Name: Frias, Abegail L.	Date Performed: Oct. 11, 2024
Course/Section: CPE212 - CPE31S21	Date Submitted: Oct. 11, 2024
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st Sem/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.

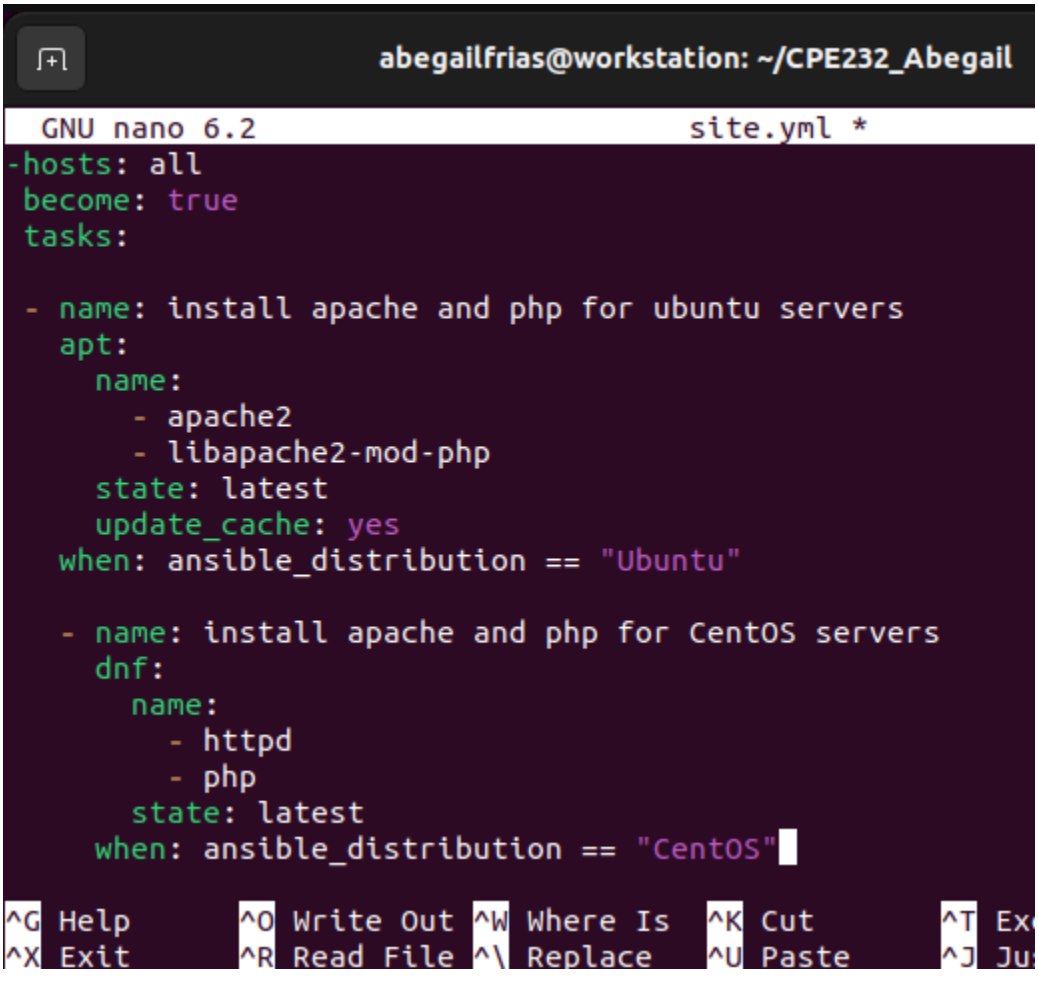
Requirement:

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:

```
abegailfrias@workstat
[web_servers]
192.168.56.107

[db_servers]
192.168.56.106
192.168.56.105

[file_servers]
192.168.56.109
```

Make sure to save the file and exit.

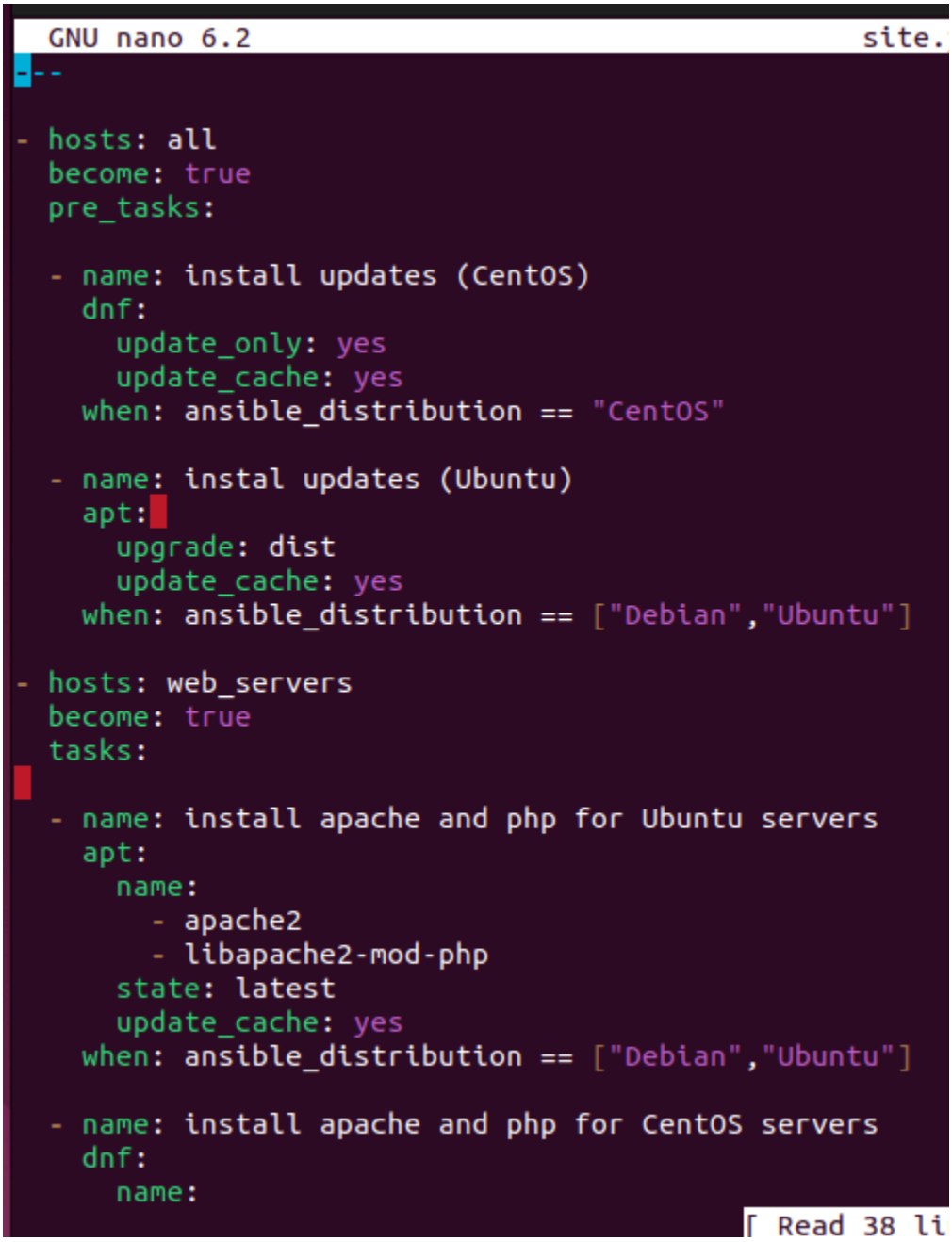
Right now, we have created groups in our inventory file and put each server in its own group. In other cases, you can have a server be a member of multiple groups, for example you have a test server that is also a web server.

```
abegailfrias@workstation:~/CPE232_Abegail$ ansible-playbook -i inventory site.yml --ask-become-pass
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.107]
ok: [192.168.56.106]
ok: [192.168.56.109]
ok: [192.168.56.105]

TASK [install apache and php for Ubuntu servers] *******************************
skipping: [192.168.56.107]
skipping: [192.168.56.106]
skipping: [192.168.56.105]
skipping: [192.168.56.109]

TASK [install apache and php for CentOS servers] *******************************
skipping: [192.168.56.107]
skipping: [192.168.56.106]
skipping: [192.168.56.109]
changed: [192.168.56.105]

PLAY RECAP *********************************************************************
192.168.56.105             : ok=2    changed=1    unreachable=0    failed=0    skipped=1    rescued=0
 ignored=0
192.168.56.106             : ok=1    changed=0    unreachable=0    failed=0    skipped=2    rescued=0
 ignored=0
192.168.56.107             : ok=1    changed=0    unreachable=0    failed=0    skipped=2    rescued=0
 ignored=0
192.168.56.109             : ok=1    changed=0    unreachable=0    failed=0    skipped=2    rescued=0
 ignored=0
```

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

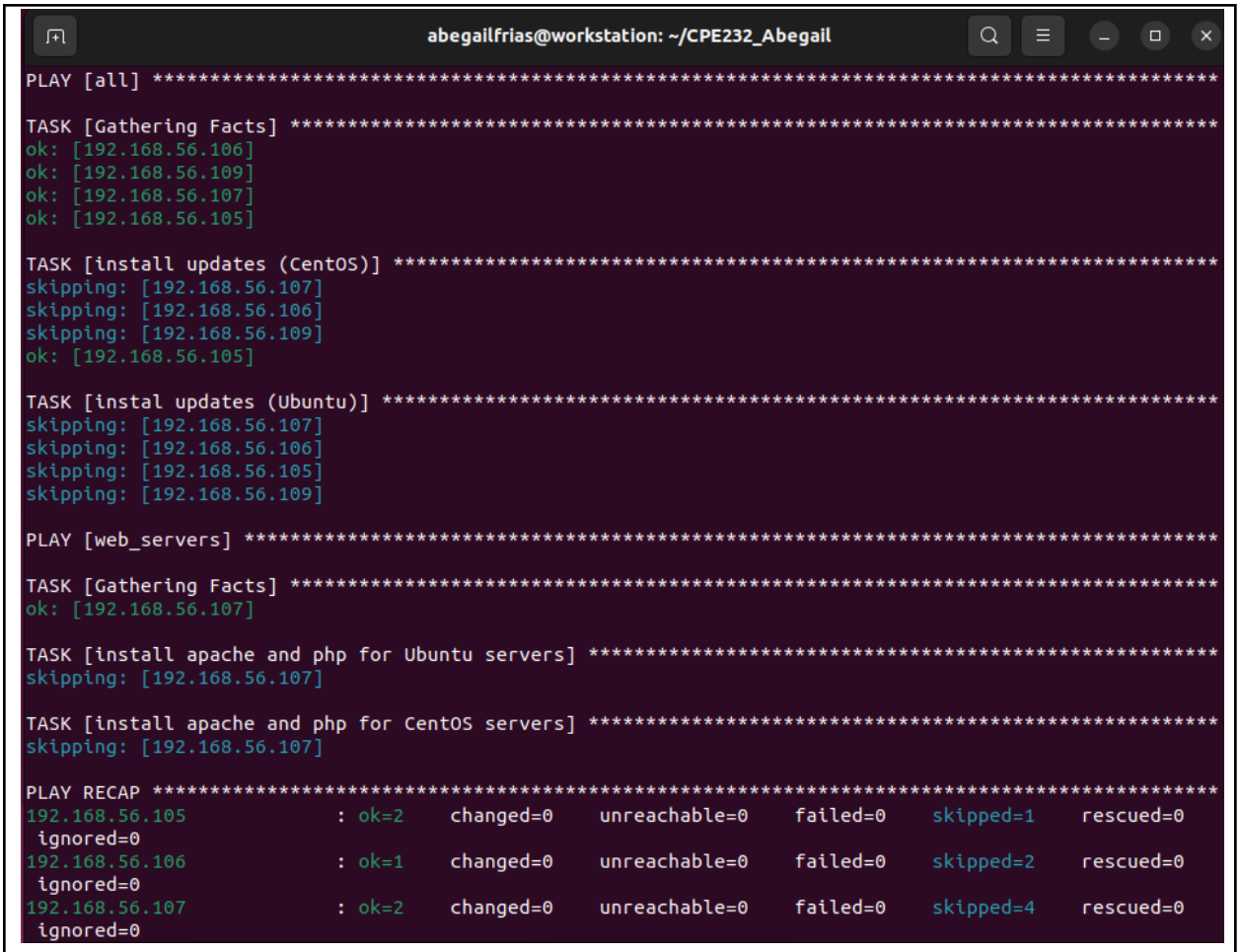
```
---
- hosts: all
  become: true
  pre_tasks:

  - 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.



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 playbook executed both the initial tasks and the group-specific tasks.

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.

```
 GNU nano 6.2                                          site.yml *
    update_cache: yes
  when: ansible_distribution == ["Debian","Ubuntu"]

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

- 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
      enable: true

  - name: install mariadb package (Ubuntu)
    apt:
      name: mariadb-server
      state: latest
    when: ansible_distribution == ["Debian","Ubuntu"]
```

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

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

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

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

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [install mariadb package (CentOS)] ****************************************************************
skipping: [192.168.56.106]
ok: [192.168.56.105]

TASK [Mariadb- Restarting/Enabling] ********************************************************************
changed: [192.168.56.106]
changed: [192.168.56.105]

TASK [install mariadb package (Ubuntu)] ****************************************************************
skipping: [192.168.56.106]
skipping: [192.168.56.105]

PLAY RECAP *********************************************************************************************
192.168.56.105             : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0
 ignored=0
192.168.56.106             : ok=3    changed=1    unreachable=0    failed=0    skipped=4    rescued=0
 ignored=0
192.168.56.107             : ok=2    changed=0    unreachable=0    failed=0    skipped=4    rescued=0
 ignored=0
192.168.56.109             : ok=1    changed=0    unreachable=0    failed=0    skipped=2    rescued=0
 ignored=0

abegailfrias@workstation:~/CPE232_Abegail$
```

- Both OS allowed for the installation, activation, and restart of 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.

```
abegailfrias@workstation:~/CPE232_Abegail$ systemctl status mariadb
● mariadb.service - MariaDB 10.6.18 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2024-10-10 00:31:40 +08; 1min 10s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 8095 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysqld (code=exited>
    Process: 8096 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exite>
    Process: 8098 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= ||   VAR=`cd /usr/b>
    Process: 8138 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exit>
    Process: 8140 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
   Main PID: 8127 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 10 (limit: 14983)
     Memory: 62.5M
        CPU: 422ms
     CGroup: /system.slice/mariadb.service
             └─8127 /usr/sbin/mariadbd
lines 1-17/17 (END)
```

```
[abegailfrias@vbox ~]$ systemctl status mariadb
● mariadb.service - MariaDB 10.5 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: >
     Active: active (running) since Thu 2024-10-10 00:31:40 PST; 1min 58s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 43580 ExecStartPre=/usr/libexec/mariadb-check-socket (code=exited, >
    Process: 43602 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir mariadb.ser>
    Process: 43649 ExecStartPost=/usr/libexec/mariadb-check-upgrade (code=exite>
   Main PID: 43637 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 8 (limit: 10950)
     Memory: 63.7M
        CPU: 189ms
     CGroup: /system.slice/mariadb.service
             └─43637 /usr/libexec/mariadbd --basedir=/usr
lines 1-15/15 (END)
```

- Mariadb has been installed and is operational on both of the systems, as evidenced by their distinct operating systems.

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.

```
  - name: install mariadb package (Ubuntu)
    apt:
      name: mariadb-server
      state: latest
    when: ansible_distribution == ["Debian","Ubuntu"]

- hosts: file_servers
  become: true
  tasks:

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

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

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

TASK [Gathering Facts] **********************************************************************
ok: [192.168.56.106]
ok: [192.168.56.105]

TASK [install mariadb package (CentOS)] *****************************************************
skipping: [192.168.56.106]
ok: [192.168.56.105]

TASK [Mariadb- Restarting/Enabling] *********************************************************
changed: [192.168.56.105]
changed: [192.168.56.106]

TASK [install mariadb package (Ubuntu)] *****************************************************
skipping: [192.168.56.106]
skipping: [192.168.56.105]

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

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

TASK [install samba package] ****************************************************************
changed: [192.168.56.109]

PLAY RECAP **********************************************************************************
192.168.56.105             : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0
 ignored=0
192.168.56.106             : ok=3    changed=1    unreachable=0    failed=0    skipped=4    rescued=0
 ignored=0
192.168.56.107             : ok=2    changed=0    unreachable=0    failed=0    skipped=4    rescued=0
 ignored=0
192.168.56.109             : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0
 ignored=0

abegailfrias@workstation:~/CPE232_Abegail$
```

- Installing the Samba package in the file_servers group was made possible by the playbook.

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.

```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: instal updates (Ubuntu)
    tags: always
    apt:
      upgrade: dist
      update_cache: yes
    when: ansible_distribution == ["Debian","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
      update_cache: yes
    when: ansible_distribution == ["Debian","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
    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 == ["Debian","Ubuntu"]
```

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

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

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

```
abegalirrias@workstation:~/CPE232_Abegalir$ ansible-playbook -i inventory site.yml --ask-become-pass
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.107]
ok: [192.168.56.106]
ok: [192.168.56.109]
ok: [192.168.56.105]

TASK [install updates (CentOS)] ************************************************************************
skipping: [192.168.56.107]
skipping: [192.168.56.106]
skipping: [192.168.56.109]
ok: [192.168.56.105]

TASK [instal updates (Ubuntu)] *************************************************************************
skipping: [192.168.56.107]
skipping: [192.168.56.106]
skipping: [192.168.56.105]
skipping: [192.168.56.109]

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

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

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

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

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.106]
ok: [192.168.56.105]
```

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

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

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

TASK [Gathering Facts] **********************************************************************************
ok: [192.168.56.106]
ok: [192.168.56.105]

TASK [install mariadb package (CentOS)] *****************************************************************
skipping: [192.168.56.106]
ok: [192.168.56.105]

TASK [Mariadb- Restarting/Enabling] *********************************************************************
changed: [192.168.56.105]
changed: [192.168.56.106]

TASK [install mariadb package (Ubuntu)] *****************************************************************
skipping: [192.168.56.106]
skipping: [192.168.56.105]

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

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

TASK [install samba package] ****************************************************************************
ok: [192.168.56.109]

PLAY RECAP **********************************************************************************************
192.168.56.105             : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignore
d=0
192.168.56.106             : ok=3    changed=1    unreachable=0    failed=0    skipped=4    rescued=0    ignore
d=0
192.168.56.107             : ok=2    changed=0    unreachable=0    failed=0    skipped=4    rescued=0    ignore
d=0
```

- The tags that were placed on the playbook allowed the tasks to be carried out in a very particular way.

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

   2.1 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]
abegailfrias@workstation:~/CPE232_Abegail$
```

- This command enumerates every tag that was discussed for every playbook process.

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

2.3

```
skipping: [192.168.56.107]
skipping: [192.168.56.105]
skipping: [192.168.56.106]
skipping: [192.168.56.109]

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

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

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

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.106]
ok: [192.168.56.105]

TASK [install mariadb package (CentOS)] ****************************************************************
skipping: [192.168.56.106]
ok: [192.168.56.105]

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

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

PLAY RECAP *********************************************************************************************
192.168.56.105             : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignore
d=0
192.168.56.106             : ok=2    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignore
d=0
192.168.56.107             : ok=2    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignore
d=0
192.168.56.109             : ok=2    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignore
d=0
```

- All playbook tasks are carried out by this command using the tag centos.

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

```
skipping: [192.168.56.105]
skipping: [192.168.56.106]
skipping: [192.168.56.109]

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

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

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.106]
ok: [192.168.56.105]

TASK [install mariadb package (CentOS)] ****************************************************************
skipping: [192.168.56.106]
ok: [192.168.56.105]

TASK [install mariadb package (Ubuntu)] ****************************************************************
skipping: [192.168.56.105]
skipping: [192.168.56.106]

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

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

PLAY RECAP *********************************************************************************************
192.168.56.105             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignore
d=0
192.168.56.106             : ok=2    changed=0    unreachable=0    failed=0    skipped=4    rescued=0    ignore
d=0
192.168.56.107             : ok=2    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignore
d=0
192.168.56.109             : ok=2    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignore
d=0
```

- With the tag db, this command completes all playbook tasks.
  2.5 ansible-playbook --tags apache --ask-become-pass site.yml

```
TASK [instal updates (Ubuntu)] ***********************************************************************
skipping: [192.168.56.107]
skipping: [192.168.56.105]
skipping: [192.168.56.106]
skipping: [192.168.56.109]

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

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

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

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

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

TASK [Gathering Facts] *******************************************************************************
ok: [192.168.56.106]
ok: [192.168.56.105]

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

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

PLAY RECAP *******************************************************************************************
192.168.56.105             : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignore
d=0
192.168.56.106             : ok=2    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignore
d=0
192.168.56.107             : ok=2    changed=0    unreachable=0    failed=0    skipped=4    rescued=0    ignore
d=0
192.168.56.109             : ok=2    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignore
d=0
```

- With the tag apache, this command completes all playbook tasks.
  2.6 *ansible-playbook --tags “apache,db” --ask-become-pass site.yml*

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

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

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

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

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

TASK [Gathering Facts] *******************************************************************************************
ok: [192.168.56.106]
ok: [192.168.56.105]

TASK [install mariadb package (CentOS)] **************************************************************************
skipping: [192.168.56.106]
ok: [192.168.56.105]

TASK [install mariadb package (Ubuntu)] **************************************************************************
skipping: [192.168.56.105]
skipping: [192.168.56.106]

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

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

PLAY RECAP *******************************************************************************************************
192.168.56.105             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignore
d=0
192.168.56.106             : ok=2    changed=0    unreachable=0    failed=0    skipped=4    rescued=0    ignore
d=0
192.168.56.107             : ok=2    changed=0    unreachable=0    failed=0    skipped=4    rescued=0    ignore
d=0
192.168.56.109             : ok=2    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignore
d=0
```

- All playbook tasks with the tags apache and db only are executed by this command.

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.

```
    when: ansible_distribution == "CentOS"

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

- hosts: db_servers
```

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

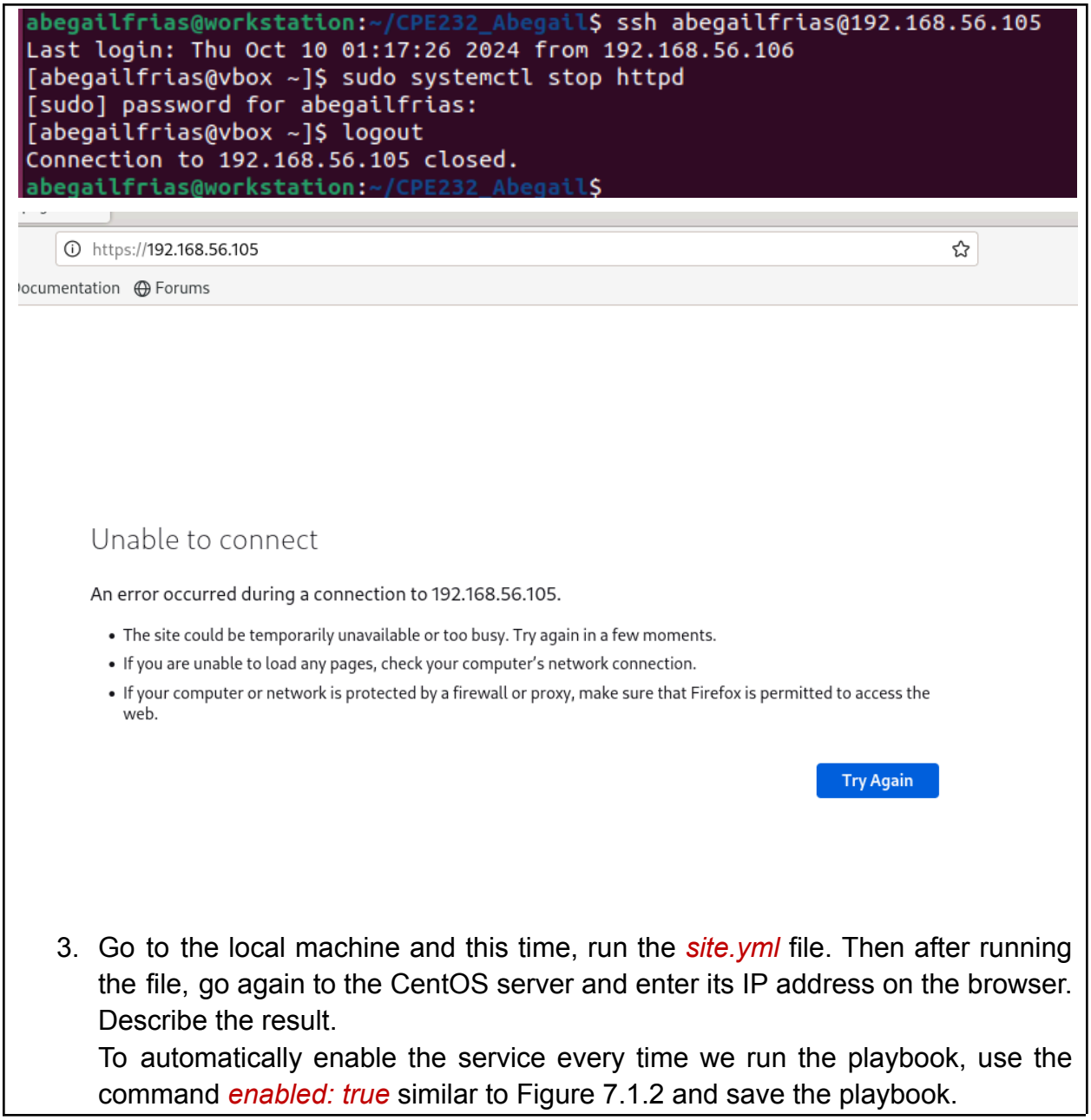
y

```
- 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

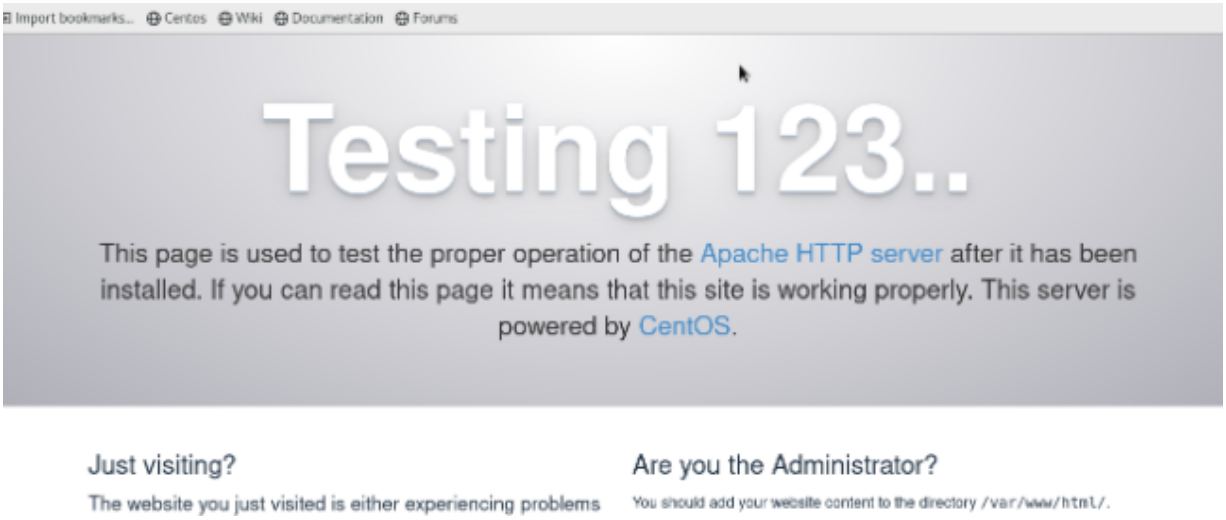
This is because in CentOS, installed packages' services are not run automatically. Thus, we need to create the module to run it automatically.

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.



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.

```
skipping: [192.168.56.107]

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

TASK [Gathering Facts] **********************************************************************************
ok: [192.168.56.106]
ok: [192.168.56.105]

TASK [install mariadb package (CentOS)] *****************************************************************
skipping: [192.168.56.106]
ok: [192.168.56.105]

TASK [Mariadb- Restarting/Enabling] *********************************************************************
changed: [192.168.56.106]
changed: [192.168.56.105]

TASK [install mariadb package (Ubuntu)] *****************************************************************
skipping: [192.168.56.105]
skipping: [192.168.56.106]

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

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

TASK [install samba package] ****************************************************************************
ok: [192.168.56.109]

PLAY RECAP **********************************************************************************************
192.168.56.105             : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignore
d=0
192.168.56.106             : ok=3    changed=1    unreachable=0    failed=0    skipped=4    rescued=0    ignore
d=0
192.168.56.107             : ok=2    changed=0    unreachable=0    failed=0    skipped=5    rescued=0    ignore
d=0
192.168.56.109             : ok=3    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignore
d=0
```



- Not only could Apache be enabled once more, but the Ansible playbook handled everything.

Reflections:

Answer the following:

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

- Task isolation is made possible by grouping remote servers, which speeds up processing and lowers costs. You can work more efficiently and quickly by allocating servers to particular groups.

2. What is the importance of tags in playbooks?
   - System administrators can run particular tasks in playbooks without running the entire playbook by using tags, which enable selective task execution. More control over the automation process is thus possible.

3. Why do think some services need to be managed automatically in playbooks?
   - It is imperative to automate the management of specific services in playbooks as this simplifies procedures that may be laborious and intricate to oversee. Automation reduces inconsistent behavior, gives system administrators more control, and guarantees that services are always current and function properly.