Name: Alexis Neil D. Deniega	**Date Performed:** September 12th, 2025
Course/Section: CPE 212 - CPE31S4	**Date Submitted:** September 19th, 2025
Instructor: Engr. Robin	**Semester and SY:** 2025-26 1st Sem

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:

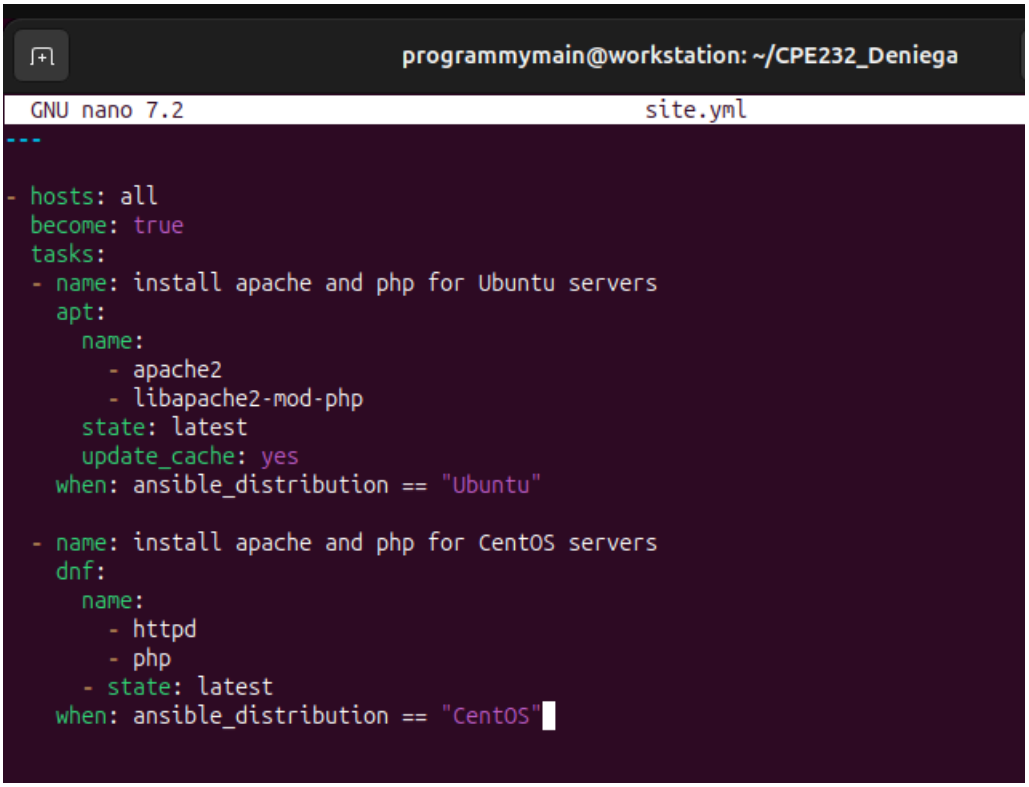
In this activity, you will need to create another Ubuntu VM and name it Server 3. Likewise, you need to activate the second adapter to a host-only adapter after the installations. Take note of the IP address of the Server 3. Make sure to use the command *ssh-copy-id* to copy the public key to Server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

```
programmymain@workstation:~$ ssh-copy-id -f -i ~/.ssh/id_rsa programmymain@192.168.56.123
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/programmymain/.ssh/id_rsa.pub"

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'programmymain@192.168.56.123'"
and check to make sure that only the key(s) you wanted were added.

programmymain@workstation:~$ ssh programmymain@192.168.56.123
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-29-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

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

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
```

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.

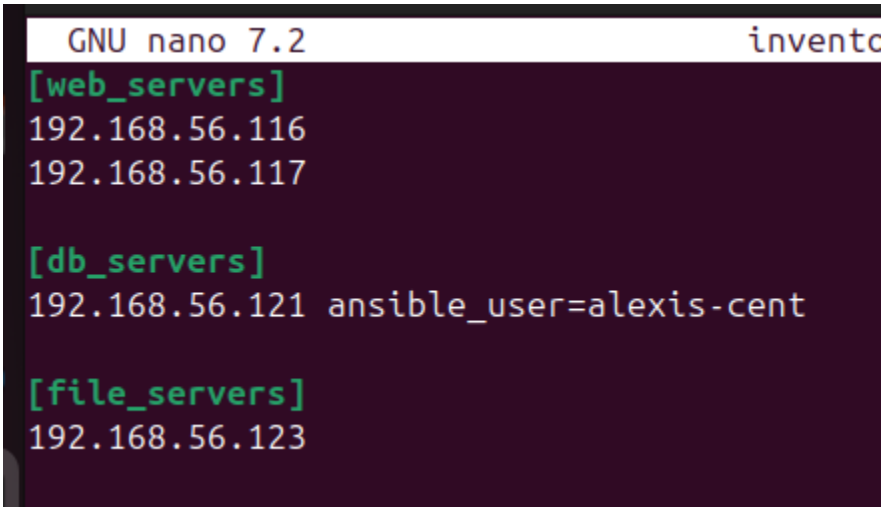
```
---

- 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.

```
  GNU nano 7.2                                        site.yml *
---
- hosts: all
  become: true
  pre_tasks:

  - name: install updates (centos)
    dnf:
      update_only: yes
      update_cahce: 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
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: install apache and php for CentOS servers
    dnf:
      name:
```

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.

```
TASK [install apache and php for CentOS servers] ************************
skipping: [192.168.56.117]
skipping: [192.168.56.116]

PLAY RECAP *************************************************************
192.168.56.116             : ok=4    changed=1    unreachable=0    failed=0
kipped=2    rescued=0    ignored=0
192.168.56.117             : ok=4    changed=1    unreachable=0    failed=0
kipped=2    rescued=0    ignored=0
192.168.56.121             : ok=2    changed=0    unreachable=0    failed=0
kipped=1    rescued=0    ignored=0
192.168.56.123             : ok=2    changed=1    unreachable=0    failed=0
kipped=1    rescued=0    ignored=0
```

```
TASK [install updates (ubuntu)] *****************************************
skipping: [192.168.56.121]
changed: [192.168.56.116]
changed: [192.168.56.117]
changed: [192.168.56.123]

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

TASK [Gathering Facts] **************************************************
ok: [192.168.56.116]
ok: [192.168.56.117]

TASK [install apache and php for Ubuntu servers] ************************
ok: [192.168.56.117]
ok: [192.168.56.116]
```

After the pre-tasks (encompassing all servers), the task proper only involved web servers (for which there were only two).

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



^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location
```

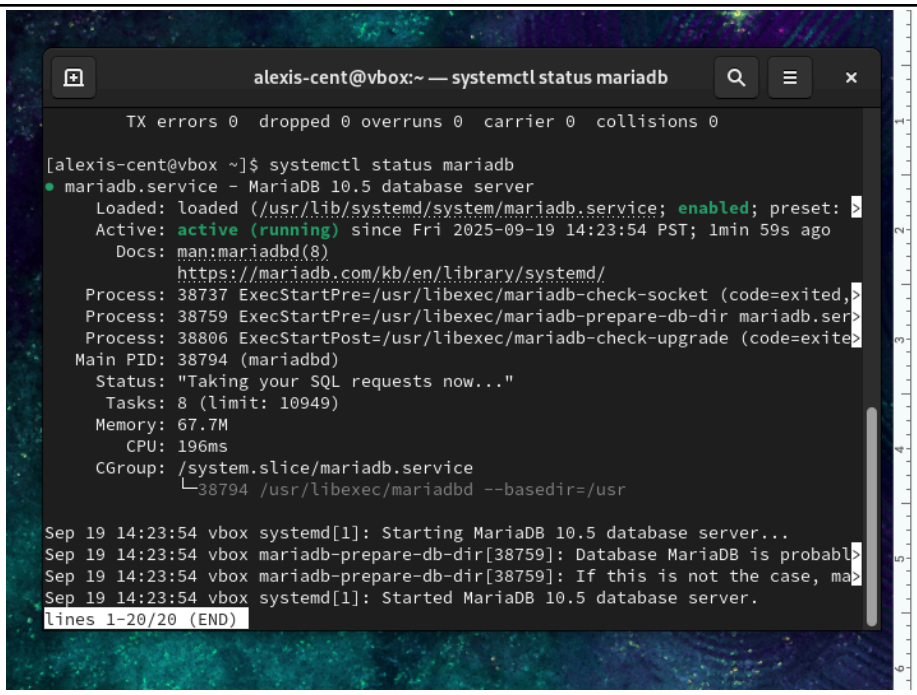
Run the *site.yml* file and describe the result.

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

TASK [Gathering Facts] *********************************************************************************************************
ok: [192.168.56.117]
ok: [192.168.56.116]

TASK [install apache and php for Ubuntu servers] *******************************************************************************
ok: [192.168.56.116]
ok: [192.168.56.117]

TASK [install apache and php for CentOS servers] *******************************************************************************
skipping: [192.168.56.117]
skipping: [192.168.56.116]

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

TASK [Gathering Facts] *********************************************************************************************************
ok: [192.168.56.121]

TASK [install mariadb package (centos)] ****************************************************************************************
ok: [192.168.56.121]

TASK [mariadb -- restarting/enabling] ******************************************************************************************
changed: [192.168.56.121]

TASK [install mariadb package (ubuntu)] ****************************************************************************************
skipping: [192.168.56.121]

PLAY RECAP *********************************************************************************************************************
192.168.56.116             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.117             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.121             : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.123             : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

programmymain@workstation:~/CPE232_Deniega$
```

Just like the instruction above, the new play filters out so that it only includes those in the database server group

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.



As you can see, it works perfectly.

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.

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

TASK [Gathering Facts] *********************************************************************************************************
ok: [192.168.56.123]

TASK [install samba package] ***************************************************************************************************
changed: [192.168.56.123]

PLAY RECAP *********************************************************************************************************************
192.168.56.116             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.117             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.121             : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.123             : ok=4    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Very much the same as any other group filter. It only affects the .123 server.

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
```

```
name: install updates (
tags: always
dnf:
  update_only: yes
  update_cache: yes
when: ansible_distribut

name: install updates (
tags: always
```

```
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
    update_cache: yes
  when: ansible_distribution == "Ubuntu"

- name: install apache and php for CentOS servers
  tags: apache,centos,httpd
  dnf:
```

```
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:
```

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

```
PLAY RECAP *********************************************************************************************************
192.168.56.116             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.117             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.121             : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.123             : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

I am going to be honest here, I don't see much.

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

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

```
programmymain@workstation:~/CPE232_Deniega$ 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]
```

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

```
TASK [install updates (ubuntu)] ************************************************************************************
skipping: [192.168.56.121]
ok: [192.168.56.123]
ok: [192.168.56.116]
ok: [192.168.56.117]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.117]
ok: [192.168.56.116]

TASK [install apache and php for CentOS servers] *******************************************************************
skipping: [192.168.56.116]
skipping: [192.168.56.117]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.121]

TASK [install mariadb package (centos)] ****************************************************************************
ok: [192.168.56.121]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.123]

PLAY RECAP *********************************************************************************************************
192.168.56.116             : ok=3    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.117             : ok=3    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.121             : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.123             : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

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

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

TASK [Gathering Facts] *****************
ok: [192.168.56.121]
ok: [192.168.56.123]
ok: [192.168.56.116]
ok: [192.168.56.117]

TASK [install updates (centos)] ********
skipping: [192.168.56.116]
skipping: [192.168.56.117]
skipping: [192.168.56.123]
ok: [192.168.56.121]

TASK [install updates (ubuntu)] ********
skipping: [192.168.56.121]
ok: [192.168.56.123]
ok: [192.168.56.117]
ok: [192.168.56.116]

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

TASK [Gathering Facts] *****************
ok: [192.168.56.116]
ok: [192.168.56.117]

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

TASK [Gathering Facts] *****************
ok: [192.168.56.121]

TASK [install mariadb package (centos)] *
ok: [192.168.56.121]

TASK [install mariadb package (ubuntu)] *
skipping: [192.168.56.121]
```

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

```
ok: [192.168.56.121]

TASK [install updates (ubuntu)] ************************************************************************
skipping: [192.168.56.121]
ok: [192.168.56.123]
ok: [192.168.56.116]
ok: [192.168.56.117]

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.117]
ok: [192.168.56.116]

TASK [install apache and php for Ubuntu servers] *******************************************************
ok: [192.168.56.116]
ok: [192.168.56.117]

TASK [install apache and php for CentOS servers] *******************************************************
skipping: [192.168.56.116]
skipping: [192.168.56.117]

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.121]
```

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

```
ok: [192.168.56.116]

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.116]
ok: [192.168.56.117]

TASK [install apache and php for Ubuntu servers] *******************************************************
ok: [192.168.56.117]
ok: [192.168.56.116]

TASK [install apache and php for CentOS servers] *******************************************************
skipping: [192.168.56.116]
skipping: [192.168.56.117]

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.121]

TASK [install mariadb package (centos)] ****************************************************************
ok: [192.168.56.121]

TASK [install mariadb package (ubuntu)] ****************************************************************
skipping: [192.168.56.121]

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.123]

PLAY RECAP *********************************************************************************************
192.168.56.116             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.117             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.121             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.123             : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

The keyword "always" means that it **always** is in **any** tag filtering. Tags are helpful to further filter out anything.

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_districution == "CentOS"
```

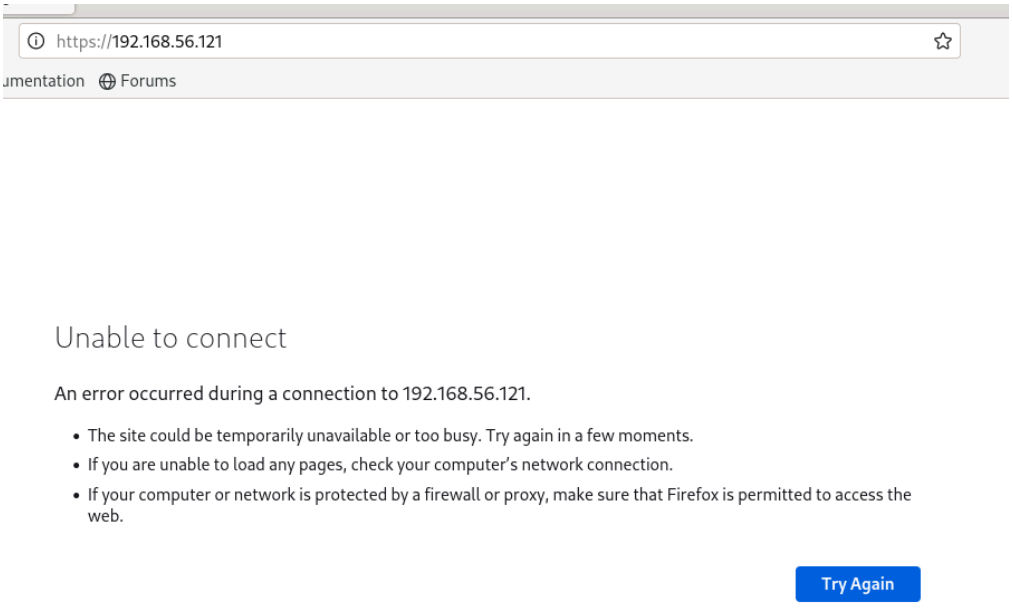
You would also notice from our previous activity that we already created a module that runs a service.

```
- 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.

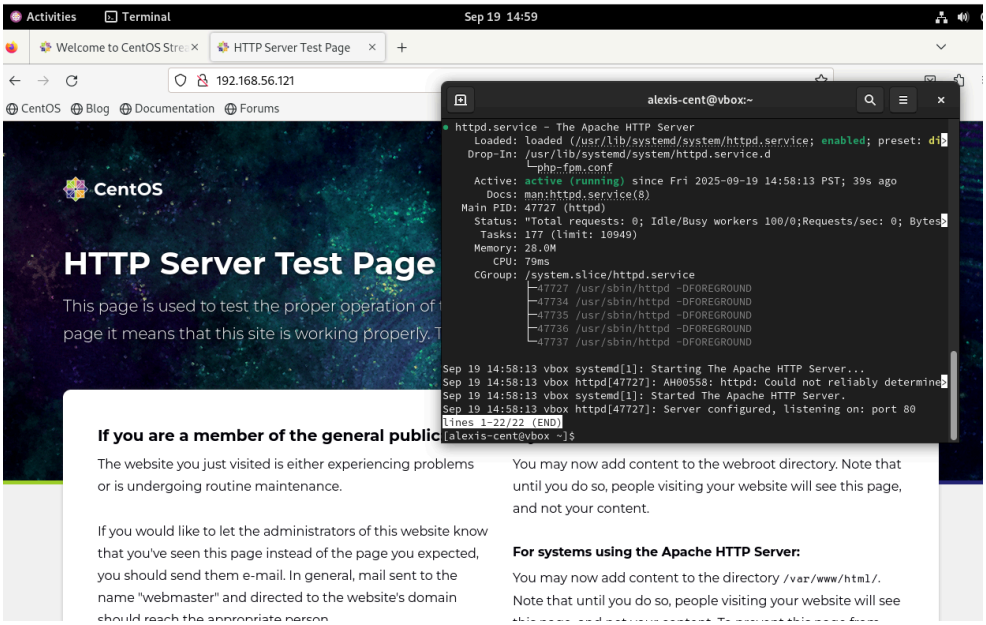


```
  GNU nano 7.2
[web_servers]
192.168.56.116
192.168.56.117
192.168.56.121 ansible_user=alexis-cent

[db_servers]
192.168.56.121 ansible_user=alexis-cent

[file_servers]
192.168.56.123

[all:vars]
ansible_ssh_common_args='-o GSSAPIAuthentication=no'
```

Also, I put .121 at web_servers because I realized that this won't work at all if I didn't put .121 at the web_servers group.

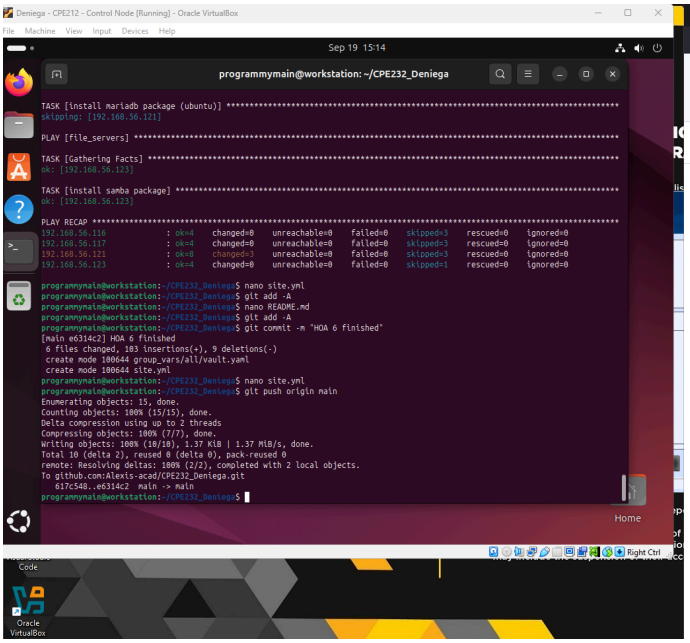
3. Go to the local machine and this time, run the *site.yml* file. Then after running the file, go again to the CentOS server and enter its IP address on the browser. Describe the result.

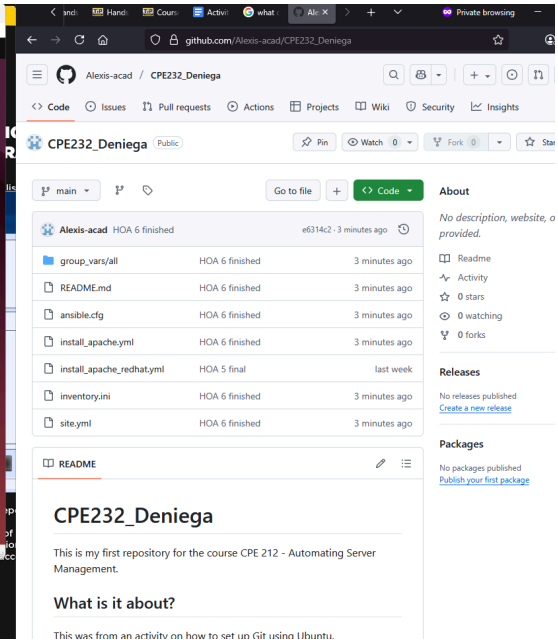
   To automatically enable the service every time we run the playbook, use the command *enabled: true* similar to Figure 7.1.2 and save the playbook.



It works!

```
      - php
    state: latest
  when: ansible_distribution == "CentOS"

- 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:
  - name: install mariadb package (centos)
```





pushed to git just in case

Reflections:

Answer the following:

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

   Each remote server has different needs and purposes that define their role and usefulness in the server-side of things. Just installing every package in every single remote server is both time-consuming, and a waste of useless space. Grouping servers into their roughly-equivalent roles will make sure that servers only get the package and tools they need to do their stuff, leaving more space and time on actual data.

2. What is the importance of tags in playbooks?
   Tags are just another way to filter out tasks and plays on playbooks like this one. However, unlike whens and groups, they can be used in individual tasks, in multiple plays simultaneously, which provides a lot of flexibility in doing/installing something. This also allows sysads and netads to troubleshoot anything wrong by just doing it tag-by-tag.
3. Why do you think some services need to be managed automatically in playbooks?

   It saves time for sysadmins that definitely has much better time to do rather than checking for the 1000th time if systemctl has enabled services for httpd.