

# Project 1 Report

## Intro

In order to facilitate the administration of network, many automation tools have been developed such as SolarWinds Network Configuration Manager, ManageEngine Network Configuration Manager, Lan-Secure Configuration Center, Ansible, BMC TrueSight Automation for Networks. A small lab was set up to demonstrate the use of ansible (network automation tool) to connect to the routers and configure them and to install program on Linux machine remotely. This will mainly demonstrate the benefits that ansible brings to the table for network engineers.

### Why to automate the network?

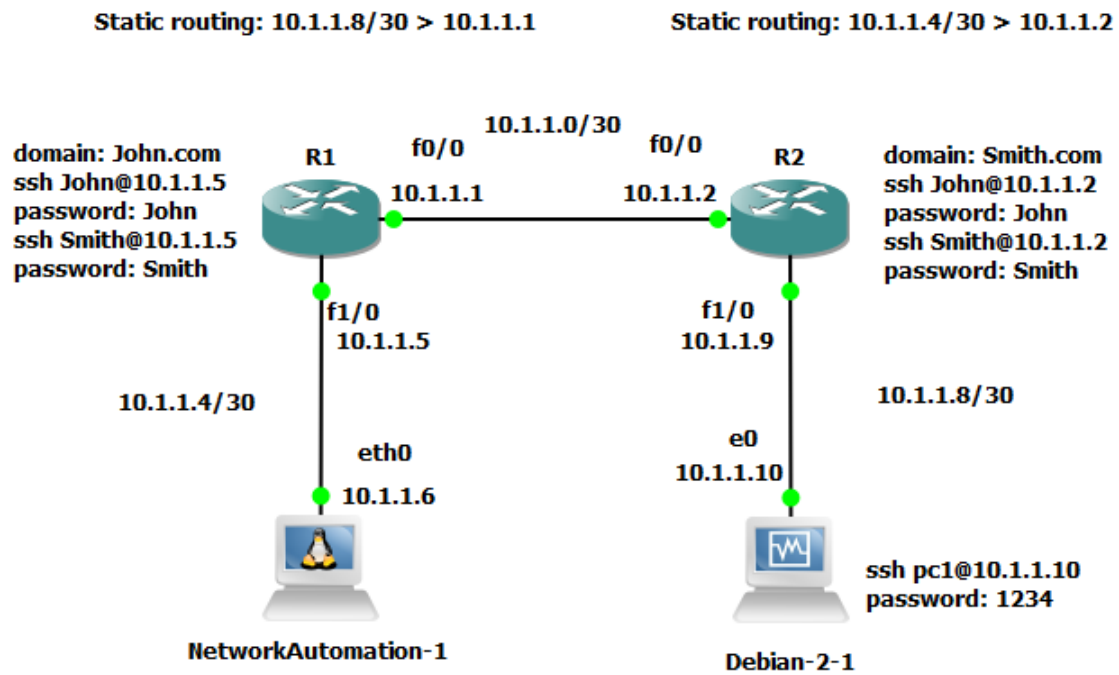
- Reduce errors
- Less time consuming
- Only one pc can deploy every device
- Reduce Risk of human errors and more reliable

### Why to automate using ansible?

- 1 script works faster than 1 person
- Automate manual work
- Automate deployment
- Configuration management
- Modules and inventory

## LAB Schema

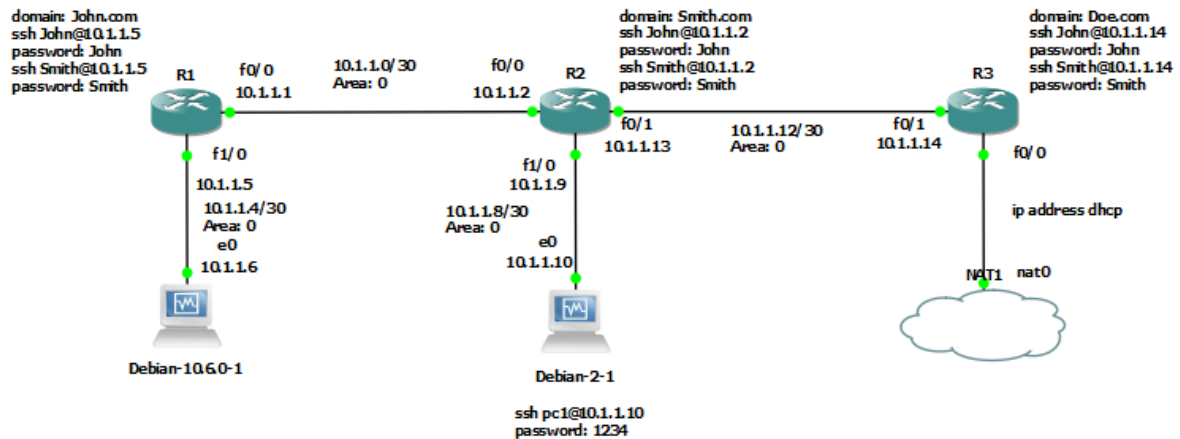
1. Configure the following network schema.



2. Use Ansible to: -
  - a) configure lo0 on R1 > 1.1.1.1 255.255.255.255 and configure lo0 on R2 > 2.2.2.2 255.255.255.255.
  - b) configure MOTD on R1 and R2.
  - c) install apache on pc2.

## Improved LAB Schema

1. Configure the following network schema.
  - a. Open Shortest Path First (OSPF) has been configured rather than static routing.
  - b. In order not to disturb the 2 initial routers, a 3<sup>rd</sup> one was added.
  - c. Use of the NAT in GNS3 to allow internet in my network.

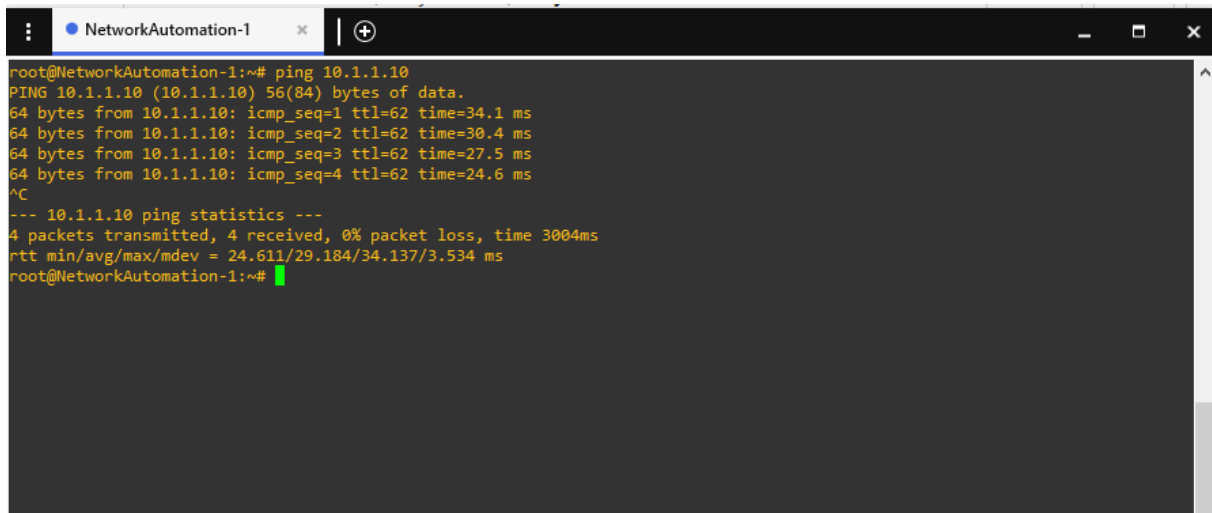


2. Use Ansible to:
  - a. configure lo0 on R1 > 1.1.1.1 255.255.255.255, configure lo0 on R2 > 2.2.2.2 255.255.255.255, and configure lo0 on R3 > 3.3.3.3 255.255.255.255.
  - b. configure MOTD on R1, R2 and R3.
  - c. install apache on pc2.

## Testing the built network

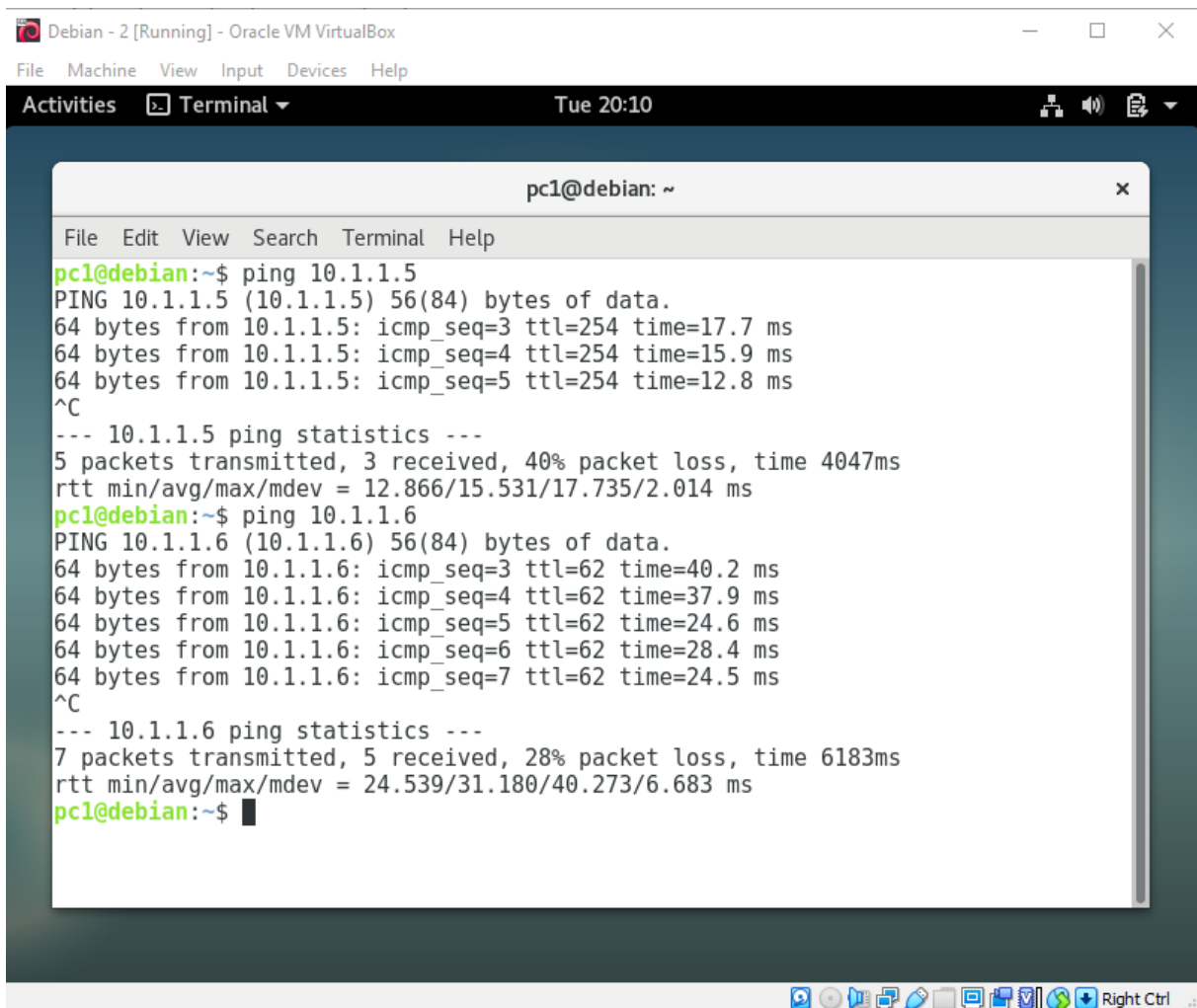
### Ping test

From pc1 being able to ping pc2:



```
root@NetworkAutomation-1:~# ping 10.1.1.10
PING 10.1.1.10 (10.1.1.10) 56(84) bytes of data.
64 bytes from 10.1.1.10: icmp_seq=1 ttl=62 time=34.1 ms
64 bytes from 10.1.1.10: icmp_seq=2 ttl=62 time=30.4 ms
64 bytes from 10.1.1.10: icmp_seq=3 ttl=62 time=27.5 ms
64 bytes from 10.1.1.10: icmp_seq=4 ttl=62 time=24.6 ms
^C
--- 10.1.1.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 24.611/29.184/34.137/3.534 ms
root@NetworkAutomation-1:~#
```

From pc2 being able to ping pc1:



```
pc1@debian: ~
File Edit View Search Terminal Help

pc1@debian:~$ ping 10.1.1.5
PING 10.1.1.5 (10.1.1.5) 56(84) bytes of data.
64 bytes from 10.1.1.5: icmp_seq=3 ttl=254 time=17.7 ms
64 bytes from 10.1.1.5: icmp_seq=4 ttl=254 time=15.9 ms
64 bytes from 10.1.1.5: icmp_seq=5 ttl=254 time=12.8 ms
^C
--- 10.1.1.5 ping statistics ---
5 packets transmitted, 3 received, 40% packet loss, time 4047ms
rtt min/avg/max/mdev = 12.866/15.531/17.735/2.014 ms
pc1@debian:~$ ping 10.1.1.6
PING 10.1.1.6 (10.1.1.6) 56(84) bytes of data.
64 bytes from 10.1.1.6: icmp_seq=3 ttl=62 time=40.2 ms
64 bytes from 10.1.1.6: icmp_seq=4 ttl=62 time=37.9 ms
64 bytes from 10.1.1.6: icmp_seq=5 ttl=62 time=24.6 ms
64 bytes from 10.1.1.6: icmp_seq=6 ttl=62 time=28.4 ms
64 bytes from 10.1.1.6: icmp_seq=7 ttl=62 time=24.5 ms
^C
--- 10.1.1.6 ping statistics ---
7 packets transmitted, 5 received, 28% packet loss, time 6183ms
rtt min/avg/max/mdev = 24.539/31.180/40.273/6.683 ms
pc1@debian:~$
```

## SSH test

From PC1 both John and Smith can SSH on R1.

```
NetworkAutomation-1 x R1
root@NetworkAutomation-1:/etc/ansible# ssh John@10.1.1.5
Password:

R1#
R1#exit
Connection to 10.1.1.5 closed.
root@NetworkAutomation-1:/etc/ansible# ssh Smith@10.1.1.5
Password:

R1#
```

From PC1 both John and Smith can SSH on R2.

```
NetworkAutomation-1 x R1 +
root@NetworkAutomation-1:/etc/ansible# ssh John@10.1.1.2
Password:
Connection closed by 10.1.1.2 port 22
root@NetworkAutomation-1:/etc/ansible# ssh John@10.1.1.2
Password:

R2#exit
Connection to 10.1.1.2 closed.
root@NetworkAutomation-1:/etc/ansible# ssh Smith@10.1.1.2
Password:

R2#exit
Connection to 10.1.1.2 closed.
root@NetworkAutomation-1:/etc/ansible#
```

From PC1 SSH on PC2.

```
root@NetworkAutomation-1:/etc/ansible# ssh pc1@10.1.1.10
pc1@10.1.1.10's password:
Linux debian 4.9.0-13-amd64 #1 SMP Debian 4.9.228-1 (2020-07-05) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Oct 27 20:34:52 2020 from 10.1.1.6
pc1@debian:~$
```

## Testing the playbook.yml

```
root@NetworkAutomation-1:/etc/ansible# ansible-playbook playbook.yml --syntax-check
playbook: playbook.yml
root@NetworkAutomation-1:/etc/ansible#
```

## Running the playbook.yml

```
root@NetworkAutomation-1:/etc/ansible# ansible-playbook playbook.yml

PLAY [Router1] *****

TASK [configure the banner for R1] *****
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host R1 should use /usr/bin/python3, but
backward compatibility with prior Ansible releases. A future Ansible release will default to
python for this host. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpre
information. This feature will be removed in version 2.12. Deprecation warnings can be disal
deprecation_warnings=False in ansible.cfg.
changed: [R1]

TASK [configure loopback interface on R1] *****
changed: [R1]

PLAY [Router2] *****

TASK [configure the banner for R2] *****
```

## Speed testing

A small speed test was carried in order to determine how much time it will take for a beginner, intermediate, and pro will be taken to perform the task without ansible.

	Routers' configurations	PC's Configurations	Total(seconds)
Beginner			
Intermediate			
Pro			

## Conclusion

The speed testing has concluded that even the pro has not been able to go faster than the ansible script (approximately 40 seconds). As the Intro stipulate this lab has demonstrate how to ease the way of configuring lots of routers and other machines at one go. Ansible has been a powerful configuration management tool for deploying software and administering remote systems in our case installing software.

## Modules used

cisco.ios.ios\_config

cisco.ios.ios\_banner

apt

## All configs

<https://github.com/Arouven/Network/>