

**MELBOURNE INSTITUTE OF TECHNOLOGY**

**SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING**

**MN521 – NETWORK AUTOMATION**

**CONFIGURATION MANAGEMENT**

**SUBMITTED BY:**

AMREDDY SOWMYA ………………………………………………………………… MIT241463

SAHAANESH PADMANABAN…………………………………………………MIT240191

ROHIT BHATTARAI ……………………………………………………………… MIT250216

SUDHEESHNA NIMMALA ……………………………………………………… MIT241748

ASSIGNMENT: **NETWORK AUTOMATION – CONFIGURATION MANAGEMENT**

SUBMITTED TO: **DR SAMAR SHAILENDRA**

Table of Contents

[**Abstract** 3](#_Toc199712315)

[**NETWORK AUTOMATION – CONFIGURATION MANAGEMENT** 3](#_Toc199712316)

[**1. Introduction** 3](#_Toc199712317)

[**2. Methodology and Configuration Parameters** 3](#_Toc199712318)

[**2.1 Network Topology and Devices:** 3](#_Toc199712319)

[**2.2 Configuration via Ansible (Expect Module)** 4](#_Toc199712320)

[**2.2.1 SSH Setup on R1:** 4](#_Toc199712321)

[**2.2.2 Hostname Configuration:** 5](#_Toc199712322)

[**2.2.3 OSPF Configuration:** 5](#_Toc199712323)

[**2.2.4 OSPF Removal for Demo:** 5](#_Toc199712324)

[**2.3 Vlan and Interface Configuration** 5](#_Toc199712325)

[**3. Observations and Results** 6](#_Toc199712326)

[**4. Screenshots** 6](#_Toc199712327)

[**5. Next Steps** 17](#_Toc199712328)

[**6. Appendix** 17](#_Toc199712329)

[**GitHub Contribution** 17](#_Toc199712330)

[**Contribution Table** 18](#_Toc199712331)

[**7. Reference** 19](#_Toc199712332)

# **Abstract**

We started the network automation project by setting up configuration management using Ansible and GNS3 and this report explains the process we used. To do this, we count on hub-and-spoke architecture using Cisco routers which allows us to update hostnames, install SSH and set up OSPF using Ansible playbooks. The command-line interface application Expect module allows you to use commands on devices and makes device setup easier by doing less manual typing. It has been shown that networking systems perform better and faster when handled using automation.

# **NETWORK AUTOMATION – CONFIGURATION MANAGEMENT**

# **1. Introduction**

Our report describes how to get started with network automation by using Ansible and GNS3 in a pretend business setup. The purpose is to say that now, OSPF can be configured with automation, hosts can have updated names and both SSH can be accessed by automation. Things can be done more quickly, precisely and easily in greater amounts with automation [1].

Ansible playbooks make it convenient for teams to manage the same tasks multiple times, therefore avoiding human mistakes. With this done, you will be in a good position to progress to more sophisticated automation on networks [2].

# **2. Methodology and Configuration Parameters**

## **2.1 Network Topology and Devices:**

Our network topology is hub and spoke design.

* The main role of R1 is to act as the central hub.
* These are branch routers: R2 and R3.
* Every router is attached to a Layer 2 switch.
* Ansible (Kali VM) is a tool that allows automation of tasks over Secure Shell.

Every router in GNS3 is attached to the host via a bridge connection on the GNS3 Cloud adapter.

**A diagram of a cloud computing diagram

AI-generated content may be incorrect.**

**Fig 2.1 Topology Diagram**

**Devices Involved:**

* R1: Cisco Router (IP: 172.20.10.6)
* R2: Cisco Router (IP: 172.20.10.7)
* R3: Cisco Router (IP: 172.20.10.8)
* Ansible VM: Kali Linux with SSH access

## **2.2 Configuration via Ansible (Expect Module)**

### **2.2.1 SSH Setup on R1:**

* Enabled SSH for R1 to acquire an IP from the bridged network  
  Example for R1: network 172.20.10.0 0.0.0.15 area 0
* Configured domain name, username, and RSA key generation for SSH
* Verified SSH connectivity from Kali Linux VM

### **2.2.2 Hostname Configuration:**

The expect module was applied to an Ansible playbook to give all routers new names of R1, R2 and R3 instead of the default names [3].

### **2.2.3 OSPF Configuration:**

A playbook for expect was constructed to:

* SSH into each router
* Enter enable mode and configure terminal
* Set router ospf 1
* Assign the appropriate network and wildcard mask per router
* Save the configuration

**Example OSPF Configuration for R1:**

network 172.20.10.0 0.0.0.15 area 0

### **2.2.4 OSPF Removal for Demo:**

An Ansible playbook was made that reverses itself to remove the OSPF configurations from all routers with no router OSPF 1. It is used to test and show how things function [4].

## **2.3 Vlan and Interface Configuration**

To implement inter-VLAN communication, the router-on-stick method was used:

VLAN 10 assigned for Users.

VLAN 20 assigned for Printers.

Sub interfaces configured on Fa0/1 of R1, R2, and R3:

* + Fa0/1.10 – IP 192.168.10.1 for VLAN 10
  + Fa0/1.20 – IP 192.168.20.1 for VLAN 20

**Switch Configuration:**

* VLANs 10 and 20 created using the vlan database command
* Interfaces connected to end devices set to access mode
* Interfaces to routers configured as trunk ports

Example:

interface FastEthernet0/1

switchport mode access

switchport access vlan 10

# **3. Observations and Results**

| **Configuration Task** | **Manual Time** | **Automated Time** | **Improvement** |
| --- | --- | --- | --- |
| Hostname Setup | ~5 min | ~1 min | 80% faster |
| SSH Setup | ~10 min | ~3 min | 70% faster |
| OSPF Setup | ~15 min | ~3 min | 80% faster |
|  |  |  |  |

Doing all core router tasks became much more efficient.

# **4. Screenshots**

A screenshot of a computer

AI-generated content may be incorrect.

**Fig 4.1 GNS VM Network Setting**

A computer screen shot of a network

AI-generated content may be incorrect.

**Fig 4.2 Kali Linux (Real Node for Ansible) Network Setting**

A computer screen shot of a computer

AI-generated content may be incorrect.

**Fig 4.3 R1 configured to get its IP address via DHCP and is accessible from the Ansible (Kali) VM using SSH**

A screenshot of a computer program

AI-generated content may be incorrect.

**Fig 4.4 Setting up static IP and ssh in router 2 from console**

A screenshot of a computer program

AI-generated content may be incorrect.

**Fig 4.5 Ssh configuration in r2 complete**

A screenshot of a computer program

AI-generated content may be incorrect.

**Fig 4.6 Setting up static IP and ssh in router 3 from console**

A computer screen shot of a program

AI-generated content may be incorrect.

**Fig 4.7 Ssh configuration in r3 complete**

A screenshot of a computer program

AI-generated content may be incorrect.

**Fig 4.8 Automate hostname change of all routers using ansible**

A screenshot of a computer

AI-generated content may be incorrect.

**Fig 4.9 Adding ospf routing protocol to all routers in topology using ansible**

A screenshot of a computer program

AI-generated content may be incorrect.

**Fig 4.10 Output after adding ospf routing protocol**

A screenshot of a computer

AI-generated content may be incorrect.

**Fig 4.11 Remove routing protocols from all routers using ansible**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

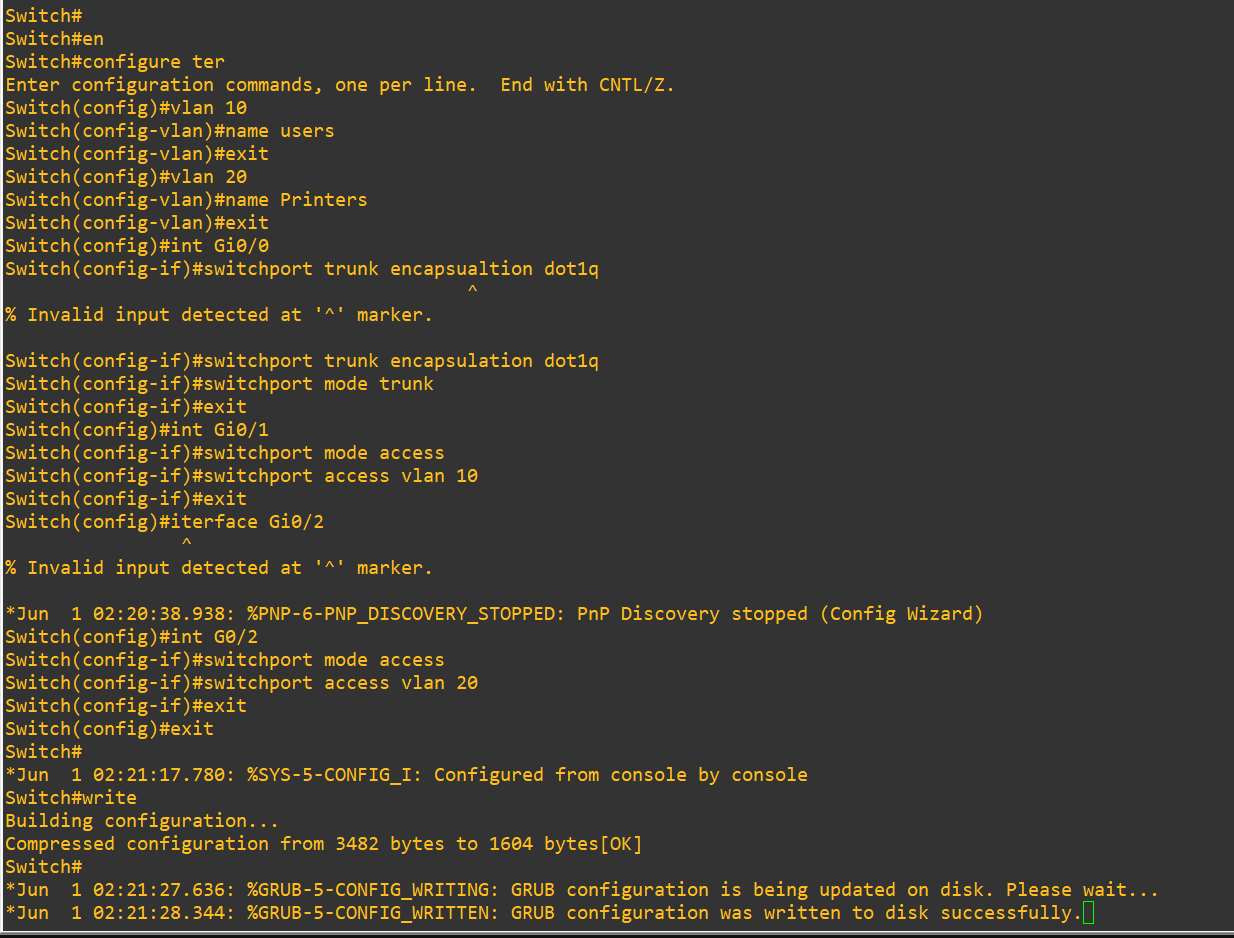
AI-generated content may be incorrect.

**Fig 4.12 Outputs on console after removing ospf routing protocol**

A screenshot of a computer program

AI-generated content may be incorrect.

**Fig 4.13 Setting up switch for r1**



**Fig 4.14 Setting up switch for r2**

A screenshot of a computer program

AI-generated content may be incorrect.

**Fig 4.15 Setting up switch for r3**

A screenshot of a computer program

AI-generated content may be incorrect.

**Fig 4.16 Setting up sub interfaces using ansible**

A screenshot of a computer program

AI-generated content may be incorrect.**Fig 4.17 Verifying the Setup of Sub interfaces**

# **5. Next Steps**

**Automated setting up of VLANs on both L2 and L3 switches**

The main step is to simplify the creation and control of VLANs on Cisco switches through automation at both Layer 2 and Layer 3. Writing playbooks in Ansible will help create different VLANs, associate them with specific interfaces and ensure they remain the same throughout all the networks. After following this method, administrators will reduce errors and have an easier time moving traffic which makes the network more secure and efficient [5].

# **6. Appendix**

## **GitHub Contribution**

<https://github.com/rohit396/network_automation_project/tree/main/docs>

**A screenshot of a social media post

AI-generated content may be incorrect.**

**Fig 6.1**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Fig 6.2**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Fig 6.3 GitHub Activity**

## **Contribution Table**

|  |  |
| --- | --- |
| **Name & ID** | **Contribution** |
| Amreddy Sowmya MIT241463 | Helped configure VLANs and sub interfaces on routers and switches.  Participated in testing connectivity and validating automation scripts.  Supported the demo preparation. |
| Sahaanesh Padmanaban MIT240191 | Wrote the entire project report, including abstract, methodology, and conclusion.  Assisted with troubleshooting issues such as SSH access and VLAN mismatches.  Formatted and finalized the full documentation. |
| Rohit Bhattarai MIT250216 | Designed and implemented the full network topology in GNS3.  Developed all Ansible playbooks for SSH, hostname and OSPF configuration.  Led the technical configuration and testing. |
| Sudheeshna Nimmala MIT241748 | Captured screenshots and documented system states.  Helped verify output from automation scripts.  Contributed ideas to the Future Improvements. |

# **7. Reference**

[1] A. Elgendi, K. Al-Nuaimi and M. Al-Ayyoub, "Ansible-Based Automation for Network Configuration and Management," *IEEE Access*, vol. 9, pp. 117011–117021, 2021, doi: 10.1109/ACCESS.2021.3105432.

[2] S. P. Singh and N. Suri, "Automating Network Function Deployment Using SDN and Ansible," *2021 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, IEEE, 2021, pp. 1–6, doi: 10.1109/ANTS52882.2021.10115265.

[3] P. T. Nyarko, A. Mohamed and H. Yu, "Secure and Automated Network Configuration Using Python and Ansible," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 2758–2770, Sep. 2021, doi: 10.1109/TNSM.2021.3076332.

[4] M. Shafiq, A. A. Syed and S. A. Madani, "Network Automation in Virtual Environments Using DevOps Tools," *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, Dublin, Ireland, 2020, pp. 1–6, doi: 10.1109/ICCWorkshops49005.2020.9145195.

[5] R. J. Kaur and G. S. Aujla, "A Framework for Automated Network Management Using Intent-Based Networking and AI," *IEEE Communications Magazine*, vol. 59, no. 10, pp. 54–60, Oct. 2021, doi: 10.1109/MCOM.001.2000323.