Network Automation in the Modern Age

Veit Heller, Marcus Stoegbauer

What can we do today easily

- Write templates (e.g. Jinja) with variables for custom input

- Set variables, merge with template, push custom configuration to device

- For example: Ansible, Python Libraries, Saltstack, ...

- Well understood, flexible usage, great documentation and many documented use cases available

- Let's call this: ***Templating***

Why not call it Automation?

Our understanding of network automation has changed over the years. What we do now:

- High level functionality:

  - Write templates, define necessary variables

  - Create variables data for devices and services

  - Call Tool to create full configuration and push to device

- Then we ask ourselves: Where do we store those variables data?

  - YAML files? Database? Documentation System?

  The place where you store your variables needs to become your Single Source of Truth.

YAML Files or Database

- YAML files: ugh

- Database

  - Better than YAML files

  - You need to define the database schema (data model)

  - You need to write custom code to look up data

  - You start at zero, and will probably change the data model a few times

  - Other people already made those mistakes and came up with a software package

Documentation Systems

- DCIMs like Netbox, Nautobot, or any kind of commercial solution

- Developers of those systems created a data model and a way how to use them

- You document via Web UI or API data in the systems (Sites, Racks, Devices, Links between devices, Services, ...), guided by the data model

- But data model introduces – by design – a third party bias

- This data model fits 70-90% of what most networks need - what happens with the remaining 10-30%?

Problem

For some parts of your network, you need to adapt an existing documentation system:

- Create additional fields, use free-text fields, redefine unused fields for other purposes

- Write documentation to make sure people are using those fields correctly

- Write code to verify that entered data is compliant to data model

Architecture Process

- Create Template

- Define Variables

- Define places where to store those variables

- Write code to write and query those variables, and merge variables with templates

- This process is done repeatedly by architects, whenever templates change, or documentation systems change (upgrades!), new products are introduced

- Change at multiple places at once, and make sure not to forget anything

Executive Summary: ***data model creation and constant updating***

In summary

- Manual work is not "getting device configured" anymore, but "setting up and maintaining systems used for configuration"

- But it's still manual labor, and a lot of it – we tried to get rid of this by automation

- Automating this is not in focus:

  - Each company is doing each step only once – no real incentive to make it easier

  - But for us network people this is the same work, over and over again

No solution, more questions

What would we need to make this work easier on us?

- Easier means: less places for changes, and reduced errors, shorter delivery times (uuh, business incentive!)

- Would need a system flexible enough to fit all kinds of backends (documentation systems), data models, networks, ...

- Not one entity can define all those requirements, we need input from a lot of network people: You.

- Let's start a conversation about it. Think about it, talk about it.

More Info and Contact

More verbose version of this talk on
https://www.lys.is/network_automation/manifesto.html

Contact me:

- Lysis on IRCNet

- Mail: marcus@grmpf.org

- Fediverse: @lysis@chaos.social