Zurich University
of Applied Sciences

School of
Engineering
InIT Institute of Applied
Information Technology

Bachelor thesis (IT)

Network Infrastructure automation with NetBox

Author	Jens Vogler
Main supervisor	Dr. Gürkan Gür
Industrial partner	Init7 (Schweiz) AG
External supervisor	Pascal Gloor ???
Date	11.06.2022

Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering

DECLARATION OF ORIGINALITY

Bachelor's Thesis at the School of Engineering

DECLARATION OF ORIGINALITY
Bachelor's Thesis at the School of Engineering

By submitting this Bachelor's thesis, the undersigned student confirms that this thesis is his/her own work and was written without the help of a third party. (Group works: the performance of the other group members are not considered as third party).

The student declares that all sources in the text (including Internet pages) and appendices have been correctly disclosed. This means that there has been no plagiarism, i.e. no sections of the Bachelor thesis have been partially or wholly taken from other texts and represented as the student's own work or included without being correctly referenced.

Any misconduct will be dealt with according to paragraphs 39 and 40 of the General Academic Regulations for Bachelor's and Master's Degree courses at the Zurich University of Applied Sciences (Rahmenprüfungsordnung ZHAW (RPO)) and subject to the provisions for disciplinary action stipulated in the University regulations.

City, Date: Winterthur, 11.06.2021

Name Student: Jens Vogler

Abstract

Foreword

Contents

1 Introduction

1.1 State of current Technology

Networking hardware makes out a huge part in our internet infrastructure. There are an increasing amount of hardware vendors varying in range of functionality. Internet Service Providers (ISPs) are obvious customers of such hardware, but businesses of all sizes may have needs for all kinds of devices. This starts to pose the question, of how these devices shall be managed, especially when the need for scaling up arises. Many vendors supply configuration management solutions, also called Network Management Systems (NMSs) along their hardware offering, such as Cisco's „Cisco DNA Center"[**noauthor_cisco_nodate**] or Juniper's „Junos Space"[**noauthor_junos_nodate**] which might be the perfect fit in an environment wherein all hardware is obtained from the same vendor. On the other hand, in mixed vendor environments, problems arise with these solutions as they primarily support their own hardware.

Since there's no „one size fits all" solution, one might want to come up with a custom solution. In order to create a custom solution, one must know how devices are configured by NMSs (or humans if that is automatable). There are numerous so called network configuration protocols, such as [**fedor_simple_1990**], CLI (e.g. via SSH or serial), NETCONF[**enns_network_2011**] and others, which can be used to remotely configure devices.

In order to configure devices, one must also know which devices exist within the infrastructure. So called infrastructure resource modeling (IRM) software is used to track the devices along with information related to how they are connected and configured. While the above mentioned NMS's usually provide such functionality, there are also standalone solutions such as NetBox, which tracks everything from hardware to IP addresses. Using the IRM as the source of information, one would then be able to use a configuration protocol to confgigure the devices according to ones needs.

1.2 Goal / Aim

The overarching goal is to provide a solution similar to an NMS while utilizing open source tools and software. This will be achieved by extending NetBox with a plugin, which will allow an engineer to author configuration templates which source their data from NetBox itself. These templates can then be applied to devices, wherein the plugin will generate the configuration and subsequently apply it via a chosen configuration protocol.

As it is usually necessary to authenticate oneself in order to apply configuration to a device, the issue of credential management and storage has to be solved.

Target open source space

be vendor agnostic

compare transport protocols

analyze compatibility

1.3 Overview

2 Theory

2.1 OSS/BSS

Differentiate OSS and BSS. Where NetBox will be the core OSS and everything connecting to it is the BSS.

includes inventory management

2.2 Secure credential storage

Need for credentials in configuration

Refer to security theory, least privileges, etc.

2.3 Network Management Systems

2.4 Configuration Methods

2.4.1 CLI Configuration

Probably the most common method of configuring devices is through its command line interface. Most modern networking hardware supports access to the CLI through SSH, but in cases where networking may not be available, the same interface is often also presented in the form of a serial connection, through a dedicated port on the device.

As there is no standardization for the CLI across vendors, it makes it a difficult target for automation in a vendor agnostic space. On the other hand it is also usually the most complete interface in terms of features for the device you are configuring, as most vendors recognise it as the primary interface for configuration[**noauthor_configuration_nodate**].

There are projects such as NAPALM (Network Automation and Programmability Abstraction Layer with Multivendor support)[**noauthor_napalm_nodate**] which aim to solve the problem of providing a similar interface to multiple vendors, but the functionality of such projects is severely limited in terms of interfaces provided for configuration across vendors.

Since the CLI is a text based interface, it makes it hard to validate the configuration before deployment. Especially since the order of configuration statements often matters (e.g. you cannot assign a VLAN before creating it), it makes it hard to stitch together multiple templates without having an engine understand the whole configuration before sending it off to the device.

2.4.2 SNMP

The Simple Network Management Protocol (SNMP)[**fedor_simple_1990**] is a protocol which was originally designed in the 1980s by a group of collaborators which sought to create a protocol which would allow large scale-deployment of the internet. As the name implies, it is quite a simple protocol based on a few key operations: GetRequest, SetRequest, GetNextRequest, GetBulkRequest and some more.

In practice it works with vendor supplied „management information base” (MIB) files, which identify a collection of of object identifiers (OIDs) with a name, description and data-type. Along

the vendor specific MIBs, there are also some standard collections, which try to define generic information which can be queried. So for example the OID „1.3.6.1.2.1.1.1" corresponds to „iso.org.dod.internet.mgmt.mib.system.sysDescr" which represents a textual description of the device which is queried.

SNMP is still largely used today for monitoring, as it is a (as the name implies) simple and light protocol which usually doesn't generate a lot of overhead. There are some shortcomings though when it comes to structured configuration data, especially with write/set operations. Altough data types can now be verified using the MIB files, a configuration section as a whole still cannot be verified before sending it to a device, as dependencies between configuration nodes cannot be specified in the MIB format. Also handling data in lists is complex, as when one wants to edit a specific node, the whole list first has to be searched with get requests in order to identify the correct offset which is to be modified.

2.4.3 NETCONF

Briefly explain the start of the NETCONF protocol

Transition into NETCONF/YANG

2.4.4 NETCONF/YANG

Explain yang modeling and why it helps

refer to useful projects: sysrepo, netopeer2, yangexplorer, yanglib, pyang, ncclient

mention RESTCONF

2.4.5 HTTP APIs

Completely vendor specific

Often not available at all

2.5 User Interface for Engineers

Existing user interfaces

Gathering information from documentation

Gathering information from Yang Models

2.6 Case Study - Init7 (Schweiz) AG

Current state: Not a lot of automation

If configuration from OSS - forces correct documentation -> documentation implies configuration

3 Methods

3.1 Literature Research Method

- Lots of papers on protocols - Difficulty in finding concrete cases - Lots of stuff is propiritary - Open Source solutions are very fragemented - Makes it hard to find examples for working solutions

3.2 Software Engineering Method

3.3 Evalution Method

4 Results

4.1 Current State

4.1.1 Init7 (Schweiz) AG

Init7 is a ISP since the year 2000. While it started with providing ADSL service, they launched their fiber internet service „Fiber7" in 2014 with competitive prices. Since then, the company has been expanding their geographical availability by equipping an increasing amount of PoPs with their own infrastructure. The increase in infrastructure means that there is an ever growing number of devices that have to be configured and managed. While they do employ automations, most of them are implemented in the provisioning process. This would be mostly sufficient if they would only provide basic internet access, as usually no configuration is needed to take a new customer online since the physical patching of the connection is done through the owner of the fiber (usually power stations or Swisscom).

On the other hand, configuration is needed for a lot of business to business (B2B) products, as for example the following services:

- Point-to-Point L2 connection: Enables the customer to create a layer 2 network across multiple sites. This can for example be implemented using a protocol like MPLS.
- IP Transit: Enables the customer to directly connect to our core through BGP.
- Business Internet Service: Very similar to a private service, but with the crucial difference that the customer uses customer premises equipment (CPE) which is managed and monitored by Init7.

In general, the business products also require Init7 to actively monitor status and quality. To be able to collect this data both the access switch within the point of presence (PoP) and the CPE at the customer need to be configured accordingly.

While this thesis aims to provide a general solution for configuration, in order to keep the scope managable, the focus and evaluation is limited to the access layer of the Init7 infrastructure. To get a basic understanding of the Init7 infrastructure, a very simplified diagram is shown in [fig]. The Interior gateway protocol (IGP) used is OSPF and is split in two main parts. The core network ecompasses a cross-connected mesh of routers which are spread all over the world. All interconnects and peerings to ISPs, customers and internet exchanges take place there.

TODO: Explain Access Layer

4.1.2 Rack Black Association

4.2 Requirements Analysis

4.3 Application structure / Architecture

To bridge the gap between the inventory management provided by NetBox and communicating with the actual hardware, the solution is divided into layers. In order to facilitate authentication and authorization there is the „Credential Access Manager" which communicates with an external credential store. The access manager provides a user interface to enter and manage credentials, as well as restricting who has access to these credentials by integrating into the NetBox permission system. The „Device Context Provider" enables a user to define what information for a specific device shall be pulled. Using the existing GraphQL API provided by NetBox, the user can easily identify what information is available and how it is accessed. After the con-

text data is defined it will subsequently be used in the „Configuration Template Editor" where the user writes a template using the previously defined context data. This template can then be applied to a class of devices utilizing constraints such as tags or locations. Lastly, the „Configuration Transport" module pulls the configuration template and the credentials for a specific device and pushes it to the actual device.

4.3.1 Credential Access Manager

4.3.2 Device Context Provider

4.3.3 Configuration Template Editor

4.3.4 Configuration Transport

4.4 Scenarios

The following sections describe three chosen real-world scenarios within Init7. More specifically, they are services which can be ordered by customers and also require some manual work by network engineers in the current process.

The goal is to show how manual labor can be replaced or supported by the created NetBox plugin.

4.4.1 Fiber7

Fiber7 is Init7's flag-ship product in the private consumer space, providing consumers a fiber-optical connection directly into Init7 infrastructure. Since this is the most commonly sold product by Init7 it is already one of the most streamlined processes throughout the department with work being largerly automated.

Service Diagram

When the work-order arrives at the network engineer, it has already been decided or defined at which , network switch and interface the customer will be connected to. The engineer will then go through the following process:

1. A new service configuration ticket arrives
2. Verify that there is no other active use on the chosen interface
3. Apply the default configuration profile for Fiber7 private customers on the interface
4. If the customer requested a static IP address, configure it on the DHCP server
5. If the customer requested a prefix, configure it on the router connected to the switch
6. If the customer does not specify a specific service activation date, enable the switchport
7. Update NetBox with the performed configuration
8. Update the service configuration ticket

While the process might have seven steps, usually not all of that work is required. When choosing the interface for the new service, it is verifed through the OSS that the port has no other active use. Interfaces have the Fiber7 private customer profile configured by default if it was previously unused or was previously used for the same service, rendering step 3 usually unnecessary. Most commonly, customers do not request static IP addresses or prefixes as they are an additional cost to the subscription and are only used by „power-users", rendering steps 4 and 5 unnecessary.

The key observation that can be made, is that there are multiple manual steps which can be prone to mistakes, yet they always performed the same way. This makes them good candidates for automation. Thus the goal is to restructure the process to the following:

1. A new service configuration ticket arrives

2. Within NetBox:
    a) Set the service field on the interface to Fiber7
    b)

(above) list not complete

Describe process to extract template information

Describe verification

4.4.2 Fiber7 Business

4.4.3 Business Optical Service

- B2B customer (standard) - Access Port with less configuration
- VLAN 600
- Static IP on router
- Static Route on router
- B2B customer (new)
- Trunk Port
- Allowed VLAN 600
- + same as standard?
- B2B Site-2-Site
- VLL through MPLS

- MikroTik ZTE

5 Discussion

6 References

6.1 List of Figures

6.2 List of Tables

7 Appendix