TRANSFORMTIKA, Vol.18, No.2, January 2021, pp. 199 - 209
ISSN: 1693-3656,
journals.usm.ac.id/index.php/transformatika

IMPLEMENTATION OF DEVOPS METHOD FOR AUTOMATION OF SERVER MANAGEMENT USING ANSIBLE

Ali Khumaidi
Universitas Krisnadwipayana
Jalan Kampus Unkris, Jatiwaringin, Pondokgede, Jakarta, e-mail: alikhumaidi@unkris.ac.id

ARTICLE INFO

Article history:

Received : 07 July 2020
Received in revised form : 27 July 2020
Accepted : 18 January 2021
Available online : 31 January 2021

ABSTRACT

The Development and Operations (DevOps) method is develops tasks that are operationalized by IT staff including server management. DevOps is a server, coding, and tester bridge so that it can be activated automatically so that the programmer does not need to do it repeatedly. Server management is still mostly done manually by doing repetitive settings on one server and another server so that it is inefficient and requires a relatively long time, the solution by adding several employees will increase the work of making users and user management on all servers, the occurrence of human errors and burdens costs for the company. Ansible is one type of configuration management tools that can be used for the infrastructure management process from a manual program to an automatic. In this study the implementation and testing of Ansibel was done in server management. Ansibel can configure and manage user account settings on all servers automatically.

Keywords: Ansible, DevOps, Management Server, SSH,User Account

1. Introduction

The increasing need for information will increase the hardware requirements to be used such as servers. The server is a type of computer that is used as an information service provider and client management. Increasing server requirements will complicate administrators in server management [1]. Quite often and it has become common if one server administrator holds more than one server. Of course, it will be more difficult for administrators if they do not have server knowledge and management.

Servers and networks have network devices that must be configured. The configuration is needed so that each server can be connected and able to manage clients. The network device is a network device used to connect one device to another device. In some companies in the management of the server is still applying the manual method by doing repetitive settings on one server and another server so that management is not good enough and requires a relatively long time because the activities carried out on server one are carried out again on the next server. The solution by adding several employees will increase the work of user creation and user

Received July 07, 2020; Revised July 27, 2020; Accepted January 18, 2021

management on all servers, human error, and cost for the company. Therefore we need an automation system that can do server management with the same work on several servers.

DevOps (development and operations) is a conceptual study of the development and delivery of software to infrastructure by taking a collaborative and integrative approach between developers (Dev) and software operations (Ops) [2]. DevOps is an organizational approach with the aim of creating collaboration, interaction, and empathy across functions and divisions [3]. The DevOps method is proven to reduce some of the development stages of the old method. DevOps also changes processes to be faster which also increases the reliability, stability, durability, and safety of the production environment [4].

DevOps as a developer in the IT field creates a new solution that is configured using the software. DevOps usually develop tasks that are operationalized by IT staff including server management. Another task of DevOps is as a server, coding, and tester bridge so that it can be activated automatically so that the programmer does not need to do it over and over. Many things can be activated DevOps so as to save processing time. One example of DevOps is Ansible. Ansible is one type of Configuration Management tool that can be used for infrastructure management processes from a manual program to an automatic [5].

The results of previous studies that Ansible is one of the automation software that can be used for solutions in configuring virtual servers [6][7]. Ansible is being developed in the IT world especially development operations for server configurations [8]. Ansible can be used to configure the Cisco network devices [9]. The use of ansible in networking and use of a Playbook on ansible and its implementation for configuration [10] [11].

Network Automation is the process of using some software to optimize capabilities, network operational effectiveness, and efficiency. New network software used by OpenFlow technology can help overcome the problem of DevOps visibility of the network. So, DevOps can be placed at the center of a managed network. Based on OpenFlow, a defined network software will make the network more programmable, allowing collaboration between the two parties, namely the network and DevOps. Here we can see together, that the DevOps environment encourages network automation. This is an important part of how we see the network evolve over time, with DevOps automating policies on the network [12].

2. Research Method

In the implementation of Ansible for automation of server management with several stages as follows.

2.1. Requirements Analysis

The requirements analysis phase is the initial stage in the system development model. In this study the user will configure a server where the configured packets are supported and ansible system on the control server. Before configuring it is first necessary to connect Ubuntu via SSH (Secure Shell). SSH is a cryptographic network protocol for secure data communication, command line interface logins, remote execution commands, and other network services between two computer networks. This is connected, through a secure channel or through an insecure network, the server and client run the SSH server and SSH client program respectively [13].

In the development of centralized management server, Ansible required main and supporting software so that the server management that is built can run well. The selection of this software is based on existing system and hardware requirements. The operating system used in this study is ubuntu 12.05 LTS ubuntu server which supports x86 / 32 bit processors. In the Ubuntu server installation process is not included in the application to support the server, the application is added so that the server can run. The hardware used in this final project is one server for control and three computers as remote servers and hubs/switches.

2.2. Architecture of Automation System

From the research that has been done, there is an important relationship between the components of computer software (software), possible, and the device to be automated, an explanation of the relationship, and how to work of the components in Figure 1.

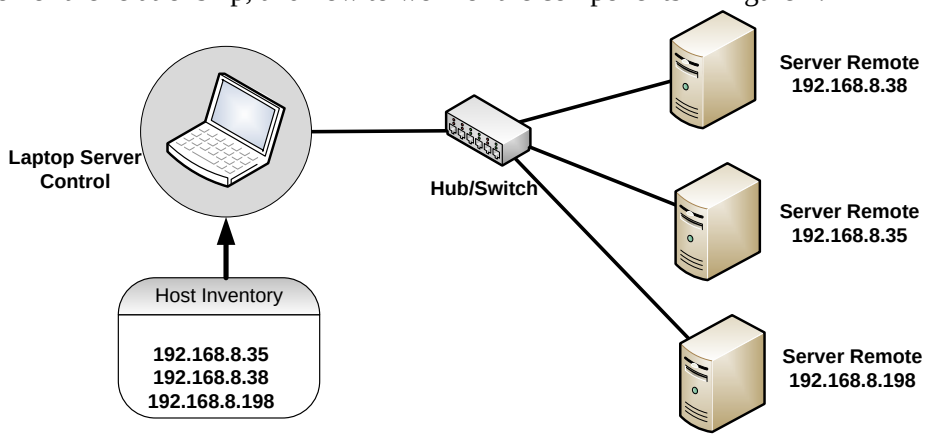

Figure. 1. Architecture Ansible management node

In Figure 1 is the relationship between software and other software or devices. Ansible will use SSH to remotely configure the server. SSH will ensure that Anonfiguration is the right destination. In this study, using 3 servers with different IPs.

2.3. Implementation of Automation System

The implementation phase is the stage where the design of the existing system is translated into computer language. To configure via ansible using 2 programming languages namely YAML and python. Here is the flow of the implementation of Centralized Management Server as a whole:

a. Ansible Installation

In this process the operating system used for the Server is Ubuntu Server 16.04 LTS. Before installing Ansible on the server, first verify the server details such as hostname and IP Address.

- Main Server:
  OS: Ubuntu Server 16.10 LTS
  IP: 192.168.8.38
  User: sinet
- IP Server to be managed:
  Ubuntu 1: 192.168.8.35
  Ubuntu 2: 192.168.8.198

Install ansible on the Main Server then log in to the server as the root user and run the commands as shown in Figure 2 and Figure 3 to confirm the system settings to be used.

```
root@sinet-video:/home/sinet# ifconfig
enp2s0f0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.8.37  netmask 255.255.255.0  broadcast 192.168.8.255
        inet6 fe80::4a7b:6bff:fe45:9954  prefixlen 64  scopeid 0x20<link>
        ether 48:7b:6b:45:99:54  txqueuelen 1000  (Ethernet)
        RX packets 22460670778  bytes 13502648828422 (13.5 TB)
        RX errors 0  dropped 7  overruns 734  frame 0
        TX packets 17912744421  bytes 4178720477640 (4.1 TB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device memory 0x92300000-923fffff
```

Figure. 2. System Details Verification After

The IP address on the control server is 192.168.8.37 with Broadcast 192.168.8.255 and subnetmask 255.255.255.0. After confirming the system settings, it's time to install the Ansible software on the system. With the command $ apt install -y ansible

```
root@sinet-video:/home/sinet# apt install -y ansible
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ieee-data python-cffi-backend python-crypto python-cryptography python-enum34
  python-httplib2 python-idna python-ipaddress python-jinja2 python-markupsafe
  python-netaddr python-paramiko python-pyasn1 python-selinux python-setuptools
  python-six python-yaml
Suggested packages:
  sshpass python-crypto-dbg python-crypto-doc python-cryptography-doc
  python-cryptography-vectors python-enum34-doc python-jinja2-doc ipython
  python-netaddr-docs python-gssapi doc-base python-setuptools-doc
The following NEW packages will be installed:
  ansible ieee-data python-cffi-backend python-crypto python-cryptography python-enum34
  python-httplib2 python-idna python-ipaddress python-jinja2 python-markupsafe
  python-netaddr python-paramiko python-pyasn1 python-selinux python-setuptools
  python-six python-yaml
0 upgraded, 18 newly installed, 0 to remove and 176 not upgraded.
1 not fully installed or removed.
Need to get 3,702 kB of archives.
After this operation, 23.8 MB of additional disk space will be used.
```

Figure. 3. Install Ansible

After being installed before the configuration is possible, ansible connection requires SSH. For that, SSH configuration first.

b. Generating SSH Keys

To do the management from the control server to the server that is remotely must first create an SSH key. First, create an SSH key by using the command. $ ssh-keygen –t rsa

```
root@sinet-video:/home/sinet# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:oGgrbuBTDi34m537sl1Q+gR8weGjG6LAvE69jrgQ9GU root@sinet-video
The key's randomart image is:
+---[RSA 2048]----+
|       .o.       |
|      . ...      |
| .   E+ =        |
|+ ..o. B .       |
|+o+.o = S        |
|o*o= . *         |
|=+B.   . o       |
|*=o+oo .         |
|+==+==o          |
+----[SHA256]-----+
root@sinet-video:/home/sinet#
```

Figure. 4. Generating SSH Keys

SSH Key has been successfully created. By default the public key is stored in the ~ / .ssh / id.rsa.pub file. and for private keys stored in ~ / .ssh / id_rsa. After generating the SSH key, now copy the public key to all three servers. With the command ssh-copy-id sinet@192.168.8.35

```
root@sinet-video:/home/sinet# ssh-copy-id sinet@192.168.8.35
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new ke
ys
sinet@192.168.8.35's password:
Permission denied, please try again.
sinet@192.168.8.35's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'sinet@192.168.8.35'"
and check to make sure that only the key(s) you wanted were added.
```

Figure. 5. Copying SSH key to the first Remote Server

Now the public key has been copied on the first remote server, and proceed to the second server. With the command ssh-copy-id sinet@192.168.8.198

```
root@sinet-video:/home/riziq# ssh-copy-id sinet@192.168.8.198
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '192.168.8.198 (192.168.8.198)' can't be established.
ECDSA key fingerprint is SHA256:aJaVsZHredT47cDLyf3yk7c+8VarlQK0OR2oNuSsav0.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new ke
ys
sinet@192.168.8.198's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'sinet@192.168.8.198'"
and check to make sure that only the key(s) you wanted were added.
```

Figure. 6. Copying SSH key to the second Remote Server

After copying all SSH Keys to the remote server, now authenticate the ssh key on all remote servers to check whether the authentication is working or not. With the command $ ssh sinet@192.168.8.35

```
root@sinet-video:/home/riziq# ssh sinet@192.168.8.35
Welcome to Ubuntu 16.10 (GNU/Linux 4.8.0-59-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

106 packages can be updated.
0 updates are security updates.

Your Ubuntu release is not supported anymore.
For upgrade information, please visit:
http://www.ubuntu.com/releaseendoflife

New release '18.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


*** System restart required ***
Last login: Mon Jan 20 22:42:46 2020 from 114.124.228.137
sinet@sinet-portal:~$
```

Figure. 7. SSH Key Authentication on the First Server

Authentication on the first server has been successfully performed by seeing the hostname is sinet @ sinet-portal no longer sinet @ sinet-video. Enter the "exit" command to exit the first server. And now authenticate on the second server. With the command $ ssh sinet@192.168.8.198

```
root@sinet-video:/home/sinet# ssh sinet@192.168.8.198
Welcome to Ubuntu 16.10 (GNU/Linux 4.8.0-22-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

194 packages can be updated.
110 updates are security updates.

Your Ubuntu release is not supported anymore.
For upgrade information, please visit:
http://www.ubuntu.com/releaseendoflife

New release '18.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Mon Jan 20 23:23:42 2020 from 114.124.228.137
sinet@sinet:~$ exit
logout
Connection to 192.168.8.198 closed.
root@sinet-video:/home/sinet#
```

Figure. 8. SSH Key Authentication on Second Server

Authentication on the third server was also successful, now exit and continue configuration on the dick server that has Ansible installed.

c. Configuring File Hosts on Ansible

Now edit the File hosts to manage the server. This hosts file holds server information such as the server needed to get a local to remote connection. The hosts file is located in / etc / ansible / hosts. Now to edit the host file with the command. $ vim / etc / ansible / hosts then add two server IP addresses:

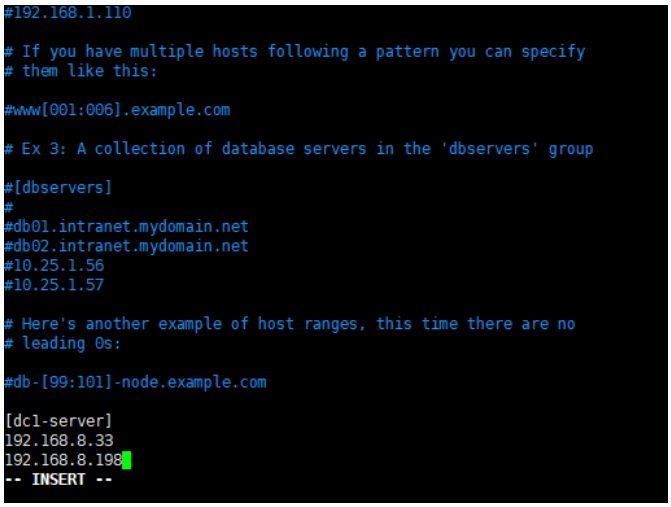

Figure. 9. Entering the server IP into the File Ansible

"Dc1-server" in parentheses indicates the name of the group, used to classify the system and determine the system to be controlled.

2.4. Testing Scenarios

To be able to manage the server properly you can use a script with the .yml extension. This script works to do something complex with the help of ansible Playbook. Ansible Playbook is very useful if we want to carry out direct orders. The first testing is checking the connection and configuration of ansibel, by checking the server access that has been registered, then checking the server partition, checking the memory usage on the server, checking the time up on the server, checking the hostname on the server, creating a directory on the server, dividing server group, check the user login on the server and check the server is on or off. The second testing is create users account by automatically on several servers and check the user usage that has been made.

3. Results and Analysis

In To manage the server, first check the server connection by executing the command $ ansible dc1-server -m ping.

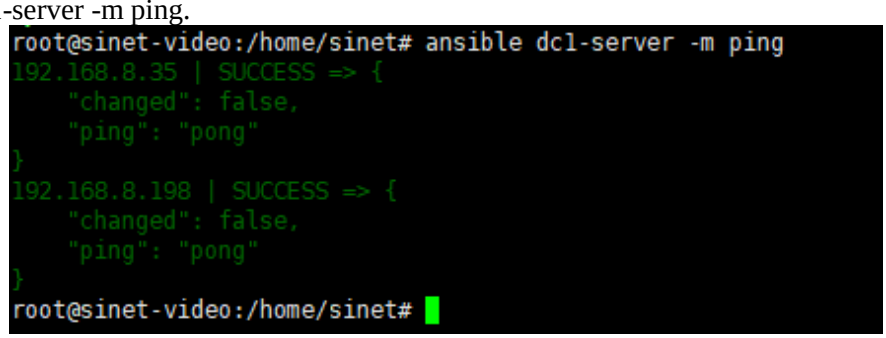

Figure. 10. Check the server connection

The results of the connection check in Figure 10, dc1-server is the name of the host group that has been registered in / etc / ansible / hosts, then -m is the command to run the module that has been provided by ansible server management software, ping is the command to find out the server that has been registered can be accessed from ansible.

3.1. Initial checking of all servers on ansible

There are several checking on all servers with Ansible, including checking the server partition can use the command $ ansible dc1-server -m shell -a "df -h". Figure 11 shows that the partition was successful.

```
root@sinet-video:/home/sinet# ansible dc1-server -m shell -a "df -h"
192.168.8.35 | SUCCESS | rc=0 >>
Filesystem      Size  Used Avail Use% Mounted on
udev             32G     0   32G   0% /dev
tmpfs           6.3G  627M  5.7G  10% /run
/dev/sda5       717G  6.1G  675G   1% /
tmpfs            32G     0   32G   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs            32G     0   32G   0% /sys/fs/cgroup
/dev/sda1       229G   11G  206G   5% /home
/dev/sda2       1.7T  1.3T  409G  76% /opt
/dev/sda3       916G  3.6G  866G   1% /var
tmpfs           6.3G     0  6.3G   0% /run/user/1000
tmpfs           6.3G     0  6.3G   0% /run/user/1002

192.168.8.198 | SUCCESS | rc=0 >>
Filesystem      Size  Used Avail Use% Mounted on
udev             63G     0   63G   0% /dev
tmpfs            13G  1.4G   12G  11% /run
/dev/sda2       722G  372G  314G  55% /
tmpfs            63G     0   63G   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs            63G     0   63G   0% /sys/fs/cgroup
/dev/sda5      1011G  579G  382G  61% /opt
/dev/sda4       674G  1.8G  638G   1% /home
tmpfs            13G     0   13G   0% /run/user/113
tmpfs            13G     0   13G   0% /run/user/1005

root@sinet-video:/home/sinet#
```

Figure. 11. Check All Server Partitions

Next, checking the memory usage on the server, can be done by executing the command $ ansible dc1-server -m shell -a "free". In Figure 12 shows the results of memory usage on each server.

```
root@sinet-video:/home/sinet# ansible dc1-server -m shell -a "free"
192.168.8.35 | SUCCESS | rc=0 >>
              total        used        free      shared  buff/cache   available
Mem:       65568440     4575884    12092164      540304    48900392    59780780
Swap:      62499836      316080    62183756

192.168.8.198 | SUCCESS | rc=0 >>
              total        used        free      shared  buff/cache   available
Mem:      131972764    48089796     1507784      201044    82375184    82615892
Swap:      67077116     1229904    65847212

root@sinet-video:/home/sinet#
```

Figure. 12. Check All Server Memory Usage

Checking the time up on the server, can be done by executing the command $ ansible dc1-server -m shell -a "uptime". Figure 13 can be seen the number of users and the average load on each server.

```
root@sinet-video:/home/sinet# ansible dc1-server -m shell -a "uptime"
192.168.8.35 | SUCCESS | rc=0 >>
 01:46:25 up 882 days, 23:07,  3 users,  load average: 1.11, 1.69, 1.10

192.168.8.198 | SUCCESS | rc=0 >>
 01:46:25 up 876 days,  2:54,  2 users,  load average: 0.21, 0.19, 0.24

root@sinet-video:/home/sinet#
```

Figure. 13. Check All Server's Timeup

Checking the hostname on the server can be done by executing the command $ ansible dc1-server -m shell -a "uname -a". Creating a directory on the server, can be done by executing the command $ ansible dc1-server -m shell -a "mkdir test". This will create a folder called test that will exist on all servers. Then to check the folders that have been successfully created on all servers using $ ansible dc1-server -m shell -a "ls". Figure 14 shows the test folder has been successfully created on all servers.

```
root@sinet-video:/home/sinet# ansible dc1-server -m shell -a "ls"
192.168.8.198 | SUCCESS | rc=0 >>
1.sh
autodump.sh
test

192.168.8.35 | SUCCESS | rc=0 >>
date-test.sh
dcblog-compress.sh
P
test
```

Figure. 14. Viewing the Directory

Divide of server groups by using the sign "[]" as a separator for each group, in each group we can add many servers at once. This serves to differentiate the server based on needs, to change it we can use the command $ vim / etc / ansible / hosts. Figure 14 shows the dividing of server groups 1 and server 2.

```
# them like this:

#www[001:006].example.com

# Ex 3: A collection of database servers in the 'dbservers' group

#[dbservers]
#
#db01.intranet.mydomain.net
#db02.intranet.mydomain.net
#10.25.1.56
#10.25.1.57

# Here's another example of host ranges, this time there are no
# leading 0s:

#db-[99:101]-node.example.com

[dc1-server1]
192.168.8.35 ansible_user=sinetansible_ssh_pass=r4m4d4n

[dc1-server2]
192.168.8.198  ansible_user=sinetansible_ssh_pass=r4m4d4n
```

Figure. 15. The dividing of the Ansible host group

User login checking on the server is done by executing the command $ ansible dc1-server -m shell -a "who". Figure 16 shows that user sinet is entered on all servers.

Figure. 16. User login checking on the server

Checking the server is off or on in the dc1-server1 group is done by executing the command $ ansible dc1-server1 -m "ping". And in the dc1-server2 group with the command $ ansible dc1-server2 -m "ping". Figure 17 shows that the servers in the dc1-server1 group and the servers in the dc1-server2 group are all on.

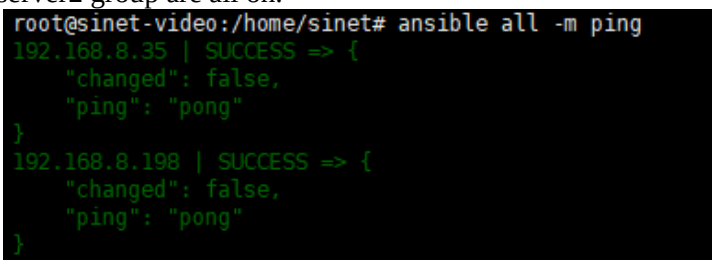

Figure. 17. Checking the server is on

3.2. Create users account by automatically on several servers

To be able to manage the server properly you can use a script with the .yml extension. This script works to do something complex with the help of Ansible Playbook. An impossible playbook is very useful if you want to do many operations directly. An example is used here by creating a user on all servers with an enabled playbook.

First, create a yml script to make an possible playbook, to do it using the $ vim create-user.yml command root @ sinet-video: / home / sinet # vim create-user.yml

- hosts: all
  become: yes
  vars:
  password: $ 1 $ SomeSalt $ aIJ0bvHJBSYd307VQuuD90
  tasks:
- user: name = unkristest password = {{password}} state = present

~

After creating the script, then run the script with the command $ ansible-playbook create-user.yml -ask-become-pass

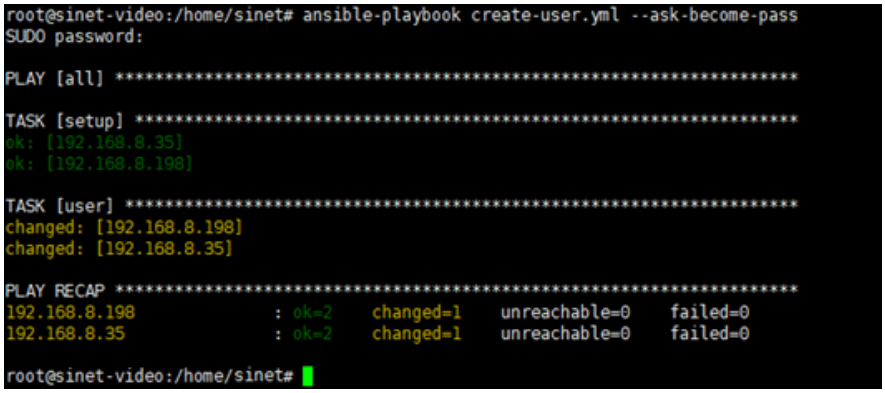

Figure. 18. Script with Ansible Playbook

Figure 18 shows an Ok status that indicates that the user was successfully created. After creating a user, can check the unkristest user that has been created with the command $ ansible all -m shell -a "cat / etc / passwd | grep unkristest"

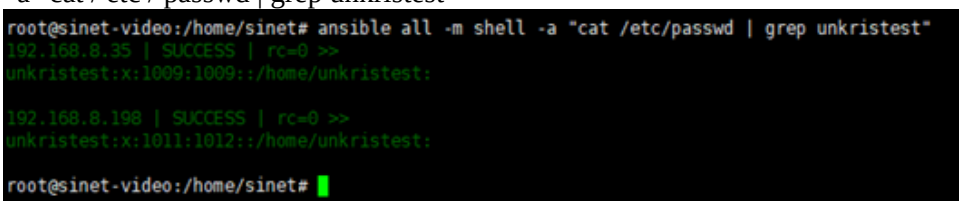

Figure. 19. Successful Anbook Playbook Script

Figure 19 shows users who have been successfully created and network admins can easily manage the server and users available on each server.

4. Conclusion

The results of the study it can be concluded several things including. Automation carried out by utilizing ansible is successfully used to repeatedly configure and manage servers on multiple servers. Ansible was successfully implemented starting the connection test, partitioning, server IP group division, login checking. Management of creating users on several servers is successful automatically so that users are available on each server. By utilizing ansible existing processes will be executed at once in sequence. The application of the DevOps method makes the work process of delivering and managing information much more practical and faster. Effective and efficient server management requires the right method. It is hoped that in the next research it will be possible to develop an enabling system to configure multiple systems through multiple accesses such as mobile or web.

References

[1] A. Widarma and Y. H. Siregar, "Analisis Kinerja Teknologi Virtualisasi Server (Study Kasus : Universitas Asahan)," in Seminar nasional Multi Disiplin Ilmu, 2019.
[2] P. Jha and R. Khan, "A Review Paper on DevOps: Beginning and More To Know," Int. J. Comput. Appl., vol. 180, no. 48, pp. 16–20, 2018.
[3] A. Dyck, R. Penners, and H. Lichter, "Towards Definitions for Release Engineering and DevOps," in Proceedings of the IEEE/ACM 3rd International Workshop on Release Engineering, 2015.
[4] S. I. Mohamed, "DevOps shifting software engineering strategy Value based perspective," IOSR J. Comput. Eng., 2015.
[5] F. Erich, C. Amrit, and M. Daneva, "A Qualitative Study of DevOps Usage in Practice," J. Softw. Evol. Process, 2017.

[6] R. Pieplu, "Ground Control Segment automated deployment and configuration with ANSIBLE and GIT," in 15th International Conference on Space Operations, 2018, doi: 10.2514/6.2018-2337.
[7] R. B. Thiyagarajan, "Single and Multi-Cloud Disaster Recovery Management using Terraform and Ansible," Dublin, National College of Ireland., 2020.
[8] J. Geerling, "Server and configuration management for humans," in Ansible for DevOps, 2015.
[9] B. Matheson, "Beginning Ansible-Cisco," in LISA18, 2020.
[10] D. Tsumak, "Large-Scale Provisioning and Configuration Management," University Of Tartu, 2016.
[11] A. Simec and A. Cukurin, "NFV And Network Security With Ansible," in 39th International Scientific Conference on Economic and Social Development, 2019.
[12] Nishant Kumar Singh, S. Thakur, H. Chaurasiya, and H. Nagdev, "Automated provisioning of application in IAAS cloud using Ansible configuration management," in 2015 1st International Conference on Next Generation Computing Technologies (NGCT), 2015, pp. 81–85, doi: 10.1109/NGCT.2015.7375087.
[13] I. P. Hariyadi and K. Marzuki, "Implementation Of Configuration Management Virtual Private Server Using Ansible," MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput., vol. 19, no. 2, pp. 347–357, May 2020, doi: 10.30812/matrik.v19i2.724.