TALLINN UNIVERSITY OF TECHNOLOGY

School of Information Technologies

Dmitri Sobolev 212941IAAB

Automating Network Infrastructure Management in a Multi-vendor Environment

Bachelor's thesis

Supervisor: Mohammad Tariq Meeran PhD

Co-supervisor: Siim Vene MSc

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Dmitri Sobolev 212941IAAB

Võrgu infrastruktuuri haldamise automatiseerimine mitme tarnija keskkonnas

Bakalaureusetöö

Juhendaja:	Mohammad Tariq Meeran PhD
Kaasjuhendaja:	Siim Vene MSc

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Dmitri Sobolev

01.03.2024

Abstract

With complexity growing within network environments and device diversity becoming the norm, it becomes quite a challenge during the configuration and management of the network. Hypothetical Company X lacks experience in automating network configuration processes. There are also as a lot of companies still configure and manage their network infrastructures manually. In case of any related configuration change or problem, it takes extra time to configure all devices manually or analyse and identify the problem cause. This approach is causing unnecessary delays in implementing measures to satisfy customer requests, which in addition leads to dissatisfaction with the Company X services.

The work aims to solve the problem faced by Company X. Automating network configuration and management activities for Company X is required to support its business needs and to reduce the amount of time spent on solving complex problems and managing the network infrastructure devices.

This research work presents the current state of network automation tools, techniques and technologies. Based on the review of relevant literature an experimental setup is designed, which simulates a real-life environment using a multi-vendor test bench, modelling an enterprise-size network in order to bring out the practical aspects of automation.

The environment was, therefore, very critical to test the interoperability of the automation tools in the platforms of various network devices, making sure that the proposed solution would manage the network infrastructure effectively and correctly. The core of this project is the development and validation of an environment for automation and monitoring solutions that must be easy to adapt to the complexities of the management and configuration of devices from different vendors. It ensures that far fewer human efforts go to the set-up and gains better uniformity across devices, generally boosting effectiveness in managing the network.

This thesis is a total number of 73 pages, and the entire thesis containing 7 chapters derives its components from 15 figures and 1 table, clearly showing the journey from

identification of the problem to development and implementation of a comprehensive automated network management solution. It provides valuable insights and practical solutions in the field of network administration and automation.

Annotatsioon

Võrgukeskkondade keerukus kasvab ja seadmed on tarnijate vahel mitmekesised, muutub see võrgu konfigureerimise ja haldamise ajal üsna keeruliseks. Hüpoteetiline Ettevõttel X puudub võrgu konfiguratsiooni protsesside automatiseerimise kogemus. Samamoodi nagu Ettevõtte X paljud ettevõtted konfigureeritavad ja haldavad endiselt oma võrguinfrastruktuure käsitsi. See tähendab, et mis tahes seotud konfiguratsioonimuudatuse või kliendi probleemi puhul kulub kõigi seadmete käsitsi konfigureerimiseks või probleemi põhjuste analüüsimiseks lisaaega. Selline lähenemine põhjustab tarbetuid viivitusi klientide soovide rahuldamiseks vajalike meetmete rakendamisel, mis lisaks põhjustab rahulolematust Ettevõte X teenustega.

Töö eesmärk on lahendada ettevõtte X ees seisev probleem. Lahendus peaks võimaldama kasutada võrguseadmete võrgu seadistamise ja haldamise automatiseerimist.

Uuring kirjeldab võrguautomatiseerimise tehnoloogiate hetkeseisu uurimisprotsessi. Kavandatakse uus eksperimentaalne seadistus, mis peaks suutma simuleerida reaalset keskkonda, kasutades reaalset mitme tootja katse stendi, modelleerides ettevõtte suurust võrku, et tuua välja automatiseerimise praktilised aspektid.

Selle projekti tuumaks on automatiseerimis- ja seirelahenduste keskkonna arendamine ja valideerimine, mida peab olema lihtne kohandada erinevate tarnijate seadmete haldamise ja konfigureerimise keerukusega. See aitab saavutada vähem käsitsi konfigureerimisel tehtavaid pingutusi ja parandada seadmete järjepidevust, tõstab üldiselt võrgu haldamise tõhusust.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 73 leheküljel, 7 peatükki, 15 joonist, 1 tabel.

List of abbreviations and terms

CLI	Command Line Interface
HDD	Hard Disk Drive
IaC	Infrastructure as code
LAN	Local area network
NETCONF	Network Configuration Protocol
ROI	Return on investment
SP	Service provider
SNMP	Simple Network Management Protocol
SFP	Small Form-factor Pluggable
SSD	Solid State Drive
UX	User Experience
UI	User Interface
VPN	Virtual private network
VRRP	Virtual Router Redundancy Protocol

Table of contents

List of figures

1 Introduction

Network Configuration and Management Automation is the implementation of provisioning, configuration, management tools on both physical and virtual network devices. Those tasks can range from setting up devices, changing configurations, upgrading software, setting compliance, and optimising networks — all done through software tools and scripts that run with no direct manual intervention.

The main objective of network configuration and management automation is to increase efficiency, reduce possible human errors and ensure organisational policy enforcement and compliance, making configuration and management processes faster in response to what the networks are demanding or changing.

In essence, automation of network configuration is part of modern network management approaches, especially with growing complex and large-scale networks. It ensures that businesses can rapidly roll out new services, adapt to changes, and keep the network performance and security at high levels.

1.1 Problem statement

For instance, as with hypothetical Company X, a lot of companies still configure and manage their network infrastructures manually. Nevertheless, with manual management and configuration, an increase in inefficiency, the possibility for error, and slow responses to changing network requirements are inevitable.

1.2 Research goals or objectives

This thesis will contribute by addressing hypothetical Company X's challenges through automation tools and technologies. The goal is to automate configuration and management tasks at the hardware layer of the network infrastructure, in order to improve efficiency, reduce the ratio of errors and quickly adapt to network changes.

The scope is limited to the available technologies and current industry practices with regard to Company X.

The primary goal of this research work is to develop and demonstrate automated solutions for managing the hardware layer of network infrastructure.

1.3 Objectives of the Research

- To evaluate the current network infrastructure management practices
- To develop test environment which will depict Company X environment
- To automate network configuration and management tasks for Company X
- To showcase the benefits of Network Automation for Company X

1.4 Research questions

- What are the current practices and challenges in network infrastructure management?
- What are the key concepts of network management?
- What are the challenges in network automation in multi-vendor environments?
- How can automated solutions be developed and implemented to improve management of the hardware layer of network infrastructure at Company X?

2 Literature review

Nowadays, companies are looking for ways to increase efficiency, reduce possible human errors and ensure organisational policies. In advance Companies concentrate their attention on making configuration and management processes faster in response to rapid changes and growing data flow.

This section analyses existing tools and technologies which are used for building the proper solution for automating network configuration and management of network devices. Literature review examines the current situation with the proposed options and selects the correct approach for implementing it with the Company X environment. Review of the existing literature provides insights into best practices for network automation solutions.

2.1 Organisation

Company X is a company, which provides LAN networking solutions to meet the clients' needs. Company X services are focused on designing and providing fast and tailor-made solutions based on client requirements even if it uses different networking vendors.

Currently, Company X uses manual configuration methods for making changes in clients' network environments. The manual processes result in spending more time to isolate problems and sometimes it introduces new problems and bugs, which are difficult to locate and solve. In addition, the number of requests from clients to expand local networks has increased recently, which inevitably leads to an increase of repetitive and monotonous work for network administrators. A good opportunity to solve those drawbacks is to implement network configuration. Author is going to do research and find a proper solution to those drawbacks.

2.2 Networking Hardware vendors and networking environment types

Today, the networking market is full of different network sizes and vendors. There are different vendors for the networking equipment, biggest of them are Cisco, Palo Alto Networks, Arista networks [1]. Other notable companies include Juniper Networks, Fortinet, Hewlett Packard Enterprise, and Ubiquiti. Also, there are small players in the market such as MikroTik. It is a good approach to implement automatization on different-sized network vendors to ensure that automatization approach works samely on different equipment.

2.3 Challenges in Manual Network Deployment and Monitoring

Businesses continue to underscore the challenges in manual network deployment and the need for a relook at innovation in the networking field. Even in such a high dependency on manual processes, there is also enough inefficiency and vulnerabilities highlighted. It defines the requirement for these new practices, which provide automation and programmability and ensures the network is able to meet flexibility, reliability, and speed requirements.[2].

The establishment of baselines in network performances in a first step due to challenges posed by manual deployment and monitoring of the networks. These operations do not only validate the changes made to the network but represent the most important step in finding performance problems. In accordance, network management is complex and requires accuracy. Traditional tools do not provide the visibility and flexibility in today's networks, driven by the cloud, and be able to report on effective network performance reporting and management [3].

Intent-based networking (IBN) and autonomous networks (AN) mark a bold step toward solving these challenges. IBN and AN both support the ability to input network intents in natural language, which is further converted into operational network configurations and policies. However, the implementation of the technologies in practice faces problems like accurate translation of intents to configurations of networks and how to make the system transparent and manageable [4].

The move toward DIY network automation is part of an increasing trend where enterprises look toward tailoring network operations to their exact desires: This customises control and comes along with its issues, like the steep learning curve and huge resources required for development and maintenance of that automation solution [5].

In a nutshell, the transition to an automated and intelligent managed network system is very important in getting rid of or at least reducing the negative aspects that come with manual deployment and monitoring of the network. DevOps, IBN, AI, and DIY automation promise to offer organisations more flexible, productive, and robust network operations.

2.4 Cisco equipment automatization approaches

Automating the Cisco network devices through several approaches and automation capabilities, giving network administrators powerful means to enhance efficiency, security, and compliance all across the network.

One popular method for automating Cisco networks is through the Red Hat Ansible Automation Platform. Besides, it is a user-friendly approach, which does not call for any high-level programming skills. The highlighted platform bears a huge number of certified integrations with Cisco products such as IOS®, IOS XE, IOS XR, NX-OS for the device OS, and Cisco ACI, DNA Centre for the SDN controllers. Ansible makes network automation easy. Administrators can automate tasks with humanly readable language to manage a network's configuration in the most diverse environments, hugely accelerating business outcomes and ROI [6].

The IOS-XE Guestshell is an on-box automation feature where network administrators can run Linux operations and open Linux applications at the device itself but within a context that is very secure It provides support for Python and other scripting languages that help in enabling automated execution on the very device, thereby enhancing the flexibility and strength of the automation scripts interfacing with the network directly [7].

Cisco also comes with support for NetConf in its automation protocol, which is attaining a lot of favour due to the standard way of management of the network provided. An augmented multi-vendor API that allows reuse is created over the basic network configuration and management APIs among many vendor APIs that support the protocol, among them being Cisco IOS-XE. It's a standardisation that points right directly towards the future of network automation and offers a way of automating all Cisco devices using any client that can SSH to devices through NetConf's port [8].

For those who want to get more details and examples on how to automate Cisco networks, in-depth, Cisco DevNet offers tremendous resources with tutorials, the latest release notes, and code samples for Ansible automation, located at the Cisco DevNet site. This is an invaluable resource for network engineers who want to take their automation capability with Cisco devices to the next level [9].

The automation of Cisco network devices will be very useful in decreasing labour tasks involving manual configuration and monitoring, improving security and compliance, and working with more efficiency.

2.5 Juniper equipment automatization approaches

Juniper Networks leads the area of network automation with advanced solutions that relieve network planning, design, and operations. Their solutions are anchored in an automation-led approach that assures quality and reliable services within a multivendor environment to make networks predictable and sustainable.

At the centre of Juniper's automation strategy, the Paragon Automation platform tightly couples with Junos OS Evolved on the ACX7000 platform to natively embed "Experience Sensors" for continuous, real-time quality monitoring of the services and application experiences across the network. On one hand, it is a cloud-based solution that allows the SP to begin with the automation of a few selected use cases and take them from there. The intuitive design of Juniper is built naturally with modern UI and UX design principles that help users interact with their solution seamlessly.

Juniper has commercially deployed over 350 solutions of Paragon Automation, providing further proof of the leadership capability presented in multi-vendor, cloud-native SDN, and traffic engineering solutions. This enables easy practice of traffic engineering in the automated designing of network paths and thus enhances network visibility and operational efficiency via real-time monitoring of the network and its capabilities to reroute traffic automatically. Official Juniper Networks Blogs.

Juniper Networks also applies the use of the Ansible IT automation framework for configuring operations and devices within Junos that enable effective running of tasks operational and other configuration duties. Ansible gives users the power to manage JunOS devices for doing a variety of tasks without Python.

The benefits of using Ansible for Junos OS include simplicity, which is rather easy in deployment, has easy-to-learn syntax, and provides acceleration in the deployment of network devices and applications with an efficient automated system. It encourages the standardisation of processes to minimise risks and improves change management, offering a scalable solution for many devices.

Ansible contains modules which will help in creation and running of Ansible playbooks on Junos devices. The playbooks could be developed to perform all types of operational duties and configurations, meaning to structure network management. The playbooks are run on the Ansible control node locally, whereby all the modules communicate back to the Junos devices through the NETCONF to make automation seamless [10][11][12][13][14].

2.6 MikroTik equipment automatization approaches

The MikroTik family of equipment, with many facets, specifically that which runs RouterOS, brings a diverse and more complex landscape in which sits network automation. Its configuration anomalies challenge the clean—and sometimes mean—configurations of its peers, such as Cisco's IOS or JunOS. Such is the case, for instance, of the RouterOS configuration system, which does not make reference to the configuration elements themselves, and identifiers of such elements may change from one command to another, making automation hard. This would require developing the same configuration commands that will provide the same result every time they are applied—an undertaking taking lots of trial and error.

Despite these challenges, Ansible, a powerful IT automation tool, can be leveraged to manage MikroTik devices. Ansible's way to automate MikroTik is via a large community-based collection: community RouterOS. It provides direct access to almost all tasks related to operation and configuration through API or the Command Line Interface. In particular, the configure backup task for the MikroTik RouterOS devices entailed executing tasks over SSH while capturing the output to be saved with device-specific identifiers and timestamps so that it could be stored and restored at a later date when needed. This step is of most importance for Ansible and MikroTik automation beginners with respect to proper communication with RouterOS. In that configuration, it includes the definition of an Ansible user, connection type (network_cli for CLI operations), network OS (RouterOS), and many other required parameters, especially the SSH password.

For MikroTik device preparation, the user needs to create a user account with full privilege and configure the IP addresses and the hostname of the device that will be able to connect through Ansible to manage the device. The capability of Ansible with RouterOS extends further into using the RouterOS API for tasks that need to make direct API calls to supplement the CLI-based automation tasks. That dual approach creates a comprehensive automation strategy that can cover a wide spectrum of management tasks, from simple configuration changes to complex operational workflows [15][16][17].

2.7 Limitations in implementing automatization approaches

There were limitations that may shape the research course and, ultimately, outcomes. These limitations are listed as following:

Diverse Network Environments: The variety of the network devices and their configurations is a massive barrier to developing universally acceptable automation.

Complexity of Implementing Automation in Physical Settings: With the physical nature of the testing environments, the hardware compatibility issues and logistic challenges in replication of diverse network scenarios only made the setup complex.

New tool learning and adaptation: Every new-generation type of network management and automation tools required appreciable efforts put into learning and post-integration learning of the test bed environment for efficient implementation.

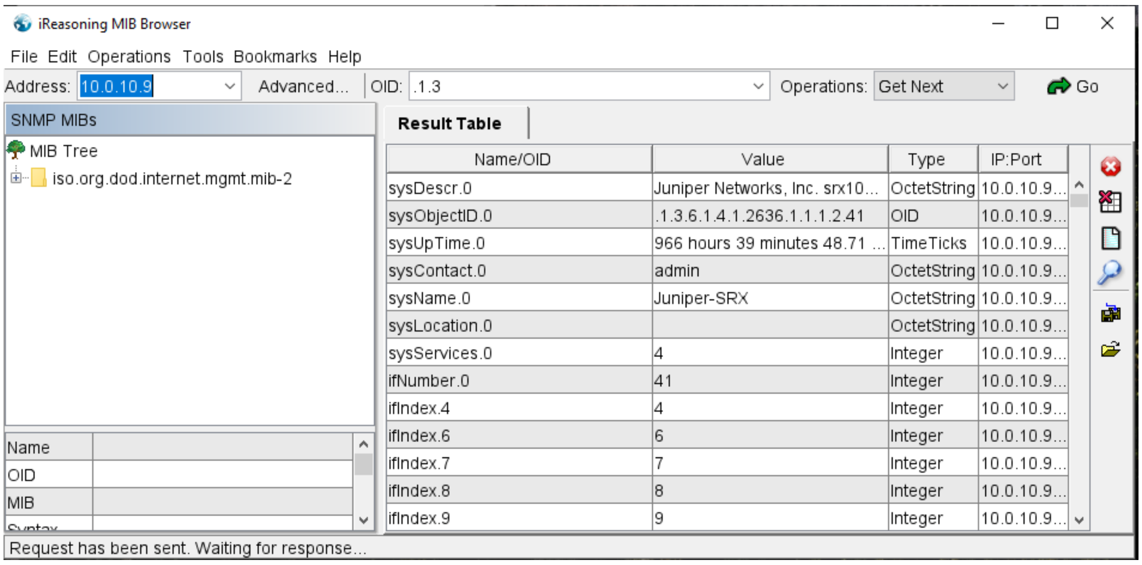
Undocumented networks: This is quite a big issue concerning many reasons. Lack of proper documentation detail makes the comprehension of the existing network infrastructure difficult. Sometimes it is impossible to understand the network infrastructure and the configurations, interconnections, and dependencies with the hardware. Without this core understanding, the design and implementation of automation solutions would otherwise devolve into a process filled with guesswork and possible misalignment from the real needs and configurations existing within the network.

2.8 Tools used for automatization, monitoring and redundancy

In the realm of network management, the deployment of sophisticated tools for automation and monitoring is critical to ensure operational efficiency and reliability. These tools are designed to handle a wide range of tasks from basic monitoring to complex automation of routine tasks, thereby reducing manual labour and increasing the accuracy and responsiveness of network operations.

2.8.1 SNMP usage for management and monitoring

SNMP is an important network monitoring and management protocol, crucial for network engineers and administrators in collecting performance, health, and status data from various network devices such as routers, switches, and firewalls. Configured in a client-server model, SNMP allows not only the monitoring of network devices through agents installed on these devices but also enables remote configuration changes from an SNMP manager. This dual capability supports comprehensive management by the retrieval and modification of device configurations, thereby enhancing network efficiency and adaptability. The collection of metrics, including device uptime, traffic load, and error rates, contributes to a comprehensive view of network health and performance used in automation. SNMP commands enable administrators to query devices for a broad variety of crucial data points, essential for effective monitoring and management. Practical guides on setting up and executing SNMP monitoring and management are available in resources such as Prometheus or Datadog, which offer detailed explanations of SNMP components, commands, metrics, and instructions for real-life scenarios. Moreover, SNMP's flexibility also supports automation, allowing for the scheduling of regular monitoring tasks, which reduces manual overhead and enhances operational efficiency.[18][19]



Name/OID	Value	Type	IP:Port
sysDescr.0	Juniper Networks, Inc. srx10...	OctetString	10.0.10.9...
sysObjectID.0	.1.3.6.1.4.1.2636.1.1.1.2.41	OID	10.0.10.9...
sysUpTime.0	966 hours 39 minutes 48.71 ...	TimeTicks	10.0.10.9...
sysContact.0	admin	OctetString	10.0.10.9...
sysName.0	Juniper-SRX	OctetString	10.0.10.9...
sysLocation.0		OctetString	10.0.10.9...
sysServices.0	4	Integer	10.0.10.9...
ifNumber.0	41	Integer	10.0.10.9...
ifIndex.4	4	Integer	10.0.10.9...
ifIndex.6	6	Integer	10.0.10.9...
ifIndex.7	7	Integer	10.0.10.9...
ifIndex.8	8	Integer	10.0.10.9...
ifIndex.9	9	Integer	10.0.10.9...

Request has been sent. Waiting for response...

Figure 1. SNMP MIB Browser on Juniper device

2.8.2 NETCONF for Network Configuration and Management

NETCONF is a protocol designed to streamline the management and configuration of network devices. Its importance is increasingly recognized in environments where dynamic, scalable, and flexible network management is crucial. NETCONF provides a standardised framework that enables network devices to be managed using a straightforward set of mechanisms. This protocol uses XML as its data encoding format, making it both extensible and suitable for complex configurations. NETCONF's operations allow for the retrieval of configuration data and the application of changes to network device configurations in a secure and predictable manner. NETCONF is designed to work with configuration data stores, which provide a powerful model for managing devices. [20][21]

2.8.3 High availability using VRRP protocol

One of the key networking protocols that enhance high availability is VRRP. When integrated with automation tools like Ansible, VRRP significantly simplifies the management of router redundancy configurations. Using Ansible, network administrators can automate the deployment and configuration of VRRP on multiple routers, ensuring that they operate simultaneously to form a single virtual router or default gateway. This setup increases the network's fault tolerance and availability, providing seamless failover capabilities. In practice, Ansible can be used to dynamically set priorities for routers, decide master elections, and configure standby routers, all through standardised playbooks. This automation not only reduces the potential for human error but also accelerates the deployment and consistency of network configurations across different environments. By leveraging Ansible with VRRP, networks become more resilient and adaptable to changes, with minimal manual intervention, thus aligning with modern automation-focused operational practices.[22][23]

3 Methodology

This section represents the research methods that author has applied in the research on the automation of network configuration and management for the network devices. It details research methods, data collection, and testing environment specifics of the subjects used in the study.

3.1 Research Method

The thesis includes the subject of technology, so the author reviewed and analysed related works. Practical experimentation has been used for literature review validation and conducting this research.

Systematic Literature Review: Author conducted a systematic literature review to establish a theoretical foundation. This involves a review of existing academic articles, technical reports, and industry publications toward various aspects of network automation, tools, frameworks, best practices, etc.

Practical research: This followed an empirical approach to apply the theoretical insights drawn from the literature. Research includes the implementation and testing of developed automation scripts and tools within the network environments. Experiments aim at preparing an environment for measuring capability, efficiency, and scalability of the automatic solutions to manage and configure network devices.

3.2 Data Collection

During writing of this research work author has used these approaches to ensure a comprehensive analysis:

Primary Data Collection: Author interactions directly within the operational network environment through physical setup reflecting real life multi-vendor environments. This is done through a controlled test setup where the network changes are deployed.

Observation and Documentation Analysis: The observation brought in the study covers IT departments to the extent of implementation, including the use of tools that enable automation of network devices configuration and management. Which helped to adopt automation approach besides the fact that the documentation analysis of the logs and the files made by the configuration automation tools was with regard to sourcing common automation tasks, success rates, and error frequency.

Secondary Data Collection: Author gathers secondary data on network automation trends, tool effectiveness, and implementation strategies by reviewing academic industry reports, and documentation on tools such as Ansible, Docker, and GitHub Actions.

3.3 Testing Environment

To assess the appropriate conditions for network automation tool features and capabilities, a test multi-vendor bench was developed as a true multi-vendor environment to measure the automation potential in the network configuration and management. Multi-vendor test bench is set with much precision to look like real models that networks meet in their operations, hence comprising network devices and software from several of the leading vendors. Such setup includes configurations on the routers and switches similar to those used in production environments. Most of the importance was brought to the testing of performance and scalability of the automation solution.

4 Experimental Setup

The experimental setup is used to establish a robust and versatile network infrastructure that represents a different-sized enterprise network. This is scaled down for practical use, allowing for simulating realistic network conditions that could be the basis for validating automated network configuration and management on a physically realistic network. Setup uses multi-vendor equipment, as it helps to implement automatization for large, middle and small size companies in 1 test setup. For the test environment 3 different-sized network equipment vendors were used to try automation solutions for different business sizes.

4.1 Requirements

For the network configuration and management of effective automation within the physical environment of a network, it is fundamental that a very well-planned testbed is being used, emulating the production settings of the testbed is crucial. Experimental setup is exactly the kind of physical environment that could enable controlled, replicable, and safe development for automation of network configuration.

4.1.1 Network Infrastructure

Physical Devices: the range of multi-vendor network devices, including routers, switches should be used to confirm that every part of the LAN network works with other devices found in the network.

Network Topology: the predefined network topology that will mimic the target production environment, and therefore, it should allow a developer to create realistic scenarios of configuration and testing.

Redundancy: Implementation of redundancy should be implemented to test failover and high availability scenarios Figure 1.

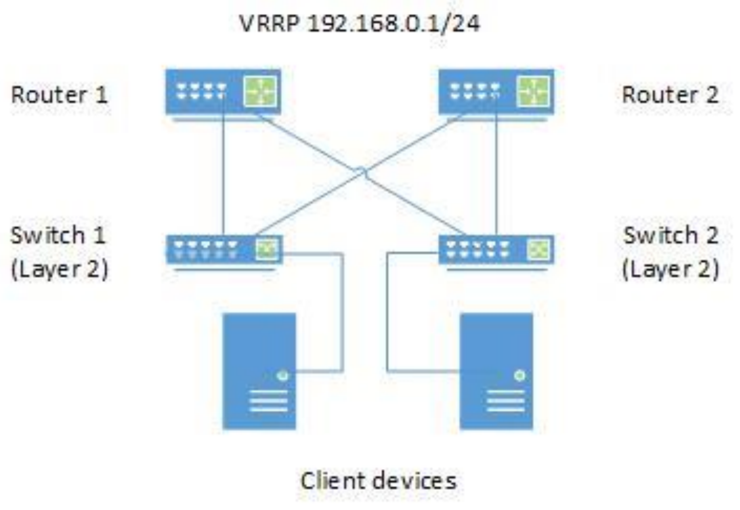
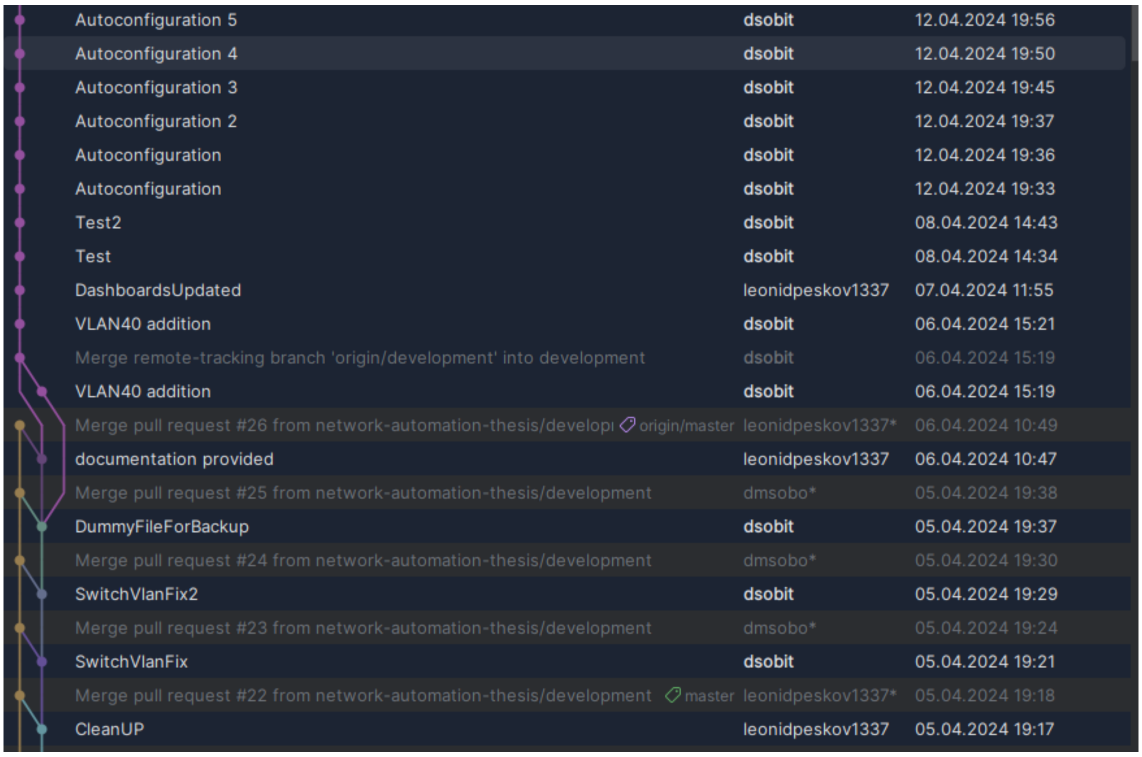


Figure 2. Example of redundant network deployment

4.1.2 Version Control

Scripts and all configuration files should be maintained through a versioning system, using GitHub to track any changes Figure 3. Repositories should be private and not accessible from the outside of the organisation to secure client network setups. Author uses the best practices of configuration management by maintaining all scripts and configuration files within a version-controlled environment using GitHub. This approach ensured that any changes were meticulously tracked and managed, enhancing the reliability and reproducibility of our network setups. Author configured the GitHub repository to be private and restricted access to them from outside the organisation, thereby securing our client network configurations against unauthorised access.



```
Autoconfiguration 5                                                    dsobit              12.04.2024 19:56
Autoconfiguration 4                                                    dsobit              12.04.2024 19:50
Autoconfiguration 3                                                    dsobit              12.04.2024 19:45
Autoconfiguration 2                                                    dsobit              12.04.2024 19:37
Autoconfiguration                                                      dsobit              12.04.2024 19:36
Autoconfiguration                                                      dsobit              12.04.2024 19:33
Test2                                                                  dsobit              08.04.2024 14:43
Test                                                                   dsobit              08.04.2024 14:34
DashboardsUpdated                                                      leonidpeskov1337    07.04.2024 11:55
VLAN40 addition                                                        dsobit              06.04.2024 15:21
Merge remote-tracking branch 'origin/development' into development     dsobit              06.04.2024 15:19
VLAN40 addition                                                        dsobit              06.04.2024 15:19
Merge pull request #26 from network-automation-thesis/develop  origin/master  leonidpeskov1337*  06.04.2024 10:49
documentation provided                                                 leonidpeskov1337    06.04.2024 10:47
Merge pull request #25 from network-automation-thesis/development      dmsobo*             05.04.2024 19:38
DummyFileForBackup                                                     dsobit              05.04.2024 19:37
Merge pull request #24 from network-automation-thesis/development      dmsobo*             05.04.2024 19:30
SwitchVlanFix2                                                         dsobit              05.04.2024 19:29
Merge pull request #23 from network-automation-thesis/development      dmsobo*             05.04.2024 19:24
SwitchVlanFix                                                          dsobit              05.04.2024 19:21
Merge pull request #22 from network-automation-thesis/development  master  leonidpeskov1337*  05.04.2024 19:18
CleanUP                                                                leonidpeskov1337    05.04.2024 19:17
```

Figure 3. Version control using GitHub

4.1.3 Monitoring and Logging

In the experiment setup the author utilised Prometheus and Grafana for monitoring and real-time visibility of network performance. Prometheus was deployed to gather metrics from various network components using SNMP, which allowed him to track everything from bandwidth usage to VLAN state proactively. Grafana is used to visualise these metrics, enabling a clear and dynamic view of network status through dashboards that were updated in real time. Examples of the dashboard can be seen on Figure 14. By integrating Prometheus and Grafana into our monitoring strategy, we ensured continuous oversight and enhanced the reliability of our network systems throughout the experimental phases.

4.1.4 Documentation and Training

All steps of the development process should be with complete documentation of the setup of the test environment including network diagrams, device configurations, and tools used.

4.2 Selection of Network Equipment

The choice of network equipment should be of great importance to develop a testing environment that best corresponds with the real characteristics of the production network. Hence, the selection of the right options would have to come with a balance between functionality, compatibility, performance, and, of course, budgetary constraints. The equipment types to be used is a mix of routers, switches, and computing devices, which together make up an entire network that would be appropriate to test for automation configurations and management.

4.3 Security of the Experimental Setup

The security of the test network environment is of prime importance since this should not only ensure the integrity of the experiments but also be reflective of security considerations that would exist in any real production environment. The following subsection provides the definition of the different security mechanisms and protocols used within our test setup.

4.3.1 Secure Remote Access

OpenVPN or other VPN service: To secure remote access, VPN should be configured to give an encrypted connection for remote management. Such a setting will ensure improved security robustness to build the tunnels over the internet, securing all exchanges.

4.3.2 Secure Device Management

Secure Connection to Devices using SSH: All connections to the devices in the network are secured using SSH with encrypted traffic that consists of both the configuration commands and management device traffic. To add more security to the connection, the risks associated with login passwords are eliminated through key-based authentication.

4.3.3 Ansible Controller Security

The Ansible controller is located inside the same network to avoid crossing potentially insecure networks unnecessarily every time a device is to be configured. This proximity guarantees that configuration-related sensitive information and commands will be used to stay within the secured network perimeter.

4.3.4 Monitoring and Auditing

Prometheus and Grafana as primary monitoring tools which the author used to oversee network activities effectively. Prometheus is adept at collecting and storing metrics from various network components, which facilitates the detection of unusual patterns that might indicate security breaches or operational anomalies. Additionally, it supports the monitoring of standard access logs and audits of configuration changes, enhancing our security posture. Grafana, on the other hand, excels in visualising these metrics through intuitive dashboards, providing us with real-time insights and alerts. This combination not only helps in identifying potential security issues but also in maintaining continuous oversight of network health and performance.[25][26].

4.3.5 Limiting Access

The author is using tools to adequately administrate the control of the access to the test environment in such a way that only relevantly authorised personnel would have the necessary privilege to change or initiate the testing cases. Such minimises, therefore, the risk of unauthorised modifications and potential breaches[25].

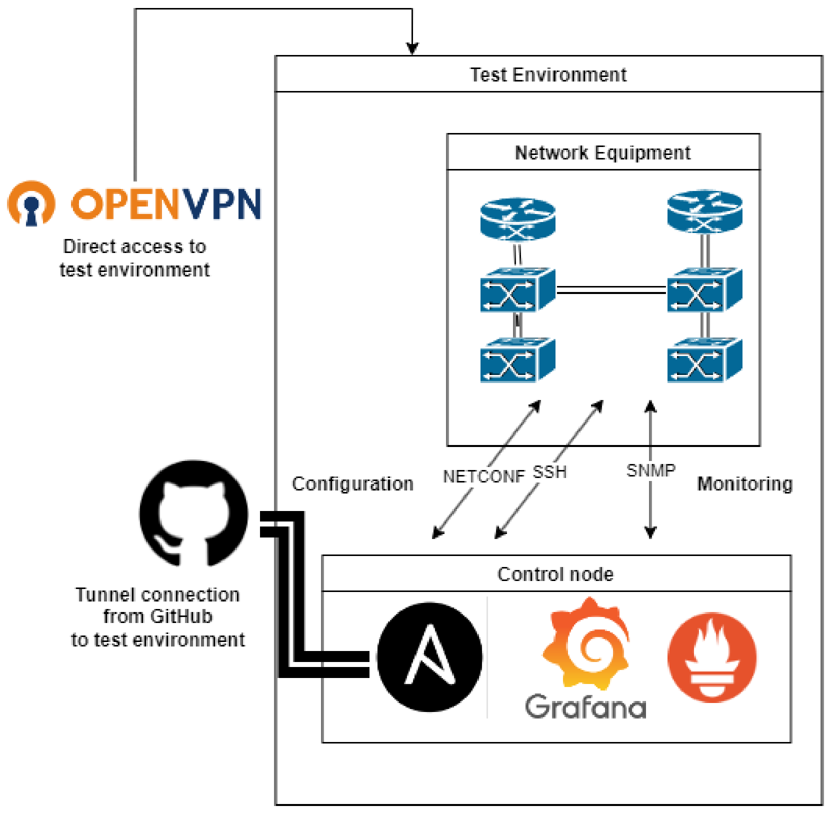


Figure 4. Test Environment component scheme

4.4 Justification for VPN Service Selection

VPNs play a crucial role in ensuring secure access to organisational network resources, particularly for configurations managed remotely. In selecting an appropriate VPN service for the author's experimental setup, the author prioritised criteria such as security features, performance, ease of integration, and compatibility with our existing network infrastructure. After evaluating several VPN solutions, OpenVPN emerged as the optimal choice due to its robust security protocols, open-source flexibility, and strong performance metrics. OpenVPN’s ability to securely connect to the author's experimental network using MikroTik devices played a crucial role in choosing a VPN solution. This decision ensures that remote access remains both secure and efficient, supporting the integrity and confidentiality of our data transmissions.

4.5 Infrastructure as Code (IaC) Implementation

This experimental setup would use Infrastructure as Code (IaC) to achieve automated and consistent configuration of the network. Considering infrastructure setup and configurations to be like software, with the help of automation tools like Ansible and GitHub for version control and CI/CD, it means most of the configurations and scripts will be deployed and managed automatically by Company X.

4.5.1 Ansible for Automated Configuration Management

Ansible is used as IaC tool, simply based on its agentless architecture, ease of use, and the broad community behind it. Ansible Playbooks use a very simple YAML configuration language to code the desired state of the network infrastructure.

Automated Configuration: Deployment is automatic and fast, making it possible to deploy the configuration to many devices quickly, with the possibility of the number scaling up as the network expands.
Change Management: Track and manage changes to the network configurations, ensuring all the changes are with intention and documentation for them.
Idempotency: Maintains that the redeploying of scripts does not create any side effects while maintaining the desired state of the network.

Ansible brings a different approach to the implementation, monitoring, and support of network configuration. Ansible's powerful IT automation engine which checks consistency over network infrastructures and does not bother with the complexity of tasks. Ansible is simple to use, without an agent, IT automation engine for provisioning and configuration management of network devices, and it natively interacts with network devices via SSH, ensuring no agent resides in devices, therefore reducing network overhead.

```
hostname {{ cisco_hostname }}
!
interface Vlan1
no ip address
shutdown
!
{% for vlan in cisco_vlans %}
vlan {{ vlan.id }}
name {{ vlan.name }}
!
{% endfor %}
```

Figure 5. Example Jinja2 Template for Cisco

This part of code shows how those VLAN configurations can be tempted to apply the same settings across your different network segments automatically.

4.5.2 GitHub for Version Control and CI/CD

GitHub serves as a collaborative platform for the storage of Ansible playbooks and other artefacts of IaC with version control and teamwork of the files by the team.

Author actively used GitHub as the central platform for managing Ansible setup and playbooks. This setup allowed the author to implement GitHub's robust version control capabilities to facilitate teamwork and ensure consistent updates to all files. Furthermore, the author utilised GitHub Actions for CI/CD pipeline, enabling automated testing and continuous deployment of network configurations. Every change underwent rigorous pre-deployment testing using Batfish to identify and resolve issues early. This integration provided a reliable version trail for tracking changes.

4.5.3 Enhancing Service Quality

Service quality is among the targets of the company X—an overall that would be otherwise used for highly strategic activities when minimising the risk of human errors and releasing valuable engineering time. Consistency is maintained through its network configurations from the experiment environment, through to the stage, and into software projects.

4.5.4 Preparing for Production

The use of Ansible and GitHub in the testing environment is just a forerunner—an indication of a wider adoption in the production environment. Implementing IaC through Ansible and GitHub is a strategic experiment befitting Company X's vision of a scalable, resilient, and efficiently managed network. Such a methodological shift to automation will, for sure, be an investment in the future as it will drastically enhance the agility and responsiveness of the network operations within the company. Furthermore, this transition underscores the necessity to continually elevate the skill of network engineers, ensuring they are well acknowledged in the latest technologies such as GitOps tools.

5 Implementation

This part details the implementation of an experimental network environment in terms of physical setup, configuration management, security concerns, and tools used for automation and remote access.

5.1 Physical setup

This chapter provides an overview of the automatisation solution implementation workflow.

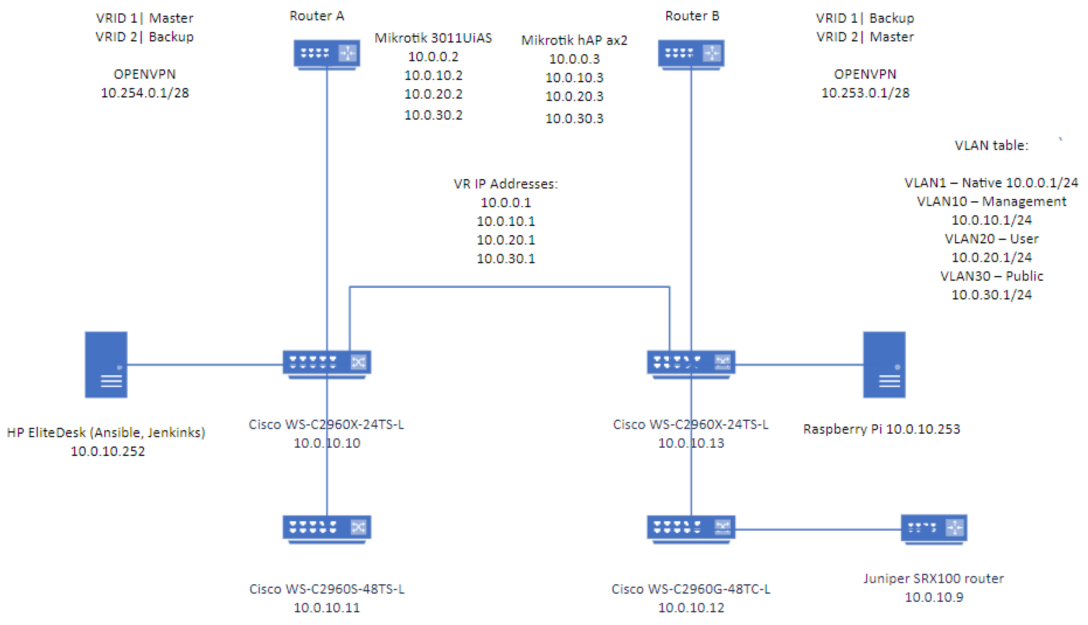


Figure 6. Network scheme

As it is depicted in Figure 5 there are different vendor equipment such as: Cisco, MikroTik, and Juniper, which are widely used in SP and customer networks industry settings.

Table 1. Company X Test setup

Device Type	**Model**	**IP Addresses**	**Role**
Control Node	HP - EliteDesk-800-G2-DM-65W	10.0.10.252/24	Central management PC runs, Ansible, Prometheus, and Grafana on the board. System runs on fast SSD. Also provides backup ability on the HDD 1TB.
MikroTik Routers	MikroTik - 3011UiAS	10.0.0.2/24 10.0.10.2/24 10.0.20.2/24 10.0.30.2/24	Main router on the setup with VRRP ability to provide redundant routing between subnets and connection with the network.
	MikroTik - hAP-ax2	10.0.0.3/24 10.0.10.3/24 10.0.20.3/24 10.0.30.3/24	Redundant router on the setup with VRRP ability to provide redundant routing between subnets and connection with network.
Cisco Switches	Cisco - WS-C2960X-24TS-L	10.0.10.10/24	Access layer switches, that interconnect devices in network
	Cisco - WS-C2960S-48TS-L	10.0.10.11/24	
	Cisco - WS-C2960S-48TC-L	10.0.10.12/24	

Device Type	**Model**	**IP Addresses**	**Role**
	Cisco - WS-C2960X-24TS-L	10.0.10.13/24	
Juniper Router	Juniper SRX100	10.0.10.9/24	Juniper router, which is used as a switch, to provide ability to test JunosOs automatization setups
Raspberry pi		10.0.10.251/24	Used in test setup for Linux endpoint/ IoT compatibility check

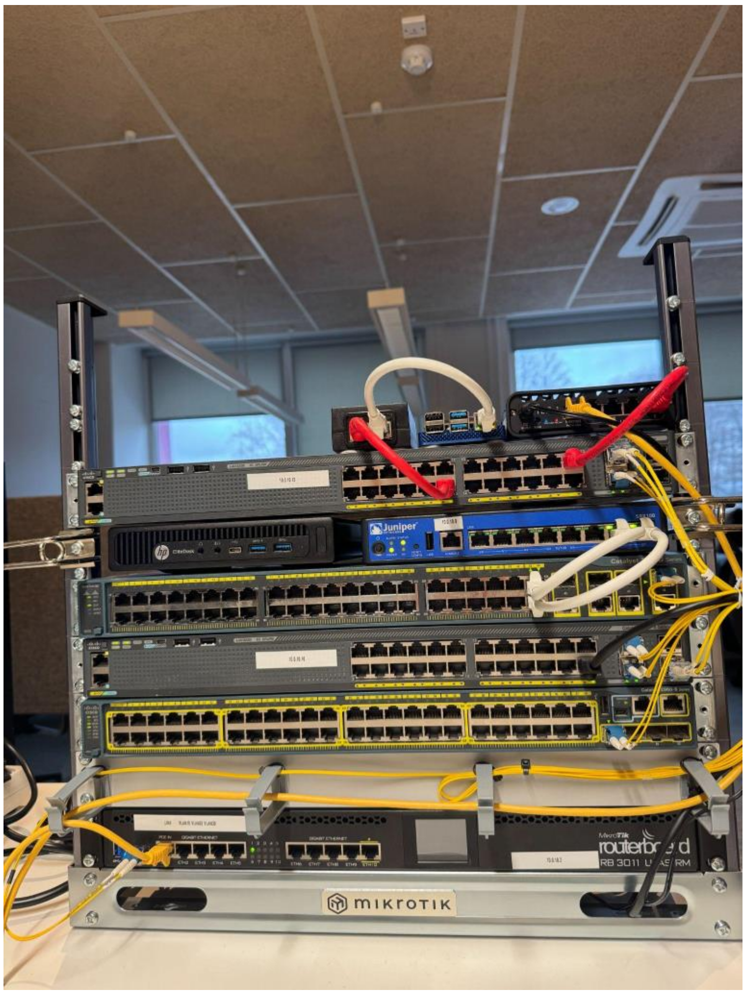


Figure 7. Physical setup

5.2 Routers

MikroTik 3011UiAS (Router A) and MikroTik hAP ax2 (Router B): These routers are selected because they have really strong performance metrics and their price is absolutely affordable. Virtual Router Redundancy Protocol (VRRP), a part of advanced routing support on the routers, was possible. Such advanced routing capabilities will need to be present on the routers for testing essential failover mechanisms while maintaining high availability. Its varied interface options provide for the extensive configuration of VLANs useful in the division of the network into management, user, and public VLANs.

Juniper SRX100: It brings a different vendor perspective into the network and allows vendor diversity to test the universality of automation scripts. The series delivers security features of SRX100 with a strong set of characteristics that come from Junos OS capabilities and commands, but are the same from device to device.

5.3 Switches

Cisco WS-C2960X Series: The switches of this series belong to the industry standard, recognized for reliability with support for a rich set of security functions.

They are integrated with tools like Ansible for automation, and their documentation is very good, with a big community support, so the decision for the author to merge them in our environment was kind of easy for troubleshooting and configuration.

The basic configuration of the Cisco switches involved:

Service Timestamps: Both debug and log entries are timestamped for accurate troubleshooting and logging purposes.

Password Encryption: The service password-encryption command is used to encrypt all plaintext passwords in the configuration, enhancing security.

Hostname: Each switch is given a unique hostname, for easy identification within the network.

SSH Configuration: SSH version 2 is enabled to provide secure administrative access over the network.

VLANs Setup: VLANs 10, 20, 30 are created with descriptive names to segregate network traffic for management (MNG), user (USER), and public (PUBLIC-NETWORK) access.

Trunk Ports: Specific Gigabit Ethernet ports are configured as trunk ports to allow traffic from multiple VLANs to traverse the links between switches.

Management Interface: VLAN 10 is assigned an IP address for switch management purposes.

Default Gateway: The default gateway is set, allowing the switch to communicate with devices outside its subnet.

Remote Access: VTY lines are configured to use local credentials and allow SSH access, with an idle timeout set for security.

Banner Message: A banner message is set to display a notice upon login, which can serve as a legal notice or a simple warning to unauthorised users.

SNMP Community: An SNMP community is configured for read-only access, enabling the monitoring of switch performance and status.

EtherChannel: EtherChannel is configured on the devices which in future can need to have a faster internet connection.

5.4 Computing Devices

HP EliteDesk (Ansible, Jenkins) - Chosen hardware specification: HP EliteDesk enterprise-level desktop as the automation server, hosting applications like Ansible for configuration management, Jenkins for continuous integration and delivery, etc. It has the expected power of computing that handles multiple automation at a time without degradation in performance.

Raspberry Pi: The reason includes Raspberry Pi in the network for dual purposes; it can serve as both a lightweight, low-power computing device to emulate network endpoints and IoT devices. At the same time, it can serve as an inexpensive testbed for execution of such scripts against networked, automated scripts targeting such devices.

5.5 OpenVPN Implementation on MikroTik Routers

In a real environment, secure, encrypted remote access maintenance is crucial. So in a test environment, which represents reality, it is also necessary. The author has developed a test setup solution with effective and efficient implementation of OpenVPN over MikroTik Routers. On the flip side, OpenVPN is rather flexible, since it has strong standards and encryption compatibility with MikroTik. The configuration yields VPN tunnels with different subnets set for mutual segregation and further enhances access control.

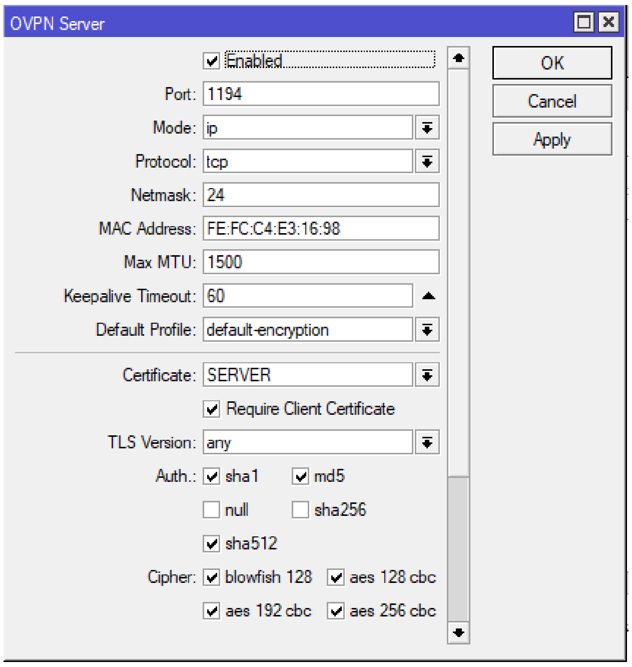
5.5.1 OpenVPN Configuration Overview

OpenVPN servers with the encrypted connections were configured to run over MikroTik routers. This was based on the OpenVPN setup interfaces on the routers, listening to some specified ports, at the same time creating secure tunnels for remote connections. The configuration for OpenVPN servers could be reviewed on Figure 5 and Figure 6.

Subnet Allocation: Two distinct subnets were designated for VPN connections with smaller pool of addresses:

10.254.0.1/28 for access from Router A, such that management and configuration tasks can be securely performed over VPN.

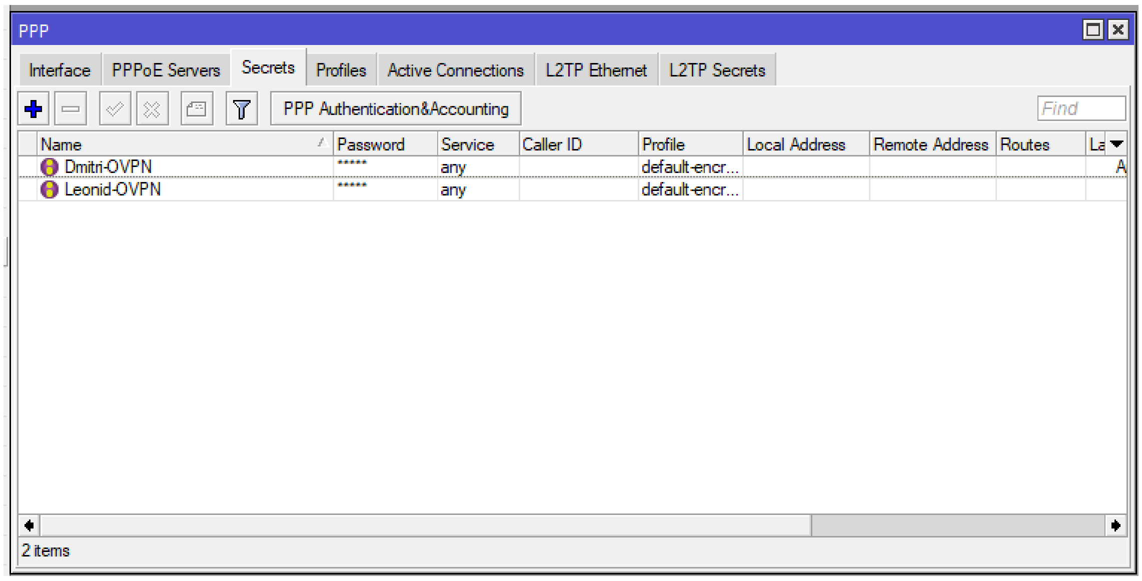
10.253.0.1/28 - on Router B safe provision for general user access to the network, also it is used as a redundant connection for network management in case Router A stops its working.


OVPN Server

[✓] Enabled

Port:	1194
Mode:	ip
Protocol:	tcp
Netmask:	24
MAC Address:	FE:FC:C4:E3:16:98
Max MTU:	1500
Keepalive Timeout:	60
Default Profile:	default-encryption
Certificate:	SERVER
	[✓] Require Client Certificate
TLS Version:	any
Auth.:	[✓] sha1 [✓] md5 [ ] null [ ] sha256 [✓] sha512
Cipher:	[✓] blowfish 128 [✓] aes 128 cbc [✓] aes 192 cbc [✓] aes 256 cbc

Figure 8. OpenVPN setup



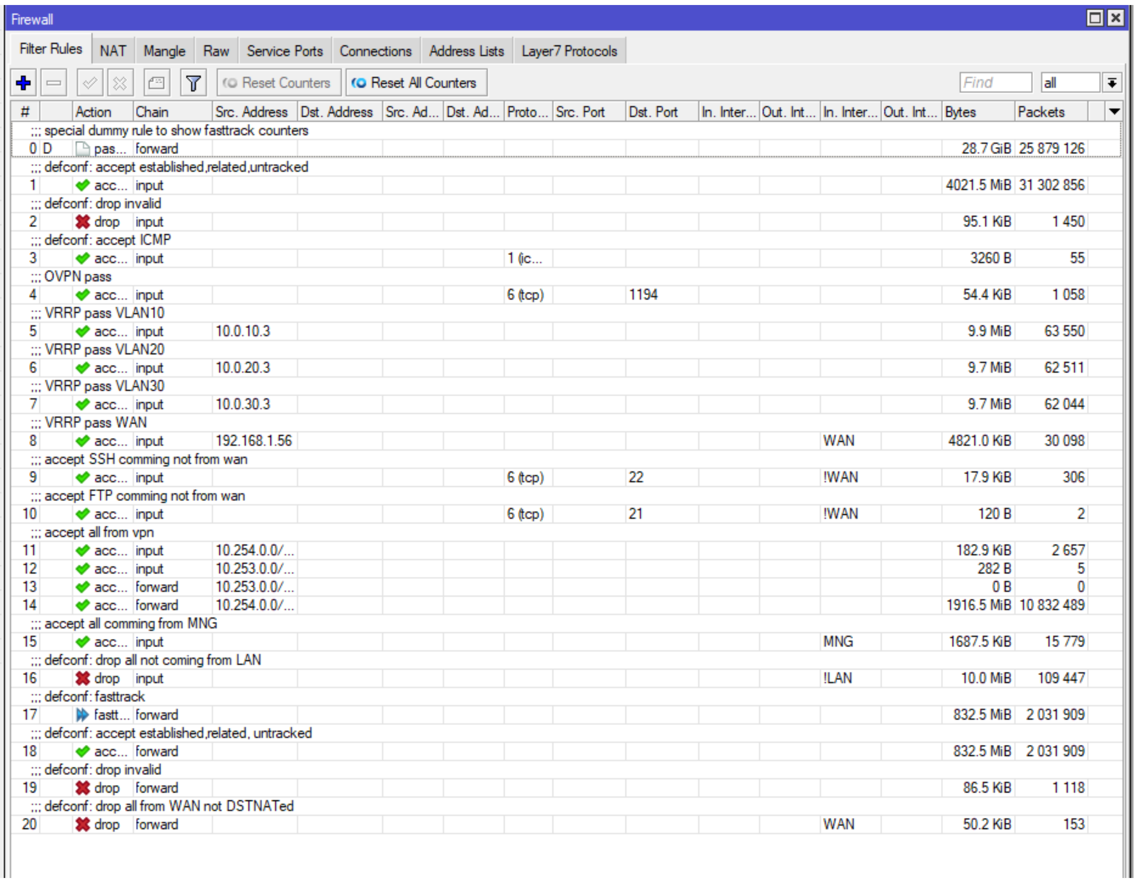
Name	Password	Service	Caller ID	Profile	Local Address	Remote Address	Routes	La
Dmitri-OVPN	*****	any		default-encr...				A
Leonid-OVPN	*****	any		default-encr...				

2 items

Figure 9. Users for VPN connections

5.5.2 Authentication and Access Control

Certificates and keys: The CA (Certificate Authority) was implemented with the capability of issuing certificates and keys to the server and clients. This ensures strong user/device authentication further, since only the devices or users bearing the right credentials will be in a position to establish a VPN connection. Specific firewall rules and access policies as on Figure 7 are also configured in VPN to control and restrict the given access; this is based on the least privilege principle of allowing users and devices access only to the resources required by their roles.



Firewall

Filter Rules | NAT | Mangle | Raw | Service Ports | Connections | Address Lists | Layer7 Protocols

Reset Counters | Reset All Counters | Find | all

#		Action	Chain	Src. Address	Dst. Address	Src. Ad...	Dst. Ad...	Proto...	Src. Port	Dst. Port	In. Inter...	Out. Int...	In. Inter...	Out. Int...	Bytes	Packets
;;; special dummy rule to show fasttrack counters																
0	D	pas...	forward												28.7 GiB	25 879 126
;;; defconf: accept established,related,untracked																
1		acc...	input												4021.5 MiB	31 302 856
;;; defconf: drop invalid																
2		drop	input												95.1 KiB	1 450
;;; defconf: accept ICMP																
3		acc...	input					1 (ic...							3260 B	55
;;; OVPN pass																
4		acc...	input					6 (tcp)		1194					54.4 KiB	1 058
;;; VRRP pass VLAN10																
5		acc...	input	10.0.10.3											9.9 MiB	63 550
;;; VRRP pass VLAN20																
6		acc...	input	10.0.20.3											9.7 MiB	62 511
;;; VRRP pass VLAN30																
7		acc...	input	10.0.30.3											9.7 MiB	62 044
;;; VRRP pass WAN																
8		acc...	input	192.168.1.56									WAN		4821.0 KiB	30 098
;;; accept SSH comming not from wan																
9		acc...	input					6 (tcp)		22			!WAN		17.9 KiB	306
;;; accept FTP comming not from wan																
10		acc...	input					6 (tcp)		21			!WAN		120 B	2
;;; accept all from vpn																
11		acc...	input	10.254.0.0/...											182.9 KiB	2 657
12		acc...	input	10.253.0.0/...											282 B	5
13		acc...	forward	10.253.0.0/...											0 B	0
14		acc...	forward	10.254.0.0/...											1916.5 MiB	10 832 489
;;; accept all comming from MNG																
15		acc...	input										MNG		1687.5 KiB	15 779
;;; defconf: drop all not coming from LAN																
16		drop	input										!LAN		10.0 MiB	109 447
;;; defconf: fasttrack																
17		fastt...	forward												832.5 MiB	2 031 909
;;; defconf: accept established,related, untracked																
18		acc...	forward												832.5 MiB	2 031 909
;;; defconf: drop invalid																
19		drop	forward												86.5 KiB	1 118
;;; defconf: drop all from WAN not DSTNATed																
20		drop	forward										WAN		50.2 KiB	153

Figure 10. Firewall rules (In code: Appendix 8)

5.5.3 Integration with Network Infrastructure

The setup integrates easily into an existing network infrastructure and allows for remote access with the least or no reconfiguration of the network. VPN access from the defined subnets was routed inside the network to allow users to be able to reach the required services and resources in a fast and secure manner.

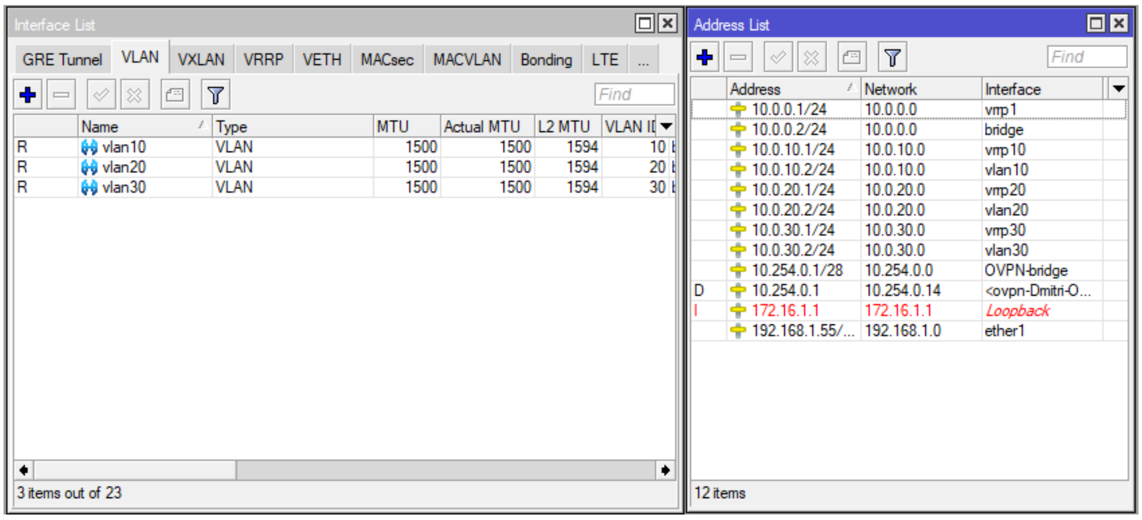
5.5.4 Challenges and Solutions

On the implementation challenge, it included the configuration of OpenVPN with the least cost and at optimal performance on MikroTik routers, while making it compatible with a wide range of client devices. This called for detailed configuration tests, adjustment of the firewalls, and support of deployment and troubleshooting with the development of comprehensive documentation.

OpenVPN is used on the MikroTik routers to make sure that the necessary, secure, and encrypted channel for remote administration and access to the lab environment is provided. In this, the experimental setup is configured with utmost care in the allocation of subnets, with rigid security policies and encryption policies, and also by dovetailing the current network infrastructure with the VPN setup. This VPN configuration is meant to not only support the current test environment requirements but also delineate the scope for further development and integration of these needs within a scalable and secure framework to be used in production environments.

5.6 VLAN and IP Addressing Scheme

VLAN and IP addressing follow an enterprise divisional practice Figure 8. It allows trying out and testing of focused automation scripts with respect to each segment without liabilities of unintended consequences for the entire network.



	Name	Type	MTU	Actual MTU	L2 MTU	VLAN II
R	vlan10	VLAN	1500	1500	1594	10
R	vlan20	VLAN	1500	1500	1594	20
R	vlan30	VLAN	1500	1500	1594	30

3 items out of 23

	Address	Network	Interface
	10.0.0.1/24	10.0.0.0	vrrp1
	10.0.0.2/24	10.0.0.0	bridge
	10.0.10.1/24	10.0.10.0	vrrp10
	10.0.10.2/24	10.0.10.0	vlan10
	10.0.20.1/24	10.0.20.0	vrrp20
	10.0.20.2/24	10.0.20.0	vlan20
	10.0.30.1/24	10.0.30.0	vrrp30
	10.0.30.2/24	10.0.30.0	vlan30
	10.254.0.1/28	10.254.0.0	OVPN-bridge
D	10.254.0.1	10.254.0.14	<ovpn-Dmitri-O...
I	172.16.1.1	172.16.1.1	*Loopback*
	192.168.1.55/...	192.168.1.0	ether1

12 items

Figure 11. VLAN and IP Addressing Scheme

5.7 Implementation of Ansible for Automating Network Configuration and Management

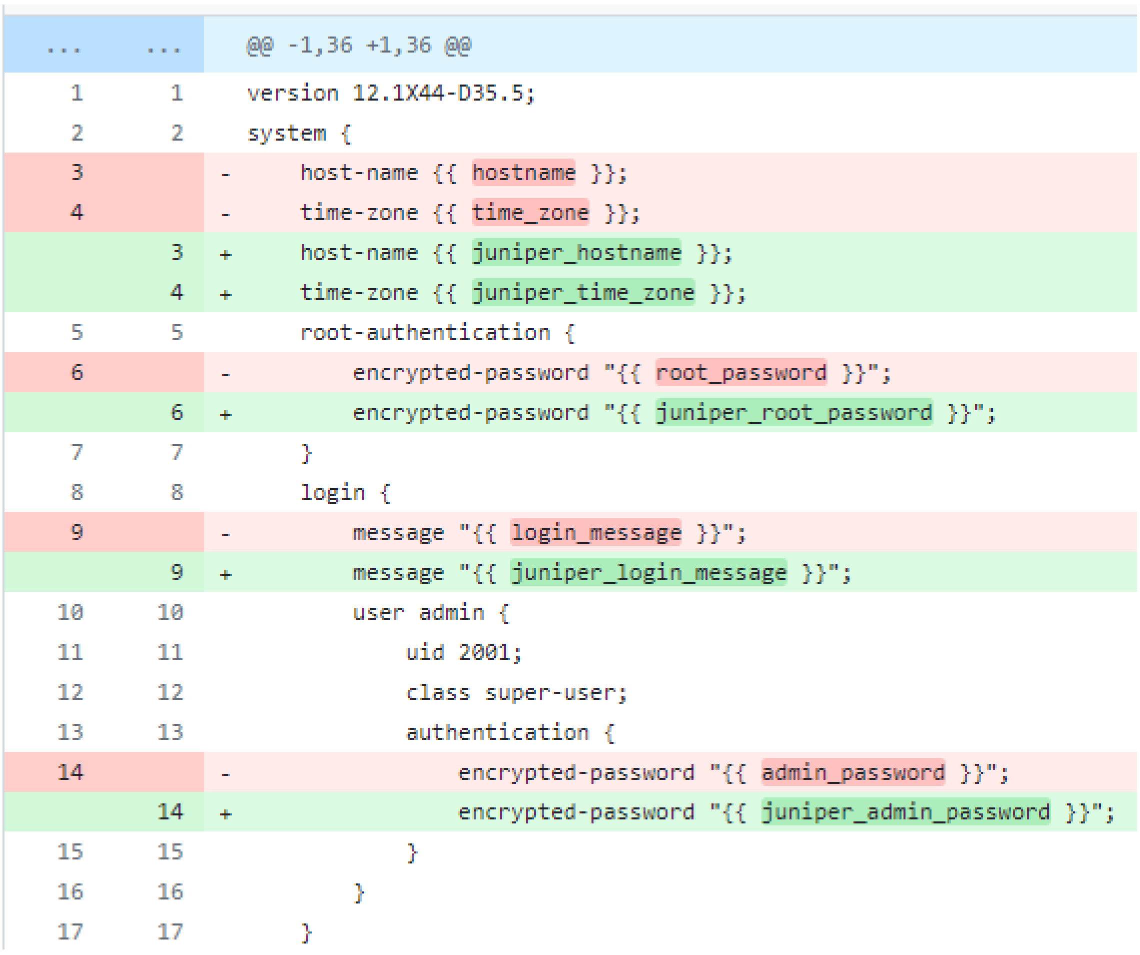
Building on the foundation laid out in section 4.5.1, this section delves into the practical implementation of Ansible for automating network configuration and management tasks. Ansible's role as a robust Infrastructure as Code (IaC) tool is instrumental in transforming how network devices are configured and managed in a dynamic and scalable environment. The following implementation details encapsulate the processes and methodologies employed to automate tasks, ensuring network configurations are consistent, efficient, and error-free.

5.7.1 Automated Configuration Deployment

Using Ansible playbooks, the deployment of network configurations has been automated to cater to an array of devices simultaneously. This is facilitated through the use of modular tasks defined in YAML. For instance, the deployment of security enhancements and VLAN configurations across Cisco switches is executed swiftly and uniformly. This allows for the rapid scaling of network infrastructure as Company X expands, without the need for manual intervention. The efficiency of playbook runs ensures that network configurations can be rolled out quickly and with fewer human errors. The example playbook provided in Appendix 3 showcases the structured approach in automating Cisco switch configurations, from VLAN setups to security settings.

5.7.2 Change Management and Documentation

Ansible also aids in change management by tracking modifications made to network configurations. This ensures all changes are intentional and documented thoroughly. For effective documentation, changes made via playbooks are uploaded to GitHub and version control of GitHub will allow developers to ensure the environment has a rollback point if necessary. This process helps maintain the integrity and reliability of network operations within Company X.



```diff
@@ -1,36 +1,36 @@
 version 12.1X44-D35.5;
 system {
-    host-name {{ hostname }};
-    time-zone {{ time_zone }};
+    host-name {{ juniper_hostname }};
+    time-zone {{ juniper_time_zone }};
     root-authentication {
-        encrypted-password "{{ root_password }}";
+        encrypted-password "{{ juniper_root_password }}";
     }
     login {
-        message "{{ login_message }}";
+        message "{{ juniper_login_message }}";
         user admin {
             uid 2001;
             class super-user;
             authentication {
-                encrypted-password "{{ admin_password }}";
+                encrypted-password "{{ juniper_admin_password }}";
             }
         }
     }
```

Figure 12. Log of changes in GitHub

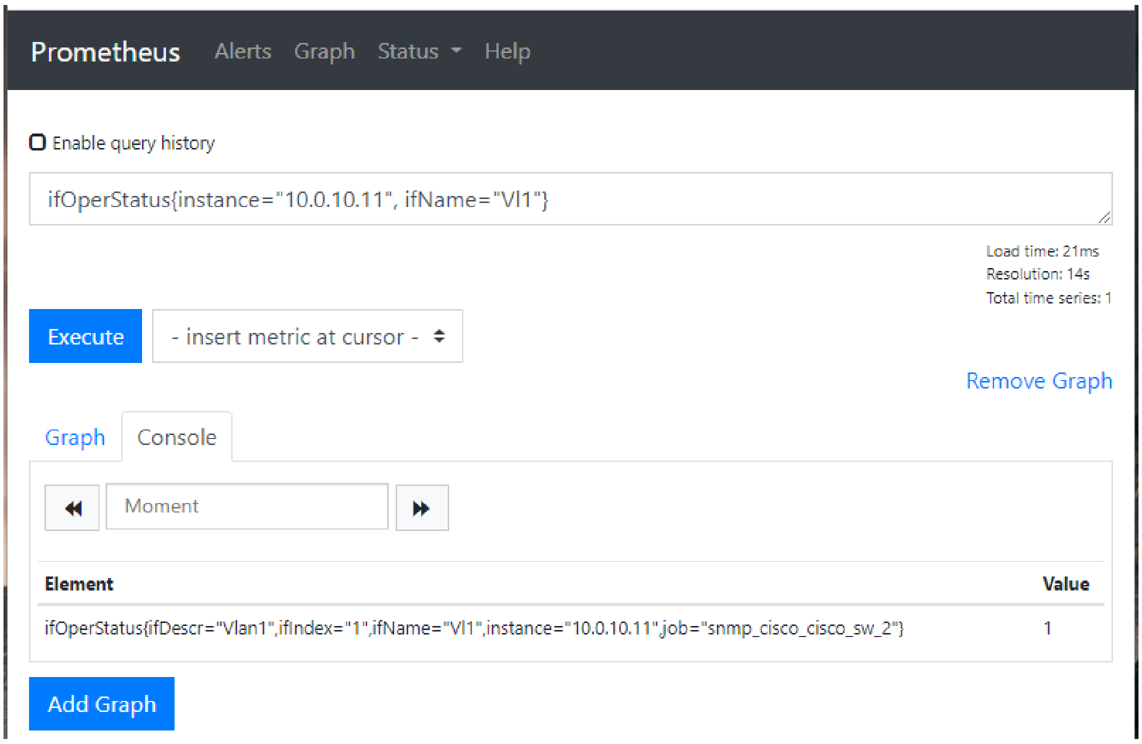
A key feature of Ansible's approach to automation is idempotency; the ability to run the same configuration multiple times without affecting the network state beyond the initial application ensures consistency across the network. This principle is crucial in maintaining long-term consistency and reliability across network deployments.

5.7.3 Real-World Application: VLAN Configuration

Consider the task of updating VLAN settings across multiple switches. Using Jinja2 templates integrated with Ansible playbooks, configurations can be templated and applied uniformly, reducing the risk of discrepancies. An example of such a template is provided in Appendix 3, demonstrating how VLANs are configured across different segments of the network, ensuring operational consistency and efficiency. Detailed examples of these templates can be found in the appendices: see Appendix 2 for the Juniper configuration and Appendix 3 for the Cisco configuration.

5.7.4 Monitoring and Validation

Post-deployment, Ansible's role extends to ongoing monitoring and validation of network configurations. Using Prometheus and Grafana (set up by a third party), network administrators can retrieve and report on current device states, comparing them against the desired configurations defined in playbooks. This ensures that the network remains in compliance with the defined standards and can quickly address deviations.

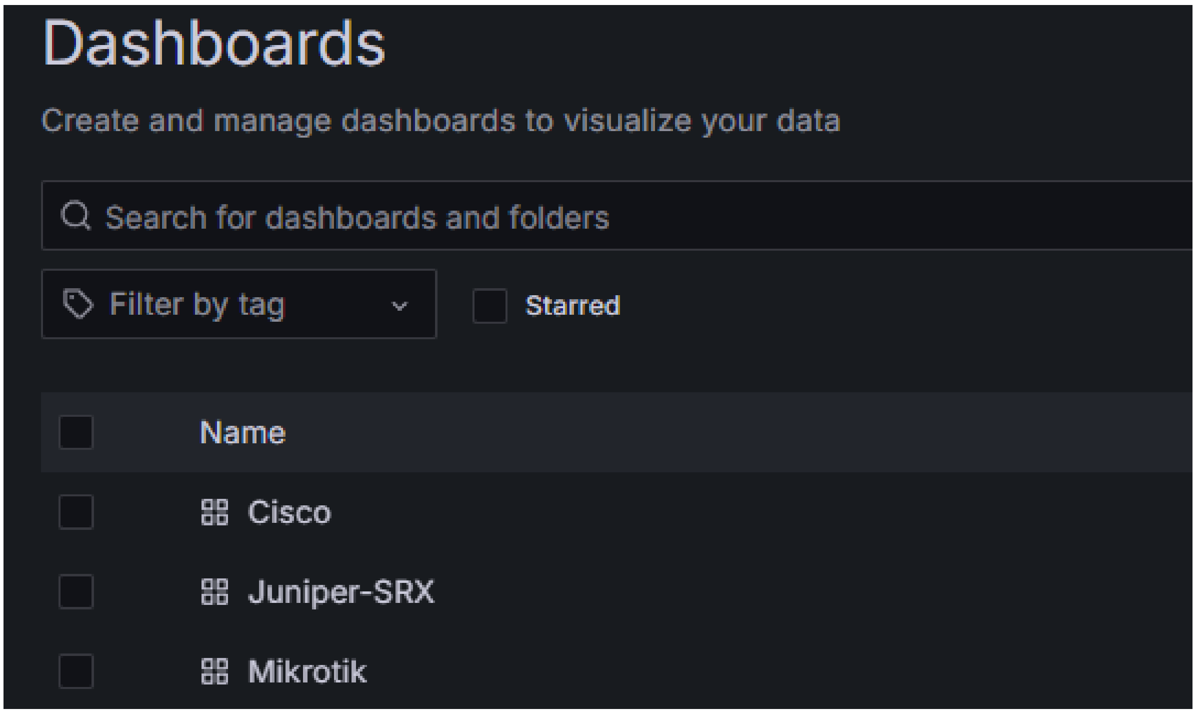


```
ifOperStatus{instance="10.0.10.11", ifName="Vl1"}
```

Load time: 21ms
Resolution: 14s
Total time series: 1

Element	Value
ifOperStatus{ifDescr="Vlan1",ifIndex="1",ifName="Vl1",instance="10.0.10.11",job="snmp_cisco_cisco_sw_2"}	1

Figure 13. Prometheus metrics page



Dashboards

Create and manage dashboards to visualize your data

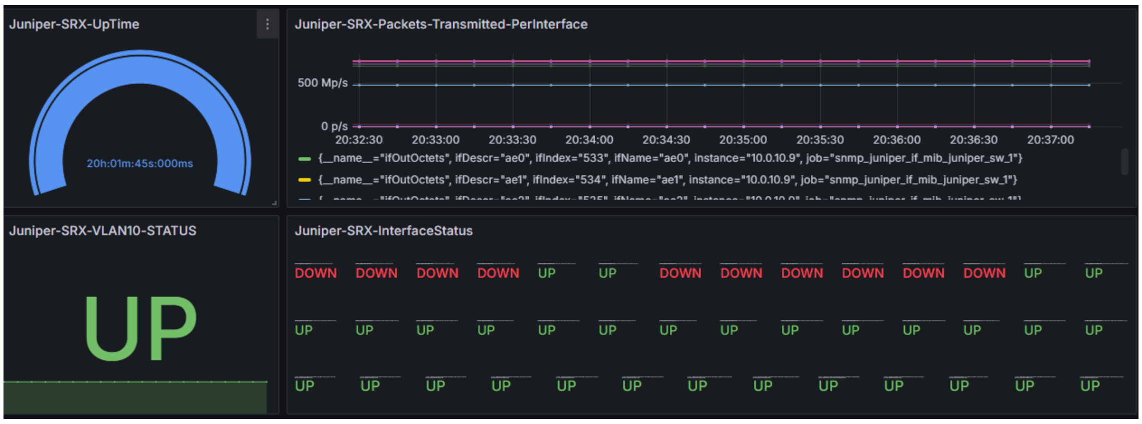
Search for dashboards and folders

Filter by tag

Starred

Name
Cisco
Juniper-SRX
Mikrotik

Figure 14. Dashboards in Grafana



Juniper-SRX-UpTime
20h:01m:45s:000ms
Juniper-SRX-Packets-Transmitted-PerInterface
500 Mp/s
0 p/s
20:32:30
20:33:00
20:33:30
20:34:00
20:34:30
20:35:00
20:35:30
20:36:00
20:36:30
20:37:00
Juniper-SRX-VLAN10-STATUS
UP
Juniper-SRX-InterfaceStatus
DOWN DOWN DOWN DOWN UP UP DOWN DOWN DOWN DOWN DOWN DOWN UP UP
UP UP UP UP UP UP UP UP UP UP UP UP UP UP
UP UP UP UP UP UP UP UP UP UP UP UP UP

```
{__name__="ifOutOctets", ifDescr="ae0", ifIndex="533", ifName="ae0", instance="10.0.10.9", job="snmp_juniper_if_mib_juniper_sw_1"}
{__name__="ifOutOctets", ifDescr="ae1", ifIndex="534", ifName="ae1", instance="10.0.10.9", job="snmp_juniper_if_mib_juniper_sw_1"}
```

Figure 15. Grafana Juniper device dashboard

6 Results and Discussion

The implementation of Ansible and the configuration and management of network devices at hypothetical Company X have been automated, as per the laid down objectives in Section 1.2 of this paper. Also the test setup depicts the whole network process. As the network scales, such capabilities will further be extended to embrace, for example, even more advanced Ansible capabilities like the ability to tune performance through modules and reach into more sophisticated network scenarios for automation coverage.

Ansible features, exploring additional modules for performance tuning, and expanding automation to implement more complex network scenarios, including those involving MikroTik devices. Unlike Cisco and Juniper, MikroTik devices do not have official Ansible modules that can directly manage configurations, so this is a challenge for the future. The author's recommendation is described in the summary section. Batfish is used for checking uploaded configurations into GitHub and if misused commands are detected it reports about it.

Utilising GitHub as version control mechanism reduces the risks associated with direct modifications on live devices and ensures that any deployment to network devices maintains the high standards of accuracy and reliability needed in a multi-vendor network setup.

7 Summary

This work investigated the automation of network configuration and management for the devices in the network in a physical testing and real environment. The thesis aim was to increase the efficiency of operations, which will reduce the margin of human error and ensure that practices in network management are consistent. The thesis has thoroughly shown the detailed investigation of existing automation opportunities to prepare the environment for its use. Also, the author prepared a testing environment for use of tools and deployment was done through the use of different sorts of scripts.

The experimental setup was very important in the testing of how automation scripts and tools like Ansible will help the administrators in implementing network configuration changes and managing network infrastructure effectively with less manual effort. The results of these tests confirmed that automation speeds up the configuration process and increases precision, reducing expenditure of time and resources in the management of the network.

The author had posed several research questions which guided this research work. the first research question was focused on:

What are the current practices and challenges in network infrastructure management?

The finding of this research highlighted that the current practices predominantly involve manual configurations, leading to inefficiencies and a high probability of errors. Challenges are particularly pronounced in environments with diverse hardware from multiple vendors, as compatibility issues can complicate management tasks. The literature review and empirical observations within Company X confirmed that manual processes are slow and error-prone, often leading to delayed response times to network demands and operational inefficiencies.

The second research question was focused on network management which fundamentally involves the monitoring, maintenance, and optimization of network resources.

What are the key concepts of network management?

Key findings identified through the research include automation, configuration management, and compliance with security standards. These are crucial for ensuring operational efficiency. Automation tools like Ansible and monitoring solutions like Prometheus and Grafana play pivotal roles in modern network management by facilitating the centralised and automated control of network components.

Multi-vendor environments introduce complexity due to varying proprietary systems and protocols, which can hinder seamless automation. The author had formulated the following research question:

What are the challenges in network automation in multi-vendor environments?

The tests conducted on the experimental setup demonstrated that without standardisation, automating configurations across different devices requires tailored scripts and configurations. The challenges are further compounded by the need for extensive testing to ensure compatibility and functionality across all devices, as evidenced by the diverse hardware setup in the test environment.

The development of automated solutions involves leveraging tools like Ansible for configuration management and using Infrastructure as Code (IaC) practices to maintain consistency and reproducibility. The following question was formulated by the author to address this aspect.

How can automated solutions be developed and implemented to improve the management of the hardware layer of network infrastructure at Company X?

The experimental results showed that implementing automation significantly reduced the time and effort required for network configurations and management. By standardising the deployment scripts and using version control systems like GitHub, Company X can manage changes more effectively, reduce errors, and quickly adapt to new network requirements. The automation layer developed provides a scalable framework that can be extended and modified as network demands evolve.

Future research directions could build on the foundation laid by this thesis in the following ways:

Improved scalability for automation solutions: The research methods concluded above can even further scale automation solutions to support a larger and more complex network infrastructure, potentially through modular scripts and tools that are easily adapted in different scenarios.

Comprehensive Testing in More Diverse Environments: Expanding the testing environments to include more types of devices and network setups that result in generalising the applicability of developed automation solutions.

Streamlining Integration Processes: The focus will be on making this integration of the automation tool with the network management framework already in deployment a pretty easy one and with quite a relatively low learning curve, so that administrators can comfortably embrace the adoption of automation.

Automation of the MikroTik devices: MikroTik devices do not have a direct automatization approach. This presents a unique challenge in the automation landscape for future works as traditional methods like direct playbook executions are not feasible. To address this, an innovative approach involving Docker and Python scripting can be utilised. A Docker container running RouterOS can simulate the MikroTik environment, allowing for safe and isolated testing of new configurations.

References

[1] "Largest Networking Hardware Companies by Market Cap," [Online]. Available: https://companiesmarketcap.com/networking-hardware/largest-companies-by-market-cap/. [Accessed 1 March 2024].

[2] Jay Ashok Shah and Dushyant Dubaria, "NetDevOps: A New Era Towards Networking & DevOps," [Online]. Available: https://ieeexplore.ieee.org/document/8992969. [Accessed 1 March 2024].

[3] Bleuwire, "10 Network Monitoring Challenges (And How to Deal With Them)," [Online]. Available: https://bleuwire.com/10-network-monitoring-challenges-and-how-to-deal/. [Accessed 10 March 2024].

[4] IBM, "Networks unchained: The shift toward intent-based autonomous operations," [Online]. Available: https://www.ibm.com/blog/networks-unchained-the-shift-toward-intent-based-autonomous-operations/. [Accessed 10 March 2024].

[5] Juniper Networks, "Enterprises DIY Data Center Network Automation - Key Motivations, Challenges, and True Costs of In-House Built Automation," [Online]. Available: https://www.juniper.net. [Accessed 10 March 2024].

[6] Red Hat, "Automate Cisco networks with Red Hat Ansible Automation Platform," in Red Hat Blog, Red Hat, 2023. [Online]. Available: https://www.redhat.com/en/resources/automate-cisco-networks-with-ansible-overview [Accessed 10 March 2024].

[7] Cisco Press, "Programming and Automating Cisco Networks: A guide to network programmability and automation in the data center, campus, and WAN," in Cisco Press

Resources, Cisco Press, 2016. [Online]. https://www.ciscopress.com/store/programming-and-automating-cisco-networks-a-guide-to-9780134436784 [Accessed 10 March 2024].

[8] Cisco Blogs, "Network Automation – Now!" in Cisco Blogs, Cisco, 2017. [Online]. Available: https://blogs.cisco.com/automation/network-automation-now [Accessed 10 March 2024].

[9] Cisco DevNet, "Ansible," in Cisco Developer Resources, Cisco, 2023. [Online]. Available: https://developer.cisco.com/automation-ansible/ [Accessed 15 March 2024].

[10] Juniper Networks, "Welcome to the Future of Network Automation: Juniper Paragon Automation as a Service," in Official Juniper Networks Blogs, Juniper Networks, 2023. [Online]. Available: https://blogs.juniper.net/en-us/automation/welcome-to-the-future-of-network-automation-juniper-paragon-automation-as-a-service [Accessed 15 March 2024].

[11] Juniper Networks, "Build the Autonomous Network of the Future Today with Paragon Automation," in Official Juniper Networks Blogs, Juniper Networks, 2023. [Online]. Available: https://blogs.juniper.net/en-us/automation/build-the-autonomous-network-of-the-future-today-with-paragon-automation [Accessed 16 March 2024].

[12] Juniper Networks, "Understanding Ansible for Junos OS," in Ansible for Junos OS, Juniper Networks, 2023. [Online]. Available: https://www.juniper.net/documentation/us/en/software/junos-ansible/ansible/topics/concept/junos-ansible-overview.html [Accessed 16 March 2024].

[13] Juniper Networks, "Understanding the Ansible for Junos OS Collections, Roles, and Modules," in Ansible for Junos OS, Juniper Networks, 2023. [Online]. Available: https://www.juniper.net/documentation/us/en/software/junos-ansible/ansible/topics/concept/junos-ansible-modules-overview.html [Accessed 16 March 2024].

[14] Juniper Networks, "Create and Execute Ansible Playbooks to Manage Junos Devices," in Ansible for Junos OS, Juniper Networks, 2023. [Online]. Available: https://www.juniper.net/documentation/us/en/software/junos-ansible/ansible/topics/task/junos-ansible-playbooks-creating-executing.html [Accessed 18 March 2024].

[15] YetiOps, "Ansible for Networking - Part 6: MikroTik RouterOS," YetiOps, 2023. [Online]. Available: https://yetiops.net/posts/ansible-for-networking-part-6-mikrotik-routeros/ [Accessed 18 March 2024].

[16] Ansible Community Documentation, "RouterOS Platform Options," Ansible, 2023. [Online]. Available: https://docs.ansible.com/ansible/latest/network/user_guide/platform_routeros.html. [Accessed 18 March 2024].

[17] Ansible Pilot, "Backup Config on Mikrotik RouterOS - Ansible Network community.routeros," Ansible Pilot, 2023. [Online]. Available: https://www.ansiblepilot.com/articles/backup-config-on-mikrotik-routeros-ansible-network-community.routeros/. [Accessed 18 March 2024].

[18] Datadog, "SNMP Monitoring: What It Is & How It Works," Datadog, 2023. [Online]. Available: https://www.datadoghq.com/knowledge-center/network-monitoring/snmp-monitoring/#:~:text=SNMP%20monitoring%20can%20be%20used,runs%20on%20a%20network%20device. [Accessed 18 March 2024].

[19] Kentik, "SNMP Monitoring: An Introduction and Practical Tutorial," Kentik, 2023. [Online]. Available: https://www.kentik.com/kentipedia/snmp-monitoring/ [Accessed 18 March 2024].

[20] "NETCONF Protocol," Cisco Systems, 2023. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/ios-

xml/ios/prog/configuration/1610/b_1610_programmability_cg/configuring_yang_datamodel.pdf . [Accessed 30 April 2024].

[21] "Understanding NETCONF and YANG," Network World, 2023. [Online]. Available: https://www.networkworld.com/article/686181/understanding-netconf-and-yang.html. [Accessed 30 April 2024].

[22] FS Community, "Understanding Virtual Router Redundancy Protocol (VRRP)," FS Community, 2024. [Online]. Available: https://community.fs.com. [Accessed 18 March 2024]

[23] Juniper Networks, "Understanding VRRP," Juniper Networks, 2023. [Online]. Available: https://www.juniper.net/documentation/us/en/software/junos/high-availability/topics/concept/vrrp-overview-ha.html. [Accessed 18 March 2024].

[24] "An Advanced Guide to Network Monitoring with Grafana and Prometheus," Grafana Documentation, 2024. [Online]. Available: https://grafana.com/blog/2022/02/01/an-advanced-guide-to-network-monitoring-with-grafana-and-prometheus/. [Accessed 30 March 2024].

[25] "Network Monitoring with Grafana and Prometheus," Earthly Blog, 2024. [Online]. Available: https://earthly.dev/blog/monitoring-system-metrics-prometheus-grafana/. [Accessed 30 March 2024].

[26] "Network Access Control (NAC) Explained," Varonis, 2024. [Online]. Available: https://www.varonis.com/blog/network-access-control-nac. [Accessed 30 March 2024].

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

[1] The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 - Configuration snippet from Juniper SRX100 router

```
version 12.1X44-D35.5;
system {
    host-name Juniper-SRX;
    time-zone GMT+2;
    root-authentication {
        encrypted-password "RealPasswordWillBeHere"; ## SECRET-DATA
    }
    login {
        message "See ruuter on Dmitri Sobolev oma, paluks mitte
puutuda!";
        user admin {
            uid 2001;
            class super-user;
            authentication {
                encrypted-password "RealPasswordWillBeHere"; ##
SECRET-DATA
            }
        }
    }
    services {
        ssh {
            root-login deny;
            protocol-version v2;
        }
        netconf {
            ssh {
                connection-limit 2;
                port 830;
            }
        }
        web-management {
            http;
            https {
                system-generated-certificate;
            }
        }
    }
}
chassis {
    aggregated-devices {
        ethernet {
            device-count 5;
        }
    }
}
interfaces {
    fe-0/0/6 {
        fastether-options {
            802.3ad ae4;
        }
    }
    fe-0/0/7 {
        fastether-options {
            802.3ad ae4;
        }
    }
    ae4 {
        aggregated-ether-options {
            lacp {
                passive;
            }
        }
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members [ USER MNG PUBLIC-NETWORK ];
                }
                native-vlan-id default;
            }
        }
    }
    vlan {
        unit 10 {
            family inet {
                address 10.0.10.9/24;
            }
        }
    }
}
snmp {
    contact admin;
    community public {
        authorization read-only;
    }
}
routing-options {
    static {
        route 0.0.0.0/0 next-hop 10.0.10.1;
    }
}
security {
    zones {
        security-zone MNG {
            host-inbound-traffic {
                system-services {
                    all;
                }
                protocols {
                    all;
                }
            }
            interfaces {
                vlan.10 {
                    host-inbound-traffic {
                        system-services {
                            all;
                        }
                    }
                }
            }
        }
        security-zone Netconf {
            host-inbound-traffic {
                system-services {
                    netconf;
                }
            }
            interfaces {
                ae4.0 {
                    host-inbound-traffic {
                        system-services {
                            netconf;
                        }
                    }
                }
            }
        }
    }
}
ethernet-switching-options {
    voip;
}
vlans {
    MNG {
        vlan-id 10;
        l3-interface vlan.10;
    }
    PUBLIC-NETWORK {
        vlan-id 30;
    }
    USER {
        vlan-id 20;
    }
    default;
}
```

Appendix 3 – Configuration snippet from one of the CISCO switches

```
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
service unsupported-transceiver
hostname SW1-dsobit
boot-start-marker
boot-end-marker
enable secret 5 RealPasswordWillBeHere.
username admin password 7  RealPasswordWillBeHere
no aaa new-model
switch 1 provision ws-c2960x-24ts-l
no ip domain-lookup
ip domain-name BAKALAVR.LOCAL
vtp domain tha
vtp mode transparent
spanning-tree mode pvst
spanning-tree extend system-id
no errdisable detect cause gbic-invalid
vlan internal allocation policy ascending
vlan 10
 name MNG
vlan 20
 name USER
vlan 30
 name PUBLIC-NETWORK
ip ssh version 2
interface FastEthernet0
 no ip address
interface GigabitEthernet1/0/1
 shutdown
interface GigabitEthernet1/0/2
 shutdown
interface GigabitEthernet1/0/3
 shutdown
interface GigabitEthernet1/0/4
 shutdown
interface GigabitEthernet1/0/5
 shutdown
interface GigabitEthernet1/0/6
 shutdown
interface GigabitEthernet1/0/7
 shutdown
interface GigabitEthernet1/0/8
 shutdown
interface GigabitEthernet1/0/9
 shutdown
```

```
interface GigabitEthernet1/0/10
 shutdown
interface GigabitEthernet1/0/11
 shutdown
interface GigabitEthernet1/0/12
 shutdown
interface GigabitEthernet1/0/13
 shutdown
interface GigabitEthernet1/0/14
 shutdown
interface GigabitEthernet1/0/15
 shutdown
interface GigabitEthernet1/0/16
 shutdown
interface GigabitEthernet1/0/17
 shutdown
interface GigabitEthernet1/0/18
 shutdown
interface GigabitEthernet1/0/19
 shutdown
interface GigabitEthernet1/0/20
 shutdown
interface GigabitEthernet1/0/21
 shutdown
interface GigabitEthernet1/0/22
 shutdown
interface GigabitEthernet1/0/23
 switchport mode trunk
interface GigabitEthernet1/0/24
 switchport access vlan 10
 switchport mode access
 spanning-tree portfast
interface GigabitEthernet1/0/25
 switchport mode trunk
interface GigabitEthernet1/0/26
 switchport mode trunk
interface GigabitEthernet1/0/27
 switchport mode trunk
interface GigabitEthernet1/0/28
 switchport mode trunk
interface Vlan1
 no ip address
interface Vlan10
 ip address 10.0.10.10 255.255.255.0
interface Vlan20
 no ip address
interface Vlan30
 no ip address
ip default-gateway 10.0.10.1
ip http server
ip http secure-server
snmp-server community public RO
banner motd ^C See switch on Dmitri Sobolev oma, paluks mitte puutuda!
^C
```

```
line con 0
 password 7 RealPasswordWillBeHere
 login
line vty 0 4
 exec-timeout 2 0
 login local
 transport input ssh
line vty 5 15
 login
End
```

Appendix 4 - Automated Cisco Jinja2 Configuration

```
version 15.0
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname {{ cisco_hostname }}
!
enable secret 5 {{ cisco_enable_secret }}
!
ip domain-name {{ cisco_domain_name }}
vtp domain {{ cisco_vtp_domain }}
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
{% for vlan in cisco_vlans %}
vlan {{ vlan.id }}
 name {{ vlan.name }}
!
{% endfor %}
!
interface Vlan1
 no ip address
 shutdown
!
{% for vlan_interface in cisco_vlan_interfaces %}
interface Vlan{{ vlan_interface.id }}
 ip address {{ vlan_interface.ip_address }} {{
vlan_interface.subnet_mask }}
 {% if not vlan_interface.shutdown %}
 no shutdown
 {% else %}
 shutdown
 {% endif %}
!
{% endfor %}
{% for interface in cisco_interfaces %}
interface {{ interface.name }}
 {% if interface.shutdown %}
 shutdown
 {% else %}
 no shutdown
 {% endif %}
 {% if interface.mode == 'access' %}
 switchport access vlan {{ interface.vlan }}
 switchport mode access
 {% if interface.portfast %}
 spanning-tree portfast
 {% endif %}
 {% elif interface.mode == 'trunk' %}
 switchport mode trunk
 {% endif %}
!
{% endfor %}
ip default-gateway {{ cisco_default_gateway }}
!
banner motd ^C{{ cisco_banner_message }}^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 login local
 stopbits 1
!
line vty 0 4
 login local
 transport input telnet ssh
!
end
```

Appendix 5 - Automated Juniper Jinja2 Configuration

```
version 12.1X44-D35.5;
system {
    host-name {{ juniper_hostname }};
    time-zone {{ juniper_time_zone }};
    root-authentication {
        encrypted-password "{{ juniper_root_password }}";
    }
    login {
        message "{{ juniper_login_message }}";
        user admin {
            uid 2001;
            class super-user;
            authentication {
                encrypted-password "{{ juniper_admin_password }}";
            }
        }
    }
    services {
        ssh {
            root-login {{ juniper_ssh_root_login }};
            protocol-version {{ juniper_ssh_protocol_version }};
        }
        netconf {
            ssh {
                connection-limit {{
juniper_netconf_ssh_connection_limit }};
                port {{ juniper_netconf_ssh_port }};
            }
        }
        web-management {
            {% if juniper_web_management_http %}
            http;
            {% endif %}
            {% if juniper_web_management_https %}
            https {
                system-generated-certificate;
            }
            {% endif %}
        }
    }
}
chassis {
    aggregated-devices {
        ethernet {
            device-count {{ juniper_chassis_device_count }};
        }
    }
}
interfaces {
    {% for interface in juniper_interfaces %}
    {{ interface.name }} {
        fastether-options {
            802.3ad {{ interface.lacp_group }};
        }
    }
    {% endfor %}
}
vlans {
    {% for vlan in juniper_vlans %}
    {{ vlan.name }} {
        vlan-id {{ vlan.id }};
        l3-interface {{ vlan.l3_interface }};
        {% if vlan.ip_address %}
        l3-interface {
            family inet {
                address {{ vlan.ip_address }};
            }
        }
        {% endif %}
    }
    {% endfor %}
}
routing-options {
    static {
        route 0.0.0.0/0 next-hop {{ juniper_default_gateway }};
    }
}
snmp {
    contact {{ juniper_snmp_contact }};
    community {{ juniper_snmp_community }} {
        authorization {{ juniper_snmp_authorization }};
    }
}
security {
    zones {
        {% for zone in juniper_security_zones %}
        security-zone {{ zone.name }} {
            host-inbound-traffic {
                system-services {
                    {{ zone.services }};
                }
                protocols {
                    {{ zone.protocols }};
                }
            }
            interfaces {
                {% for interface in zone.interfaces %}
                {{ interface }} {
                    host-inbound-traffic {
                        system-services {
                            {{ zone.services }};
                        }
                    }
                }
                {% endfor %}
            }
        }
        {% endfor %}
    }
}
ethernet-switching-options {
    voip;
}
```

Appendix 6 - File with variables for Automated Cisco Jinja2 Configuration

```
cisco_ansible_user_id: admin

cisco_hostname: SW1-dsobit

cisco_enable_secret: $1$/Rqu$qHK5E/RzbKGtiP7AcR.R4.

cisco_domain_name: BAKALAVR.LOCAL

cisco_vtp_domain: tha

cisco_vlans:

  - id: 10

    name: MNG

  - id: 20

    name: USER

  - id: 30

    name: PUBLIC-NETWORK

  - id: 40

    name: Accountant

cisco_interfaces:

  - name: GigabitEthernet1/0/1

    shutdown: true
    mode: access
  - name: GigabitEthernet1/0/2
    shutdown: true
    mode: access
  - name: GigabitEthernet1/0/3
    shutdown: true
    mode: access
  - name: GigabitEthernet1/0/4
    shutdown: true
    mode: access
  - name: GigabitEthernet1/0/5
    shutdown: true
    mode: access
  - name: GigabitEthernet1/0/6
    shutdown: true
    mode: access
  - name: GigabitEthernet1/0/7
    shutdown: true
    mode: access
  - name: GigabitEthernet1/0/8
    shutdown: true
    mode: access
  - name: GigabitEthernet1/0/9
    shutdown: true
    mode: access
  - name: GigabitEthernet1/0/10
    shutdown: true
    mode: access
```

```
- name: GigabitEthernet1/0/11
  shutdown: true
  mode: access
- name: GigabitEthernet1/0/12
  shutdown: true
  mode: access
- name: GigabitEthernet1/0/13
  shutdown: true
  mode: access
- name: GigabitEthernet1/0/14
  shutdown: true
  mode: access
- name: GigabitEthernet1/0/15
  shutdown: true
  mode: access
- name: GigabitEthernet1/0/16
  shutdown: true
  mode: access
- name: GigabitEthernet1/0/17
  shutdown: true
  mode: access
- name: GigabitEthernet1/0/18
  shutdown: true
  mode: access
- name: GigabitEthernet1/0/19
  shutdown: true
  mode: access
- name: GigabitEthernet1/0/20
```

```
  shutdown: true
  mode: access
- name: GigabitEthernet1/0/21
  shutdown: true
  mode: access
- name: GigabitEthernet1/0/22
  shutdown: true
  mode: access
- name: GigabitEthernet1/0/23
  shutdown: false
  mode: trunk
- name: GigabitEthernet1/0/24
  shutdown: false
  vlan: 10
  mode: access
  portfast: true
- name: GigabitEthernet1/0/25
  shutdown: false
  mode: trunk
- name: GigabitEthernet1/0/26
  shutdown: false
  mode: trunk
- name: GigabitEthernet1/0/27
  shutdown: false
  mode: trunk
- name: GigabitEthernet1/0/28
  shutdown: false
  mode: trunk
```

```
cisco_vlan_interfaces:

  - id: 10

    ip_address: 10.0.10.10

    subnet_mask: 255.255.255.0

    shutdown: false

cisco_default_gateway: 10.0.10.1

cisco_banner_message: "See switch on Dmitri Sobolev oma, paluks mitte
puutuda!"

cisco_console_password: 094F471A1A0A5A100A0705787B767C
```

Appendix 7 - File with variables for Automated Juniper Jinja2 Configuration

```
juniper_ansible_user_id: admin

juniper_hostname: Juniper-SRX

juniper_time_zone: GMT+2

juniper_root_password: "$1$Uq/lq0HW$bNl9kgIm3aZgr1o0X8mVN1"

juniper_admin_password: "$1$/evImjG7$xMMl7BkvGobgDR95fPXci/"

juniper_login_message: "See ruuter on Dmitri Sobolev oma, paluks mitte
puutuda!"

juniper_ssh_root_login: deny

juniper_ssh_protocol_version: v2

juniper_netconf_ssh_connection_limit: 2

juniper_netconf_ssh_port: 830

juniper_web_management_http: true

juniper_web_management_https: true

juniper_chassis_device_count: 5
```

```
juniper_interfaces:
  - name: fe-0/0/6
    lacp_group: ae4
  - name: fe-0/0/7
    lacp_group: ae4
  - name: ae4
    port_mode: trunk
    vlan_members: [USER, MNG, PUBLIC-NETWORK]
    native_vlan_id: default
juniper_vlans:
  - id: 10
    name: MNG
    l3_interface: vlan.10
    ip_address: 10.0.10.9/24
  - id: 20
    name: USER
  - id: 30
    name: PUBLIC-NETWORK
  - id: 40
    name: Accountant
juniper_default_gateway: 10.0.10.1
juniper_snmp_contact: admin
juniper_snmp_community: public
juniper_snmp_authorization: read-only
juniper_security_zones:
  - name: MNG
    services: all
    protocols: all
    interfaces: [vlan.10]

  - name: Netconf

    services: netconf

    interfaces: [ae4.0]

juniper_ethernet_switching_options: voip
```

Appendix 8 – MikroTik router firewall rules

```
/ip firewall filter

add action=accept chain=input comment=\

    "defconf: accept established,related,untracked" connection-state=\

    established,related,untracked

add action=drop chain=input comment="defconf: drop invalid"
connection-state=\

    invalid

add action=accept chain=input comment="defconf: accept ICMP"
protocol=icmp

add action=accept chain=input comment="OVPN pass" dst-port=1194
protocol=tcp

add action=accept chain=input comment="VRRP pass VLAN10" src-address=\

    10.0.10.3

add action=accept chain=input comment="VRRP pass VLAN20" src-address=\

    10.0.20.3

add action=accept chain=input comment="VRRP pass VLAN30" src-address=\

    10.0.30.3

add action=accept chain=input comment="VRRP pass WAN" in-interface-
list=WAN \

    src-address=192.168.1.56
```

```
add action=accept chain=input comment="accept SSH comming not from
wan" \

    dst-port=22 in-interface-list=!WAN protocol=tcp

add action=accept chain=input comment="accept FTP comming not from
wan" \

    dst-port=21 in-interface-list=!WAN protocol=tcp

add action=accept chain=input comment="accept all from vpn" src-
address=\

    10.254.0.0/28

add action=accept chain=input src-address=10.253.0.0/28

add action=accept chain=forward src-address=10.253.0.0/28

add action=accept chain=forward src-address=10.254.0.0/28

add action=accept chain=input comment="accept all comming from MNG" \

    in-interface-list=MNG

add action=drop chain=input comment="defconf: drop all not coming from
LAN" \

    in-interface-list=!LAN log=yes log-prefix=drop-not-lan

add action=fasttrack-connection chain=forward comment="defconf:
fasttrack" \

    connection-state=established,related hw-offload=yes

add action=accept chain=forward comment=\

    "defconf: accept established,related, untracked" connection-
state=\

    established,related,untracked

add action=drop chain=forward comment="defconf: drop invalid" \

    connection-state=invalid

add action=drop chain=forward comment=\

    "defconf: drop all from WAN not DSTNATed" connection-nat-
state=!dstnat \

    connection-state=new in-interface-list=WAN

/ip firewall nat

add action=masquerade chain=srcnat comment="defconf: masquerade" \
    ipsec-policy=out,none out-interface-list=WAN

/ip route

add disabled=no distance=10 dst-address=0.0.0.0/0 gateway=192.168.1.1
\

    pref-src="" routing-table=main scope=30 suppress-hw-offload=no \

    target-scope=10

add disabled=no distance=15 dst-address=0.0.0.0/0 gateway=10.0.0.3
pref-src=\

    "" routing-table=main scope=30 suppress-hw-offload=no target-
scope=10

add disabled=no dst-address=10.253.0.0/28 gateway=10.0.0.3 routing-
table=main \

    suppress-hw-offload=no
```