Name: Zamora, Angelo E.	**Date Performed:** 09-30-2024
Course/Section: CpE31S2	**Date Submitted:** 10-02-2024
Instructor: Engr. Robin Valenzuela	**Semester and SY:** 1st Semester 2024 - 2025

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.

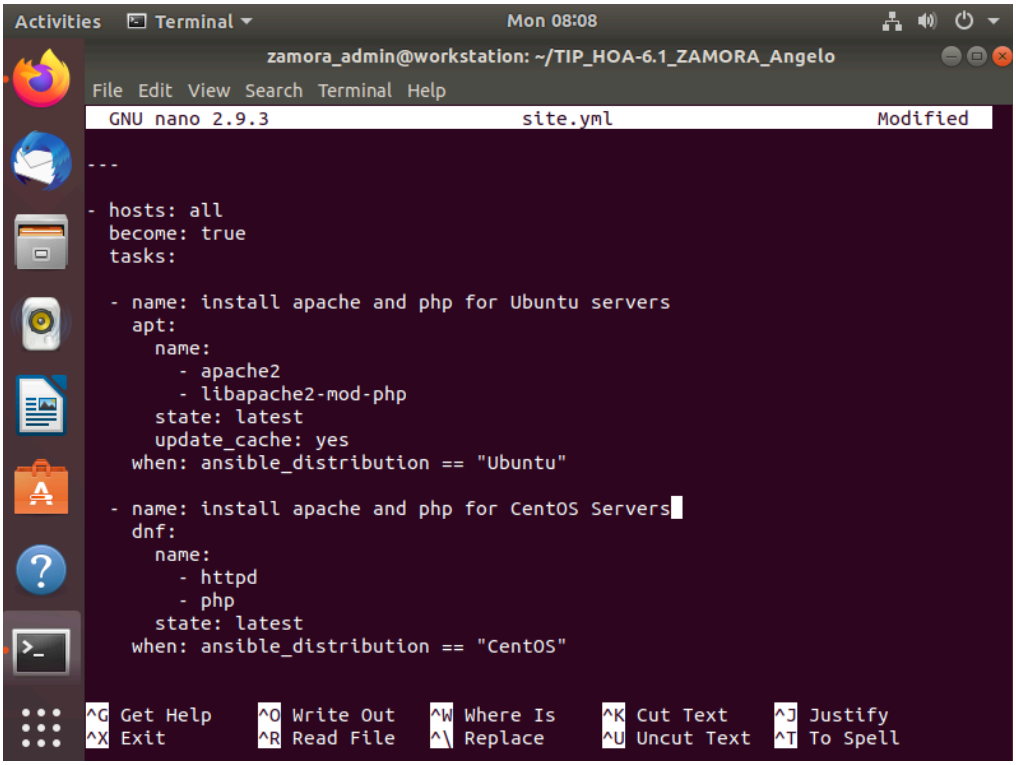
We also try to manage services that do not automatically run using the automations in playbook. For example, when we install web servers or httpd for CentOS, we notice that the service did not start automatically.

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.

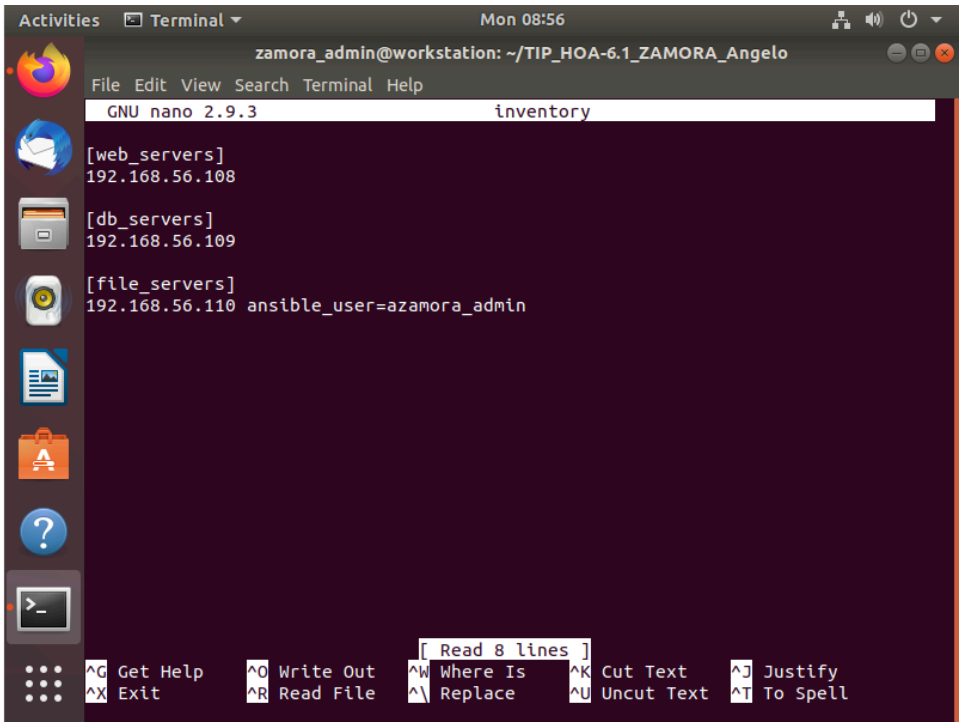
```
---

- 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"
```



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.



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.

```
- 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
```

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.

```
zamora_admin@workstation:~/TIP_HOA-6.1_ZAMORA_Angelo$ ansible-playbook --ask-be
come-pass site.yml
SUDO password:

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.108]
ok: [192.168.56.109]
ok: [192.168.56.110]

TASK [install updates (CentOS)] ***********************************************
*
skipping: [192.168.56.109]
skipping: [192.168.56.108]
ok: [192.168.56.110]

TASK [install updates (Ubuntu)] ***********************************************
*
skipping: [192.168.56.110]
ok: [192.168.56.108]
ok: [192.168.56.109]

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

- **It runs but there is no play for web_servers below.**

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.

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

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

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

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

PLAY RECAP ******************************************************************
*
192.168.56.108             : ok=4    changed=0    unreachable=0    failed=0
192.168.56.109             : ok=5    changed=1    unreachable=0    failed=0
192.168.56.110             : ok=2    changed=0    unreachable=0    failed=0

zamora_admin@workstation:~/TIP_HOA-6.1_ZAMORA_Angelo$
```

- **The playbook runs successfully and it installs the mariaDB.**

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.

```
zamora_admin@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 Mon 2024-09-30 08:57:37 +08; 2min 59s ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
  Process: 9987 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_STA
  Process: 9984 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SU
  Process: 9882 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VA
  Process: 9880 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_STAR
  Process: 9879 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/r
 Main PID: 9957 (mysqld)
   Status: "Taking your SQL requests now..."
    Tasks: 27 (limit: 4656)
   CGroup: /system.slice/mariadb.service
           └─9957 /usr/sbin/mysqld
lines 1-15/15 (END)
```

   - **The mariadb service is active since we added a task to enable the service.**

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
```

```
- 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.

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.110]

TASK [install samba package] **************************************************
*
changed: [192.168.56.110]

PLAY RECAP ********************************************************************
*
192.168.56.108             : ok=4    changed=0    unreachable=0    failed=0
192.168.56.109             : ok=5    changed=1    unreachable=0    failed=0
192.168.56.110             : ok=4    changed=1    unreachable=0    failed=0

zamora_admin@workstation:~/TIP_HOA-6.1_ZAMORA_Angelo$
```

- **The samba package runs successfully.**

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.

```yaml
---

- 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"
```

```yaml
- 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"
```

```yaml
- 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
```

```
- hosts: all
  become: true
  pre_tasks:

  - name: install updates (CentOS)
    tags: always
    dnf:
      name: "*"
      state: latest
    when: ansible_distribution == "CentOS"

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

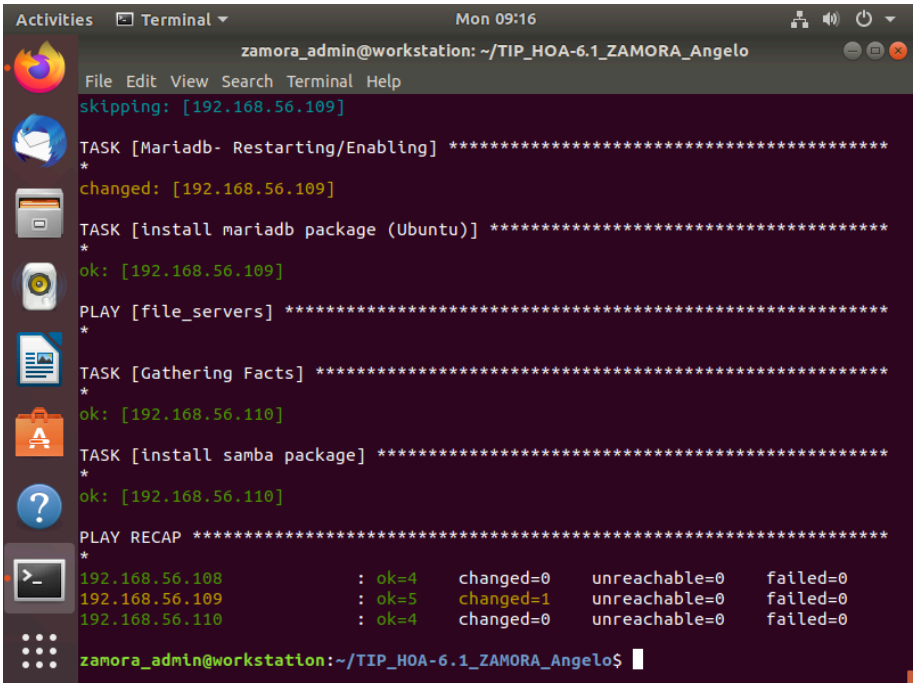
```
zamora_admin@workstation: ~/TIP_HOA-6.1_ZAMORA_Angelo
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                    site.yml                    Modifiec

- name: install apache and php for Ubuntu servers
  tags: apache,apache2,ubuntu
  apt:
    name:
      - apache2
      - libapache2-mod-php
    state: latest
    update_cache: yes
  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"
```

```
zamora_admin@workstation: ~/TIP_HOA-6.1_ZAMORA_Angelo
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                    site.yml                    Modifie

  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 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
```

Make sure to save the file and exit.

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



- **It runs normally since we only added tags.**

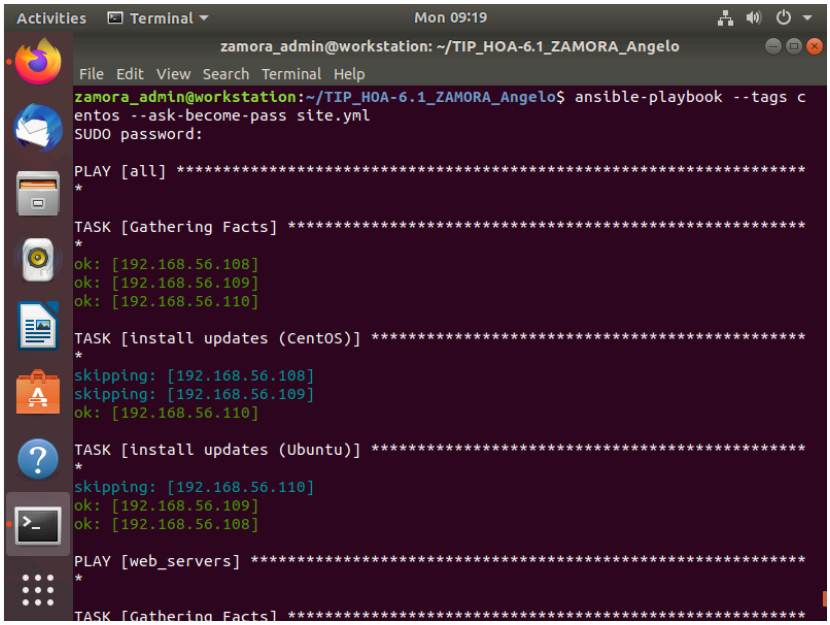
2. On the local machine, try to issue the following commands and describe each result:

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

```
zamora_admin@workstation:~/TIP_HOA-6.1_ZAMORA_Angelo$ ansible-playbook --list-t
ags 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]
zamora_admin@workstation:~/TIP_HOA-6.1_ZAMORA_Angelo$
```

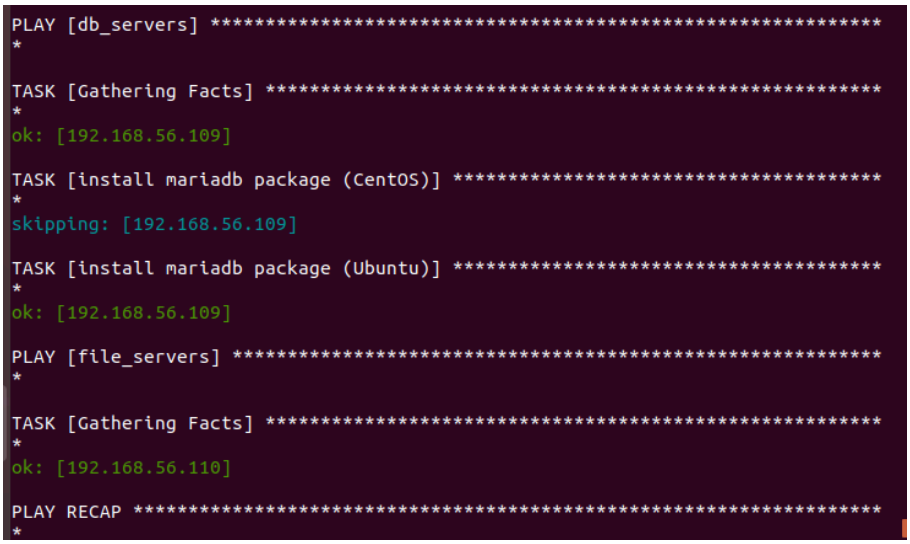
- **It list all the tags written in the playbook**

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



- **It runs tasks that have the tag centOS.**

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



- **The playbook runs but only the task that has db tags in it.**

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

```
TASK [install updates (CentOS)] ***********************************************
*
skipping: [192.168.56.108]
skipping: [192.168.56.109]
ok: [192.168.56.110]

TASK [install updates (Ubuntu)] ***********************************************
*
skipping: [192.168.56.110]
ok: [192.168.56.108]
ok: [192.168.56.109]

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.108]

TASK [install apache and php for Ubuntu servers] ******************************
*
ok: [192.168.56.108]

TASK [install apache and php for CentOS Servers] ******************************
*
skipping: [192.168.56.108]

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

- The playbook runs everything that involves apache.

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

```
zamora_admin@workstation:~/TIP_HOA-6.1_ZAMORA_Angelo$ ansible-playbook --tags "
apache,db" --ask-become-pass site.yml
SUDO password:

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.108]
ok: [192.168.56.109]
ok: [192.168.56.110]

TASK [install updates (CentOS)] ***********************************************
*
skipping: [192.168.56.108]
skipping: [192.168.56.109]
ok: [192.168.56.110]

TASK [install updates (Ubuntu)] ***********************************************
*
skipping: [192.168.56.110]
ok: [192.168.56.109]
ok: [192.168.56.108]

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

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

TASK [Gathering Facts] *****************************************************
*
ok: [192.168.56.108]

TASK [install apache and php for Ubuntu servers] ****************************
*
ok: [192.168.56.108]

TASK [install apache and php for CentOS Servers] ****************************
*
skipping: [192.168.56.108]

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

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

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

TASK [install mariadb package (Ubuntu)] *************************************
```

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

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

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

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

TASK [Gathering Facts] *****************************************************
*
ok: [192.168.56.110]

PLAY RECAP *****************************************************************
*
192.168.56.108             : ok=4    changed=0    unreachable=0    failed=0
192.168.56.109             : ok=4    changed=0    unreachable=0    failed=0
192.168.56.110             : ok=3    changed=0    unreachable=0    failed=0

zamora_admin@workstation:~/TIP_HOA-6.1_ZAMORA_Angelo$
```

- This time we used combinations of tags to run task simultaneously

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.

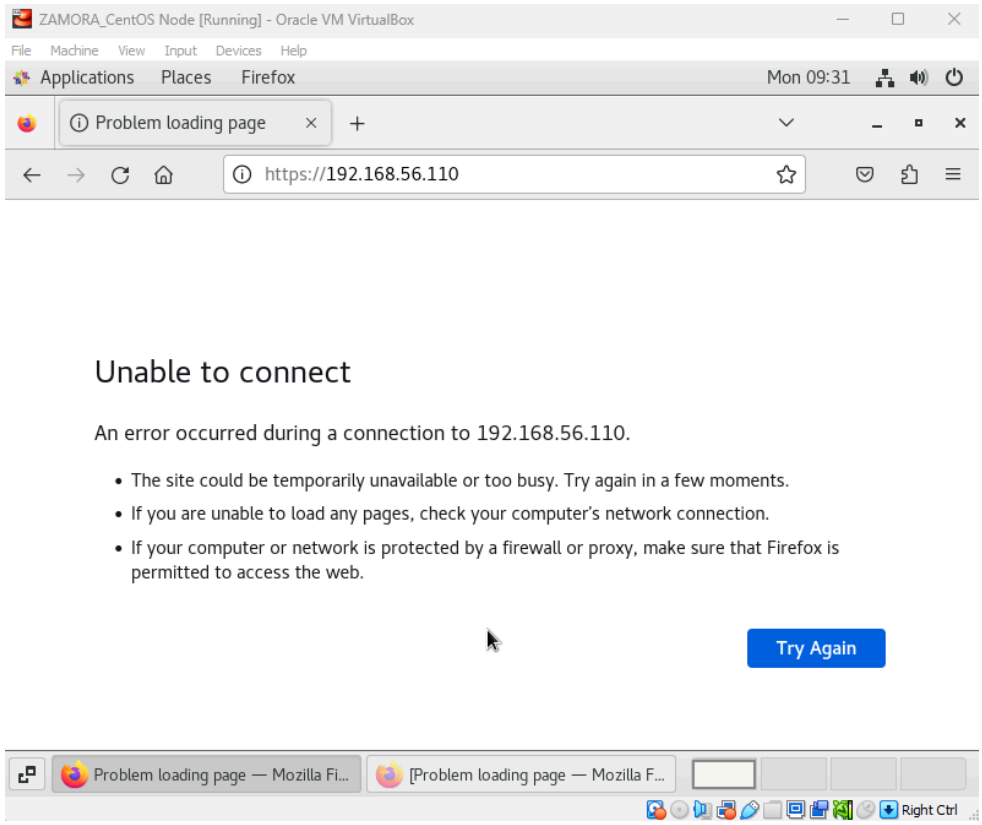
```
- hosts: db_servers
  become: true
  tasks:

  - 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

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

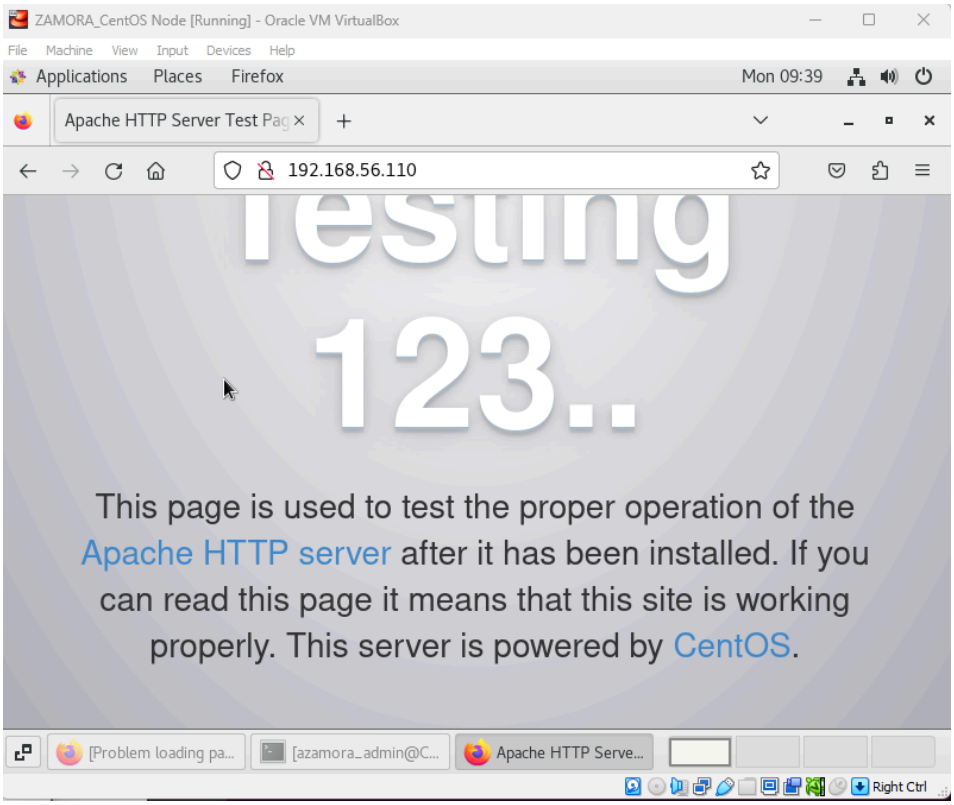
2. To test it, before you run the saved playbook, go to the CentOS server and stop the currently running httpd using the command *sudo systemctl stop httpd.* When prompted, enter the sudo password. After that, open the browser and enter the CentOS server's IP address. You should not be getting a display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

```
zamora_admin@workstation:~/TIP_HOA-6.1_ZAMORA_Angelo$ sudo nano site.yml
zamora_admin@workstation:~/TIP_HOA-6.1_ZAMORA_Angelo$ ssh azamora_admin@CentOS
Last login: Mon Sep 30 09:25:41 2024 from 192.168.56.107
[azamora_admin@CentOS ~]$ sudo systemctl stop httpd
[sudo] password for azamora_admin:
```



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.

```
TASK [start httpd (CentOS)] ***
skipping: [192.168.56.108]
skipping: [192.168.56.109]
ok: [192.168.56.110]
```




```
zamora_admin@workstation:~/TIP_HOA-6.1_ZAMORA_Angelo$ git add inventory ansible
.cfg site.yml
zamora_admin@workstation:~/TIP_HOA-6.1_ZAMORA_Angelo$ git commit -m "DONE"
[main 19883b0] DONE
 3 files changed, 96 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 inventory
 create mode 100644 site.yml
zamora_admin@workstation:~/TIP_HOA-6.1_ZAMORA_Angelo$ git push
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 977 bytes | 977.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0)
To github.com:GeloaceRT/TIP_HOA-6.1_ZAMORA_Angelo.git
   6d53562..19883b0  main -> main
zamora_admin@workstation:~/TIP_HOA-6.1_ZAMORA_Angelo$
```

Reflections:

Answer the following:

1. What is the importance of putting our remote servers into groups? The main purpose is grouping these servers for better arrangement and organization. The main advantage of this is when running a playbook you'll know which servers are doing or what server is affected in every task in the playbook. This is the versatility of a playbook in ansible where you'll see the servers is doing or what is the play and task for each group.
2. What is the importance of tags in playbooks? Tags gives you the ability to run the playbook like one task, multiple task, or a specific task you want to do. It can be tagged by their distribution, by what package, and etc.
3. Why do think some services need to be managed automatically in playbooks? So that you will save time in setting up when you have a new machine, you'll just run the playbook and the playbook will take care of the installation and starting or setting up the service of the package you installed. So the advantage of setting up the startup service will save you time.

Github Link: https://github.com/GeloaceRT/TIP_HOA-6.1_ZAMORA_Angelo