Name: Gabiano, Chris Leonard A.	Date Performed: Sept 28, 2023
Course/Section: CPE31S6	Date Submitted:
Instructor: Dr. Jonathan Taylar	Semester and SY: 2023 - 2024

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.

Task 1: Targeting Specific Nodes

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

```
---

- hosts: all
  become: true
  tasks:

  - name: install apache and php for Ubuntu servers
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

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

```
leonard@workstation: ~/Gabiano_Mod6
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                        site.yml

---

- hosts: all
  become: true
  tasks:

  - name: install apache and php for Ubuntu servers
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: install apache and php for CentOS servers
    dnf:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
      update_cache: yes
    when: ansible_distribution == "CentOS"
                                [ Wrote 23 lines ]
```

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

```
[web_servers]
192.168.56.120
192.168.56.121

[db_servers]
192.168.56.122


[file_servers]
192.168.56.123
```

```
leonard@workstation:~/Gabiano_Mod6$ cat inventory
[web_servers]
192.168.56.102
192.168.56.109

[db_servers]
192.168.56.103
192.168.56.109

[file_servers]
192.168.56.102
```

Make sure to save the file and exit.

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:

- hosts: all
become: true
pre_tasks:
- name: install updates (CentOS)
dnf:
update_only: yes
update_cache: yes
when: ansible_distribution == "CentOS"
- name: install updates (Ubuntu)
apt:
upgrade: dist
update_cache: yes
when: ansible_distribution == "Ubuntu"
- hosts: web_servers
become: true
tasks:
- name: install apache and php for Ubuntu servers
apt:
name:
- apache2
- libapache2-mod-php
state: latest
when: ansible_distribution == "Ubuntu"
- name: install apache and php for CentOS servers
dnf:
name:
- httpd
- php
state: latest
when: ansible_distribution == "CentOS"

Make sure to save the file and exit.

```
leonard@workstation:~/Gabiano_Mod6$ cat site.yml
---

- hosts: all
  become: true
  pre_tasks:

  - name: install updates (CentOS)
    dnf:
      update_only: yes
      update_cache: yes
    when: ansible_ditribution == "CentOS"

   - name: install updates (Ubuntu)
    dnf:
      update_only: yes
      update_cache: yes
    when: ansible_ditribution == "Ubuntu"

- hosts: web_servers
  become: true
  tasks:

  - name: install apache and php for Ubuntu servers
    apt:
      name:
        - apache2
```

```
      name:
        - apache2
        - libapache2-mod-php
      state: latest
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: install apache and php for CentOS servers
    dnf:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
      update_cache: yes
    when: ansible_distribution == "CentOS"
```

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.

```
leonard@workstation:~/Gabiano_Mod6$ ansible-playbook --ask-become-pass site.yml
SUDO password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.103]
ok: [192.168.56.102]
ok: [192.168.56.109]

TASK [install apache and php for Ubuntu servers] *******************************
*
skipping: [192.168.56.109]
ok: [192.168.56.103]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] *******************************
*
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.109]
```

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.102]
ok: [192.168.56.109]

TASK [install updates (CentOS)] ************************************************
*
skipping: [192.168.56.102]
ok: [192.168.56.109]

TASK [install updates (Ubuntu)] ************************************************
*
skipping: [192.168.56.109]
ok: [192.168.56.102]
```

```
TASK [start httpd (CentOS)] ****************************************************
*
skipping: [192.168.56.102]
ok: [192.168.56.109]

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.103]
ok: [192.168.56.109]

TASK [install mariadb package (CentOS)] ****************************************
*
skipping: [192.168.56.103]
ok: [192.168.56.109]

TASK [Mariadb - Restarting/Enabling] *******************************************
*
changed: [192.168.56.109]
changed: [192.168.56.103]
```

```
TASK [install mariadb package (Ubuntu)] ****************************************
*
skipping: [192.168.56.109]
ok: [192.168.56.103]

TASK [Mariadb- Restaring/Enabling] *********************************************
*
changed: [192.168.56.103]
changed: [192.168.56.109]

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.103]

TASK [install samba package] ***************************************************
*
ok: [192.168.56.103]
```

```
TASK [start httpd (CentOS)] ****************************************************
*
skipping: [192.168.56.103]

PLAY RECAP *********************************************************************
*
192.168.56.102             : ok=4    changed=0    unreachable=0    failed=0
192.168.56.103             : ok=8    changed=2    unreachable=0    failed=0
192.168.56.109             : ok=9    changed=2    unreachable=0    failed=0
```

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.

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

  - name: install mariadb package (CentOS)
    yum:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "CentOS"

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

  - name: install mariadb packege (Ubuntu)
    apt:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "Ubuntu"
```

Make sure to save the file and exit.

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

```
Leonard@workstation:~/Gabiano_Mod6$ ansible-playbook --ask-become-pass site.yml
SUDO password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.103]
ok: [192.168.56.102]
ok: [192.168.56.109]

TASK [install apache and php for Ubuntu servers] *******************************
*
skipping: [192.168.56.109]
ok: [192.168.56.103]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] *******************************
*
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.109]
```

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.102]
ok: [192.168.56.109]

TASK [install updates (CentOS)] ***********************************************
*
skipping: [192.168.56.102]
ok: [192.168.56.109]

TASK [install updates (Ubuntu)] ***********************************************
*
skipping: [192.168.56.109]
ok: [192.168.56.102]
```

```
TASK [start httpd (CentOS)] ***************************************************
*
skipping: [192.168.56.102]
ok: [192.168.56.109]

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.103]
ok: [192.168.56.109]

TASK [install mariadb package (CentOS)] ***************************************
*
skipping: [192.168.56.103]
ok: [192.168.56.109]

TASK [Mariadb - Restarting/Enabling] ******************************************
*
changed: [192.168.56.109]
changed: [192.168.56.103]
```

```
TASK [install mariadb package (Ubuntu)] ***************************************
*
skipping: [192.168.56.109]
ok: [192.168.56.103]

TASK [Mariadb- Restaring/Enabling] ********************************************
*
changed: [192.168.56.103]
changed: [192.168.56.109]

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.103]

TASK [install samba package] **************************************************
*
ok: [192.168.56.103]
```

```
TASK [start httpd (CentOS)] ****************************************************
*
skipping: [192.168.56.103]

PLAY RECAP *********************************************************************
*
192.168.56.102             : ok=4    changed=0    unreachable=0    failed=0
192.168.56.103             : ok=8    changed=2    unreachable=0    failed=0
192.168.56.109             : ok=9    changed=2    unreachable=0    failed=0
```

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.

```
[leonard@localhost ~]$ systemctl status mariadb
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: dis
abled)
   Active: active (running) since Thu 2023-09-28 07:24:06 EDT; 2min 40s ago
  Process: 19165 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited, s
tatus=0/SUCCESS)
  Process: 19129 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, stat
us=0/SUCCESS)
 Main PID: 19164 (mysqld_safe)
    Tasks: 20
   CGroup: /system.slice/mariadb.service
           ├─19164 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
           └─19329 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --plu...

Sep 28 07:24:04 localhost.localdomain systemd[1]: Starting MariaDB database server...
Sep 28 07:24:04 localhost.localdomain mariadb-prepare-db-dir[19129]: Database MariaD...
Sep 28 07:24:04 localhost.localdomain mariadb-prepare-db-dir[19129]: If this is not ...
Sep 28 07:24:04 localhost.localdomain mysqld_safe[19164]: 230928 07:24:04 mysqld_saf...
Sep 28 07:24:04 localhost.localdomain mysqld_safe[19164]: 230928 07:24:04 mysqld_saf...
Sep 28 07:24:06 localhost.localdomain systemd[1]: Started MariaDB database server.
Hint: Some lines were ellipsized, use -l to show in full.
[leonard@localhost ~]$
```

```
leonard@SERVER1:~$ systemctl status mariadb
● mariadb.service - MariaDB 10.1.48 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset:
   Active: active (running) since Thu 2023-09-28 19:30:28 PST; 1min 10s ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
 Main PID: 32684 (mysqld)
   Status: "Taking your SQL requests now..."
    Tasks: 27 (limit: 4884)
   CGroup: /system.slice/mariadb.service
           └─32684 /usr/sbin/mysqld

Sep 28 19:30:28 SERVER1 /etc/mysql/debian-start[32717]: information_schema
Sep 28 19:30:28 SERVER1 /etc/mysql/debian-start[32717]: mysql
Sep 28 19:30:28 SERVER1 /etc/mysql/debian-start[32717]: performance_schema
Sep 28 19:30:28 SERVER1 /etc/mysql/debian-start[32717]: Phase 6/7: Checking and
Sep 28 19:30:28 SERVER1 /etc/mysql/debian-start[32717]: Processing databases
Sep 28 19:30:28 SERVER1 /etc/mysql/debian-start[32717]: information_schema
Sep 28 19:30:28 SERVER1 /etc/mysql/debian-start[32717]: performance_schema
Sep 28 19:30:28 SERVER1 /etc/mysql/debian-start[32717]: Phase 7/7: Running 'FLU
Sep 28 19:30:28 SERVER1 /etc/mysql/debian-start[32717]: OK
Sep 28 19:30:28 SERVER1 /etc/mysql/debian-start[323]: Checking for insecure roo
lines 1-21/21 (END)
```

```
leonard@SERVER2:~$ systemctl status mariadb
● mariadb.service - MariaDB 10.1.48 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset:
   Active: active (running) since Thu 2023-09-28 19:33:02 PST; 17s ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
 Main PID: 14674 (mysqld)
   Status: "Taking your SQL requests now..."
    Tasks: 27 (limit: 4884)
   CGroup: /system.slice/mariadb.service
           └─14674 /usr/sbin/mysqld

Sep 28 19:33:02 SERVER2 systemd[1]: Starting MariaDB 10.1.48 database server...
Sep 28 19:33:02 SERVER2 mysqld[14674]: 2023-09-28 19:33:02 140456800177280 [Not
Sep 28 19:33:02 SERVER2 /etc/mysql/debian-start[14706]: Upgrading MySQL tables
Sep 28 19:33:02 SERVER2 systemd[1]: Started MariaDB 10.1.48 database server.
Sep 28 19:33:02 SERVER2 /etc/mysql/debian-start[14768]: Checking for insecure r
Sep 28 19:33:02 SERVER2 /etc/mysql/debian-start[14774]: Triggering myisam-recov
lines 1-17/17 (END)
```

Describe the output.

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.

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

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

Make sure to save the file and exit.

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

```
leonard@workstation:~/Gabiano_Mod6$ ansible-playbook --ask-become-pass site.yml
SUDO password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.103]
ok: [192.168.56.102]
ok: [192.168.56.109]

TASK [install apache and php for Ubuntu servers] *******************************
*
skipping: [192.168.56.109]
ok: [192.168.56.103]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] *******************************
*
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.109]
```

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.102]
ok: [192.168.56.109]

TASK [install updates (CentOS)] ***********************************************
*
skipping: [192.168.56.102]
ok: [192.168.56.109]

TASK [install updates (Ubuntu)] ***********************************************
*
skipping: [192.168.56.109]
ok: [192.168.56.102]
```

```
TASK [start httpd (CentOS)] ***************************************************
*
skipping: [192.168.56.102]
ok: [192.168.56.109]

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.103]
ok: [192.168.56.109]

TASK [install mariadb package (CentOS)] ***************************************
*
skipping: [192.168.56.103]
ok: [192.168.56.109]

TASK [Mariadb - Restarting/Enabling] ******************************************
*
changed: [192.168.56.109]
changed: [192.168.56.103]
```

```
TASK [install mariadb package (Ubuntu)] ***************************************
*
skipping: [192.168.56.109]
ok: [192.168.56.103]

TASK [Mariadb- Restaring/Enabling] ********************************************
*
changed: [192.168.56.103]
changed: [192.168.56.109]

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.103]

TASK [install samba package] **************************************************
*
ok: [192.168.56.103]
```

```
TASK [start httpd (CentOS)] ****************************************************
*
skipping: [192.168.56.103]

PLAY RECAP *********************************************************************
*
192.168.56.102             : ok=4    changed=0    unreachable=0    failed=0
192.168.56.103             : ok=8    changed=2    unreachable=0    failed=0
192.168.56.109             : ok=9    changed=2    unreachable=0    failed=0
```

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.

```
---

- hosts: all
  become: true
  pre_tasks:

  - name: install updates (CentOS)
    tags: always
    dnf:
      update_only: yes
      update_cache: yes
    when: ansible_distribution == "CentOS"

  - name: install updates (Ubuntu)
    tags: always
    apt:
      upgrade: dist
      update_cache: yes
    when: ansible_distribution == "Ubuntu"
```

```
- hosts: web_servers
  become: true
  tasks:

  - name: install apache and php for Ubuntu servers
    tags: apache,apache2,ubuntu
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
    when: ansible_distribution == "Ubuntu"

  - 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
    dnf:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "CentOS"

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

  - name: install mariadb packege (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
```

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

```
leonard@workstation:~/Gabiano_Mod6$ ansible-playbook --ask-become-pass site.yml
SUDO password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.103]
ok: [192.168.56.102]
ok: [192.168.56.109]

TASK [install apache and php for Ubuntu servers] *******************************
*
skipping: [192.168.56.109]
ok: [192.168.56.103]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] *******************************
*
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.109]

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.102]
ok: [192.168.56.109]

TASK [install updates (CentOS)] ************************************************
*
skipping: [192.168.56.102]
ok: [192.168.56.109]

TASK [install updates (Ubuntu)] ************************************************
*
skipping: [192.168.56.109]
ok: [192.168.56.102]
```

```
TASK [start httpd (CentOS)] ***************************************************
*
skipping: [192.168.56.102]
ok: [192.168.56.109]

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.103]
ok: [192.168.56.109]

TASK [install mariadb package (CentOS)] ***************************************
*
skipping: [192.168.56.103]
ok: [192.168.56.109]

TASK [Mariadb - Restarting/Enabling] ******************************************
*
changed: [192.168.56.109]
changed: [192.168.56.103]
```

```
TASK [install mariadb package (Ubuntu)] ***************************************
*
skipping: [192.168.56.109]
ok: [192.168.56.103]

TASK [Mariadb- Restaring/Enabling] ********************************************
*
changed: [192.168.56.103]
changed: [192.168.56.109]

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.103]

TASK [install samba package] **************************************************
*
ok: [192.168.56.103]
```

```
TASK [start httpd (CentOS)] ***************************************************
*
skipping: [192.168.56.103]

PLAY RECAP ********************************************************************
*
192.168.56.102             : ok=4    changed=0    unreachable=0    failed=0
192.168.56.103             : ok=8    changed=2    unreachable=0    failed=0
192.168.56.109             : ok=9    changed=2    unreachable=0    failed=0
```

2. On the local machine, try to issue the following commands and describe each result:
   2.1 *ansible-playbook --list-tags site.yml*
   2.2 *ansible-playbook --tags centos --ask-become-pass site.yml*

2.3 *ansible-playbook --tags db --ask-become-pass site.yml*
2.4 *ansible-playbook --tags apache --ask-become-pass site.yml*
2.5 *ansible-playbook --tags "apache,db" --ask-become-pass site.yml*

Task 3: Managing Services

1. Edit the file site.yml and add a play that will automatically start the httpd on CentOS server.

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

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

Figure 3.1.1

Make sure to save the file and exit.

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

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

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

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

Figure 3.1.2

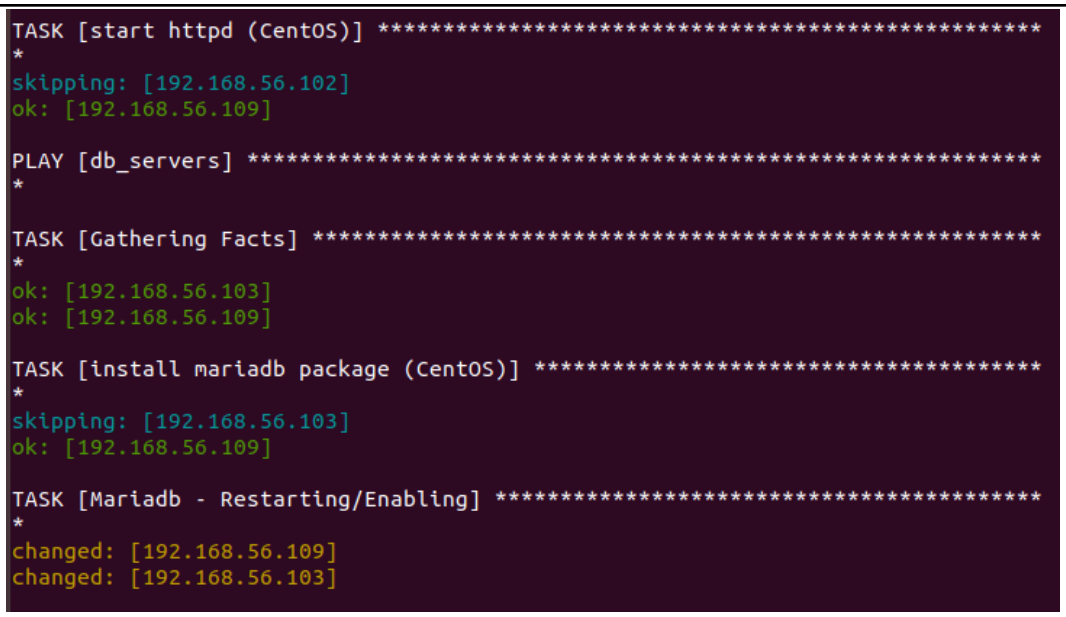
```
TASK [Mariadb- Restaring/Enabling] ********************************************
*
changed: [192.168.56.103]
```

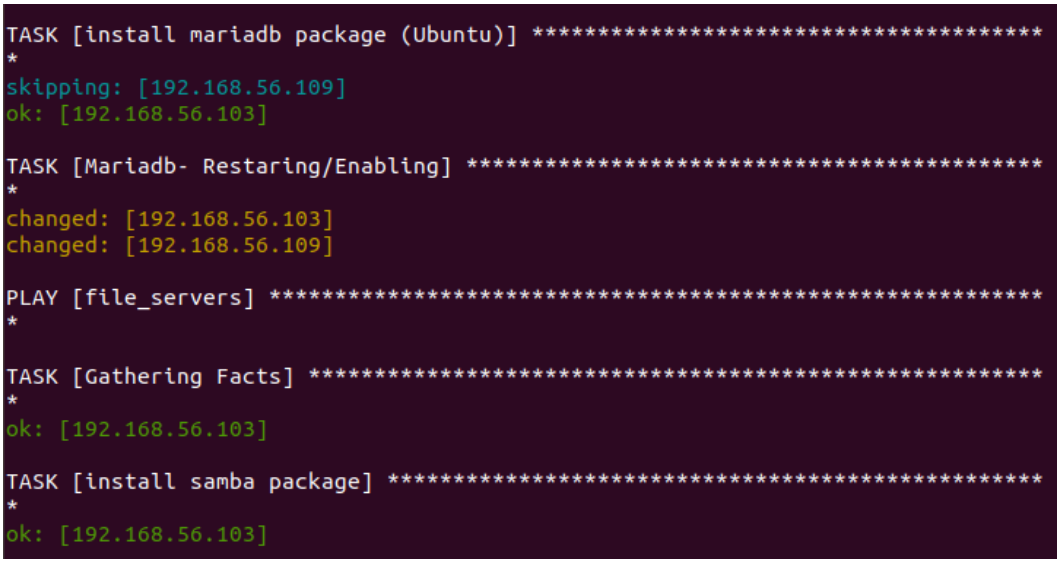
This is because in CentOS, installed packages' services are not run automatically. Thus, we need to create the module to run it automatically.

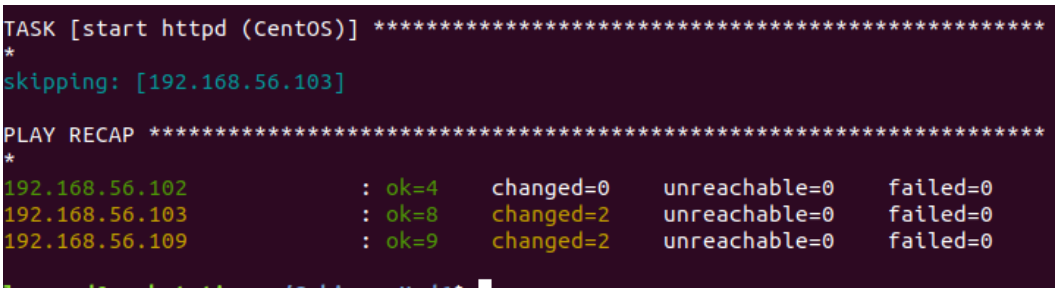
2. To test it, before you run the saved playbook, go to the CentOS server and stop the currently running httpd using the command *sudo systemctl stop httpd.* When prompted, enter the sudo password. After that, open the browser and enter the CentOS server's IP address. You should not be getting a display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

   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.

```
leonard@workstation:~/Gabiano_Mod6$ ansible-playbook --ask-become-pass site.yml
SUDO password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.103]
ok: [192.168.56.102]
ok: [192.168.56.109]

TASK [install apache and php for Ubuntu servers] *******************************
*
skipping: [192.168.56.109]
ok: [192.168.56.103]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] *******************************
*
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.109]

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.102]
ok: [192.168.56.109]

TASK [install updates (CentOS)] ************************************************
*
skipping: [192.168.56.102]
ok: [192.168.56.109]

TASK [install updates (Ubuntu)] ************************************************
*
skipping: [192.168.56.109]
ok: [192.168.56.102]
```







Reflections:

Answer the following:

1. What is the importance of putting our remote servers into groups?

   Grouping remote servers is essential for effective server management and automation. It enables streamlined configuration, maintenance, and task execution by allowing you to apply changes, updates, and security policies consistently across servers with similar functions or roles. This grouping also enhances scalability and simplifies troubleshooting, making it easier to isolate and address issues within specific server clusters, ultimately improving the overall efficiency and reliability of your infrastructure.

2. What is the importance of tags in playbooks?

   Tags in playbooks are important because they provide a way to selectively execute specific tasks or groups of tasks within a playbook. This functionality allows for better control over the playbook's execution, making it easier to perform tasks like testing, debugging, or running only certain parts of the playbook when needed. Tags enhance playbook flexibility and efficiency by enabling users to focus on relevant tasks and streamline automation processes.

3. Why do think some services need to be managed automatically in playbooks?

   Automatically managing services in playbooks is essential for several reasons. Firstly, automation ensures consistency and reduces human error by following predefined configurations and workflows. Secondly, it enhances scalability and efficiency, allowing for rapid deployment and scaling of services as needed, especially in dynamic and cloud-based environments. Lastly, automation facilitates monitoring and maintenance, enabling continuous health checks and automatic remediation, which is crucial for maintaining service availability and reliability.

   conclusion:

   In this activity, the objectives were to individualize hosts, apply tags for selecting plays, and manage services on remote servers using playbooks. We achieved the goal of individualizing hosts by configuring different servers with specific roles. Additionally, we used tags to select which plays to run based on the requirements of each server. Lastly, we managed services that didn't start automatically, ensuring our infrastructure was properly configured. To complete the activity, we also created a new Ubuntu VM, configured its networking, and verified SSH

connectivity, which further expanded our skills in server management and automation.