Implementation of Network Automation Using Ansible to Configure Routing Protocol in Cisco and Mikrotik Router with Raspberry PI

Muhammad Fauzi Islami, Purnawarman Musa and Missa Lamsani

Computer System
Faculty of Computer Science and Information Technology
Gunadarma University
E-mail: islamifauzi@gmail.com

Abstrak

Teknik Jaringan tidak pernah dipisahkan dengan konfigurasi yang merupakan tugas utama yang harus dilakukan dengan tujuan untuk membuatnya berfungsi. Seperti yang kita ketahui, untuk mengkonfigurasi beberapa perangkat baik server, switch, atau router memiliki beberapa metode jika kita mengetahui bagaimana kita berkomunikasi dengan sistem. Kita dapat menggunakan port konsol untuk berkomunikasi, menggunakan telnet untuk berkomunikasi jarak jauh, atau bahkan menggunakan SSH jika kita ingin berkomunikasi dari jarak jauh dan juga dengan aman dengan semua informasi yang dienkripsi. Secara umum, metode konvensional seperti metode yang baru saja dijelaskan sebelumnya selalu digunakan. Masalahnya belum datang karena hanya ada 2 sampai 5 perangkat, tetapi akan melelahkan ketika kita masuk ke setiap perangkat. Makalah ini menggambarkan metode baru dalam mengkonfigurasi perangkat jaringan dengan menggunakan otomatisasi, mengurangi waktu untuk konfigurasi peralatan dan lebih mudah untuk pemeliharaan. Ini mengurangi kesalahan manusia dalam sintaks konfigurasi dan bagaimana cara perangkat dikonfigurasi. Inovasi akan mewujudkannya. Ada alat untuk melakukan tugas-tugas ini seperti: Ansible dan Terraform. Dalam tulisan ini, Ansible dipilih karena kesederhanaannya.

***Kata Kunci*: :** Otomasi jaringan, rekayasa jaringan, Python, Ansible, jaringan.

Abstract

Network Engineering is never separated with configuration which is the main task must be committed in the purpose to put it into function. As we know, to configure some devices either server, switch, or router have some methods if we ob-serve by how we communicate to the system. We can use the console port to communicate, using telnet to communicate remotely, or even using SSH if we want to communicate remotely and also securely with all of the information is encrypted. Generally, the conventional method such as the methods has just explained before is always be used. The problem has not come yet since there are just 2 until 5 devices, but it will be exhausting when we log in to each network device. This paper is illustrating a new method in configuring network devices by using automation, reducing time for equipment configuration and easier to maintenance. It reduces human error in configuration syntaxes and how the way the device is configured. The innovation is going to make it happen. There are tools to do these tasks such as : Ansible and Terraform. In this paper, Ansible is chosen because of its simplicity.

Keywords : network automation, network engineering, Python, Ansible, network.

Introduction

Network Automation is common sense. It's a culture and collaborative effort to operate networks in a way, which is in line with the organization goals and mission. The network engineering has been progressing year by year and there are many problems on it. For example, is in configuration management. It has always been an exhausting occupation if we have to configure many network devices one by one in a large network area. Therefore, some innovations to handle this problem has to be invented[1].

Network Automation is a solution that offered these days which is efficiency in every aspect is priority. The term of network automation comes from how the way we can automate network management, testing, deployment, and either operating physical devices or virtual in a network with a run-

DOI : http://dx.doi.org/10.32409/jikstik.19.2.2766

ning program in a control node. With this new method, a network engineer's task is eased in configuring many network devices just with a program which runs on control node. It helps many tasks to be done, it will not waste our time, and it reduces human errors because all of the configurations will be written and run in a script. Every tasks and functions are performed automatically. It will simplify the network configurations into a file. The file which is used will accomplish repeatable tasks and orchestrate the process sequentially.

Automating network is a more efficient method than traditional method to support the rapid growth of network traffic generated by social media, video, and another data. Net-work automation has become more available to many businesses. There are many implementations of network automation which can be applied to virtualized environments, data centres, public and private clouds.

Many methods to do this kind of automation such as, utilizing network programmability which uses some libraries that exist in python programming language such as, Netmiko and Paramiko. There are also some tools to do this such as, Ansible, Terraform, SaltStack, Chef, Puppet, and Napalm. In this paper, ansible is chosen[2][5].

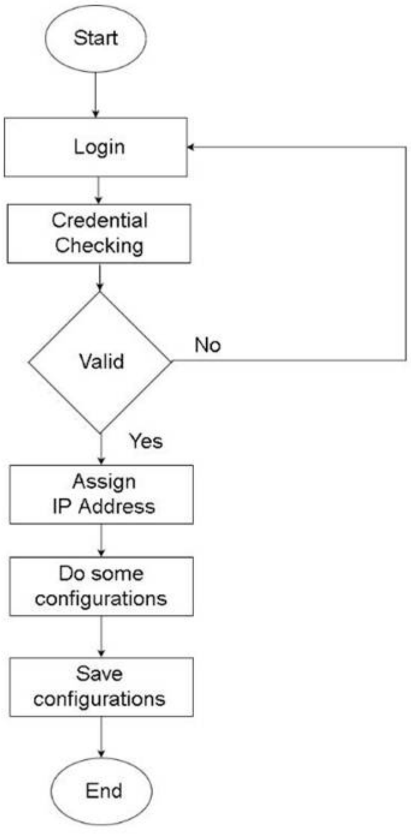


Figure 1: Conventional Step

Method

Before automation joining into network engineering, we have to configure one by one devices either routers or switches. It is a wasting time if we want to configure some network devices with some similarities to other devices. Therefore, we need some methods to do it repetitively with-out missing any configurations as well. Here is Fig.1 that depicts how we configure network devices manually.

Based on the picture, it tells us that we have to do it manually step by step just for one device, it will be an annoying task if we face more than 10 devices to be configured. With automation we can configure many devices concurrently just within seconds. Repetitive is power of efficiency in this case, it will not replace device's configuration that has been con-figured earlier except there are some changes on that configurations itself.

Since the Fourth Industrial Revolution has come, it has changed many aspects in technology into efficiency and close on more simplicity approach. Network automation is one of the developments in this circumstance. Here is a flowchart for network automation in general:

Fig. 2 is a flowchart how network automation works generally. As we can see that network automation is completely simpler than the conventional method. We just need to write the configuration of the devices in it then connect it via SSH, hence it can deploy configuration concurrently into devices just with a single click.

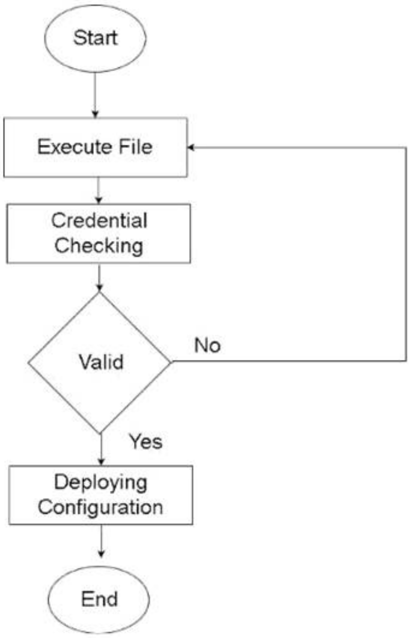


Figure 2: Network Automation Step

Ansible

Figure 3: Ansible[1]

Ansible is a provisioning tool, configuration management, and also an application deployment which is developed by RedHat. A System Administrator or even Network Engineer are able to automate and write the deployment or even configuration that is usually repetitively to some devices such as server or router. Certainly, as a person who has responsibility in network infrastructure management, either the installation of a service, configuration or keep the system running well.

YAML File

Network automation eases network engineer to make it simple as it just executes a single file to deploy many con-figurations that is written in certain format such as YAML. YAML is a file that is written in human-readable format with indentation as an alternate of curly bracket which normally exists in common programming language. It seems like python's rule but there is a difference between them. That is the indentation in YAML file does not support tabulation to be used except spaces. It will be easier to read even a beginner can read it.

YAML file has some rules that we do obediently, thereare:

1. Spaces are used to tag the writing structure
2. Tabulation is prohibited as permitted as python
3. The using of — sign is written on the first line
4. The using of ... sign is used if we want to end the document within the same file)
5. Using # (hashtag) to give a comment on a certain line
6. The sign - (strip) shows the list member on each object sub-divisions
7. Using: (colon) sign to give a value to every key or variable
8. Using; (semi-colon) sign to make an array

Benefit of Using Ansible

The benefit of using ansible as an automation tool in infrastructure is explained below:

1. Simple

   Ansible uses simple syntax to be written in YAML for-mat which is called as playbook.

2. Agentless

   Software as an agent is not necessary to be installed on the host target.

3. Powerful & Flexible

   Ansible has many modules to manage network infrastructure and its service in operating system.

4. Efficient

   It does not need any space in hard drive of host because ansible is agentless.

Terminology of Ansible

In ansible there are some terminologies which is important to understand because it will make ansible easier to use. Here are some terminologies with its explanation:

1. Controller Node

   Node or also called as machine where ansible is installed and responsible to do provisioning on the managed infrastructure.

2. Inventory

   Initialization file which contains information about de-vices on the managed infrastructure.

3. Playbook

   The entrance node for ansible to do provisioning where automation is defined through tasks that is written in YAML format.

4. Module

   Abstraction of the entire system, such as something related with package or in making and modifying file. Ansible has many built-in modules or some modules which has installed when the installation.

5. Role

   This is how playbook is managed and another file to facilitate any other usage and use it back from provisioning step.

6. Play

   Provisioning that is executed early until the end is called as play. In other word, the execution of playbook is called as play.

7. Facts

   The global variable which contains information about system, such as the installed interfaces or system operation.

8. Handlers

   Handlers is used to trigger some changes status from the service, such as restarting or stopping the service.

9. Vault

   Vault is used to secure sensitive information such as username and password with AES256 encryption. 2.1.4 Ansible Structure net protocol networks which divide them into some au tonomous system.

Ansible Structure

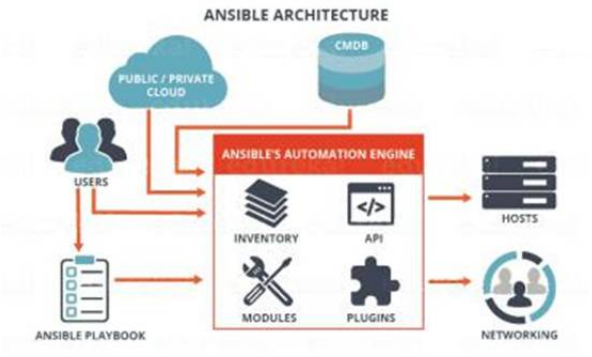


Figure 4: Ansible Architecture [1]

Based on the picture above, it explains how ansible has a simple architecture but has many functionalities which can handle the complexity of network infrastructure.

Routing Protocol

Thousands of devices are connected to each other around the world. Therefore, internet exists for people to do their businesses more efficient and faster. These devices are named router. It has many capabilities to manage the network traffic, one of them is routing. Routing is simply defined as a method of forwarding the network packet from source node to destination node. There are some types of routing protocol and each of them have differences in deciding the best route to reach the destination.

Types of Routing Protocol

1. RIP (Routing Information Protocol) RIP is a routing protocol type which often be used for local network only because this routing protocol is categorized as interior gateway protocol which means only local area network is covered. It has distance vector algorithm in its implementation. It uses the distances of router's hop as metrics to decide the best path.

2. OSPF (Open Shortest Path First) OSPF is particularly categorized as interior gateway protocol. It covers more than RIP does, hence it is the right decision to consider OSPF as routing protocol for high scalability and dividing area by autonomous systems. OSPF uses dijkstra algorithm to decide the best path to reach the destination source

3. EIGRP (Enhanced Interior Gateway Routing Protocol) Enhanced Interior Gateway Routing Protocol (EIGRP) based on their original IGRP while it is a Cisco proprietary routing protocol. It is the next level of distance-vector routing protocol that optimize it by mixing it with link-state advantages.

4. BGP (Border Routing Protocol) Border Gateway Protocol (BGP) are the only core routing protocol of internet as known as exterior routing protocol and responsible to maintain the table of inter

Raspberry Pi

Raspberry PI Foundation is the first non-profit founda-tion who develop this product.The main purpose of this product is to be used for whoever want to learn programming with its powerful GPIO (General Purpose Input Output)[3][4].

Figure 5: Raspberry Pi[3]

GNS3 (Graphical Network Simulator 3)

Figure 6: GNS3 Logo[6]

GNS3 is a network emulator with GUI (Graphical User Interface) which was released in 2008. With GNS3 we can simulate and emulate as we use a real device. This emulator is able to run in many operation system such as Windows, Linux, and OSX.

The differences between GNS3 and any other network simulator like Cisco Packet Tracer is GNS3 can simulate and emulate different vendor devices. Therefore, not only cisco device which can be simulated in GNS3. Furthermore in GNS3, user can explore more because it is created as close as in the real world of network engineering with real devices[6].

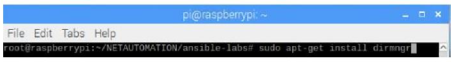
Design and Analysis

Raspberry Pi is similar to any other computer in com-mon but for this case has a limitation related to ansible version. In raspberry, version 2.2.0 is the latest version but commonly ansible has the latest version, that is version 2.8.0. The different between both is in version 2.2.0 has some modules which has not been made like in version 2.8.0. In this re-search, 2.8.0 is needed because there are many module which has been made.

Setup

First, the latest version of ansible must be installed in raspberry pi. There is certain way to install the latest version of ansible in raspberry pi. Here is step by step to do it:

1. Installing Directory Manager

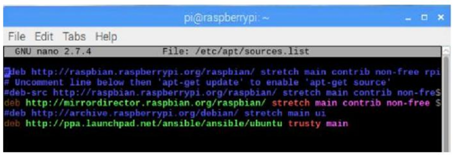


```
root@raspberrypi:~/NETAUTOMATION/ansible-labs# sudo apt-get install dirmngr
```

Figure 7: Installing Directory Manager

It is necessary to download and manage certificate revocation. As what has just told before, the latest version of ansible for raspberry pi is out of date if it is compared to the latest version for computer generally. Therefore, breaking the rules is a way.

2. Appending Repository



```
  GNU nano 2.7.4              File: /etc/apt/sources.list

#deb http://raspbian.raspberrypi.org/raspbian/ stretch main contrib non-free rpi
# Uncomment line below then 'apt-get update' to enable 'apt-get source'
#deb-src http://raspbian.raspberrypi.org/raspbian/ stretch main contrib non-fre$
deb http://mirrordirector.raspbian.org/raspbian/ stretch main contrib non-free $
#deb http://archive.raspberrypi.org/debian/ stretch main ui
deb http://ppa.launchpad.net/ansible/ansible/ubuntu trusty main
```

Figure 8: Appending Repository

This repository will fetch ansible from ubuntu. The final result is the version of ansible in ubuntu is added in raspberry pi and ready to update and install. Before updating repository, run this command for the certificate revocation sudo apt-key adv –keyserver keyserver.ubuntu.com –recv-keys 93C4A3FD7BB9C367.

3. Updating Repository

Update repository using the command below:

```
sudo apt update -y
```

Wait until the repository is finished to fetch ansible from ubuntu repository.

4. Installing Ansible

Install ansible using the command below:

```
sudo apt install ansible -y
```

This is the final step to install ansible and it will take less than 10 minutes to be installed successfully. It is necessary to download and manage certificate revocation. As what has just told before, the latest version

Designing Topology

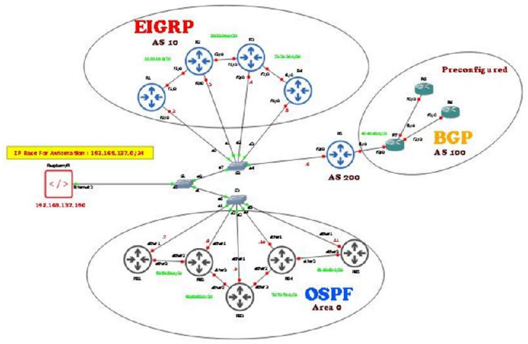


Figure 9: Topology

Based on the picture above there are 5 cisco devices and 5 mikrotik devices that will be configured automatically just the way executing the script. In the middle of this topology is the representation of raspberry pi which is connected to GNS3 through the physical port of the computer which runs GNS3.

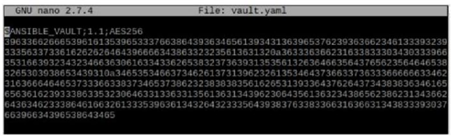
First, we have to setup the device manually by giving them an IP Address directly connected to the raspberry pi to manage it. Second, we have to install SSH service and create a username to grant the access to raspberry pi when it tries to connect. Third, we have to copy the SSH public key to each device to make it password less.

Preparing File

In configuring devices, ansible only needs a YAML File to deploy the configuration to all of the listed devices. There are some files which is related with YAML file. There are:

1. Vault File

   This file is used to save sensitive string such as username and password, because of this file, string is encrypted. To open it, we will be asked to input the password we put while the process of create vault file. If we want to see what is inside of it, there are just dump of random characters and numbers combined.



```
  GNU nano 2.7.4                    File: vault.yaml

$ANSIBLE_VAULT;1.1;AES256
39633662666539616135396533376638643936346561393431363965376239363662346133393239
33356337336162626264643966663438633232356136313 20a3633363662316338333034303339 66
35316639323432346636306163343362653832373639313535613263646635643765623564646538
32653039386534393 10a3465353466373462613731396232613534643736633736333666666633462
31636664646537333663383734653738623238383835616265313933643762643734383836346165
65636162393338633532306463313363313561363134396230643561363234386562386231343662
64363462333864616632613335396361343264323335643938376338336631636631343833393037
663966343965386434 65
```

Figure 10: Inside of Vault File

2. Hosts File

   This file contains the information of the managed de-vices such as its IP Address and DNS.

3. YAML File

   This file has just explained on the prior section. This file contains the configuration of devices which will be deployed.

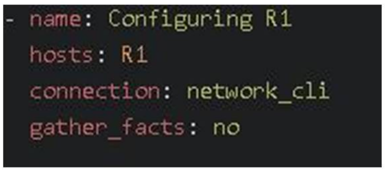
Creating YAML File

YAML file must be created to deploy configuration into the target devices. Here are the snippet of YAML file related to each vendor and function.

Cisco

This YAML file is dedicated to contain cisco configuration which will configure IP Address, EIGRP with Autonomous System 10 and eBGP Routing with Autonomous System 200. It is mentioned below:

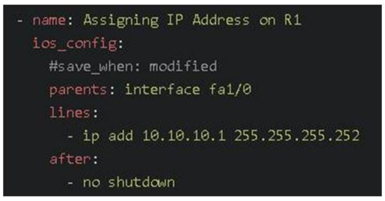
1. Determining which connection will be used and giving an explanation what will it do



```
- name: Configuring R1
  hosts: R1
  connection: network_cli
  gather_facts: no
```

Figure 11: YAML Information for Cisco

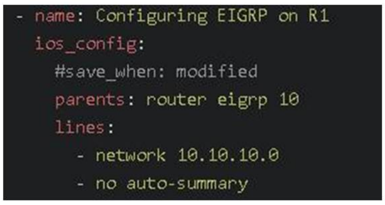
2. Assigning IP Address specifically into the interface



```
- name: Assigning IP Address on R1
  ios_config:
    #save_when: modified
    parents: interface fa1/0
    lines:
      - ip add 10.10.10.1 255.255.255.252
    after:
      - no shutdown
```

Figure 12: Assigning IP Address for Cisco o

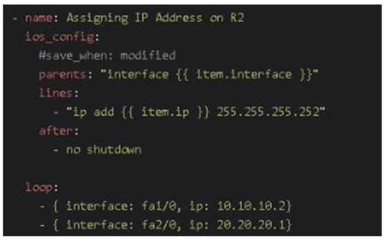
3. Configuring EIGRP to the router



```
- name: Configuring EIGRP on R1
  ios_config:
    #save_when: modified
    parents: router eigrp 10
    lines:
      - network 10.10.10.0
      - no auto-summary
```

Figure 13: Configuring EIGRP

4. Assigning IP Address if there are more than one IP Ad-dress will be assigned. It uses loop to configure the router repetitively.

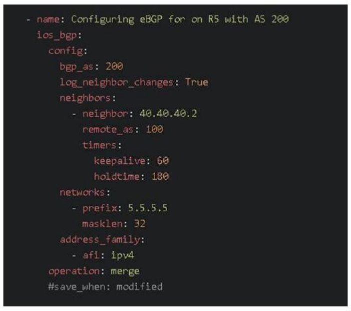


```
- name: Assigning IP Address on R2
  ios_config:
    #save_when: modified
    parents: "interface {{ item.interface }}"
    lines:
      - "ip add {{ item.ip }} 255.255.255.252"
    after:
      - no shutdown

  loop:
    - { interface: fa1/0, ip: 10.10.10.2}
    - { interface: fa2/0, ip: 20.20.20.1}
```

Figure 14: Assigning Two IP Address for Cisco

5. Configuring eBGP to the router.



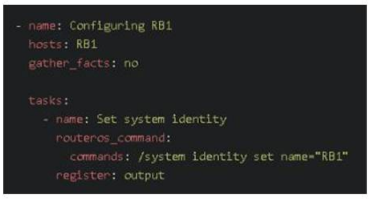
```
- name: Configuring eBGP for on R5 with AS 200
  ios_bgp:
    config:
      bgp_as: 200
      log_neighbor_changes: True
      neighbors:
        - neighbor: 40.40.40.2
          remote_as: 100
          timers:
            keepalive: 60
            holdtime: 180
      networks:
        - prefix: 5.5.5.5
          masklen: 32
      address_family:
        - afi: ipv4
    operation: merge
    #save_when: modified
```

Figure 15: Configuring eBGP

Mikrotik

This YAML file is dedicated to contain mikrotik con-figuration which will configure IP Address, Loopback Inter-face, Hostname, and OSPF Routing. It is mentioned below:

1. Determining which connection will be used, setting hostname and giving an explanation what will it do

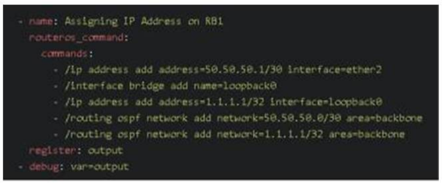


```
- name: Configuring RB1
  hosts: RB1
  gather_facts: no

  tasks:
    - name: Set system identity
      routeros_command:
        commands: /system identity set name="RB1"
      register: output
```

Figure 16: YAML Information for Mikrotik

2. Assigning IP Address specifically into the interface and configuring OSPF



```
- name: Assigning IP Address on RB1
  routeros_command:
    commands:
      - /ip address add address=50.50.50.1/30 interface=ether2
      - /interface bridge add name=loopback0
      - /ip address add address=1.1.1.1/32 interface=loopback0
      - /routing ospf network add network=50.50.50.0/30 area=backbone
      - /routing ospf network add network=1.1.1.1/32 area=backbone
  register: output
- debug: var=output
```

Figure 17: Assigning IP Address for Mikrotik and OSPF

Executing YAML File with Ansible

1. Here is the command to execute YAML file with Ansible to Cisco Routers:

   ansible-playbook cisco.yaml –ask-vault-pass -e@./vault.yaml

   after that there will prompt us to input the vault password.

2. Here is how to execute YAML file with Ansible to Mikrotik Routers:

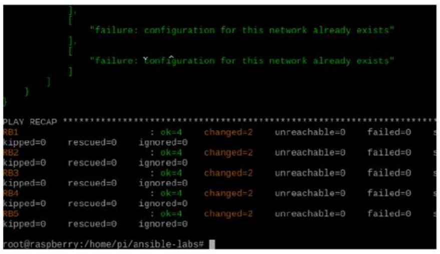
   ansible-playbook mikrotik.yaml –ask-vault-pass - e@./vault.yaml

   after that there will prompt us to input the vault password.

Result

After doing some steps from installation until executing the YAML file, it is time to see how the result looks like. In this section will be divided into two subsections, they are report of ansible after executing file and connection testing.

1. Report



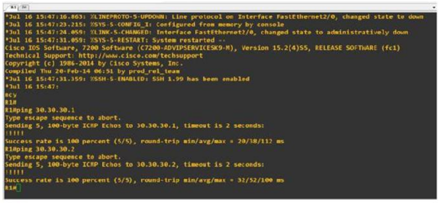
```
                ],
                [
                    "failure: configuration for this network already exists"
                ],
                [
                    "failure: configuration for this network already exists"
                ]
            ]
        }
}

PLAY RECAP *********************************************************************
RB1                        : ok=4    changed=2    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
RB2                        : ok=4    changed=2    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
RB3                        : ok=4    changed=2    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
RB4                        : ok=4    changed=2    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
RB5                        : ok=4    changed=2    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0

root@raspberry:/home/pi/ansible-labs#
```

Figure 18: Report for Mikrotik

Based on the picture above, there are 4 OK as a status of connection. If there are some changes on the configuration, the 'changed' parameter will show how many changes has been implemented. If there is some failure, the 'unreachable' and 'failed' parameter will count how many failures occurred. In this case only the 'OK' status that is showed up, it means there is either no any changes or failures occurred.

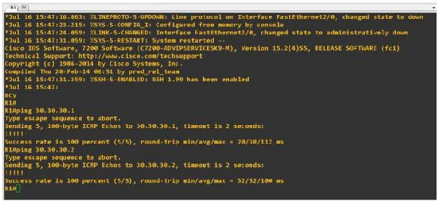


```
*Jul 16 15:47:16.863: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet2/0, changed state to down
*Jul 16 15:47:23.215: %SYS-5-CONFIG_I: Configured from memory by console
*Jul 16 15:47:24.059: %LINK-5-CHANGED: Interface FastEthernet2/0, changed state to administratively down
*Jul 16 15:47:31.059: %SYS-5-RESTART: System restarted --
Cisco IOS Software, 7200 Software (C7200-ADVIPSERVICESK9-M), Version 15.2(4)S5, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2014 by Cisco Systems, Inc.
Compiled Thu 20-Feb-14 06:51 by prod_rel_team
*Jul 16 15:47:31.359: %SSH-5-ENABLED: SSH 1.99 has been enabled
*Jul 16 15:47:
ncy
R1#
R1#ping 30.30.30.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.30.30.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/38/112 ms
R1#ping 30.30.30.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.30.30.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/52/100 ms
R1#
```

Figure 19: Report for Cisco

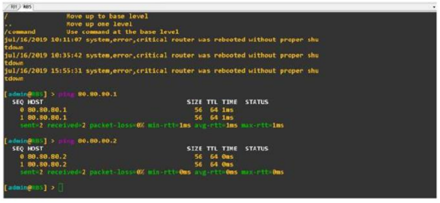
Fig. 19 depicts there are 4 OK and 4 Changes. It means 4 well-established connection and 4 changes occurred. It indicates this script is run successfully as expected.

2. Connection Testing



```
*Jul 16 15:47:16.863: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet2/0, changed state to down
*Jul 16 15:47:23.215: %SYS-5-CONFIG_I: Configured from memory by console
*Jul 16 15:47:24.059: %LINK-5-CHANGED: Interface FastEthernet2/0, changed state to administratively down
*Jul 16 15:47:31.059: %SYS-5-RESTART: System restarted --
Cisco IOS Software, 7200 Software (C7200-ADVIPSERVICESK9-M), Version 15.2(4)S5, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2014 by Cisco Systems, Inc.
Compiled Thu 20-Feb-14 06:51 by prod_rel_team
*Jul 16 15:47:31.359: %SSH-5-ENABLED: SSH 1.99 has been enabled
*Jul 16 15:47:
ncy
R1#
R1#ping 30.30.30.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.30.30.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/38/112 ms
R1#ping 30.30.30.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.30.30.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/52/100 ms
R1#
```

Figure 20: Connection Testing for Cisco



```
/               Move up to base level
..              Move up one level
/command        Use command at the base level
jul/16/2019 10:11:07 system,error,critical router was rebooted without proper shu
tdown
jul/16/2019 10:35:42 system,error,critical router was rebooted without proper shu
tdown
jul/16/2019 15:55:31 system,error,critical router was rebooted without proper shu
tdown

[admin@RB5] > ping 80.80.80.1
  SEQ HOST                                     SIZE TTL TIME  STATUS
    0 80.80.80.1                                 56  64 1ms
    1 80.80.80.1                                 56  64 1ms
    sent=2 received=2 packet-loss=0% min-rtt=1ms avg-rtt=1ms max-rtt=1ms

[admin@RB5] > ping 80.80.80.2
  SEQ HOST                                     SIZE TTL TIME  STATUS
    0 80.80.80.2                                 56  64 0ms
    1 80.80.80.2                                 56  64 0ms
    sent=2 received=2 packet-loss=0% min-rtt=0ms avg-rtt=0ms max-rtt=0ms

[admin@RB5] > 
```

Figure 21: Connection Testing for Mikrotik

Now the time has come to check the connection between devices. In this case for cisco, R1 do PING to R5 to make sure that the routing table between R1 and R5 is saved properly. For mikrotik, RB1 do PING to RB5 as well and the purpose is same like prior. The result there are replies from R5 to R1 and RB5 to RB1 that means the connection has been well-established. In this research, raspberry pi is used to be a controller node. Therefore, it will handle many tasks to automate routers. All of the configuration has been deployed well after about 3 minutes in raspberry pi to configure 10 routers. It is quick enough rather than configuring it manually, we have to connect our laptop to the remote device than do some configurations.

Conclusions

Network automation is a new breakthrough in technology, especially in network engineering which starts to enter the new world of network engineering itself, is network programmability and automation. Surely this really helps the network engineer to reduce their struggle in complexity of the existing network infrastructure. This kind of break-through ensure all of the tech enthusiast that technology will always be developed time by time. We have to be able to equalize it if we want to stay in IT.

Acknowledgements

First I would like to thank God to Allah subhanahu wata'ala because His blessings and His help, I can be confident to pass all many obstacles along this research. I don't forget to say thank you to Gunadarma University where I am facilitated as a student and who has supported me to do this research. I dedicate it to my parents, my lecturers and colleagues. Let me mention here who they are. My mother and my father are two special person who have given their best effort to support their son to be well-educated and well-comprehended as long as my study, especially special attention to my mother who is the strongest woman against cancer, everytime I feel lucky I believe there is her prayer which Allah subhanahu wa ta'ala grant. Next is Mr. Purnawarman Musa and Mrs. Missa Lamsani who have given me their trust to do this researh and they always support me when I was down, they always understand how their student's circustance and they always spend time for us to supervise our minithesis.The last is my friends and I can not mention them one by one in here because I have many good friends who always support whatever my situation is.

Daftar Pustaka

[1] Jason Edelman, "Network Automation with Ansible", O'Reilly Media, Inc., 2016.

[2] Widodo Budiharto, "Pemrograman Python Untuk Ilmu Komputer Dan Teknik", Yogyakarta : Andi Pub-lisher, 2018.

[3] Abdul Kadir, "Buku Dasar Raspberry Pi", Yogyakarta : Andi Publisher, 2017.

[4] Edi Rakhman, "Faisal Candrasyah dan Fajar D. Sutera, Raspberry Pi : Mikrokontroler Mungil Yang Serba Bisa", Yogyakarta : Andi Publisher, 2015.

[5] Yuniar Supardi, "Python Basic : Semua Bisa Jadi Programmer Python", Jakarta : PT.Elex Media Komputindo, 2017.

[6] Jason C. Neumann,"The Book of GNS3: Build Virtual Network Labs Using Cisco, Juniper, and More", No Starch Press. ISBN 9781593276959, 2015.