Name: Nealian Beth B. Nanquil	**Date Performed:** October 06, 2022
Course/Section: CPE 232-CPE31S23	**Date Submitted:** October 07, 2022
Instructor: Dr. Jonathan Taylar	**Semester and SY:** 1st semester 2022-2023

Activity 6: Targeting Specific Nodes and Managing Services

1. **Objectives:**

1.1 Individualize hosts

1.2 Apply tags in selecting plays to run

1.3 Managing Services from remote servers using playbooks

2. **Discussion**:

In this activity, we try to individualize hosts. For example, we don't want apache on all our servers, or maybe only one of our servers is a web server, or maybe we have different servers like database or file servers running different things on different categories of servers and that is what we are going to take a look at in this activity.

We also try to manage services that do not automatically run using the automations in playbook. For example, when we install web servers or httpd for CentOS, we notice that the service did not start automatically.

Requirement:

In this activity, you will need to create another Ubuntu VM and name it Server 3. Likewise, you need to activate the second adapter to a host-only adapter after the installations. Take note of the IP address of the Server 3. Make sure to use the command *ssh-copy-id* to copy the public key to Server 3. Verify if you can successfully SSH to Server 3.

Completing the requirement:

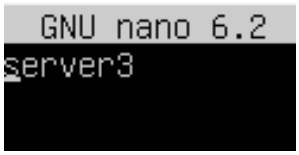

Figure 1.1: Changing the hostname to Server3.

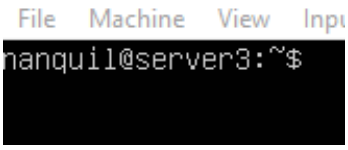

Figure 1.2: Successfully changed the hostname.

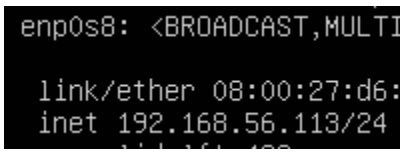

Figure 1.3: IP address of the Server 3.

```
nanquil@workstation:~$ ssh-copy-id nanquil@192.168.56.113
The authenticity of host '192.168.56.113 (192.168.56.113)' can't be established
.
ED25519 key fingerprint is SHA256:05QWqkbyfVO9hhYmzA72c97KTmI5A1Ie6kqO05eSIIA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
 out any that are already installed
The authenticity of host '192.168.56.113 (192.168.56.113)' can't be established
.
ED25519 key fingerprint is SHA256:05QWqkbyfVO9hhYmzA72c97KTmI5A1Ie6kqO05eSIIA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are promp
ted now it is to install the new keys
nanquil@192.168.56.113's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'nanquil@192.168.56.113'"
and check to make sure that only the key(s) you wanted were added.
```

Figure 1.4: Using the command *ssh-copy-id* to copy the public key to Server3.

```
nanquil@workstation:~$ ssh nanquil@192.168.56.113
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-48-generic x86_64)

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

  System information as of Thu Oct  6 01:30:33 AM UTC 2022

  System load:  0.0               Processes:               102
  Usage of /:   45.8% of 9.75GB   Users logged in:         1
  Memory usage: 21%               IPv4 address for enp0s3: 10.0.2.15
  Swap usage:   0%                IPv4 address for enp0s8: 192.168.56.113


38 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


Last login: Thu Oct  6 01:24:29 2022
nanquil@server3:~$
```

Figure 1.5: SSH to Server3.

Task 1: Targeting Specific Nodes

1. Create a new playbook and name it site.yml. Follow the commands as shown in the image below. Make sure to save the file and exit.

```
nanquil@workstation:~$ mkdir nanquil_HOA6
nanquil@workstation:~$ cd nanquil_HOA6
nanquil@workstation:~/nanquil_HOA6$ nano site.yml
```

Figure 2.1: Created a new folder named nanquil_HOA6 and a new playbook named site.yml.

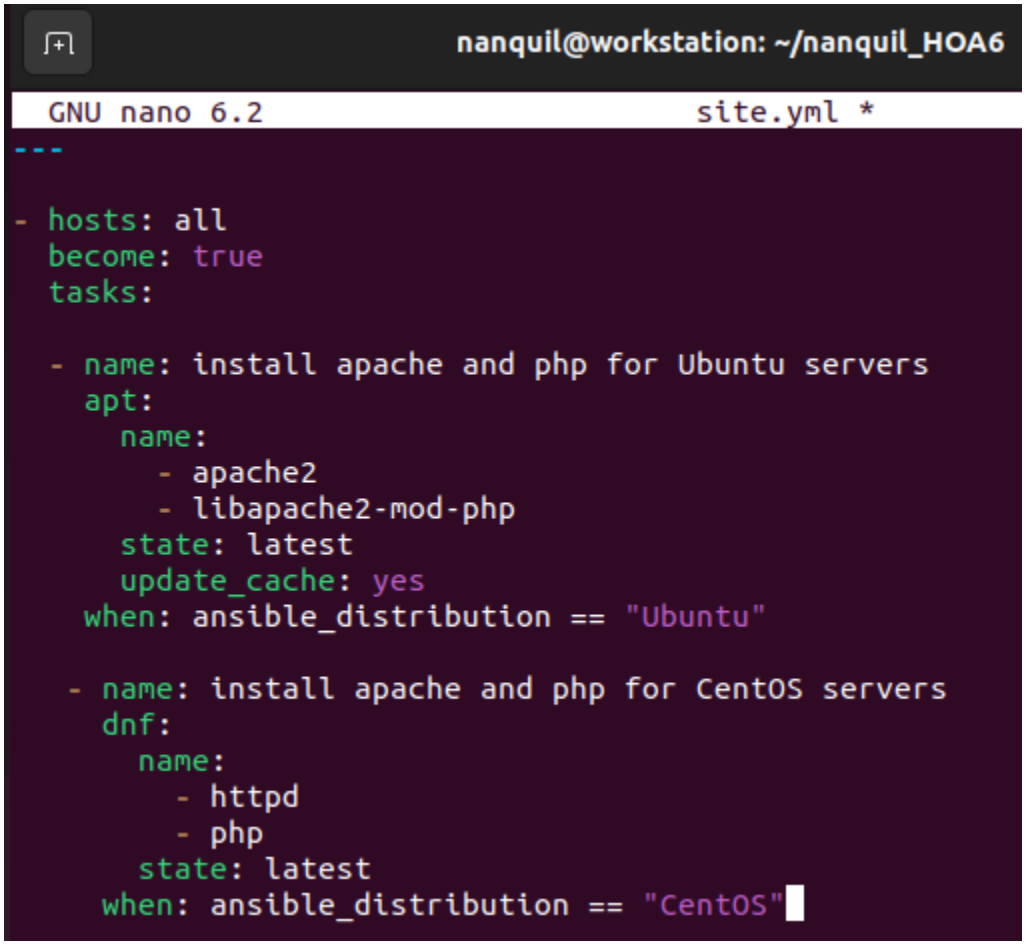

Figure 2.2: Entered the commands given in the new playbook.

2. Edit the inventory file. Remove the variables we put in our last activity and group according to the image shown below:

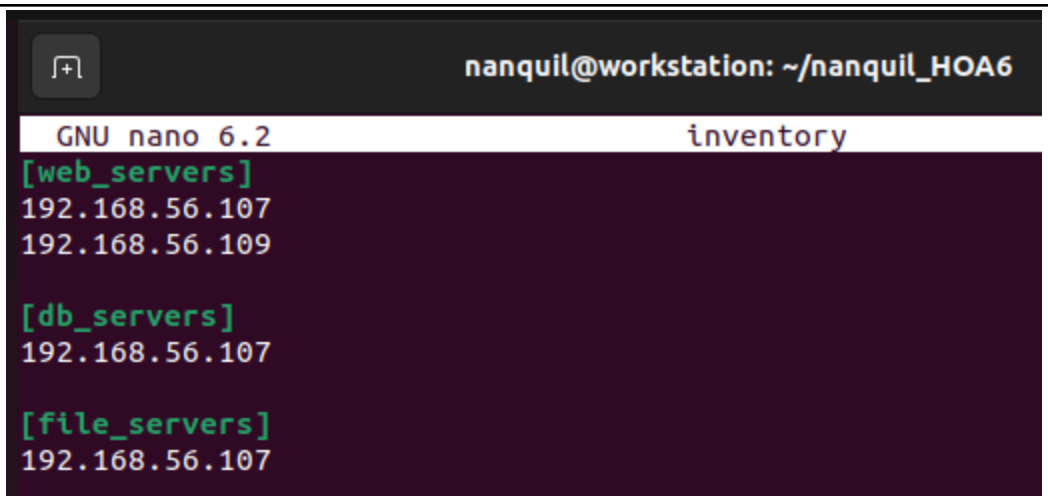

Figure 2.3: Inserted the variables needed in the inventory file.

Make sure to save the file and exit.

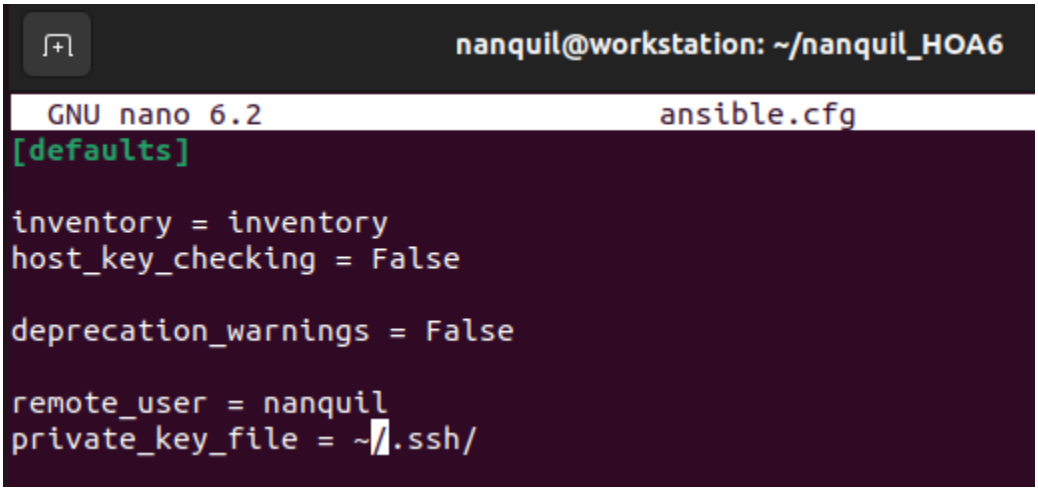

Figure 2.4: Creating the ansible.cfg.

Right now, we have created groups in our inventory file and put each server in its own group. In other cases, you can have a server be a member of multiple groups, for example you have a test server that is also a web server.

3. Edit the *site.yml* by following the image below:

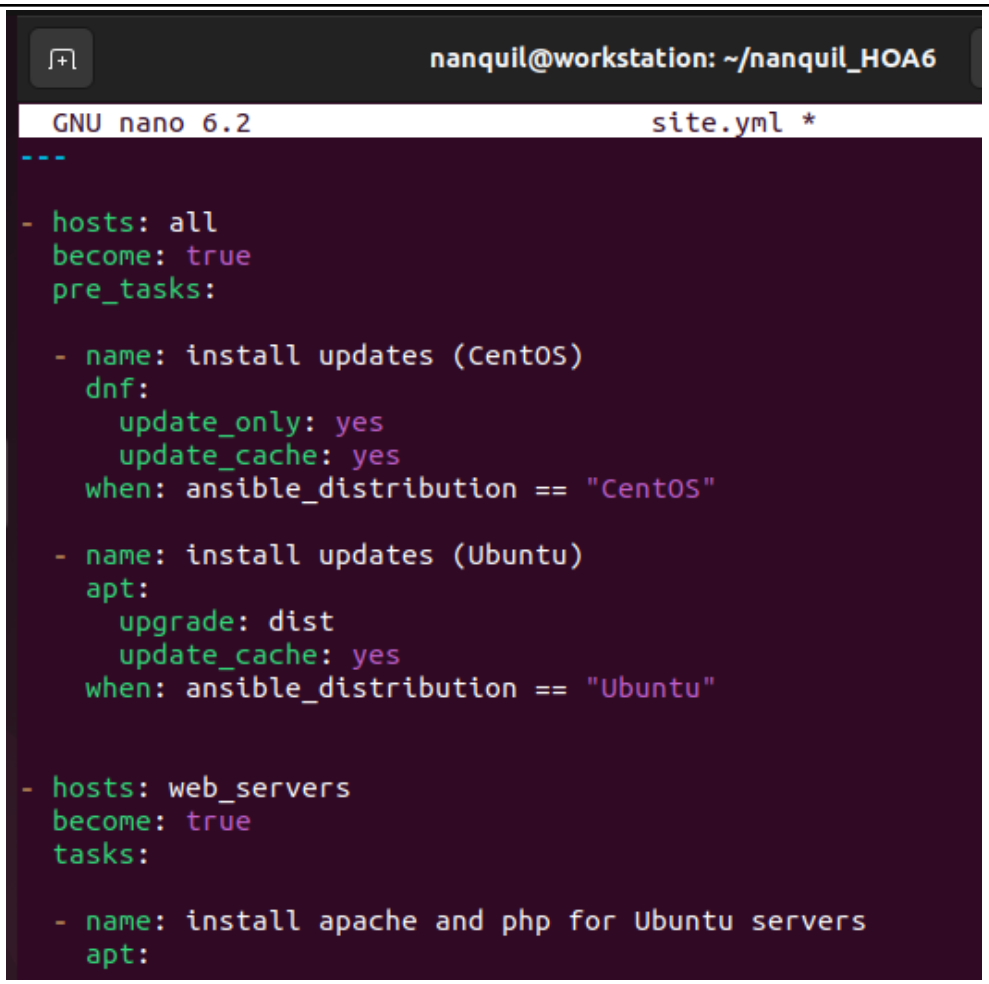

Figure 2.5: Edited the playbook named site.yml.

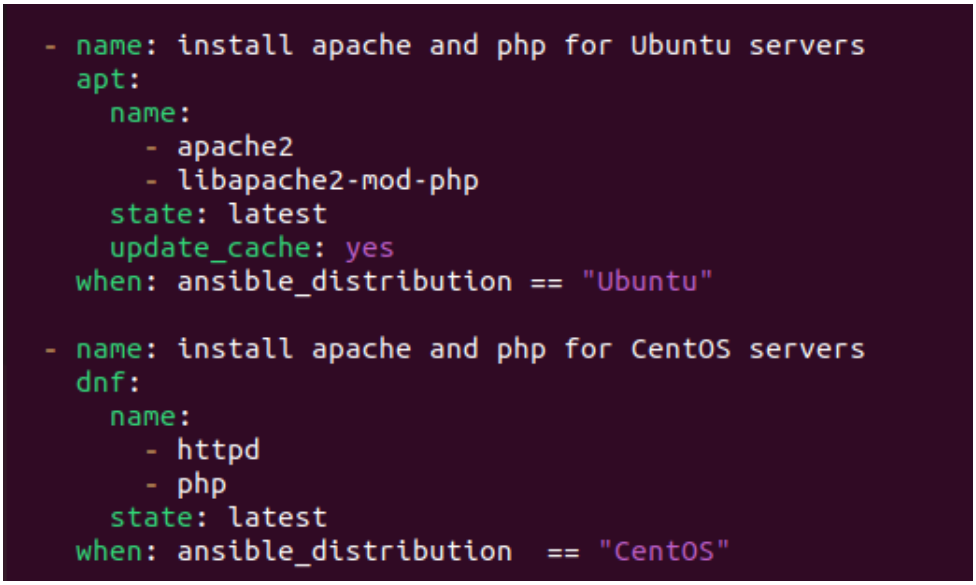

Figure 2.6: Second part of the edited playbook named site.yml.

Make sure to save the file and exit.

The *pre-tasks* command tells the ansible to run it before any other thing. In the *pre-tasks*, CentOS will install updates while Ubuntu will upgrade its distribution package. This will run before running the second play, which is targeted at *web_servers*. In the second play, apache and php will be installed on both Ubuntu servers and CentOS servers.

Run the *site.yml* file and describe the result.

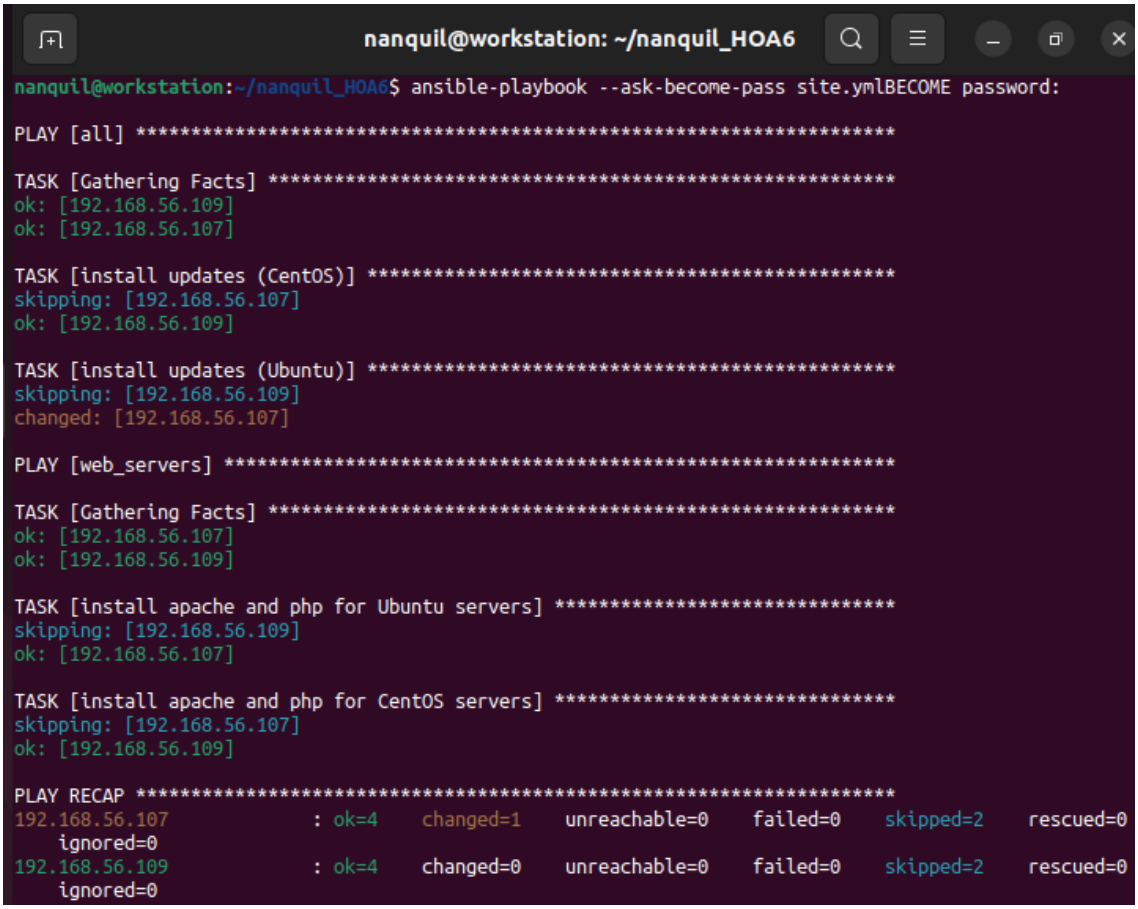

Figure 2.7: Executed the site.yml file.

4. Let's try to edit again the *site.yml* file. This time, we are going to add plays targeting the other servers. This time we target the *db_servers* by adding it on the current *site.yml*. Below is an example: (Note add this at the end of the playbooks from task 1.3.

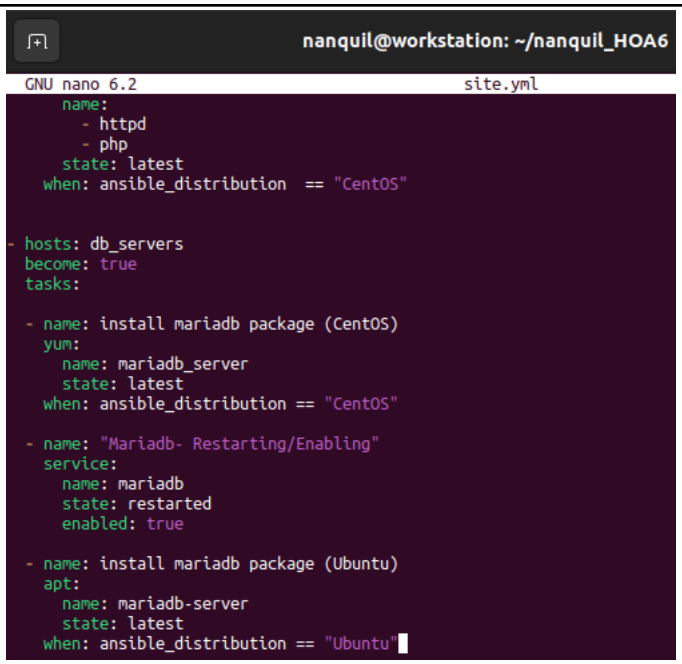

Figure 2.8: Added the db_servers in the site.yml file.

Make sure to save the file and exit.
Run the *site.yml* file and describe the result.

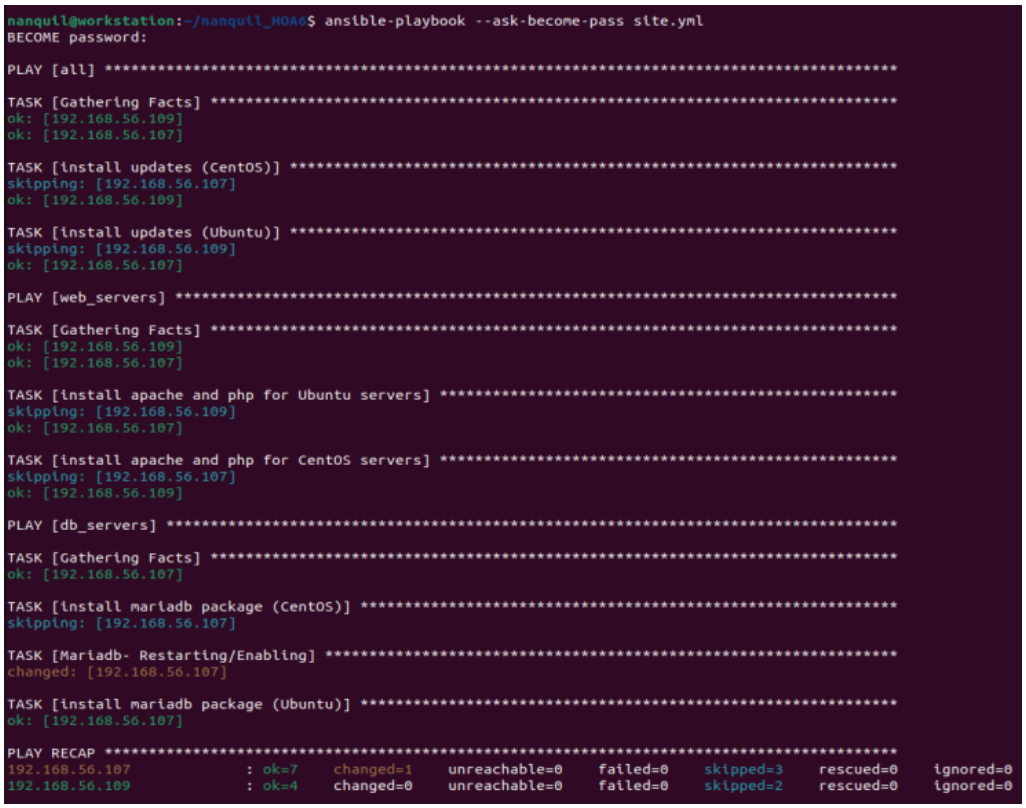

Figure 2.9: Executed the site.yml file.

5. Go to the remote server (Ubuntu) terminal that belongs to the db_servers group and check the status for mariadb installation using the command: *systemctl status mariadb.* Do this on the CentOS server also.

   Describe the output.

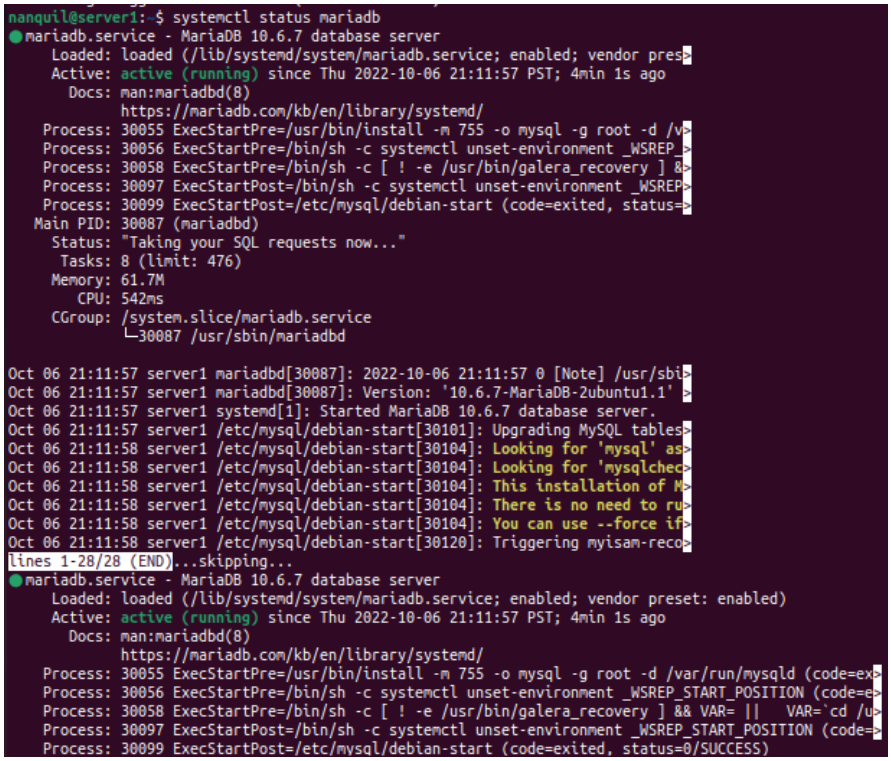

Figure 2.10: Entering the command *systemctl status mariadb* in Ubuntu to check the status of Mariadb installation. The installation shows that it is active(running).

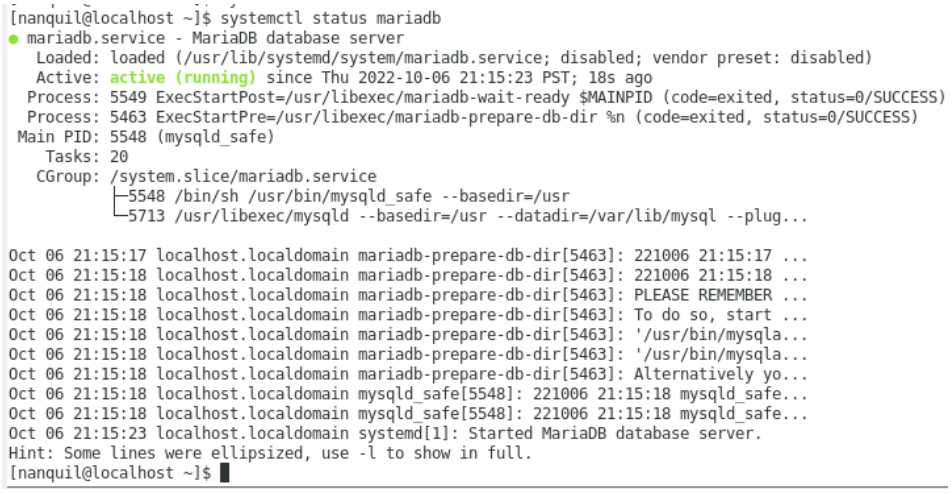

Figure 2.11: Entering the command *systemctl status mariadb* in CentOS to check the status of Mariadb installation. The installation shows that it is active(running).

6. Edit the *site.yml* again. This time we will append the code to configure installation on the *file_servers* group. We can add the following on our file. Make sure to save the file and exit.

```
- hosts: file_servers
  become: true
  tasks:

  - name: install samba package
    package:
      name: samba
      state: latest
```

Figure 2.12: Added the installation on the file_servers in site.yml file.

Run the *site.yml* file and describe the result.

```
nanquil@workstation:~/nanquil_HOA6$ ansible-playbook --ask-become-pass site.yml
BECOME password:

PLAY [all] *********************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.109]
ok: [192.168.56.107]

TASK [install updates (CentOS)] ************************************************************************************
skipping: [192.168.56.107]
ok: [192.168.56.109]

TASK [install updates (Ubuntu)] ************************************************************************************
skipping: [192.168.56.109]
ok: [192.168.56.107]

PLAY [web_servers] *************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.107]
ok: [192.168.56.109]

TASK [install apache and php for Ubuntu servers] *******************************************************************
skipping: [192.168.56.109]
ok: [192.168.56.107]

TASK [install apache and php for CentOS servers] *******************************************************************
skipping: [192.168.56.107]
ok: [192.168.56.109]

PLAY [db_servers] **************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.107]

TASK [install mariadb package (CentOS)] ****************************************************************************
skipping: [192.168.56.107]

TASK [Mariadb- Restarting/Enabling] ********************************************************************************
changed: [192.168.56.107]

TASK [install mariadb package (Ubuntu)] ****************************************************************************
ok: [192.168.56.107]

PLAY [file_servers] ************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.107]

TASK [install samba package] ***************************************************************************************
changed: [192.168.56.107]

PLAY RECAP *********************************************************************************************************
192.168.56.107             : ok=9    changed=2    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.109             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

Figure 2.13: Executed the edited site.yml file.

The testing of the *file_servers* is beyond the scope of this activity, and as well as our topics and objectives. However, in this activity we were able to show that we can target hosts or servers using grouping in ansible playbooks.

Task 2: Using Tags in running playbooks

In this task, our goal is to add metadata to our plays so that we can only run the plays that we want to run, and not all the plays in our playbook.

1. Edit the *site.yml* file. Add tags to the playbook. After the name, we can place the tags: *name_of_tag*. This is an arbitrary command, which means you can use any name for a tag.

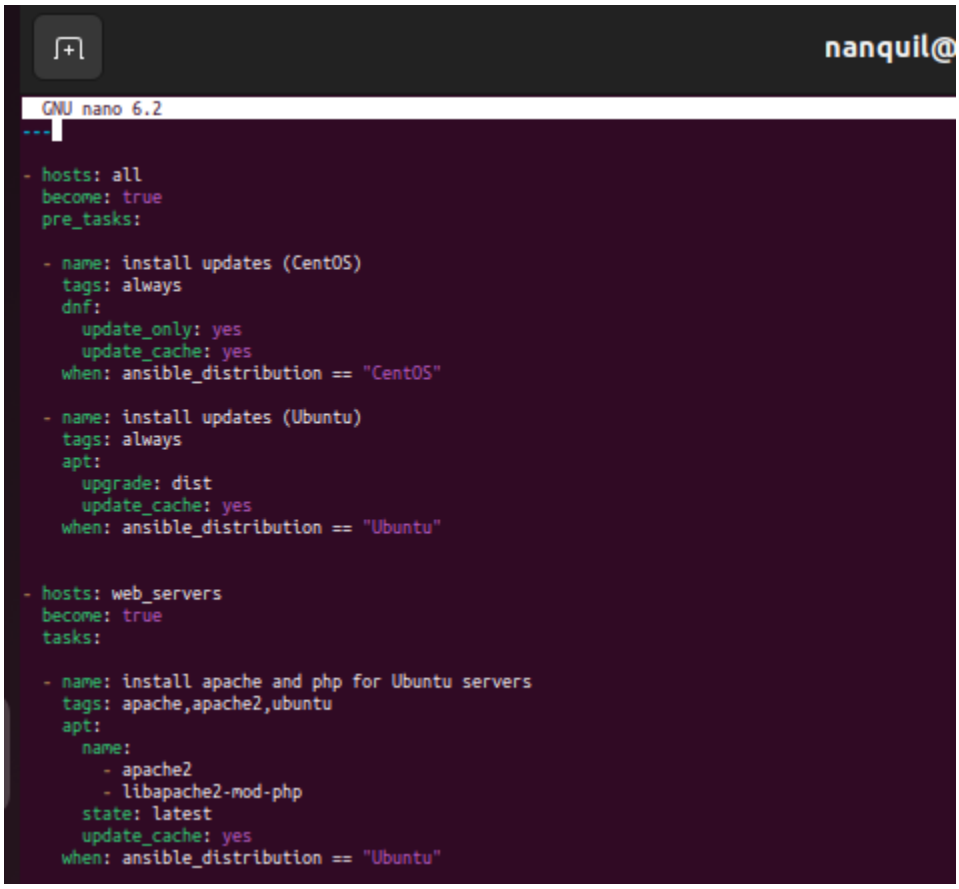

Figure 3.1: Placing the tags in site.yml file.

```
  - name: install apache and php for CentOS servers
    tags: apache,centos,httpd
    dnf:
      name:
        - httpd
        - php
      state: latest
    when: ansible_distribution  == "CentOS"


- hosts: db_servers
  become: true
  tasks:

  - name: install mariadb package (CentOS)
    tags: centos, db,mariadb
    yum:
      name: mariadb_server
      state: latest
    when: ansible_distribution == "CentOS"

  - name: "Mariadb- Restarting/Enabling"
    service:
      name: mariadb
      state: restarted
      enabled: true

  - name: install mariadb package (Ubuntu)
    tags: db, mariadb,ubuntu
    apt:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "Ubuntu"


- hosts: file_servers
  become: true
  tasks:

  - name: install samba package
    tags: samba
    package:
      name: samba
      state: latest
```

Figure 3.2: Second part of placing the tags in site.yml file.

Make sure to save the file and exit.
Run the *site.yml* file and describe the result.

```
nanquil@workstation:~/nanquil_HOA6$ ansible-playbook --ask-become-pass site.yml
BECOME password:

PLAY [all] *********************************************************************
**********************************

TASK [Gathering Facts] *********************************************************
**********************************
ok: [192.168.56.109]
ok: [192.168.56.107]

TASK [install updates (CentOS)] ************************************************
**********************************
skipping: [192.168.56.107]
ok: [192.168.56.109]
```

Figure 3.3: Executed the playbook.

Figure 3.4: Executed the playbook.

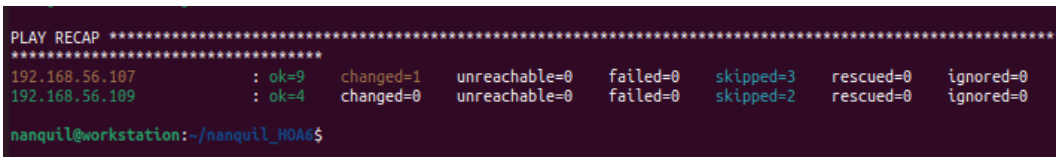

Figure 3.5: Executed the playbook.

2. On the local machine, try to issue the following commands and describe each result:

2.1 ansible-playbook --list-tags site.yml

```
nanquil@workstation:~/nanquil_HOA6$ ansible-playbook --list-tags site.yml

playbook: site.yml

  play #1 (all): all    TAGS: []
      TASK TAGS: [always]

  play #2 (web_servers): web_servers    TAGS: []
      TASK TAGS: [apache, apache2, centos, httpd, ubuntu]

  play #3 (db_servers): db_servers      TAGS: []
      TASK TAGS: [centos, db, mariadb, ubuntu]

  play #4 (file_servers): file_servers  TAGS: []
      TASK TAGS: [samba]
```

Figure 3.6: The image shows the lists of plays that have tags.

2.2 ansible-playbook --tags centos --ask-become-pass site.yml

```
nanquil@workstation:~/nanquil_HOA6$ ansible-playbook --tags centos --ask-become-pass site.yml
BECOME password:

PLAY [all] *********************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.109]
ok: [192.168.56.107]

TASK [install updates (CentOS)] ************************************************************************************
skipping: [192.168.56.107]
ok: [192.168.56.109]

TASK [install updates (Ubuntu)] ************************************************************************************
skipping: [192.168.56.109]
ok: [192.168.56.107]

PLAY [web_servers] *************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.109]
ok: [192.168.56.107]

TASK [install apache and php for CentOS servers] *******************************************************************
skipping: [192.168.56.107]
ok: [192.168.56.109]

PLAY [db_servers] **************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.107]

TASK [install mariadb package (CentOS)] ****************************************************************************
skipping: [192.168.56.107]

PLAY [file_servers] ************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.107]

PLAY RECAP *********************************************************************************************************
192.168.56.107             : ok=5    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.109             : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Figure 3.7: The image shows that we only executed CentOS.

2.3 ansible-playbook --tags db --ask-become-pass site.yml

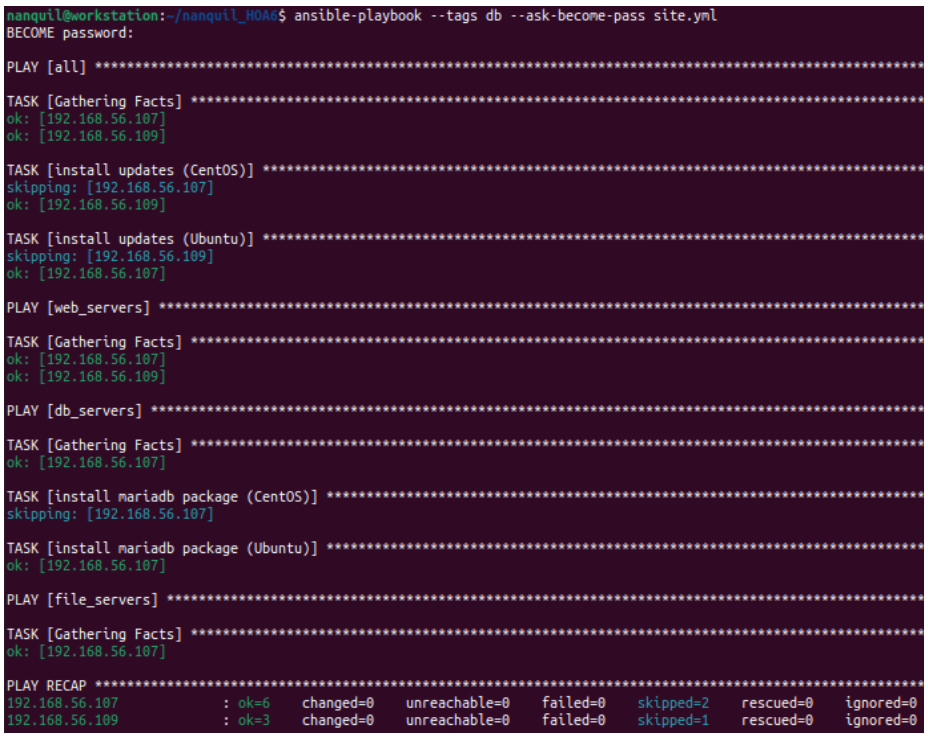

Figure 3.8: The image shows that we only executed db_server.

2.4 ansible-playbook --tags apache --ask-become-pass site.yml

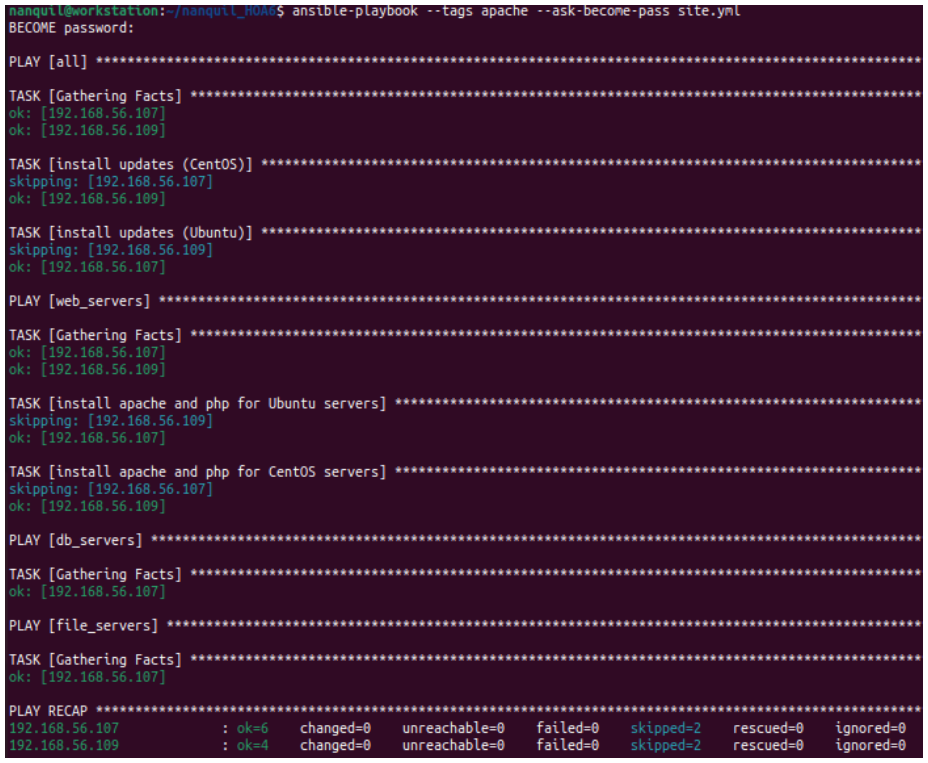

Figure 3.9: The image shows that we only executed apache.

2.5 *ansible-playbook --tags “apache,db” --ask-become-pass site.yml*

```
nanquil@workstation:~/nanquil_HOA6$ ansible-playbook --tags "apache,db" --ask-become-pass site.yml
BECOME password:

PLAY [all] *********************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.107]
ok: [192.168.56.109]

TASK [install updates (CentOS)] ************************************************************************************
skipping: [192.168.56.107]
ok: [192.168.56.109]

TASK [install updates (Ubuntu)] ************************************************************************************
skipping: [192.168.56.109]
ok: [192.168.56.107]

PLAY [web_servers] *************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.107]
ok: [192.168.56.109]

TASK [install apache and php for Ubuntu servers] *******************************************************************
skipping: [192.168.56.109]
ok: [192.168.56.107]

TASK [install apache and php for CentOS servers] *******************************************************************
skipping: [192.168.56.107]
ok: [192.168.56.109]

PLAY [db_servers] **************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.107]

TASK [install mariadb package (CentOS)] ****************************************************************************
skipping: [192.168.56.107]

TASK [install mariadb package (Ubuntu)] ****************************************************************************
ok: [192.168.56.107]

PLAY [file_servers] ************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.107]

PLAY RECAP *********************************************************************************************************
192.168.56.107             : ok=7    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.109             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

Figure 3.10: The image shows that we only executed apache,db.

Task 3: Managing Services

1. Edit the file site.yml and add a play that will automatically start the httpd on CentOS server.
   Make sure to save the file and exit.

```
- name: start httpd (CentOS)
  tags: apache, centos,httpd
  service:
    name: httpd
    state: started
  when: ansible_distribution == "CentOS"
```

Figure 4.1: Added the new play that automatically starts the httpd on CentOS server.

You would also notice from our previous activity that we already created a module that runs a service.

```
- name: install mariadb package (Ubuntu)
  tags: db, mariadb,ubuntu
  apt:
    name: mariadb-server
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: "Mariadb - Restarting/Enabling"
  service:
    name: mariadb
    state: restarted
    enabled: true
```

Figure 4.2: Added another new play under the installation of mariadb package in site.yml file.

This is because in CentOS, installed packages' services are not run automatically. Thus, we need to create the module to run it automatically.

2. To test it, before you run the saved playbook, go to the CentOS server and stop the currently running httpd using the command *sudo systemctl stop httpd*. When prompted, enter the sudo password. After that, open the browser and enter the CentOS server's IP address. You should not be getting a display because we stopped the httpd service already.

```
[nanquil@localhost ~]$ sudo systemctl stop httpd
[nanquil@localhost ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabl
   Active: inactive (dead)
     Docs: man:httpd(8)
           man:apachectl(8)
[nanquil@localhost ~]$
```

Figure 4.3: In the CentOS server, we stopped the httpd.

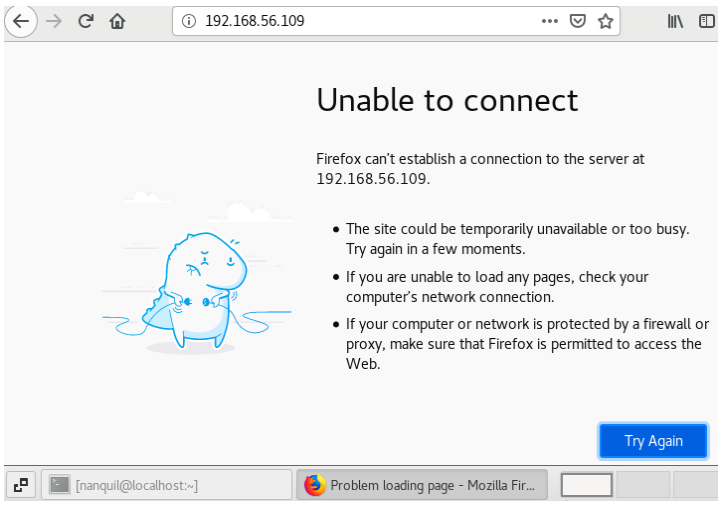

Figure 4.4: Opened the Mozilla Firefox and entered the IP address of the CentOS. It shows that there is no display since we stopped the httpd service.

3. Go to the local machine and this time, run the *site.yml* file. Then after running the file, go again to the CentOS server and enter its IP address on the browser. Describe the result.

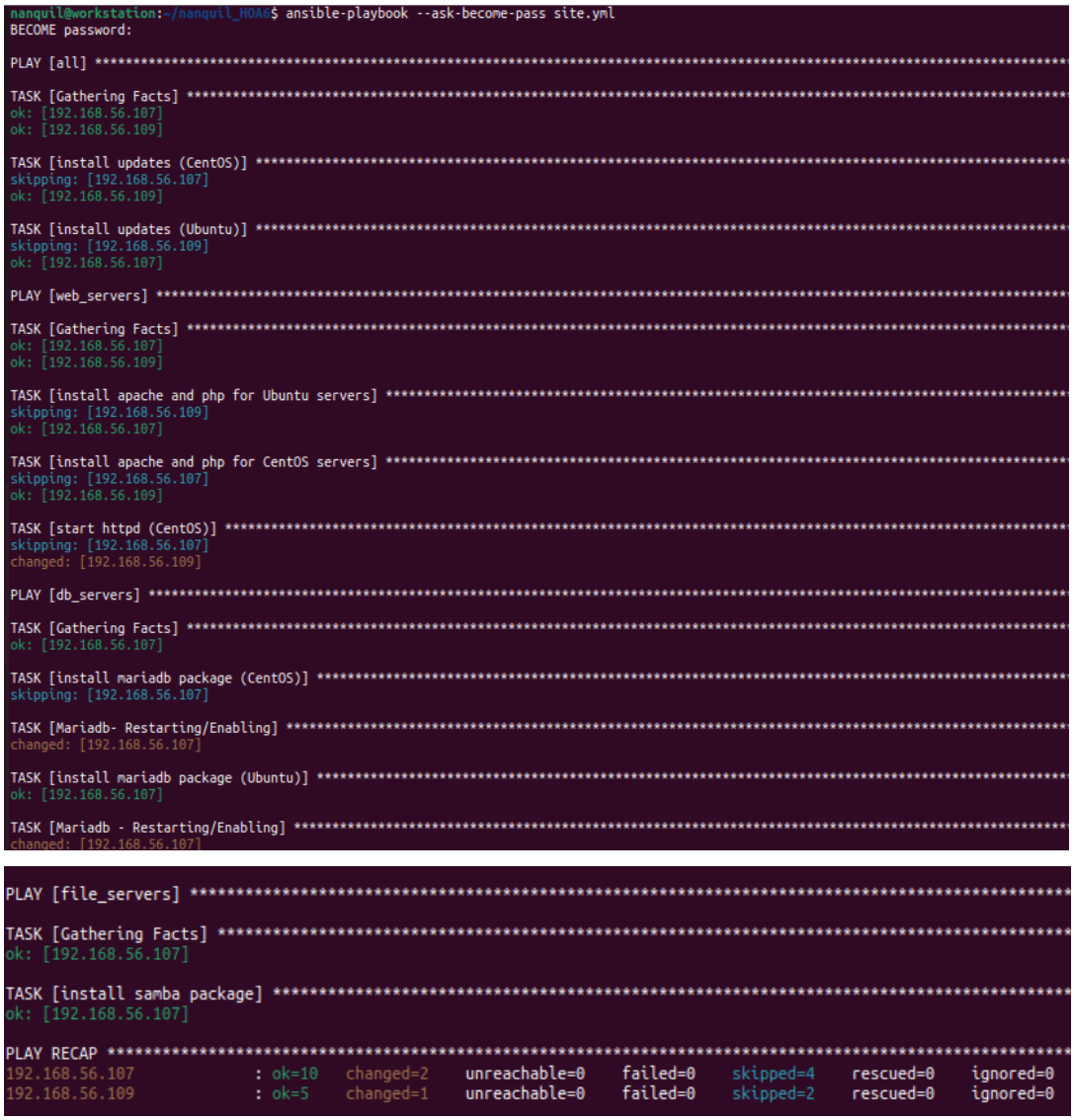

Figure 4.5: Executed the site.yml in the Local Machine.

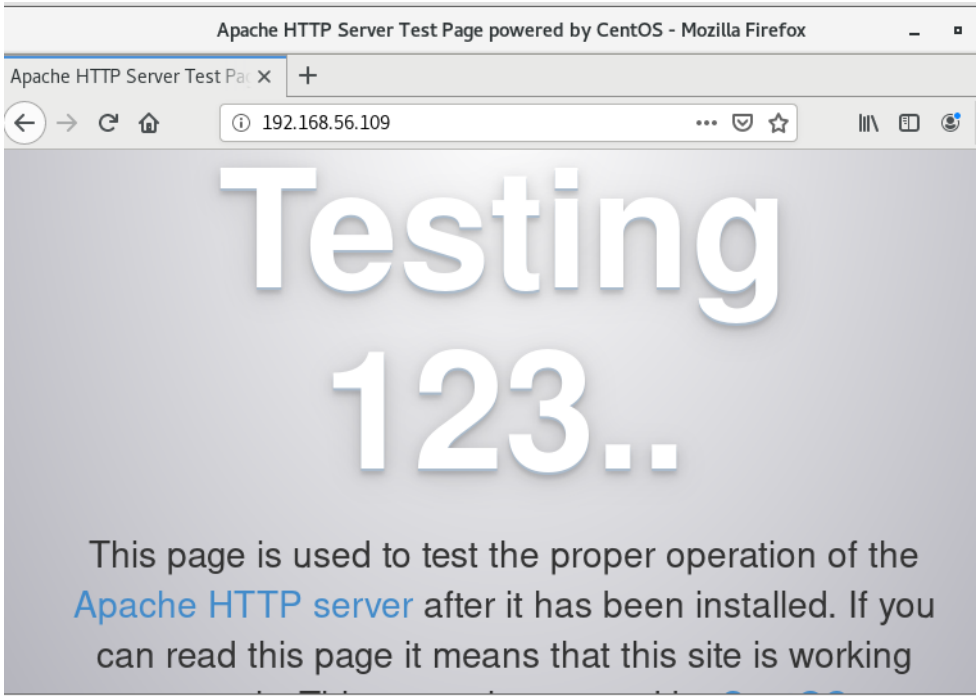

Figure 4.6: In the CentOS server, entering the IP address on the mozilla firefox, the result shows that the Apache HTTP server is now testing.

To automatically enable the service every time we run the playbook, use the command *enabled: true* similar to Figure 7.1.2 and save the playbook.

```
  - name: start httpd (CentOS)
    tags: apache, centos,httpd
    service:
      name: httpd
      state: started
      enabled: true
    when: ansible_distribution == "CentOS"

- hosts: db_servers
  become: true
  tasks:
```

Figure 4.7: Adding the enabled:true to automatically enable the service every time we run the playbook.

```
nanquil@workstation:~/nanquil_HOA6$ ansible-playbook --ask-become-pass site.yml
BECOME password:

PLAY [all] *********************************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.109]
ok: [192.168.56.107]

TASK [install updates (CentOS)] ************************************************
skipping: [192.168.56.107]
ok: [192.168.56.109]

TASK [install updates (Ubuntu)] ************************************************
skipping: [192.168.56.109]
ok: [192.168.56.107]

PLAY [web_servers] *************************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.107]
ok: [192.168.56.109]

TASK [install apache and php for Ubuntu servers] *******************************
skipping: [192.168.56.109]
ok: [192.168.56.107]

TASK [install apache and php for CentOS servers] *******************************
skipping: [192.168.56.107]
ok: [192.168.56.109]

TASK [start httpd (CentOS)] ****************************************************
skipping: [192.168.56.107]
changed: [192.168.56.109]

PLAY [db_servers] **************************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.107]

TASK [install mariadb package (CentOS)] ****************************************
skipping: [192.168.56.107]

TASK [Mariadb- Restarting/Enabling] ********************************************
changed: [192.168.56.107]

TASK [install mariadb package (Ubuntu)] ****************************************
ok: [192.168.56.107]

TASK [Mariadb - Restarting/Enabling] *******************************************
changed: [192.168.56.107]
```

```
PLAY [file_servers] ************************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.107]

TASK [install samba package] ***************************************************
ok: [192.168.56.107]

PLAY RECAP *********************************************************************
192.168.56.107             : ok=10   changed=2    unreachable=0    failed=0    skipped=4    rescued=0    ignored=0
192.168.56.109             : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0

nanquil@workstation:~/nanquil_HOA6$
```

Figure 2.8: Executing again the site.yml file.

Reflections:

Answer the following:

1. What is the importance of putting our remote servers into groups? Putting our remote servers into groups is important because in this way, we can easily modify the remote servers individually. Also, it helps us to limit syntax errors.
2. What is the importance of tags in playbooks? Tags in playbooks are important because inserting tags in servers will allow us to play a specific task that has been tagged.
3. Why do you think some services need to be managed automatically in playbooks? I think some services need to be managed automatically in playbooks because in this method, we will not be needed to enter custom codes to automate some services.

Conclusion:

As I have finished this activity, I have successfully accomplished the three(3) objectives. I have individualized the hosts by inserting the IP addresses of the remote servers into different groups such as web servers, db servers, and file servers. Also, we can repeat the IP address of a certain remote server into another group in which this other group can perform different tasks from other groups.

In this activity, I have applied tags in the playbook and these tags can be found in every task under different groups. When tags are called in executing the playbook, it will perform the task which includes the tags you have called or declared while ignoring the other task that does not include the declared tags when running the playbook.

I was able to manage services from remote servers using playbooks like re-enabling the httpd service from CentOS. In this activity, I was able to demonstrate stopping the httpd service of the CentOS server and verify the result using the browser in the CentOS. Using the playbook that I have created, I was able to enable the httpd service in the CentOS server and I added additional code like enabled:true, so it will run automatically every time I execute my playbook.

Therefore, I conclude that I met all the objectives in this activity such as individualizing hosts, applying tags in playbook, and managing services from remote servers using playbooks. Since I have successfully performed all the tasks in this activity, I am ready for the next activity and hopefully I can apply the skills that I learned here for the next activity.

"I affirm that I will not give or receive any unauthorized help on this activity and that all work will be my own". - Nealian Beth B. Nanquil