

**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**

# **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 DHCP 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.

Table of Contents

[**Abstract** 2](#_Toc199634561)

[**NETWORK AUTOMATION – CONFIGURATION MANAGEMENT** 3](#_Toc199634562)

[**1. Introduction** 3](#_Toc199634563)

[**2. Methodology and Configuration Parameters** 3](#_Toc199634564)

[**2.1 Device Involved:** 3](#_Toc199634565)

[**2.2 Configuration via Ansible (Expect Module)** 3](#_Toc199634566)

[**2.2.1 SSH and DHCP Setup on R1:** 3](#_Toc199634567)

[**2.2.2 Hostname Configuration:** 3](#_Toc199634568)

[**2.2.3 OSPF Configuration:** 3](#_Toc199634569)

[**2.2.4 OSPF Removal for Demo:** 4](#_Toc199634570)

[**3. Observations and Results** 4](#_Toc199634571)

[**4. Screenshots** 5](#_Toc199634572)

[**5. Next Steps** 12](#_Toc199634573)

[**6. Appendix** 12](#_Toc199634574)

# **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 and DHCP 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 Device 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 and DHCP Setup on R1:**

* Enabled DHCP for R1 to acquire an IP from the bridged network
* 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].

# **3. Observations and Results**

| **Configuration Task** | **Manual Time** | **Automated Time** | **Improvement** |
| --- | --- | --- | --- |
| Hostname Setup | ~5 min | ~1 min | 80% faster |
| SSH/DHCP 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 diagram of a cloud computing diagram

AI-generated content may be incorrect.**

**Fig 4.1 Topology Diagram**

A screenshot of a computer

AI-generated content may be incorrect.

**Fig 4.2 GNS VM Network Setting**

A computer screen shot of a network

AI-generated content may be incorrect.

**Fig 4.3 Kali Linux (Real Node for Ansible) Network Setting**

A computer screen shot of a computer

AI-generated content may be incorrect.

**Fig 4.4 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 4.5 Setting up static IP and ssh in router 2 from console**

A screenshot of a computer program

AI-generated content may be incorrect.

**Fig 4.6 Ssh configuration in r2 complete**

A screenshot of a computer program

AI-generated content may be incorrect.

**Fig 4.7 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 4.8 Ssh configuration in r3 complete**

A screenshot of a computer program

AI-generated content may be incorrect.

**Fig 4.9 Automate hostname change of all routers using ansible**

A screenshot of a computer

AI-generated content may be incorrect.

**Fig 4.10 Adding ospf routing protocol to all routers in topology using ansible**

A screenshot of a computer program

AI-generated content may be incorrect.

**Fig 4.11 Output after adding ospf routing protocol**

A screenshot of a computer

AI-generated content may be incorrect.

**Fig 4.12 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 4.13 Outputs on console after removing ospf routing protocol**

# **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].

**Cisco 3640 for SVI Routing**

Setting up Switched Virtual Interfaces (SVIs) will enable inter-VLAN routing on the Cisco 3640 routers. SVIs will be created for every VLAN by automation which will also allot each one an IP address and switch on the necessary routing. Ansible allows you to automate actions at both Layer 2 and Layer 3 which makes it easier for a large enterprise with many segments to talk between VLANs.

**Automating How Web Interfaces Are Managed**

You need to now add both simple and advanced settings for configuring all the network device interfaces at this stage. Part of your job is giving out IP addresses, turning on interfaces, providing descriptions and ensuring port-security or changing the mode for non-essential ports. With automation, the environment stays the same every time, setup is quicker and large deployments become simpler.

# **6. Appendix**

**Git history**

**Contribution**

# **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.