networkhero

Network Automation Today and Tomorrow

CATNIX 49 Workshop

Pau Nadeu Rabat
Network Hero

What is Network Hero?

Network Hero is...

- International IT Services Integrator:
  - Network technologies
  - Public and private clouds, virtualization
  - Storage solutions
- Projects in:
  - Spain, Portugal, UK, Netherlands, Argentina
- Customers:
  - Service Providers, Hosting, Enterprise
- Partner of:
  - Nokia, 6WIND

networkhero

Which services do we provide?

IT/Network Design and Integration:

- Audit of the existing IT/network infrastructure
- Design of solutions per customer equipment
- Implementation/installation/configuration
- Knowledge transfer and trainings

Network Operations:

- Remote NOC
- Monitoring

networkhero

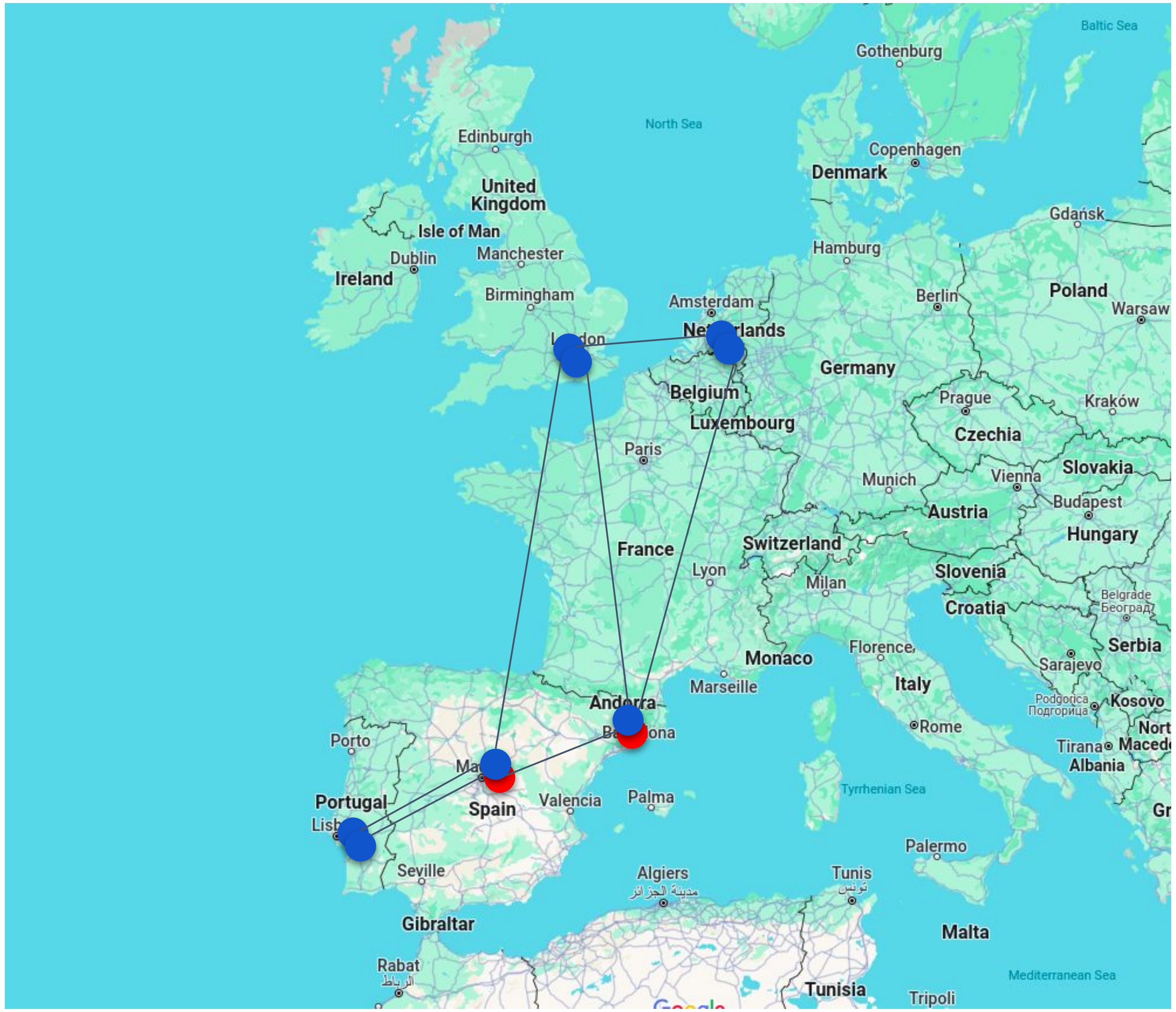
Where do we use network automation?

Everywhere. But today we talk about one specific project:

- Customer:
  - International Service Provider
  - Presence in Spain, Portugal, UK, Netherlands
- Requirements:
  - Speedup and reduce errors during provisioning of customer services (automate provisioning of L2 services)
  - Ensure only relevant services are in network (automate decommissioning)
  - Provisioning shall rely on central documentation

networkhero

Network to automate

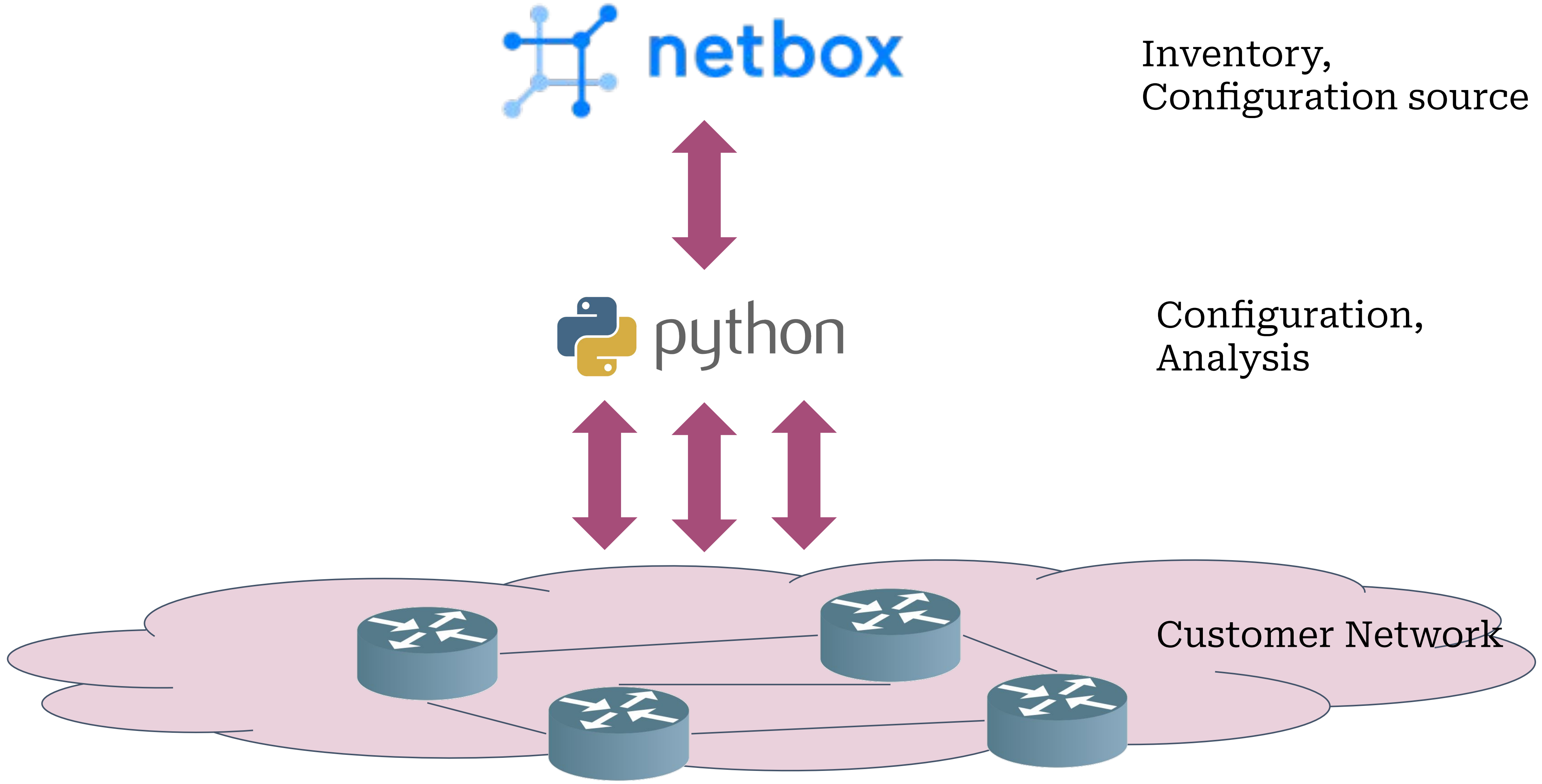


● PoP with Cisco

● PoP with Nokia

networkhero

High-level solution



networkhero

How did it work?

➕ Pros:

- Centralized system for provisioning of customers
- One source of truth - one view of network configuration
- Reduced time for configuration
- Open Source

➖ Cons:

- Efforts to build and maintain system
- Open Source (some packages brook over time)

How can I build such system myself?

Before you start, some questions to consider:

- What is my current inventory database/system?
- How accurate/adequate is that? Is that updated?
- What are the capabilities of my devices?
- Do they support modern management protocols? Which?
- How much time/effort do I have?

And now we start talking tech!

networkhero

NetBox: What is it and what are alternatives?

netbox

Modern Network source of Truth. Capable to do DCIM, IPAM, Services documentation, trigger automation jobs and much more.
Highly extensible via plugins. REST/GraphQL APIs.
Open Source/Commercial.

Deployment Options

You manage

Docker Compose

Managed

netbox labs

Alternatives

Some other systems

>>> nautobot

NetBox: Interaction via REST API

{ REST:API }

REST (Representational state transfer) is a software architectural style that was created to guide the design and development of the architecture for the World Wide Web. De-facto standard for m2m communication via HTTP

Method	URL	Endpoint
GET / POST / PUT/ PATCH / DELETE	`https://netbox`	`/api/dcim/devices/`

Metadata (headers)

```
{
   "Authorization": "Token 1234567890abcdef",
}
```

Response Body

```
[
  {
    "id": 1,
    "name": "router1",
    "device_type": {
      "id": 1,
      "model": "7750-SR1"
    },
    ...
}
```

networkhero

NetBox: Interaction via GraphQL

GraphQL

GraphQL is a query language for API, and a server-side runtime for executing queries using a type system defined for data. A GraphQL service is created by defining types and fields on those types, then providing functions for each field on each type.

Method	URL	Endpoint
POST	https://netbox	/api/graphql/

Metadata (headers)

```
{
   "Authorization": "Token 1234567890abcdef",
}
```

Request Body

```
{
  query get_devices($site:[bcn]){
    device_list(site: $site){
      name
      model {
        name
      }
    }
  }
}
```

Response Body

```
{
  "data": {
    "device_list": [
      {
        "name": "router1",
        "model": {"model": "7750-SR1"}
      }
    ]
  }
}
```

networkhero

NetBox: Hands-on demo

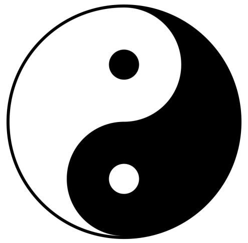
networkhero

Devices: How to manage devices

Many ways to manage network devices:

	SSH	NETCONF	RESTCONF	GNMI
Penetration	Very high	High	Low	Moderate
Data model	CLI	YANG	YANG	YANG
Transport	SSH	SSH	HTTP/HTTPS	GRPC
Serialization	clear-text	XML	JSON	Protobuf
Telemetry support	No	No	No	Yes
Standard	RFC	RFC	RFC	No (info RFC)

Devices: What is YANG?



YANG [RFC7950] is a data modelling language originally designed to model configuration and state data manipulated by the Network Configuration Protocol (NETCONF) [RFC6241]. Since the publication of YANG version 1 [RFC6020], YANG has been used or proposed to be used for other protocols

Defines

- How data structured?
- What are data types?
- What are possible values?
- How to connect to other modules?

```
$ git clone https://github.com/nokia/7x50_YangModels.git
$ cd 7x50_YangModels/latest_sros_20.10/
$ cat nokia-conf.yang
```

https://github.com/YangModels, https://github.com/nokia/7x50_YangModels, https://github.com/aristanetworks/yang

Devices: What are building blocks YANG?

Definition statements

/*** the data model itself ***/

Leaf node

```
leaf user-name {
    type types-sros:display-string {
        length "1..64";
    }
    description "Username for health check";
}
```

Leaf-list node

```
leaf-list apply-groups {
    type leafref {
        path "../../groups/group/name";
    }
    max-elements 8;
    ordered-by user;
}
```

Container node

```
container aaa {
    description "Enter the aaa context";
    leaf-list apply-groups {
    ...
    container radius {
        description "Enter the radius context";
        leaf coa-port {
        ...
}
```

List node

```
list acct-on-off-group {
    key "name";
    max-elements 32;
    description "Enter the acct-on-off-group
list instance";
    leaf name {
    ...
}
```

networkhero

Devices: What is GNMI?

gNMI is a specification for network management via gRPC. Specification is a description of services, calls and messages. gRPC is a modern open source high performance RPC framework that can run in any environment. It can efficiently connect services in and across data centers with pluggable support for load balancing, tracing, health checking and authentication.

Operation	gNMI	NETCONF	RESTCONF
Collect information about the YANG modules supported by the endpoint	Capabilities	get-schema	GET to a specific endpoint
Collect the configuration or operational data	Get	get, get-config	GET
Modify the configuration	Set	edit-config, delete-config	POST, PUT, PATCH, DELETE
Collect the operational data on a push-basis	Subscribe	create-subscription, notification	-

https://github.com/openconfig/reference/blob/master/rpc/gnmi/gnmi-specification.md

Devices: How to use GNMI?

With Python

With Go

Library

Library

https://github.com/akarneliuk/pygnmi

https://github.com/openconfig/gnmic

networkhero

Devices: What is pyGNMI?

pyGNMI

Pure Python implementation of GNMI client.
Import in Python code or use as CLI.

Ready

✓

All RPCs:

- Capabilities
- Get
- Set
- Subscribe (JSON, Protobuf)

Channels:

- Insecure
- Secure with certificate

Integration with Nornir

Use your favorite automation framework

Simplicity

0+

Easy to use

Devices: Code with pyGNMI

Install

```
$ pip install pygnmi
```

Use

```
#!/usr/bin/env python

# Modules
from pygnmi.client import gNMIclient

# Variables
host = ('fd17:625c:f037:2::100',6030)

# Body
if __name__ == '__main__':
    with gNMIclient(target=host, username='aaa', password='aaa', insecure=True) as gc:
        response = gc.capabilities()

    print(response)
```

networkhero

Devices: Hands-on with pyGNMI

networkhero

Together: How to integrate

Just Python

Benefits:
- Easy to start
- Quick to apply first configurations

Drawbacks:
- Difficult to maintain
- Clashes between engineers
- Slow-ish

Python with Nornir

Benefits:
- Quick to apply first configurations
- Quick execution
- Many built-in functions

Drawbacks:
- More complicated
- Difficult to maintain
- Clashes between engineers

NetBox Script

Benefits:
- Natural integration
- UI
- Job queue

Drawbacks:
- Most complexity

Together: NetBox Scripts

Custom scripting was introduced to provide a way for users to execute custom logic from within the NetBox UI. Custom scripts enable the user to directly and conveniently manipulate NetBox data in a prescribed fashion.

Code

```
from extras.scripts import Script, ObjectVar, MultiObjectVar
from dcim.models import Site, Device

class ConfigDevice(Script):
    class Meta:
        name = "Configure Device"
        description = "Sample Script to configure network device"
        field_order = ("vpns", "device")

    devices = MultiObjectVar(
        model=Device,
        query_params={
            "site_id": "$site"
        }
    )

    def run(self, data, commit):
        ...
```

https://docs.netbox.dev/en/stable/customization/custom-scripts/

Together: Requirements for NetBox Scripts

Relevant Python packages available within NetBox

Install dependencies via pip

Rebuild NetBox Container Image

Together: Hands-on NetBox Scripts

networkhero

Conclusion

Main takeaways:

- Think on end-to-end business process to automate
- There are many ways to do things with different pros/cons
- NetBox is a modern and future-proof source of truth
- Use modern management protocols where possible
- Integrations of tools/solutions is a key

networkhero

Questions

networkhero

networkhero

Thanks for your attention

Contacts:
+34 900 525 795
https://www.linkedin.com/company/nhero/
team@networkhero.ad