24 p.m. · Updated 2 months, 4 weeks ago

Cable

Type	Multimode Fiber (OM4)
Status	Connected
Tenant	PRO-ZETA a.s.
Label	—
Description	—
Color	🟪
Length	1 Meters

Tags

No tags assigned

Comments

Termination

Site
Location
Rack
Device
Interface

Termination

Site
Location
Rack
Device
Interface

PROZETA

netbox

Netbox: Good Aspects

1

All in one place
with a "usable" user interface

We have consolidated a number of data sources in one place. This entails many customizations and plugins.

2

Detailed network configuration
– configure almost anything:

Device type templates
IP configuration
Overlays
VRFs
Modern IPAM

3

Endless customization

Custom fields
Custom validations
Scripts
Tags
Plugins
API

Netbox: Bad Aspects

1

Non-intuitive “usable” user interface

The interface was probably fine when Netbox started and in the early stages. It lags behind commercial products.

Home work: Connect the first cable

2

No versioning & rollback

There is no versioning available. After making a change, you cannot revert to the previous state. A list of changes is available but not sufficient.

3

Rapid development

It's actually a good thing, but upgrades can be challenging and we need multi-level testing.

Dynamic Data and Microservices

Netbox is not a good place to store dynamic data.

- IRR, RPKI, IX peers (IP, ASN, as-set, prefix count), preferred paths
- Need to be stored somewhere other than
- Many different things emerged during development -> microservices

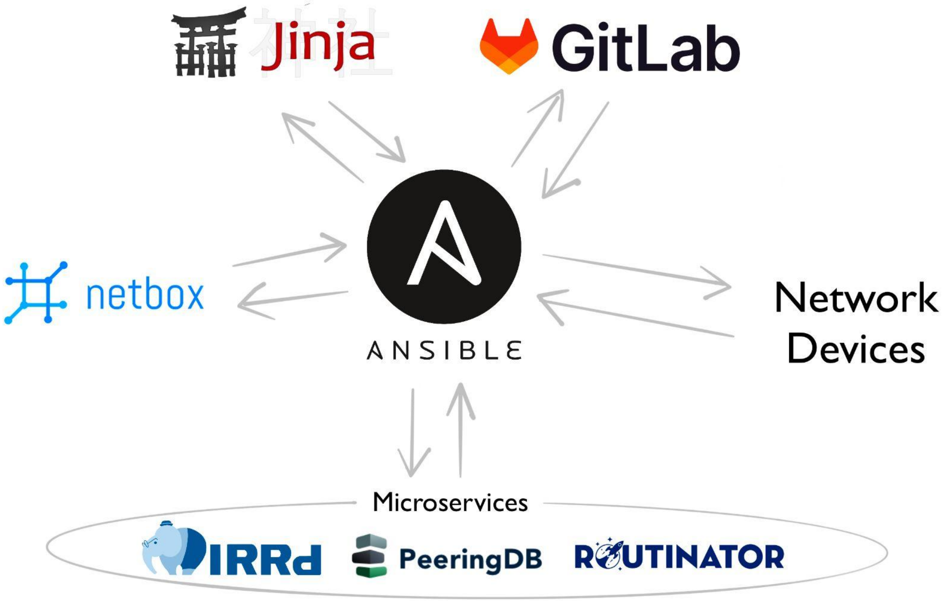
PeeringDB

ROUTINATOR

IRRd

PROZETA

Solution diagram



Monitoring

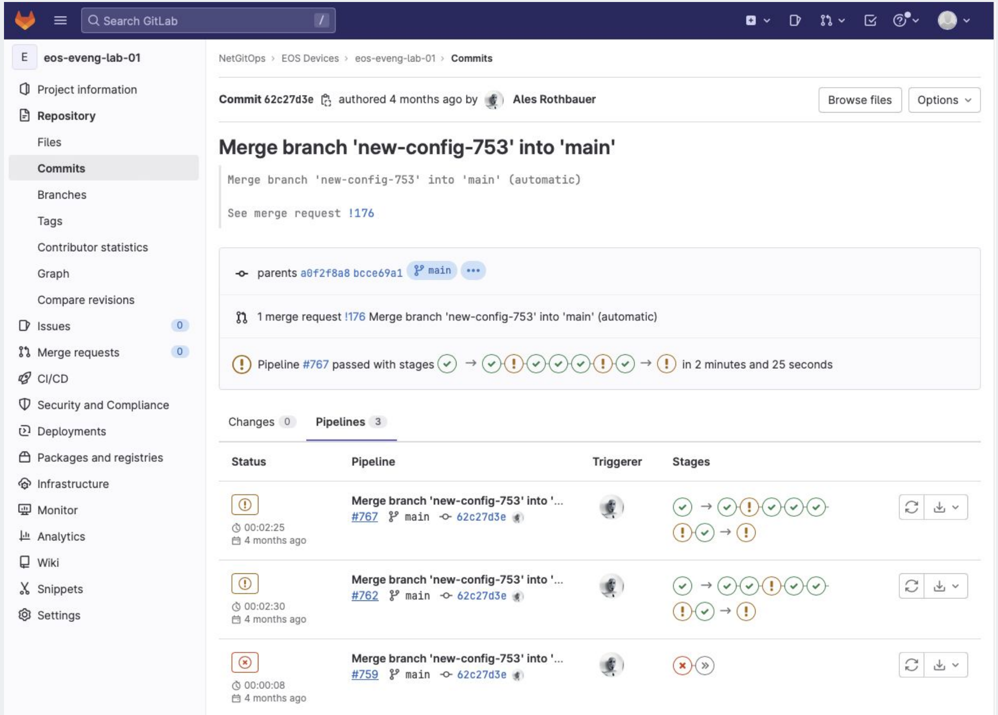
Akvorado

ZABBIX

Logstash,
Elasticsearch,
and many more….

GitLab: Configuration Management

- Git – the industry standard for version control
- No versioning in Netbox
  (in fact, it's one of Nautobot's features)
- So we need to be able to check in and revert changes at least for network configuration.
- GitLab: Store configurations
  - History
  - Running configuration
  - New configuration



NetGitOps › EOS Devices › eos-eveng-lab-01 › **Commits**

Commit 62c27d3e authored 4 months ago by **Ales Rothbauer**

Merge branch 'new-config-753' into 'main'

```
Merge branch 'new-config-753' into 'main' (automatic)

See merge request !176
```

parents a0f2f8a8 bcce69a1 main •••

1 merge request !176 Merge branch 'new-config-753' into 'main' (automatic)

! Pipeline #767 passed with stages ✓ → ✓ ! ✓ ✓ ✓ ! ✓ → ! in 2 minutes and 25 seconds

Changes 0 | **Pipelines 3**

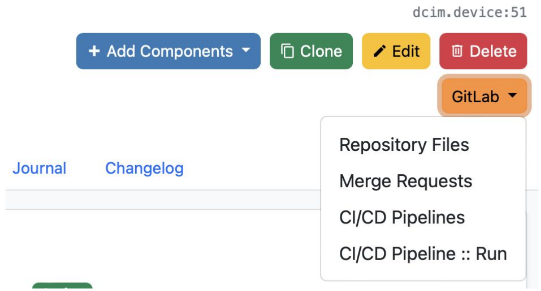
Status	Pipeline	Triggerer	Stages
! 00:02:25 4 months ago	**Merge branch 'new-config-753' into '...** #767 main 62c27d3e		✓ → ✓ ! ✓ ✓ ✓ ! ✓ → !
! 00:02:30 4 months ago	**Merge branch 'new-config-753' into '...** #762 main 62c27d3e		✓ → ✓ ✓ ! ✓ ✓ ! ✓ → !
✗ 00:00:08 4 months ago	**Merge branch 'new-config-753' into '...** #759 main 62c27d3e		✗ »

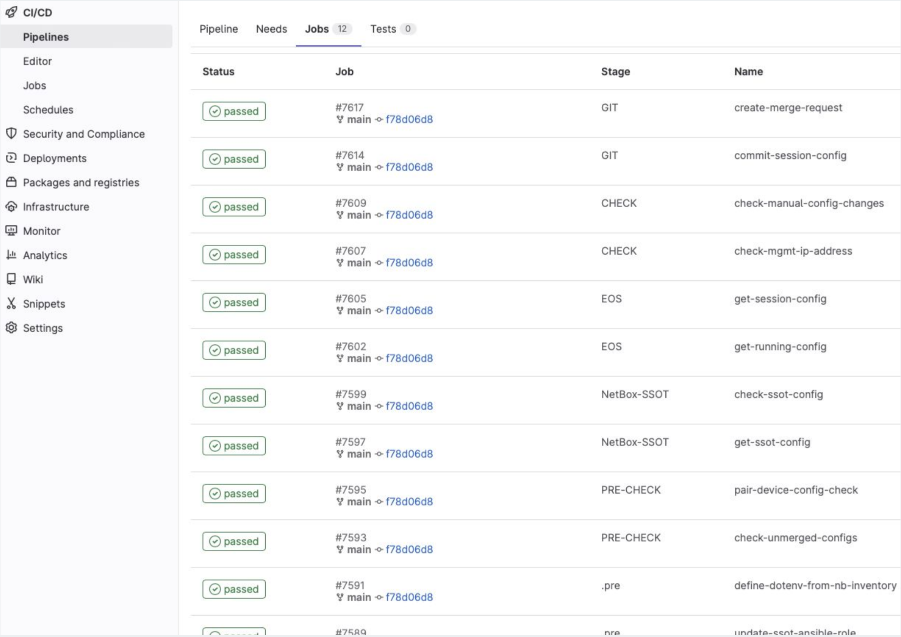
PROZETA

GitLab: Pipelines

- We created the state machine in GitLab pipelines
- Complete workflow: what to do and in what order
- Exception and error handling
- Very powerful in combination with Ansible playbooks

Integration in Netbox





Status	Job	Stage	Name
passed	#7617 main f78d06d8	GIT	create-merge-request
passed	#7614 main f78d06d8	GIT	commit-session-config
passed	#7609 main f78d06d8	CHECK	check-manual-config-changes
passed	#7607 main f78d06d8	CHECK	check-mgmt-ip-address
passed	#7605 main f78d06d8	EOS	get-session-config
passed	#7602 main f78d06d8	EOS	get-running-config
passed	#7599 main f78d06d8	NetBox-SSOT	check-ssot-config
passed	#7597 main f78d06d8	NetBox-SSOT	get-ssot-config
passed	#7595 main f78d06d8	PRE-CHECK	pair-device-config-check
passed	#7593 main f78d06d8	PRE-CHECK	check-unmerged-configs
passed	#7591 main f78d06d8	.pre	define-dotenv-from-nb-inventory
passed	#7589	.pre	update-ssot-ansible-role

Available actions via pipelines

- Approval of new configurations
- Rollback (single device or entire network)
- Diff of configurations

Network Devices

Required Functionality

1

Approve changes

Approval of changes built by the automation engine. This is the role of our network engineers.

2

Rollbacks

Roll back the configuration to any point in time – for a single device or for the entire network.

3

Exception handling

Handling possible exceptions – manual changes in the network, incompatible configurations, etc...

Network Devices

Configuration Challenges

1

Different Network OS version -> different configurations

Network OS upgrades bring different running configurations. A configuration built using a template does not look like the running configuration.

2

Manual changes

Network engineers must make changes to the systems in operation when necessary. This needs to be enabled and tools should address such situations.

3

Different configuration models

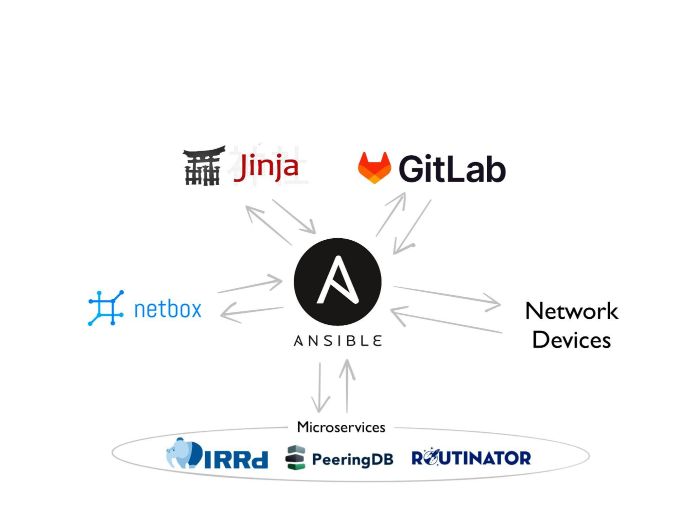
Different vendors have different configuration models. This is doubly true for software-defined routers such as VPP or OpenStack routers.

How it works

- Change in NetBox

- Run GitLab Pipeline from Netbox
  - Ansible w/ nb_inventory & nb_lookup (netbox collection)
  - Jinja2 template -> new configuration
  - Ansible w/ eos_config & eos_command (arista.eos collection)

- Review configuration change in GitLab
  - Approve configuration change (merge)
  - Automated deployment to the infrastructure



It’s Demo Time !!!

From manual to fully automated

1

First deployment in new PoPs

No legacy configurations

Just create templates & Configure Netbox

Easy job!

2

Migration of existing configurations

Difficult to create templates for existing configurations - many non-standard customizations

We don't want to migrate so-called tweaks!

Still migrating legacy infrastructure

3

Service-oriented infrastructure

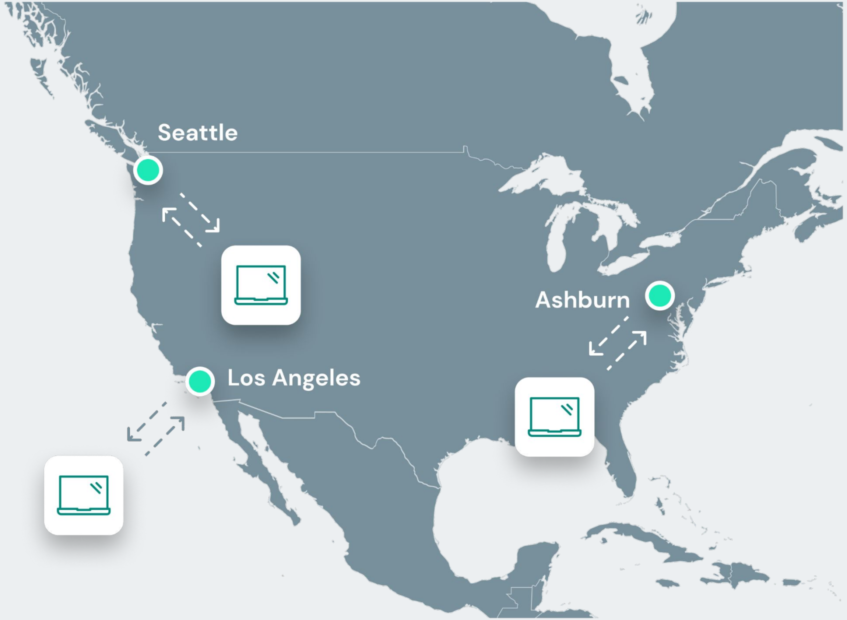
Instead of configuring individual ports in Netbox, we want to configure services for customers.

So far, only a few customer services are deployed this way

PROZETA

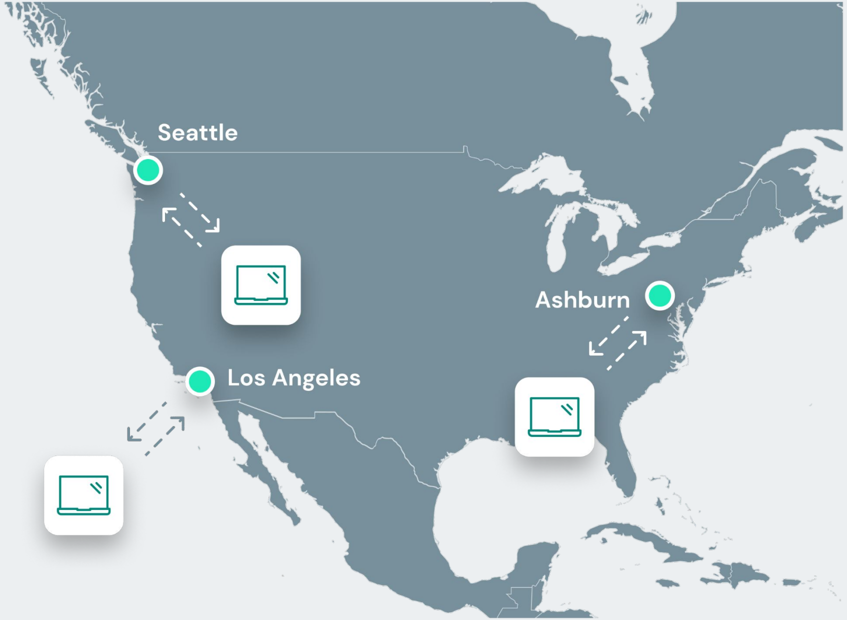
Lessons Learned (Part 1)

- Overcoming challenges
  - Retrieving data from NetBox can be a pain
  - DB relationship != REST/GraphQL API relationship
- Best practices
  - Don't allow device configuration to move away from SSOT -> automatic configuration checks
  - On the next deployment, the configuration should be aligned with SSOT.



Lessons Learned (Part 2)

- Continuous improvement
    - Better to start with something than wait for a large and complex solution
- Recreate configuration in Netbox by hand (if feasible with your scale)
- What is not feasible
    - Continuous integration
    - Automated testing



PROZETA

Out-of-Band (OOB) Management

Implementation strategy: unified design

- Same IP address for network devices management ports as with in-band access
- LTE (+ WiFi/Ethernet when available)
- Road-warrior WireGuard client to two independent cloud-based VPN concentrators
- ALIX-based x86 w/ custom chassis and configuration

Benefits

- No need to switch between in-band and out-of-band
- Full redundancy even for OOB
- Automated deployment & upgrades

PROZETA

Benefits of Automation

A little of what we have already learned

Improved efficiency and flexibility

Do we need to add anything?

Enhanced reliability

More predictable performance and availability with assured configurations. We can deploy changes with little risk.

Access to anyone

Every member of our team can understand how our network works and how it relates to our services. Systems engineers have a full overview and fundamental understanding of our network services.

Services can be delivered across our entire infrastructure without a single touch.

That's the most important thing!

Reduced operational costs

We are already experiencing a much faster rollout of changes and new services with the same team. We can scale faster with the same costs.

Multi-vendor / software-defined

We can freely test and mix and match new technologies and vendors and implement them into our network in almost no time.

Future Outlook

- Removal of manual configurations (in Netbox) in favour of explicit service definitions.
- Management dashboard (plugin for Netbox)
- Continuous integration
- Automated testing
- Staying ahead of industry trends

Conclusion

Network automation using open source technologies and CI/CD pipelines is a game-changer.

Our successful implementation demonstrates improved efficiency, reliability, and flexibility.

Continuous improvement and adaptation are essential for modern network management.

Visit blindspot.cloud

Q&A

PROZETA