Name: Espiritu, Diego Angelo G.	**Date Performed: 28/09/2023**
Course/Section: CPE232/CPE31S6	**Date Submitted: 28/09/2023**
Instructor: Dr. Jonathan Vidal Taylar	**Semester and SY:**

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"
```

```
diego@workstation: ~/HOA6
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:
        - httpd
        - php
      state: latest
    when: ansible_distribution == "CentOS"
```

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
```

Make sure to save the file and exit.

```
diego@workstation: ~/HOA6
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                    inventory

[web_server]
192.168.56.102
192.168.56.107

[db_server]
192.168.56.103
192.168.56.107

[file_server]
192.168.56.102
```

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.

```
diego@workstation: ~/HOA6
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                          site.yml

---

- 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_server
  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"
```

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.
-The command is successful and there are no errors.

```
diego@workstation: ~/HOA6
File  Edit  View  Search  Terminal  Help
PLAY [web_server] ***********************************************************
*

TASK [Gathering Facts] ******************************************************
*
ok: [192.168.56.102]
ok: [192.168.56.107]

TASK [install apache and php for Ubuntu servers] ****************************
*
skipping: [192.168.56.107]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] ****************************
*
skipping: [192.168.56.102]
ok: [192.168.56.107]

PLAY RECAP ******************************************************************
*
192.168.56.102             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.103             : ok=2    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.107             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0

diego@workstation:~/HOA6$
```

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.

```
- hosts: db_server
  become: true
  tasks:

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

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

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

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

-There are some changes in the code. I first put the installation of maria db before restarting code.

```
diego@workstation: ~/HOA6
File  Edit  View  Search  Terminal  Help
ok: [192.168.56.103]
ok: [192.168.56.107]

TASK [install mariadb package (CentOS)] ************************************
*
skipping: [192.168.56.103]
ok: [192.168.56.107]

TASK [install mariadb package (Ubuntu)] ************************************
*
skipping: [192.168.56.107]
changed: [192.168.56.103]

TASK [Mariadb- Restarting/Enabling] ****************************************
*
changed: [192.168.56.103]
changed: [192.168.56.107]

PLAY RECAP *********************************************************************
*
192.168.56.102             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.103             : ok=5    changed=2    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.107             : ok=7    changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0

diego@workstation:~/HOA6$
```

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.

   -After using the command on db_servers the mariadb is active (running)

```
diego@server2: ~
File  Edit  View  Search  Terminal  Help
diego@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 18:14:09 PST; 1min 23s ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
  Process: 10239 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_ST
  Process: 10236 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/S
  Process: 10135 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && V
  Process: 10133 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_STA
```

```
diego@localhost:~
File  Edit  View  Search  Terminal  Help
    link/ether 52:54:00:78:73:48 brd ff:ff:ff:ff:ff:ff
[diego@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 06:14:10 EDT; 2min 13s ago
  Process: 13865 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited, s
tatus=0/SUCCESS)
  Process: 13829 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, stat
us=0/SUCCESS)
 Main PID: 13864 (mysqld_safe)
    Tasks: 20
```

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.

```
- hosts: file_server
  become: true
  tasks:

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

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

-After running the site.yml the installation of samba package is successful.

```
diego@workstation: ~/HOA6
File  Edit  View  Search  Terminal  Help
ok: [192.168.56.103]

TASK [Mariadb- Restarting/Enabling] ******************************************
*
changed: [192.168.56.103]
changed: [192.168.56.107]

PLAY [file_server] ***********************************************************
*

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

TASK [install samba package] *************************************************
*
changed: [192.168.56.102]

PLAY RECAP *******************************************************************
*
192.168.56.102             : ok=6    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.103             : ok=5    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.107             : ok=7    changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0

diego@workstation:~/HOA6$
```

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.

```
  - 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"
```

```
- 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"
```

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

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

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

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

- There were many skips because there were no tags on them.

```
diego@workstation: ~/HOA6
File  Edit  View  Search  Terminal  Help
ok: [192.168.56.103]

TASK [Mariadb- Restarting/Enabling] ****************************************
*
changed: [192.168.56.103]
changed: [192.168.56.107]

PLAY [file_server] *********************************************************
*

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

TASK [install samba package] ***********************************************
*
ok: [192.168.56.102]

PLAY RECAP *****************************************************************
*
192.168.56.102             : ok=6    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.103             : ok=5    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.107             : ok=7    changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0

diego@workstation:~/HOA6$
```

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

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

```
diego@workstation:~/HOA6$ ansible-playbook --list-tags site.yml

playbook: site.yml

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

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

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

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

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

```
TASK [Gathering Facts] ******************************************************
*
ok: [192.168.56.107]
ok: [192.168.56.103]

TASK [install mariadb package (CentOS)] *************************************
*
skipping: [192.168.56.103]
ok: [192.168.56.107]

PLAY [file_server] **********************************************************
*

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

PLAY RECAP ******************************************************************
*
192.168.56.102             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.103             : ok=3    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.107             : ok=6    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
```

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

```
diego@workstation: ~/HOA6
File  Edit  View  Search  Terminal  Help

TASK [install mariadb package (CentOS)] ************************************
*
skipping: [192.168.56.103]
ok: [192.168.56.107]

TASK [install mariadb package (Ubuntu)] ************************************
*
skipping: [192.168.56.107]
ok: [192.168.56.103]

PLAY [file_server] *********************************************************
*

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

PLAY RECAP *****************************************************************
*
192.168.56.102             : ok=4    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.103             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.107             : ok=5    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
```

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

```
diego@workstation: ~/HOA6
File  Edit  View  Search  Terminal  Help
skipping: [192.168.56.102]
ok: [192.168.56.107]

PLAY [db_server] ***********************************************************
*

TASK [Gathering Facts] *****************************************************
*
ok: [192.168.56.103]
ok: [192.168.56.107]

PLAY [file_server] *********************************************************
*

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

PLAY RECAP *****************************************************************
*
192.168.56.102             : ok=5    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.103             : ok=3    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.107             : ok=5    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
```

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

```
diego@workstation: ~/HOA6
File  Edit  View  Search  Terminal  Help

TASK [install mariadb package (CentOS)] ************************************
*
skipping: [192.168.56.103]
ok: [192.168.56.107]

TASK [install mariadb package (Ubuntu)] ************************************
*
skipping: [192.168.56.107]
ok: [192.168.56.103]

PLAY [file_server] *********************************************************
*

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

PLAY RECAP *****************************************************************
*
192.168.56.102             : ok=5    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.103             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.107             : ok=6    changed=0    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
```

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.

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

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

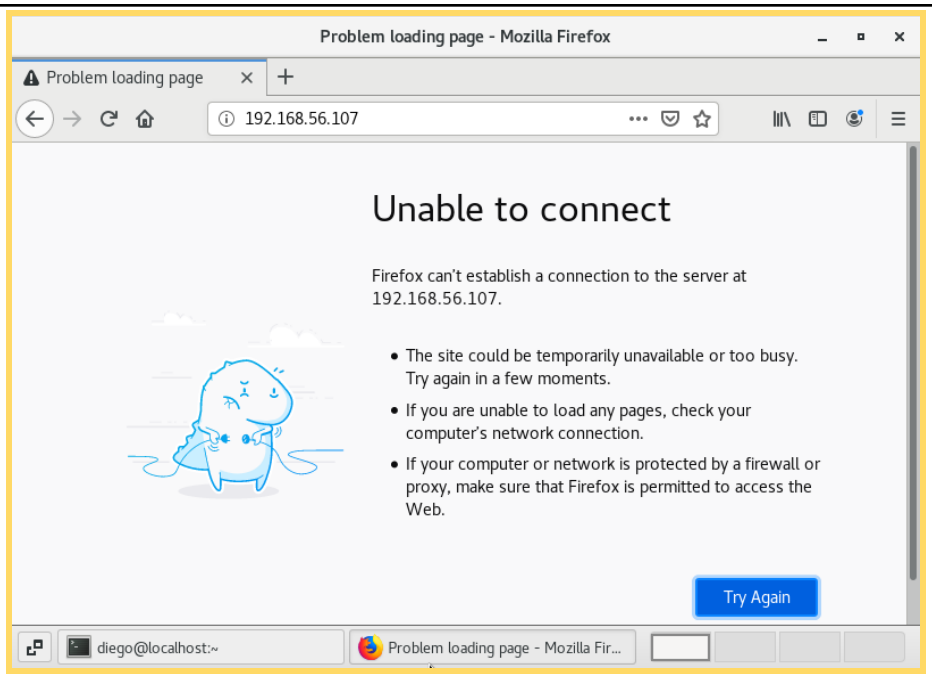
```
- hosts: db_servers
  become: true
  tasks:

  - 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

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

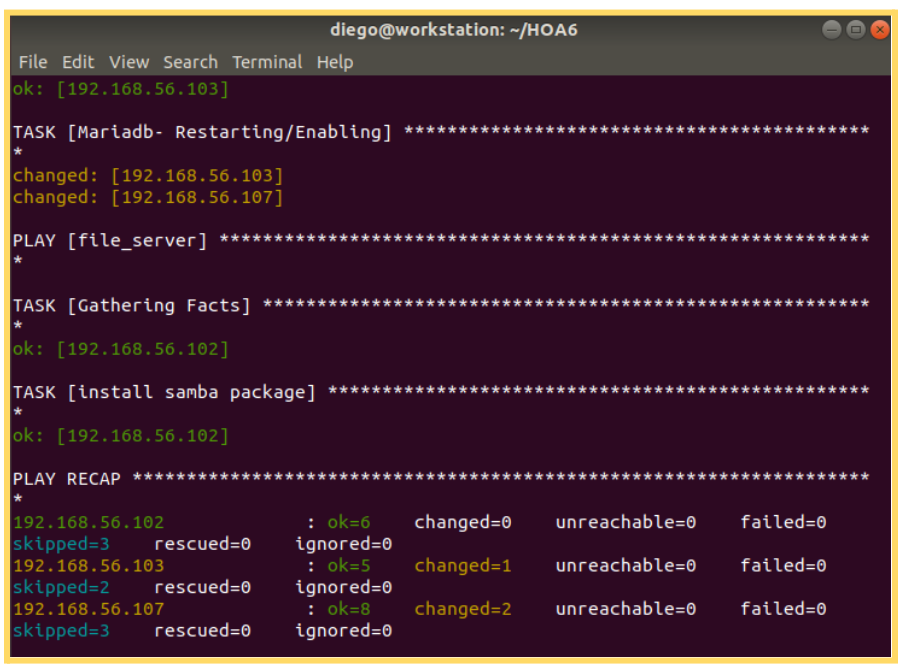
This is because in CentOS, installed packages' services are not run automatically. Thus, we need to create the module to run it automatically.

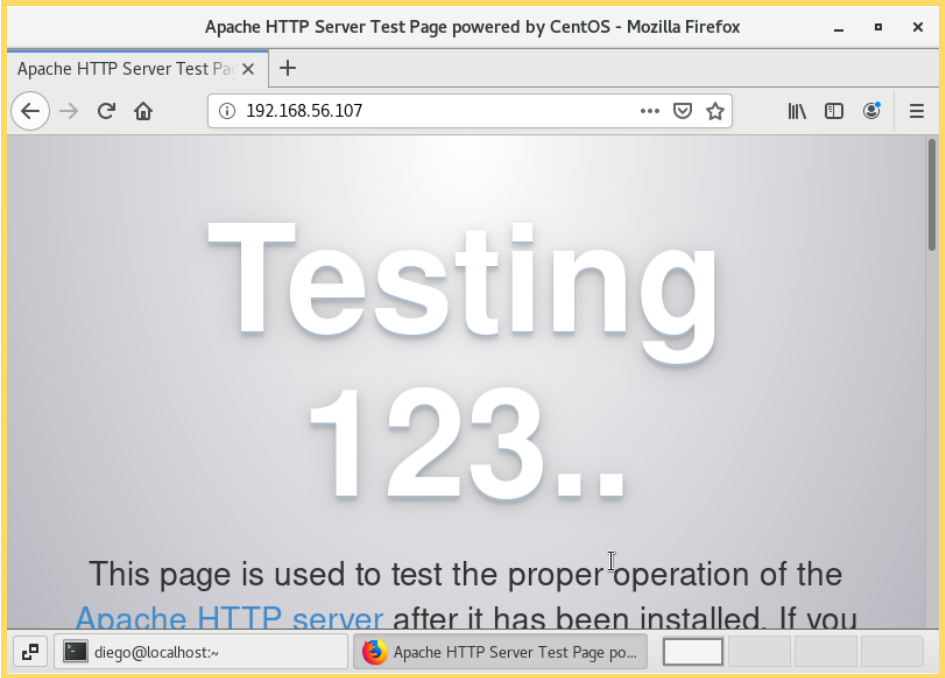
2. To test it, before you run the saved playbook, go to the CentOS server and stop the currently running httpd using the command *sudo systemctl stop httpd*. When prompted, enter the sudo password. After that, open the browser and enter the CentOS server's IP address. You should not be getting a display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

-There are some changes in the code and running the code has no errors.





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.

https://github.com/DiegoEspiritu11/HOA6

Reflections:

Answer the following:

1. What is the importance of putting our remote servers into groups?
   Managing servers remotely allows the creator or managed service providers to solve problems from anywhere, reducing the probability of downtime. They can quickly address server issues without waiting for someone to be physically present.
2. What is the importance of tags in playbooks?
   Tags provide a way to organize tasks into logical groups, making it easier to manage and maintain playbooks. You can assign tags to individual tasks, blocks, plays.
3. Why do think some services need to be managed automatically in playbooks?
   Automated playbooks follow predefined workflows and best practices, minimizing the risk of misconfigurations or other issues that can lead to service disruptions. This improves overall system reliability and availability.

Conclusions:

Doing this activity targeting specific nodes allows us to focus on a subset of servers. By grouping related nodes together, we can streamline management tasks, improve efficiency, and ensure consistent configurations across similar systems. This approach simplifies troubleshooting, reduces the risk of errors, and enhances overall system reliability.