

Network automation and orchestration

CMP7221



[Date]

student name

Student ID

Table of Contents

[Introduction 2](#_Toc129880419)

[Proposed solution 2](#_Toc129880420)

[Design of the network 11](#_Toc129880421)

[Critique of the proposed solution 12](#_Toc129880422)

[References 14](#_Toc129880423)

**List of figures**

[Figure 1 Creating the git hub repository 4](#_Toc129880424)

[Figure 2 Cloning the repo 4](#_Toc129880425)

[Figure 3 Creating inventory 5](#_Toc129880426)

[Figure 4 Creating playbook file for the OSPF 6](#_Toc129880427)

[Figure 5 Creating EIGRP playbook file for R1 and R2 7](#_Toc129880428)

[Figure 6 Testing OSPF 9](#_Toc129880429)

[Figure 7 Testing credentials security 10](#_Toc129880430)

[Figure 8 Console logs at CI/CD 11](#_Toc129880431)

# Introduction

The automation and orchestration technologies are playing a critical role in the evolving network. This work aims at the development of the CI/CD pipeline for the network device configuration for the BCU network. The goal behind this is to perceive the advantages of having a agile approach for the management of the infrastructure. This provides the benefit of dynamic configuration with minimal input by the network engineer. This work includes the range of the orchestration and automation solutions. The proof-of-concept implementation is provided. For this scenario, the tools like Ansible, Git and Jenkins can be used. The Ansible is effective in the configuration of the network devices as pet the network topology with the help of playbooks. To automate the configuration of the elements in the BCU network, the orchestration and automation is designed using the infrastructure as Code approach.

# Proposed solution

The proposed solution to meet the orchestration and automation requirements of the BCU network is to implement the CI/CD pipeline with the help of GitHub and Ansible. The proposed solution aims to offer the ability to dynamically configure the new devices in the network to minimize the amount of effort required in the configuration. The proposed solution comprises of the version control system for maintain the code and the configuration files of the network infrastructure. The code is stored on Git to track and manage the changes easily. This is also helpful in the collaboration, version control and audit trails. The Ansible is used as the configuration management tool for the automation of the configuration of the network devices. This uses the configuration code from the Git repository and push the change using SSH. After this, the Pytest is used for the validation of the network configuration changes this is done to check the test cases for the EIGRP and OSPF configuration.

There are mainly four stages in the pipeline that are development, testing, staging and production. Following is the explanation of the proposed solution and stages:

1. Development stage: In this stage, the Anisible playbooks are written for the configuration of the network devices which are committed to the GitHub repository’s development branch.
2. Testing: The virtual box can be used for testing the playbooks and the test results can be reported to the developers for the correction or feedback. This stage can be automated using Ansible test which is the testing framework of ansible.
3. Staging: The playbooks can be tested in the stage network environment which is similar to the production environment. Any issues errors or issues that occurred during testing can be corrected prior to the deployment of playbooks to production.
4. Production stage: The playbooks are deployed to the production network environment which can be automated using Ansible pull that the deployment framework of Ansible.

Following are the tools and technologies that are used in the proposed solution:

* GitHub: This is the tool which is used for the source code management to get the deployment, configuration management and orchestration.
* Ansible: This is the automation tool which is used for configuration management.
* Virtual box: This is the open source virtualized software that enables the execution of the virtual machines.
* Ansible test: This is the testing framework provided by Ansible to test the functionalities and effectiveness of the Ansible playbooks.
* Ansible-pull: This is the deployment framework provided by Ansible for the execution of the Ansible playbooks.

Following are the steps, that are performed to automate the task of automation and orchestration.

Firstly, the repository is created on the GitHub platform with the name “BCUrepo”. Then the repository is cloned on the local machine using the ‘git clone’ command.

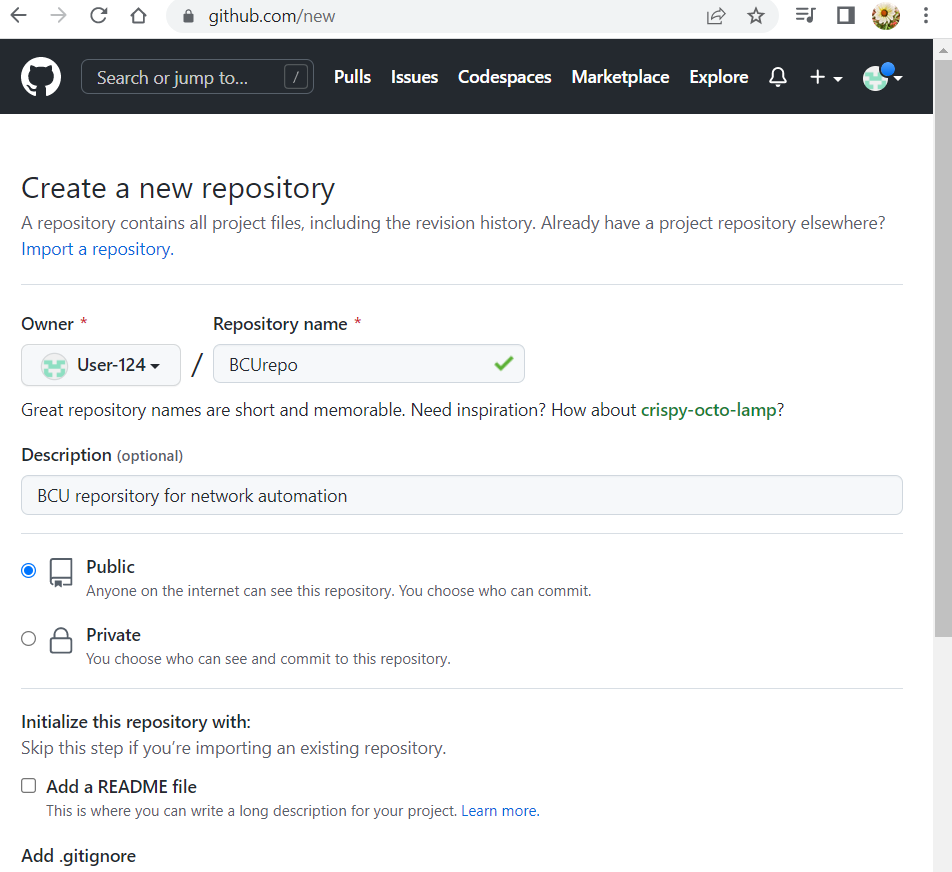


Figure Creating the git hub repository

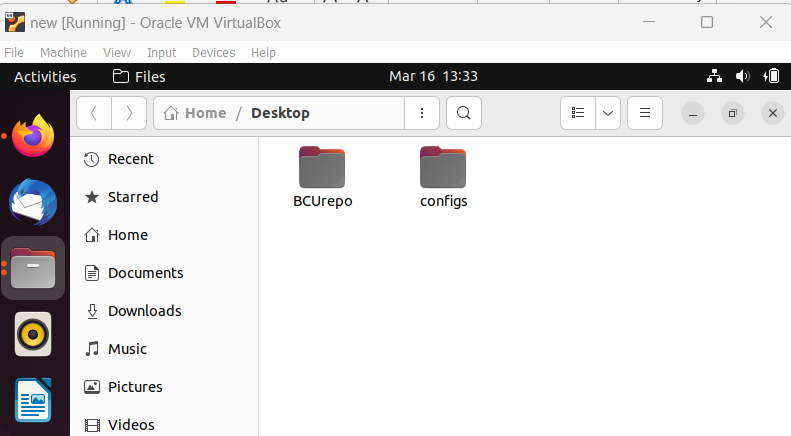


Figure Cloning the repo

Then a new inventory file is created in the local Git repository for storing the IP addresses of the network devices. This file is used by Ansible to get the IP address of the host for configurations.

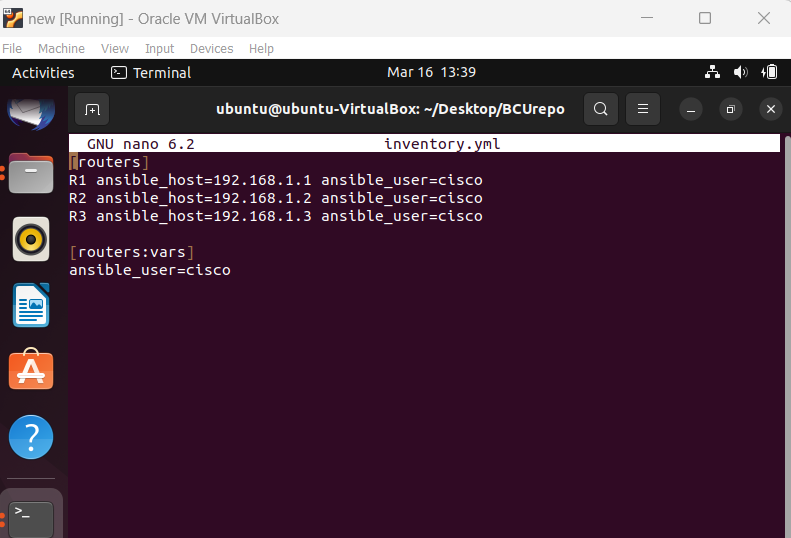


Figure Creating inventory

After this a new playbook is created with the name ‘ospf.yml’. this includes the details of the host that the playbook will run and the task to configure the OSPF on the R3 router. This Ansible playbook is used for the configuration of the OSPF routing protocol on the R3 router. This starts by defining the target host and then defines the tasks that will be executed at the target host including setting the router ID and advertising the suitable network prefixes.

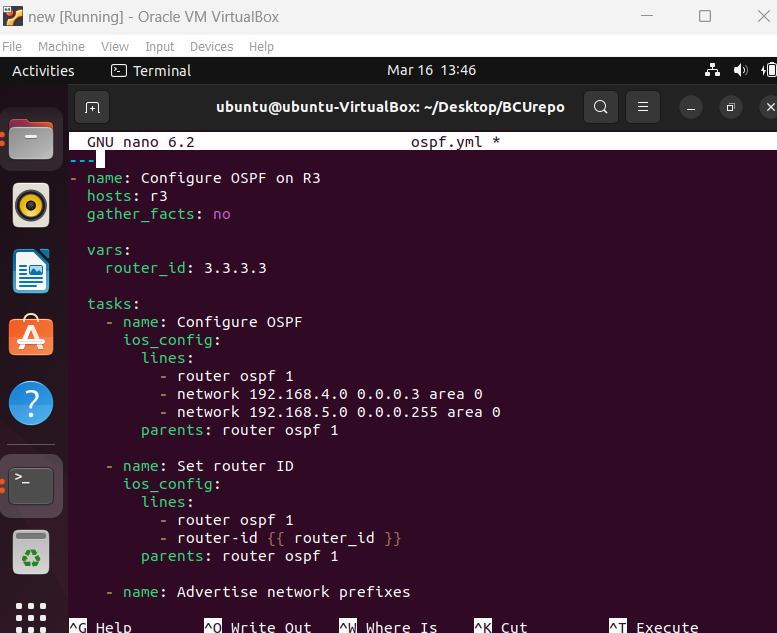


Figure Creating playbook file for the OSPF

Then the playbook file is executed using the command ‘ansible-playbook -i inventory.yml ospf.yml’. This configures the OSPF routing as defined in the notebook. This ansible playbook is used for the configuration of the OSPF routing protocol on the network devices. This also starts by defining the playbook which defines the set of tasks that will be

Then, another playbook is created with the name ‘eigrp.yml’ for configuring the EIGRP routing on R1 and R2. This includes the tasks to configure the EIGRP and set the ID of the router.

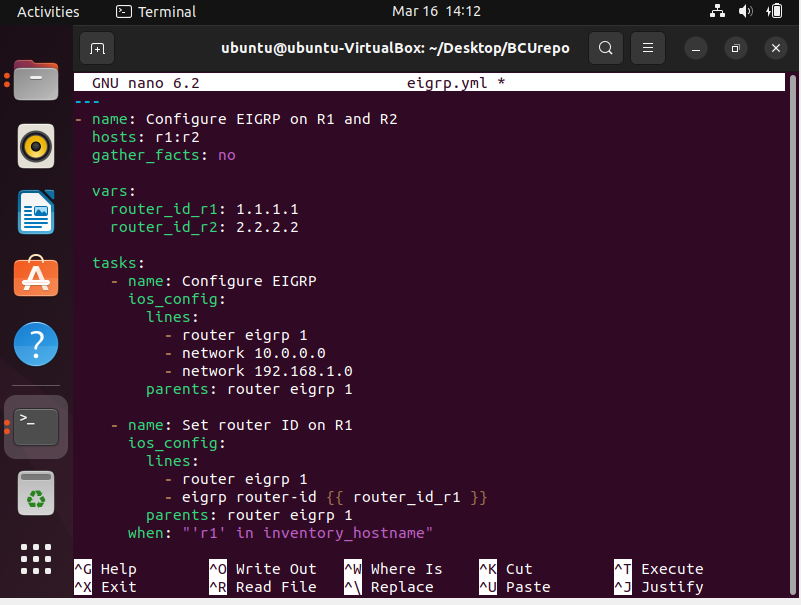


Figure Creating EIGRP playbook file for R1 and R2

The code is the Ansible playbook code for the configuration of the EIGRP on R1 and R2. This starts with defining the variables for the router ID as well as network prefixes. After this the playbook includes the task for the configuration of EIGRP routing, and setting the ID of the router. The task 1 of this configures the EIGRP on the both R1 and R2 with the ASN number and the second task set the ID of the router with the defined value. The final task advertises the connected network prefixes to other routers.

Both of these Ansible playbooks uses the ios\_config module to push the configuration commands to the network devices which are the example of showing the network automation to be used by Ansible. These are also allowed to b extended for meeting the specific requirements for the BCU environment. These Ansible playbooks are important for the automation and orchestration process for the BCU network.

**Setting up the CI/CD pipeline**: The following steps are performed for the automation of the network:

The GitLab CI/CD is used that can best fit to the requirement of the BCU network. Then after this, the tool is integrated with the Git repository. Under this, the new pipeline is created and the triggers are configuring for the initiation of the pipeline when the new code is pushed to the git repository. This also used for the configuration of the required authentication and access credentials. The CI/CD tools are used for the automation of the delivery process of the system which are helpful in the building, deployment of the software and streamlining the process. The GitLab CI/CD tool is used for this work.

Then after this, the configuration files are compiled into template to be used for deployment. To do this, firstly the git repository is cloned that included the network configuration files. The ansible allowed to write the code in YAML to run it as the playbook for the configuration of the network infrastructure. The ansible.cfg file is the configuration file that defines the pipeline for ansible. In the first stage of this, the git repository is cloned along with the ansible playbooks and test files.

Then in the testing stage, the tests are executed by running the ansible playbooks to assuring that the change in the network does not create any disruption. The Ansible playbooks are used for this. There are several testing tools available such as Pytest, netmiko and so on which allow to write the test cases and validating the changes to meet the desired requirements.

The OSPF configuration is tested to ensure that these are properly applied to the routers. It is done to determine the errors in the pipeline to assure that the issues can be addressed before moving to the deployment phase.

The testing files are created for defining the expected EIGRP neighbors, metrics and routes. Similar tests are created for the expected OSPF neighbor.

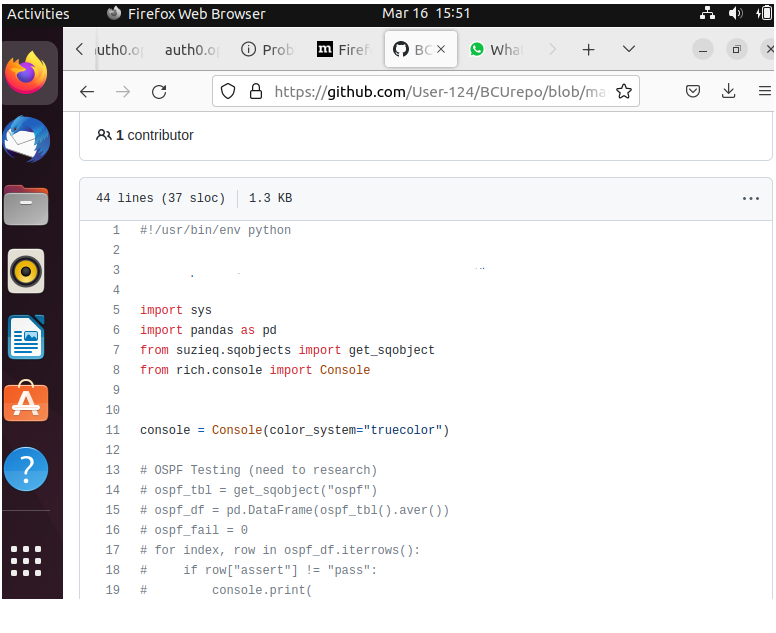


Figure Testing OSPF

The deployment stage is the final step of any CI/CD pipeline. After the success of the build and test stages, the changes can be implemented effectively in the network infrastructure. This includes the execution of the ansible playbooks, and scripts for making changes in the network devices. This is also important to have the plan for rolling back the change if anything wrong happens. This included taking regular backups of the configuration.

Setting up the alters and the notification is important for keeping track of the pipeline status and getting the notification about the alerts and getting the details of the failure and errors. In order to trigger the pipeline on the git repository updates, the first step is to establish the connection with the Git repository. This is done by adding the URL and the authentication credentials in the CI/CD tool. After this, the webhook is configured which is the mechanism for enabling the Git repository to notify the CI/CD tool when the changes are pushed to the git repository. This is done by adding the webhook URL to the git repository (De Sousa, et al., 2017).

After this, the pipeline trigger is configured which is done with the help of CI/CD settings by specifying the Git branch which is for monitoring the changes. If the changes are pushed to the branch, then the pipeline will be triggered automatically. The following is the handler that is used to assure that the credentials are not stored in clear text form to assure security and data privacy in the network. the function is accountable to check if the username is passed as the argument or otherwise it will take the current user name. a similar process is done with the password

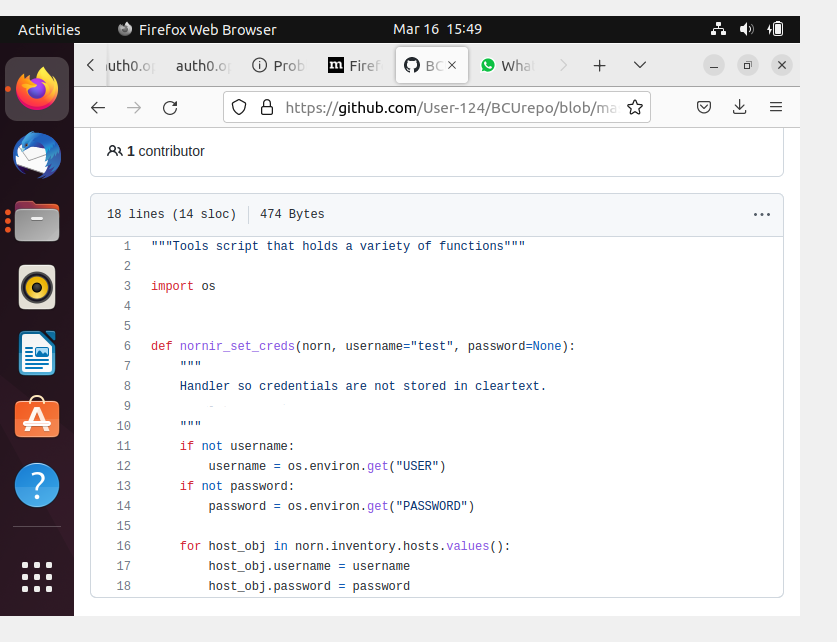
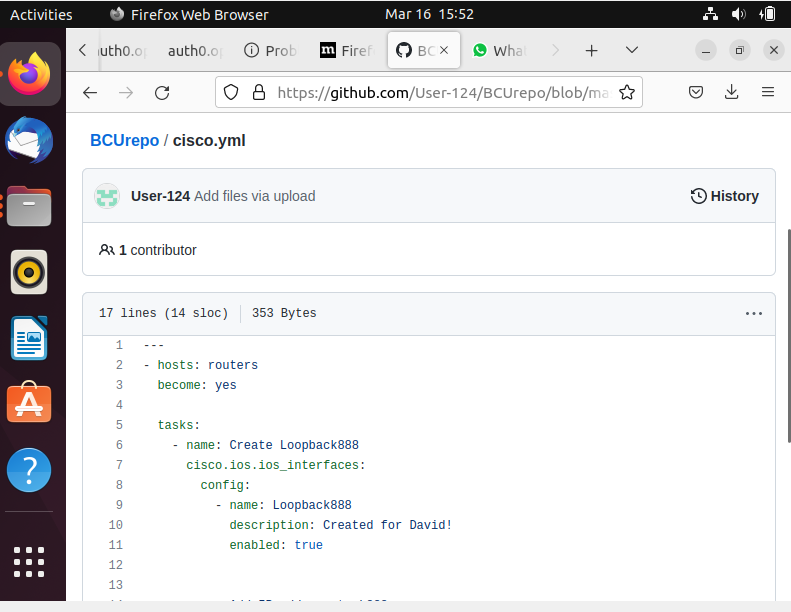


Figure Testing credentials security



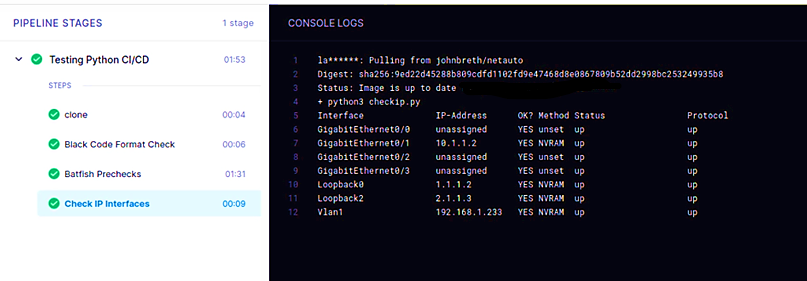


Figure Console logs at CI/CD

After this, the CI/CD interface is used for monitoring the interface to get insights into the progress of the pipeline and the different stages.

# Design of the network

The proposed design for the BCU network includes the following components:

Network development PC: This is accountable to support the installation of the orchestration and automation tools. This supports the development, testing, and deployment of the changes in the network configuration. The dynamic configuration allows to automatically configure of the network devices with minimal effort. The Ansible playbooks and modules are used to push the configuration.

Management of the source control: This is used for storing and controlling the version of the files of the network configuration. GitHub is used in the proposed solution for storing files.

CI/CD pipeline: This refers to the automated process of creating, building, and deploying the network configuration. Jenkins is used as the continuous integration server, Git as the version control system and Ansible as the configuration management tool.

Configuration management: Ansible is used as the configuration management platform for the management of the network statement. For the configuration of routers, Ansible playbooks and module is used.

Network devices: The network devices that are part of the proposed solution include the routers and ISP which are configured using the Ansible playbooks and modules.

# Critique of the proposed solution

The proposed solution addresses the network requirements and proves to be an effective and feasible solution. This may offer several benefits to the BCU networks including:

* **Better consistency:** The use of the standardized approach for the configuration assures consistency across the network and its devices and minimizes the risk of network downtime and errors in the configuration.
* **Improved efficiency:** The proposed orchestration and automation solution enables the automatic configuration of the network devices that can reduce the requirement of manual intervention by the network engineers which can be helpful in time and cost savings and also improve the efficiency of the process of managing the network.
* **Quick deployment:** The orchestration and automation enable the quick deployment of the new components and services so that the network can become more responsive and robust to meet the changing business requirements.
* **Reduced risk:** The orchestration and automation solution is helpful in the reduction of the risks related to security breaches and network downtime by assuring secure configuration across all the devices.
* **Better agility:** This orchestration and automation solution allows to be responsive and agile to evolving requirements of BCU to adopt new solutions and technologies quickly.
* **Improved visibility:** The centralized management platform offers better visibility of the network-based operations so that the network engineers can determine and resolve the issues quickly.

While the proposed automation solution can meet the network requirements, there are still some areas that can be improved or can be considered for future improvement.

* **Vendor-specific automation tool:** While ansible is a commonly used and robust tool for enabling automation, some network device vendors also have their vendor-specific tools which are suitable for their products. It is known that the BCU network has devices from one particular vendor that is Cisco. Cisco DNA Center is the platform for centralized management. This offers automation tools Cisco NSO (network Service orchestrator) to get network automation (Rotsos, et al., 2017). The vendor-specific tools can offer better integration and other advanced options.
* **Disaster discovery:** The proposed solution does not offer any specific measure for the recovery from disaster including restoration and backup procedures along with the failover methods. This is critical to have a disaster recovery plan to assuring the continuity of the operations at BCU in case any failure or outage occurs (Filiposka, and Łapacz, 2022).
* **Monitoring and reporting:** The monitoring tools like Nagios can be used for monitoring the network devices in the BCU network. This can generate an alert for the network engineers in case of any failure or errors. This can use SNMP to monitor the device metrics and generate alerts as per predefined thresholds. Similarly, reporting tools such as ELK stack can be used for the generation of reports and insights on the network infrastructure. ELK stack can use log data from the network devices and then create visualizations and insights on the performance network, changes in configuration, and other metrics.

# References

Masek, P., Stusek, M., Krejci, J., Zeman, K., Pokorny, J. and Kudlacek, M., 2018, May. Unleashing full potential of ansible framework: University labs administration. In *2018 22nd conference of open innovations association (FRUCT)* (pp. 144-150). IEEE.

Rahman, A., Rahman, M.R., Parnin, C. and Williams, L., 2021. Security smells in ansible and chef scripts: A replication study. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, *30*(1), pp.1-31.

Dalla Palma, S., Di Nucci, D. and Tamburri, D.A., 2020. AnsibleMetrics: A Python library for measuring Infrastructure-as-Code blueprints in Ansible. *SoftwareX*, *12*, p.100633.

González, G. and Evans, C.L., 2019. Biomedical Image Processing with Containers and Deep Learning: An Automated Analysis Pipeline: Data architecture, artificial intelligence, automated processing, containerization, and clusters orchestration ease the transition from data acquisition to insights in medium‐to‐large datasets. *BioEssays*, *41*(6), p.1900004.

De Sousa, N.F.S., Perez, D.A.L., Rosa, R.V., Santos, M.A. and Rothenberg, C.E., 2019. Network service orchestration: A survey. *Computer Communications*, *142*, pp.69-94.

Rotsos, C., King, D., Farshad, A., Bird, J., Fawcett, L., Georgalas, N., Gunkel, M., Shiomoto, K., Wang, A., Mauthe, A. and Race, N., 2017. Network service orchestration standardization: A technology survey. *Computer Standards & Interfaces*, *54*, pp.203-215.

Filiposka, S. and Łapacz, R., 2022, November. Challenges and Benefits of Using Open Standards for Network Automation and Orchestration. In *2022 30th Telecommunications Forum (TELFOR)* (pp. 1-4). IEEE.