**University of Mumbai**

**Computer Network Automation**

**Using Python and GNS3**

Submitted at the end of semester VII in partial fulfillment of requirements

For the degree of

**Bachelor of Technology**

by

**Dhruti Sangal**

**Roll No: 1913076**

**Aastha Joshi**

**Roll No: 1913084**

**Krishna Kumar Pal**

**Roll No: 1913097**

**Vrunda Patel**

**Roll No: 1913101**

Guide

**Mrs. Jyoti Varavadekar**



**Department of Electronics and Telecommunication Engineering**

**K. J. Somaiya College of Engineering, Mumbai-77**

**(Autonomous College Affiliated to University of Mumbai)**

**Batch 2019 -2023**

**Certificate**

This is to certify that the dissertation report entitled **Computer Network Automation**

**Using Python and GNS3** submitted by Dhruti Sangal, Aastha Joshi, Krishna Kumar Pal and Vrunda Patel at the end of semester VII of LY B. Tech is a bonafide record for partial fulfillment of requirements for the degree of Bachelors in Technology in Electronics and Telecommunication Engineering of University of Mumbai

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Guide Head of the Department

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Principal

Date: 16-12-2022

Place: Mumbai-77

**Certificate of Approval of Examiners**

We certify that this dissertation report entitled **Computer Network Automation**

**Using Python and GNS3** is a bonafide record of project work done by Dhruti Sangal, Aastha Joshi, Krishna Kumar Pal and Vrunda Patel during semester VII.

This project work is submitted at the end of semester VII in partial fulfillment of requirements for the degree of Bachelors in Technology in Electronics and Telecommunication Engineering of University of Mumbai.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Internal Examiners

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

External/Internal Examiners

Date: 16-12-2022

Place: Mumbai-77

**DECLARATION**

We declare that this written report submission represents the work done based on our and / or others’ ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty and integrity as we have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/ matter in my submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Signature of the Student**  **\_\_\_\_\_\_\_\_\_\_\_1913076\_\_\_\_\_\_\_\_\_**  **Roll No.** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Signature of the Student**  **\_\_\_\_\_\_\_\_\_\_\_\_\_1913084\_\_\_\_\_\_\_\_\_\_\_\_**  **Roll No.** |
| --- | --- |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Signature of the Student**  **\_\_\_\_\_\_\_\_\_\_\_1913097\_\_\_\_\_\_\_\_**  **Roll No.** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Signature of the Student**  **\_\_\_\_\_\_\_\_\_\_\_\_\_1913101\_\_\_\_\_\_\_\_\_\_\_**  **Roll No.** |

**Date: 16-12-2022**

**Place: Mumbai-77**

**ABSTRACT**

Network programmability is a trend that is based on scripting techniques and traditional programming languages used for managing and monitoring of network parts. It has been enhanced and inspired by Software Defined Networks (SDN). In order to speed up equipment configuration and make maintenance simpler, we have presented some novel techniques for automating the configuration of network devices. By identifying and addressing security flaws, it also strengthens network stability and enhances network security. These approaches, which enable the unitary control of an expanding number of devices, are what networks will look like in the future.

The software's Graphical User Interface (GUI) allows users to conduct both fundamental network automation activities, such as backing up and restoring configuration files on several devices at once, and more complex processes, such as configuring security and configuration settings. The numerous options that the user has to connect to and configure network devices using Python and its libraries are demonstrated and discussed in the application's code.

**Keywords:** Network automation, software-defined networks, computer network operations, network management, python scripting.

**Contents**

| **Abstract……………………………………………………………………………………** | | | |  |
| --- | --- | --- | --- | --- |
| **List of Figures**……………………………………………………………………………. | | | |  |
| **List of Tables**……………………………………………………………………………… | | | |  |
| **Introduction ………………………………………………………………………………** | | | |  |
| **1** | **Network Automation**…………………………………….……………………... | | | 1 |
|  | 1.1 | | Challenges in network configuration……………………………………... | 1 |
|  | 1.2 | | Problem Statement ………………………………………………………... | 4 |
|  | 1.3 | | Objectives………………………………………………………………... | 5 |
| **2** | **Literature Survey**………………………………………………………………. | | | 6 |
| **3** | **Technologies and Techniques** …...…………………………………………… | | | 9 |
|  | 3.1 | | Program development information……………………………………………… | 9 |
|  | 3.2 | | Python modules and libraries……………………………………………. |  |
|  | 3.3 | | VScode …………………………………...……………………………… | 9 |
|  | 3.3 | | Virtual Environment………….…………………………………………. | 12 |
| **4** | **Program Implementation** ….………………………………………………… | | | 28 |
|  | 4.1 | Network Configuration …………………………………………………... | |  |
|  | 4.2 | Application Demo ……………………………………………………….. | |  |
| **5** | **Conclusions** …….……………………………… | | | 55 |
| **6** | **Scope for further work ………………………………………………………...** | | |  |
| **Bibliography** ……………………………………………………………………………... | | | | 57 |
| **Acknowledgments**………………………………………………………………………. | | | |  |

**List of Figures**

**List of Tables**

**INTRODUCTION**

Automation is the technology that allows a process or procedure to be carried out with little help from humans. The use of various control systems for equipment such as machinery, factory processes, boilers, heat-treating ovens, and other applications with little or no human intervention is known as automation or automatic control [1].

Similar to this, there are various levels of networking automation, ranging from automating tasks in a single device to automating workflows, such as backing up configurations or configuring a routing protocol across several devices, to higher levels of the hierarchy known as cross-domain automation.

Computer networks have expanded in complexity and dynamism. The difficulty for computer network providers then arises from the availability and dependability of network devices. Network engineers utilize a well-known utility as a secured shell to set up network equipment (SSH). However, the manual configuration takes time because it requires entering passwords, logging in and out, and other repetitive procedures for each device.

An application programming interface (API) for network automation can cut down on the repetition and time required for network maintenance [2]. One of the tasks is to keep an eye on the network for vulnerabilities. The automation network can be used to configure users as well as alter static and dynamic routing. Thus, we may say that network automation manages network devices through the application of programming logic, enabling network managers to configure network devices automatically.

Begin constructing automation at the device level by establishing tasks to automate the essential operations and growing it up from there to the Domain level [3]. Any task that is clearly defined and often repeated can be automated. Device Automation refers to the grouping of these tasks. It has been utilized for fault management and service level monitoring, but as the demands of the business have increased, so have the problems and opportunities.

Up until recently, network engineers had to employ time-consuming techniques and be familiar with proprietary protocols and technology.

Network engineers created Network Automation techniques to automate repetitive daily processes in an effort to cut costs and increase efficiency. With the help of significant networking firms, an open-source community was established with the aim of implementing automation applications, primarily using common programming languages like Python and standard interfaces (SSH, REST).

By utilizing industry-standard protocols like OpenFlow, SDN was able to minimize vendor dependence to a certain extent. A pioneer of the software-defined network revolution, OpenFlow is a low-level hardware-based protocol. It allows the network device's control plane to be deconstructed from the data plane.

The control plane is also engaged in the network's data flow. The manager that directs traffic through a device is referred to as the data plane.

SDN makes use of the distinction between the control plane and the data plane, removes the control plane from the device, and assumes control of how the network acts so that the network devices only concentrate on forwarding.

IT administrators may simply manage all network components and provision network services using an SDN application. Some of the benefits of SDN include the ability to automate networks, program traffic, and boost agility. On the other hand, there is regrettably no backward compatibility with legacy networks or non-SDN networks.

**1 Network Automation**

This chapter briefly describes network automation concepts, the problem statement and enterprises to network automation, and various types of automation and objective.

* 1. **Challenges in Network configuration**

The main reason for an organization to adopt network automation is to reduce the time that is needed to maintain and deploy changes on the network. Although time is crucial not all organizations choose to move to network automation. Each network nowadays is unique and the same goes for the network devices, this is something that discourages enterprises from moving to automation on their network because it usually involves the upgrade of the current equipment or moving for example on SDN because these unique devices are sometimes difficult if not impossible to be part of an automated network. Another hitch on moving towards network automation is the lack of a vendor-free standardized schema in conjunction with an affordable environment for testing. Except for time-saving networks, automation helps on troubleshooting problems on a network. The network administrators had to manually configure each device through CLI and in the case of a change that had to be done in all the devices, such as an addition of a new VLAN, they had to go to its device and set it up. This was not only time-consuming, but it also maximizes the possibility of an error. Moreover, it is dangerous to apply changes on a network during work hours and there are enterprises that work every day and there is a tight window, usually on holidays, for applying changes. Surveys that have been conducted showed that the most usual reason for downtime on a network is human errors. The most common human error in networking is the misconfiguration of network devices. It is a common task of a network administrator to apply an updated configuration file to a bunch of network devices

* 1. **Problem Statement**

Information technology is very high because of the COVID-19 pandemic. Organizations from business through education tend to use this technology most of the time. Information technology uses computer networks for the integration and management of data. A manageable network configuration for networked devices will be easier to maintain and reduce communication problems. Traditionally, network administrators must configure each network device manually. This process takes time and is inefficient. Automated network configuration can overcome the repetitive process, but it is relatively slow.

* 1. **Objective**

**1. Eliminate manual tasks:** Automating the network replaces manual tasks with predictable, repeatable network changes.

**2. Accelerate service delivery:** A critical benefit: provide key data services faster, optimize network performance, and speed the rollout of new services and applications.

**3. Standardize processes with templates:** Leverage standardized templates to drive efficiency among network stakeholders and streamline network changes.

**4. Make changes faster:** More frequently execute network changes that were previously rarely performed because they were manual, time-consuming, and resource-intensive.

**5. Gain network visibility:** Access wide-ranging performance monitoring capabilities to pinpoint performance issues, spiking resource utilization levels, and errors on the network.

**6. Analyze and resolve issues:** Rely on network analytics for insight into performance, utilization, security, and resource allocation that help resolve issues far faster than via manual

**2 LITERATURE SURVEY**

In **‘Performance analysis on network automation interaction with network devices using python**’[5] the main aim was that automation is becoming a trend these days due to its tremendous benefits, especially with the increasing numbers of network devices. The word automatic is being defined as acting or operating in a manner independent of external influence and human control, network automation offers thousands of benefits for companies, it allows the configuration of many devices within minutes, and eliminates the chances of human error.

In **‘Network automation and abstraction using Python Programming Methods’**[6] we looked into network programmability based on scripting techniques and traditional programming languages. It has been enhanced and inspired by SDNs. To speed up equipment configuration and make maintenance simpler, this paper presented some novel techniques for automating the configuration of network devices: by identifying and addressing the security flaws. These approaches, which enable the unitary control of an expanding number of devices, are what networks will look like in the future. It is possible to automate the configuration and monitoring of any device, irrespective of the vendor, on SDN devices as well as other networking solutions.

In ‘**As-RaD System as a Design Model of the Network Automation Configuration System Based on the REST API and Django Framework**’[7] they proposed an alternative model of a network automation system. The model system was implemented with a controller application that used REST API (Representational State Transfer Application Programming Interface) architecture and built by the Django framework with Python programming language to increase the performance of network automation. The design model, called the As-RaD System, uses a web-based application for maintenance and automates networking tasks with easy GUI. The network devices used in this research include the Cisco CSR1000V because it supports REST API communication to manage its network configuration and could be placed on the server either.

In **‘Network Automation Methodology for detecting Rogue Switch’**[8] it provided a solution to detect malicious switches on a network using GNS3 network emulator, Python for automation, Cisco ios configurations, WireShark packet analyzer and Docker. Goal is achieved by continuously filtering and analyzing network traffic for any broadcast storms or new ARP packets using Packet Analyzers and then effectively tracing the malicious host connected to the rogue switch by deploying automation techniques.

**3 Technologies and Techniques**

**3.1 Program development information**

The application’s purpose is to demonstrate different ways to connect and configure network devices thus commented code will be present in the code to exhibit an alternative solution. The implementation will be done in python and the code will follow most of the pep 8 instructions. PEP 8 is a style guide for python code that gives coding conventions. Following a style guide when coding an application improves the readability of code, and makes it consistent and easily maintained.

Some of the recommendations of the PEP 8 style guide that was followed are below.

* Indentation 4 spaces per indentation level, continuation lines align wrapped elements
* Blank lines two lines for top-level functions single line for methods
* Imports Each import on a separate line except if it is in the format “from library import something”. All the imports are at the top of the code.
* snake case naming style refers to the style of variable name writing. Each word starts with a lowercase letter and the space is replaced with an underscore.

Secure methods will be used, for example, the connectivity with the network devices will be accomplished through SSH instead of Telnet and if there will be a need for data serialization JSON will be used instead of Pickle. By using methods and functions of the imported modules the application’s security feature is enhanced. For instance, the password that the application needs to connect to the network devices is encoded and encrypted, and when the user is prompted to input a password, the password is masked with asterisks. Multithreading will be used to speed up the execution of the code.

**3.2 PYTHON MODULES AND LIBRARIES**

1. **Paramiko :** The SSH protocol version 2 is implemented in Python by Paramiko, a pure Python interface that offers client and server functions. Paramiko can achieve excellent performance with simple cryptographic ideas. This module enables Python scripts to configure any device that can be configured via SSH.
2. **Netmiko :** It is an open-source multi-vendor library, it enables the configuration of devices from several vendors using Python. Cisco IOS, Juniper, Arista, HP, and Linux are a few of the platforms that Netmiko supports. It might also work with other vendors including Alcatel, Huawei, and Ubiquity, though there hasn't been much testing done with them. To make SSH connections to network devices less complicated, more adaptable, and user-friendly, Netmiko operates on top of Paramiko. Although Netmiko is simple to use and it only supports a few suppliers' devices. Contrarily, Paramiko can be used to communicate with any SSH-compatible device.
3. **JSON** : A lightweight interchangeable data format called JavaScript Object Notation Module is used to convert Python objects, like lists and dictionaries, into a form that can be stored in a text file, a database, or transmitted over the network, and then to convert the data back to a Python object or other environments.
4. **Ip address** : The addresses that the user enters into the programme will be checked to ensure that they are legitimate IP addresses using this module library. The power of Python is demonstrated by modules like these. Using this module, it just takes three to four lines to determine whether an IP address is genuine as opposed to tenths of lines.
5. **Base64 and cryptography** : In addition to cryptography and some of its recipes, like Fernet, Base64 is used to encode and decode the passwords that are used to connect to networking devices, applying a high-level symmetric encryption to the passwords.
6. **Tkinter**: This is the module library that will be used to create the application's graphical user interface.

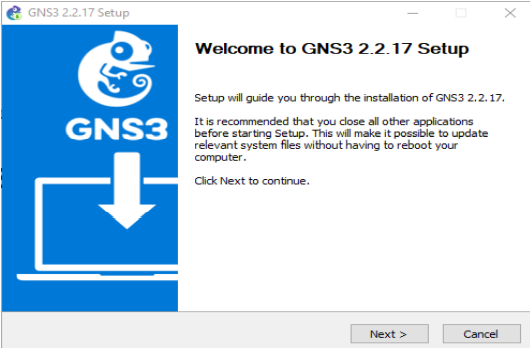
**3.3 VScode**

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS, and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages and runtimes.

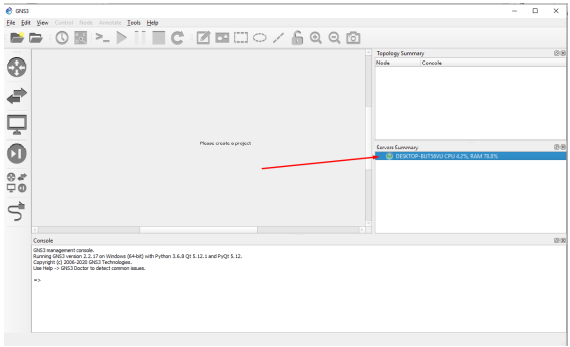
**3.4 Virtual Environment**

GNS3 was selected as the virtual environment application that will be used to test the application. Another option could be Cisco’s DevNet or as a third option EVE-NG.

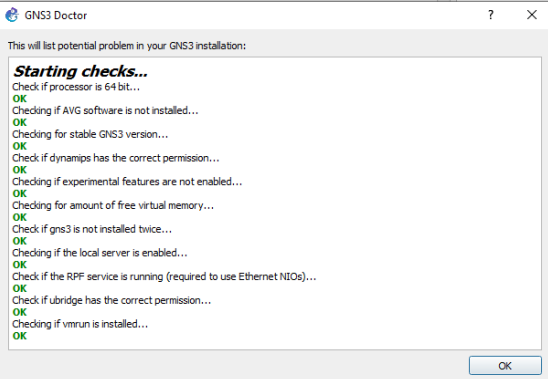
GNS3 is software that is used to emulate, configure and test a network environment. It is open-source free software and can be downloaded from the official web site. GNS3 consists of two components. The all-in-one software (GUI) is a graphical user interface and the Virtual Machine (VM) is a server that runs in a virtual environment and provides better topology size and device support. The installation is straightforward, and the default options should be used.

****

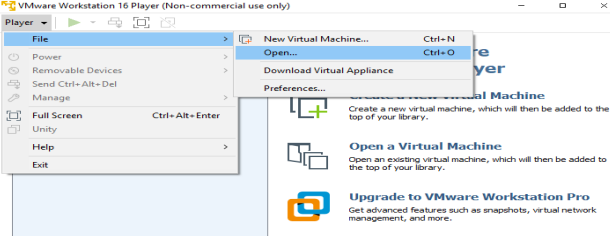
After the installation and booting of the GNS3 GUI on the Servers Summary window the PC’s name that the GNS3 is installed must be shown and it should also have a green light on the left.

****

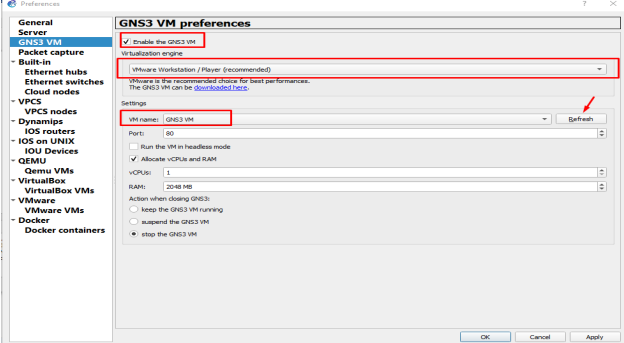
If there is nothing on that window or the light is red try restarting the GNS3, restarting the PC or check if the firewall or antivirus stopped the GNS3 service from running. If it is not running, make sure if necessary that permission was given to GNS3 through firewall and antivirus. GNS3 doctor (Help -> GNS3 Doctor) is a helpful module to check if there is something that blocks GNS3 from working correctly



GNS3 VM can be downloaded also from the GNS3 site. To install GNS3 VM a VM player must be used, for this project, the VMWare workstation Player will be used but VirtualBox or HyperV could also be used. Attention must be given to the fact that GNS3 GUI and GNS3 VM should be in the same version.

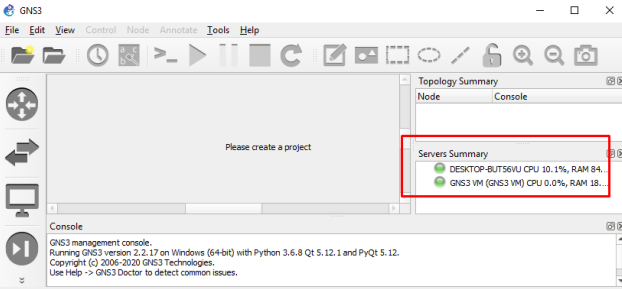


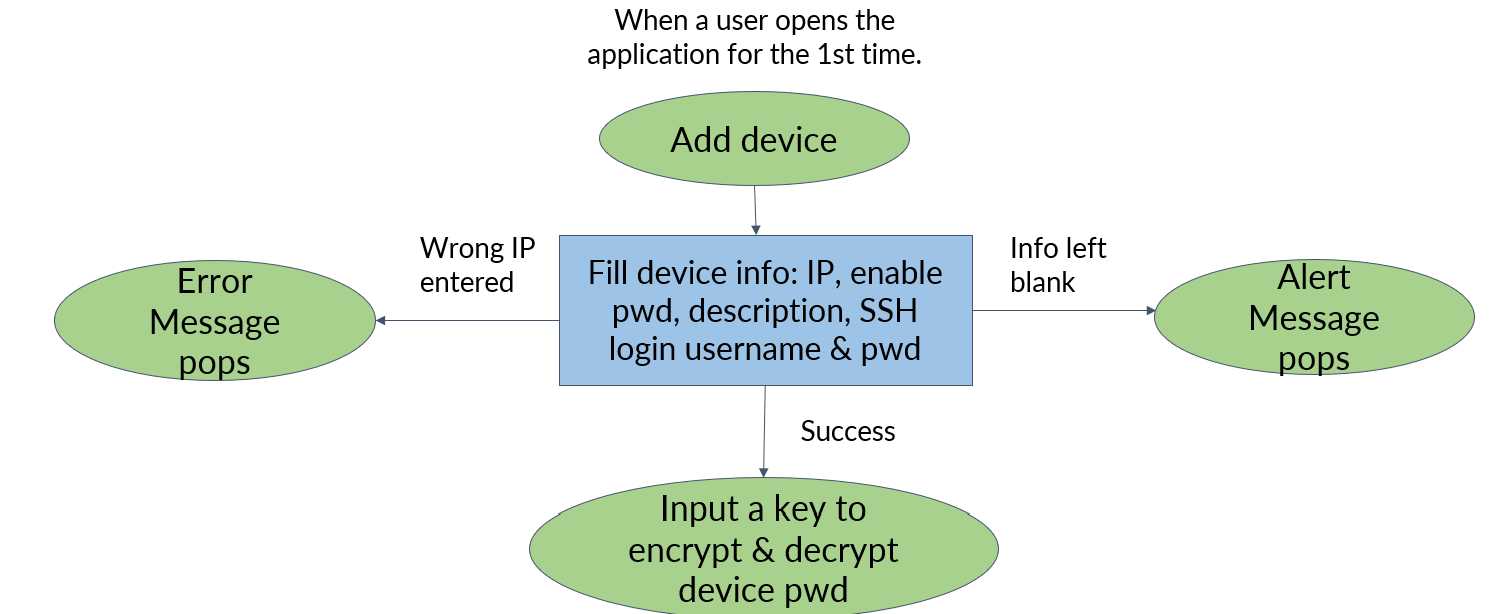
To install GNS3 VM the OVA file has to be opened in VMWare workstation player, GNS3 GUI should not be running through the process, from player->file->open menu. It is important to keep the path in the red rectangle at the default. After creating a new blank project (If GNS3 GUI was running it should be restarted) GNS3 VM should be enabled from virtualization engine select which can be found under edit -> preferences menu on VMWare workstation Player

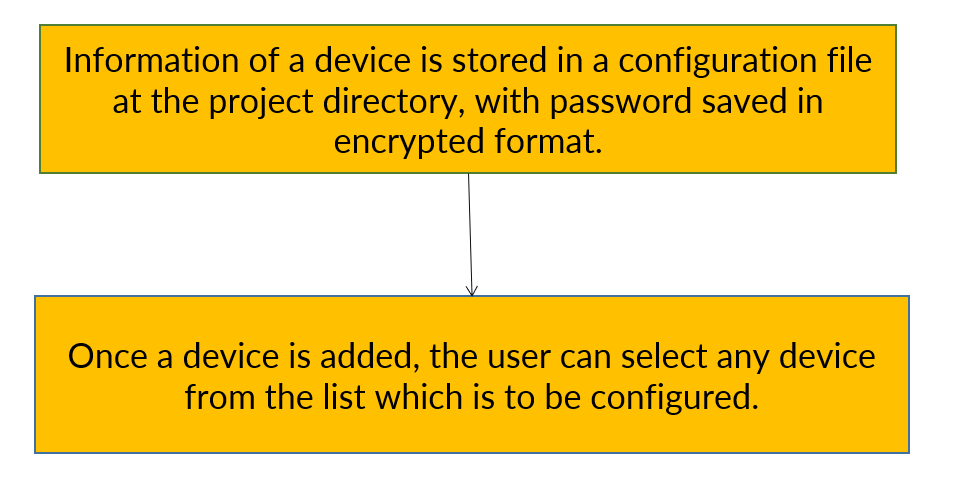


On VM name the previously downloaded the OVA GNS3 VM ought to be selected but if nothing is displayed try to restart the GNS3 GUI and go to help->setup wizard. In the wizard choose Run modern IOS and click next until completion. If an error message appears the vix-api probably needs to be installed.

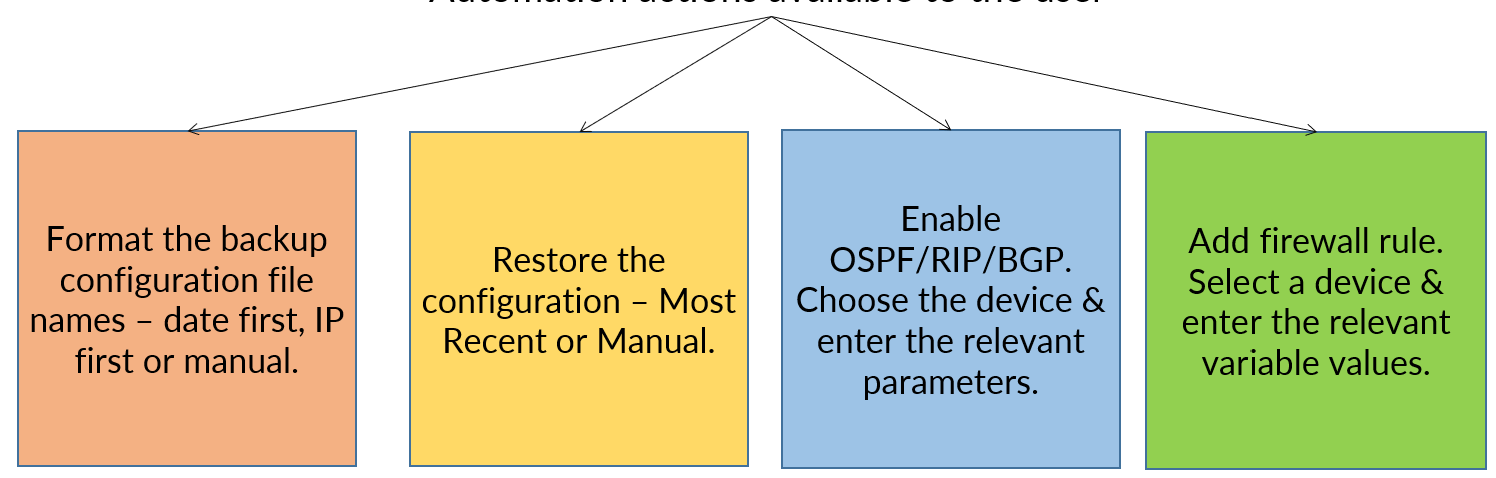
When GNS3 VM boots if the above message appears take Ownership. If at this stage of the process an error message appears restarting both VMWare Player and the GNS3 VM GUI may solve it. If everything is installed correctly under servers summary there should be now 2 servers, the PC that the GNS3 GUI is installed and the GNS3 VM.



**FLOW CHART**

****

**AUTOMATION ACTIONS**

****

**PROGRAM’S DESIGN**

**-OBJECTIVE**

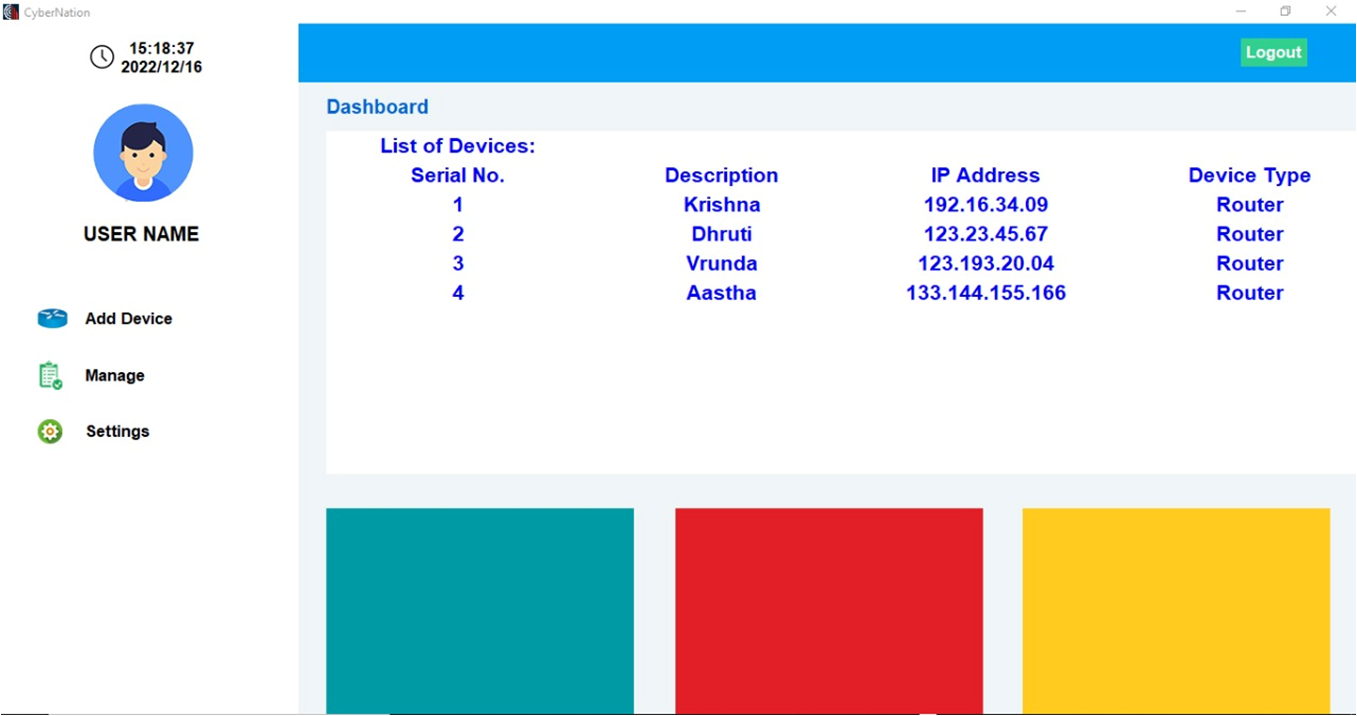
The user will be able to add a device's information (IP, enable password, type, hostname, SSH username, SSH password), which will be recorded in a CSV type file, through the application's graphical interface. When the credentials are used to open an SSH connection with the device, they are decrypted after being stored in an encrypted manner. A list of all the saved devices will be accessible for selection. It is possible to choose one or more devices. The user will then need to pick which automated method will be used after selecting the devices, and depending on the operation, additional parameters may need to be configured.

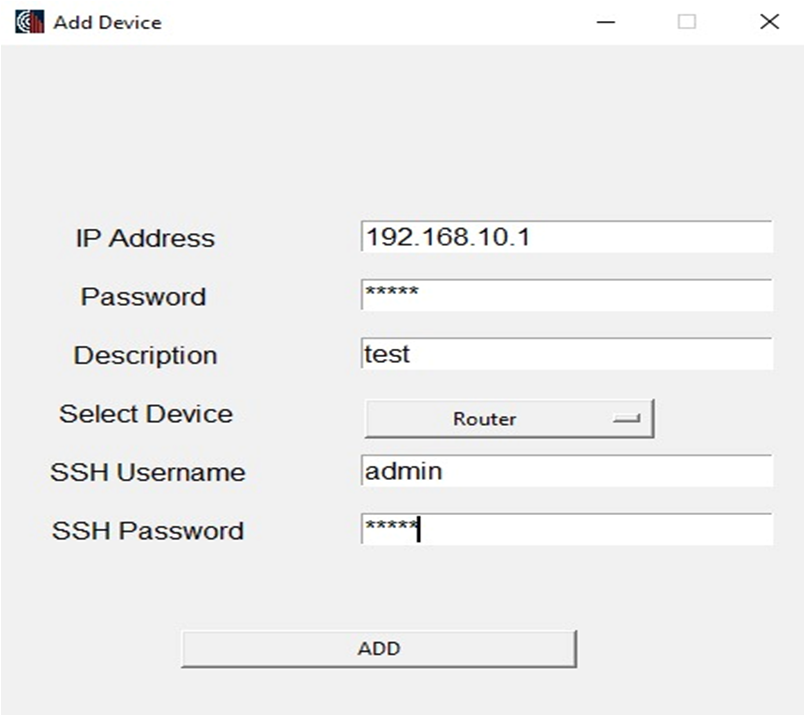
**-SOFTWARES AND HARDWARES USED**

* Python and VsCode community editions were used for coding and executing the script.
* GNS3, cisco IOS images, were used in order to set up a testing environment and windows 10 laptop from which all the applications were running.

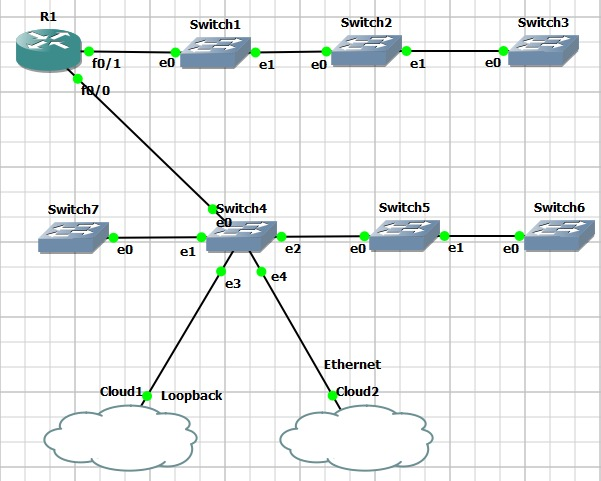
**-DEVELOPMENT**

**APPLICATION DEMO**

****

****

**NETWORK TOPOLOGY**

****

**5 CONCLUSIONS**

The goal of this application is to serve as both a starting point for an application that can be applied in a real setting and a demonstration of the fundamental concept surrounding network automation with Python. It needs to be adapted to meet the requirements of the network environment in which it will operate and the user inputs check needs to be improved. Additionally, some of the code needs to be updated because of the majority of the application's demonstration purposes.

An automation plan will help organizations with change control, architecture, security, and operational management. When automated systems continuously scan the network, troubleshooting may be done fast and easily.

The idea of software controllability is growing throughout the networking industry thanks to the creative applications of SDN. It is possible to automate the configuration and monitoring of any device, irrespective of the vendor, on SDN devices as well as other networking solutions.

We have shown that network engineers may use Python to automate the configuration of devices rather than having to manually configure each one. They only need to set up the necessary infrastructure and automation scripts. As a result of network events, network controllability improves and changes can be implemented more quickly. So, with SDNs, the components of older networks are becoming comparable.

**FUTURE SCOPE**

This application’s purpose is to demonstrate the basic idea around network automation with python and as a starting point for an application that can be used in real environment. It needs improvements on the user inputs check and it needs to be customized to fill to the needs of the network environment that it will run. Also, because most of the demonstration purpose of the application there are parts of the code that needs to be improved.

**BIBLIOGRAPHY**

1. Mihăilă, Paul, et al. "Network Automation and Abstraction using Python Programming Methods." *MACRo 2015* 2.1 (2017): 95-103.
2. Choi, Brendan. "GNS3 Basics." *Introduction to Python Network Automation*. Apress, Berkeley, CA, 2021. 415-489.
3. Santyadiputra, G. S., I. M. E. Listartha, and G. A. J. Saskara. "The effectiveness of Automatic Network Administration (ANA) in network automation simulation at Universitas Pendidikan Ganesha." *Journal of Physics: Conference Series*. Vol. 1810. No. 1. IOP Publishing, 2021.
4. Choi, Brendan. "Introduction to Python Network Automation." *Introduction to Python Network Automation*. Apress, Berkeley, CA, 2021. 1-21.
5. Milios, George. "Network Automation using Python." (2021).
6. Abdelhak, B. O. U. M. E. Z. R. A. G. "A Proposed of Novel Network Management Platform for Network Automation and Programmability with Implementation on GNS3."

**Acknowledgements**

**ACKNOWLEDGEMENTS**

We appreciate the opportunity to work on this topic for our major-project at K. J. Somaiya College of Engineering. We are grateful to Dr. Shubha Pandit, our principal, and Dr. Bharati Singh, our HOD, for giving us with this opportunity.

We would like to thank Prof. Jyoti Varavdekar, our project guide, for allowing us to complete this project under her expert guidance. We are incredibly grateful for her time and care. Throughout the course of this short project, she taught us extremely valuable skills. Her helpful comments and frequent support helped us to focus and shape our learning throughout the process. Each of us researched the subject using various sources of information and did our best to make this major-project a success.