Name: Ilagan, Carlo Hideki D.	Date Performed: 9-30-2024
Course/Section: CpE31S2	Date Submitted: 10-2-2024
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"
```

Open ▾ | site.yml [Read-Only] ~/HOA6.1-chilagan | Save

```
1  ---
2
3  - hosts: all
4    become: true
5    tasks:
6
7    - name: install apache and ph for Ubuntu servers
8      apt:
9       name:
10        - apache2
11        - libapache2-mod-php
12      state: latest
13      update_cache: yes
14     when: ansible_disctribution == "Ubuntu"
15
16   - name: install apache and php for CENTOS servers
17     dnf:
18       name:
19         - httpd
20         - php
21       state: latest
22     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.

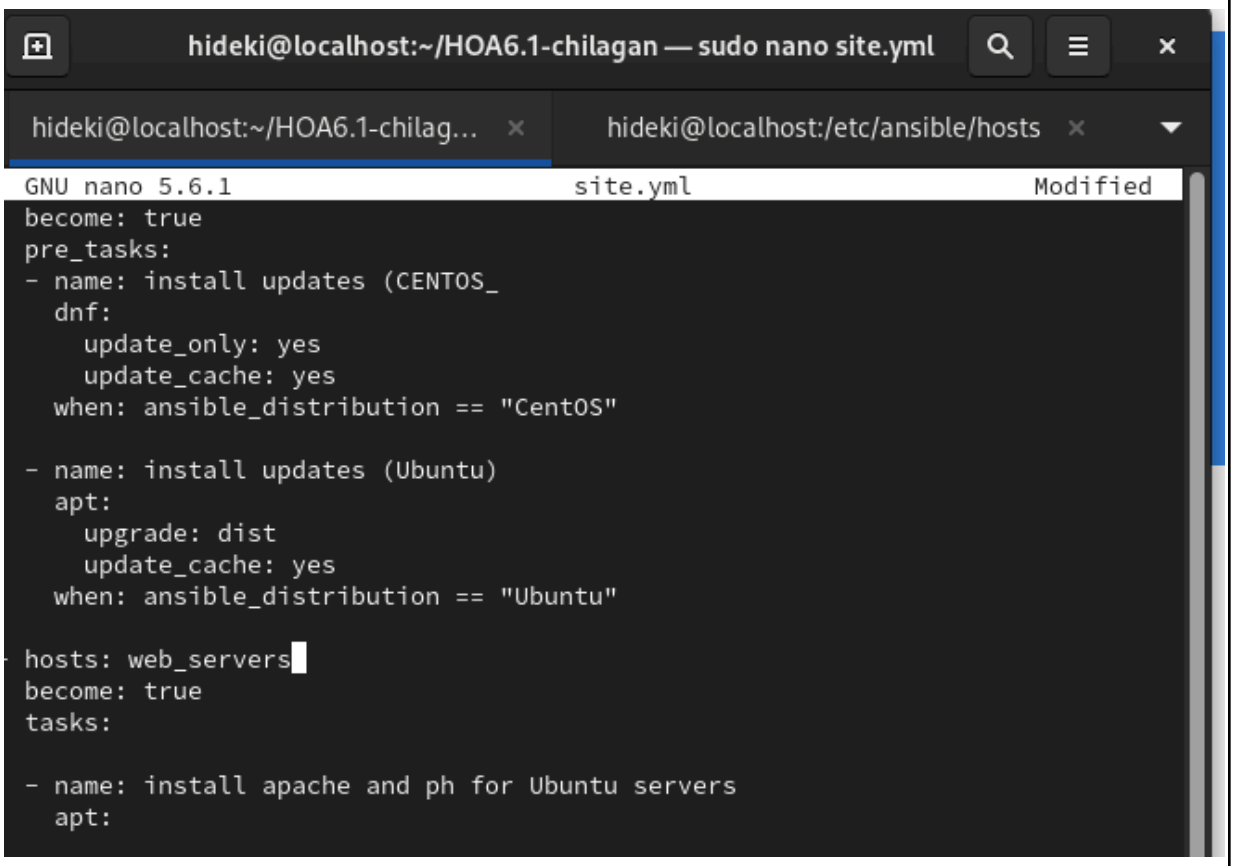
Right now, we have created groups in our inventory file and put each server in its own group. In other cases, you can have a server be a member of multiple groups, for example you have a test server that is also a web server.

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

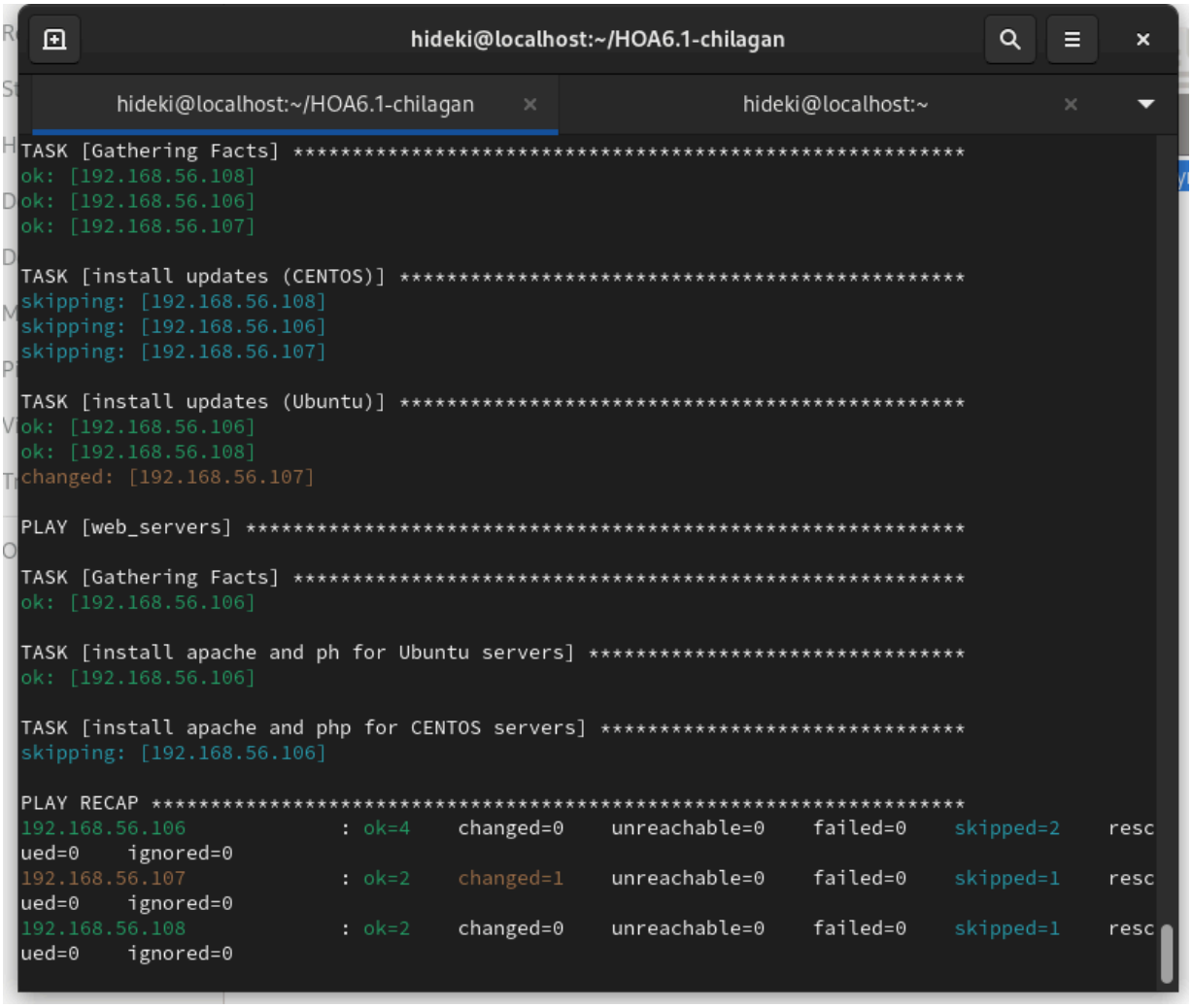
```
---

- 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.

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.

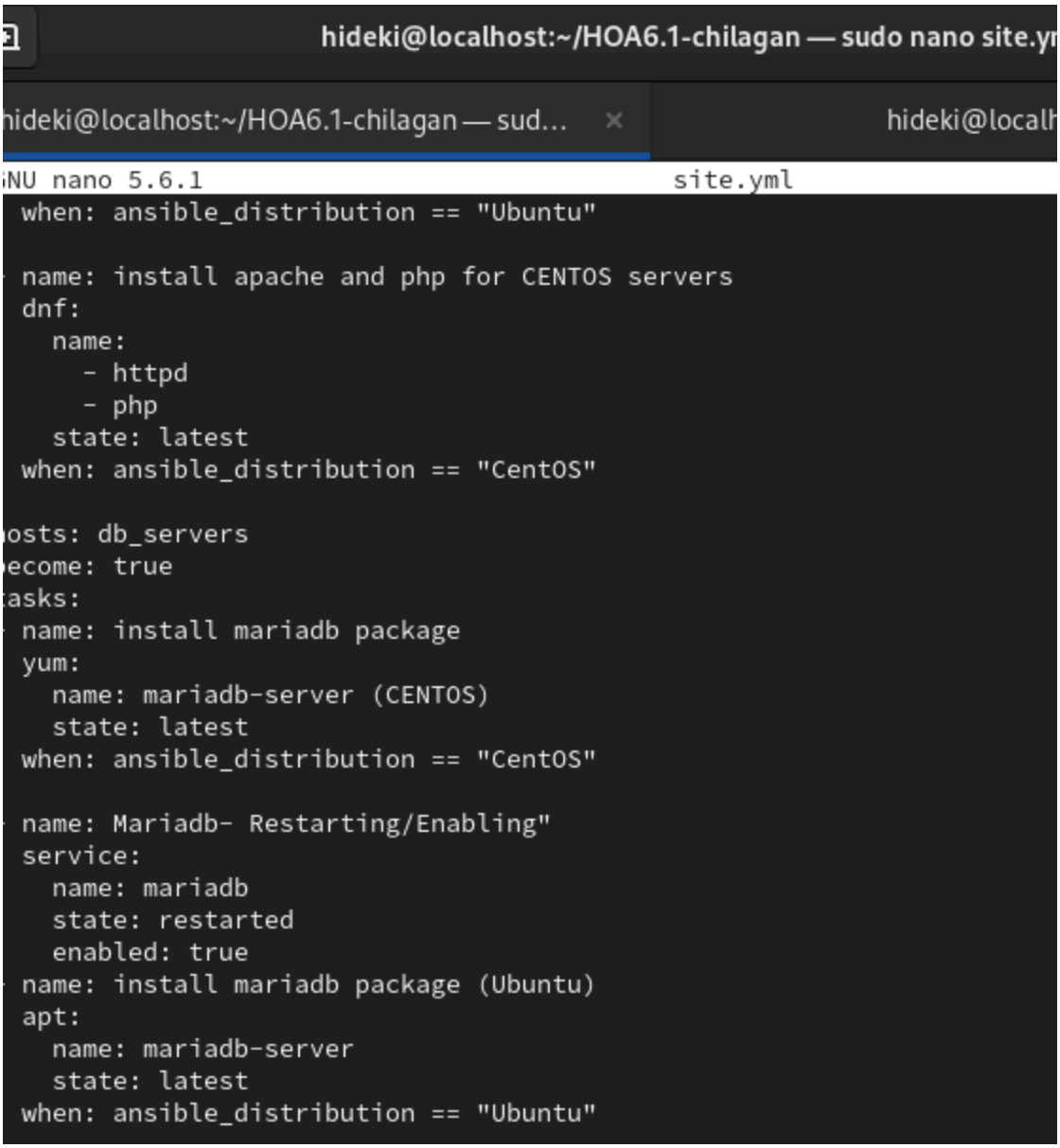


- The result showed that updates and install apache and ph for ubuntu servers ran since all of the ip addresses that were used inside the inventory were all ubuntu servers. All tasks for CENTOS are skipped since there is no detected CENTOS to be update and installed.

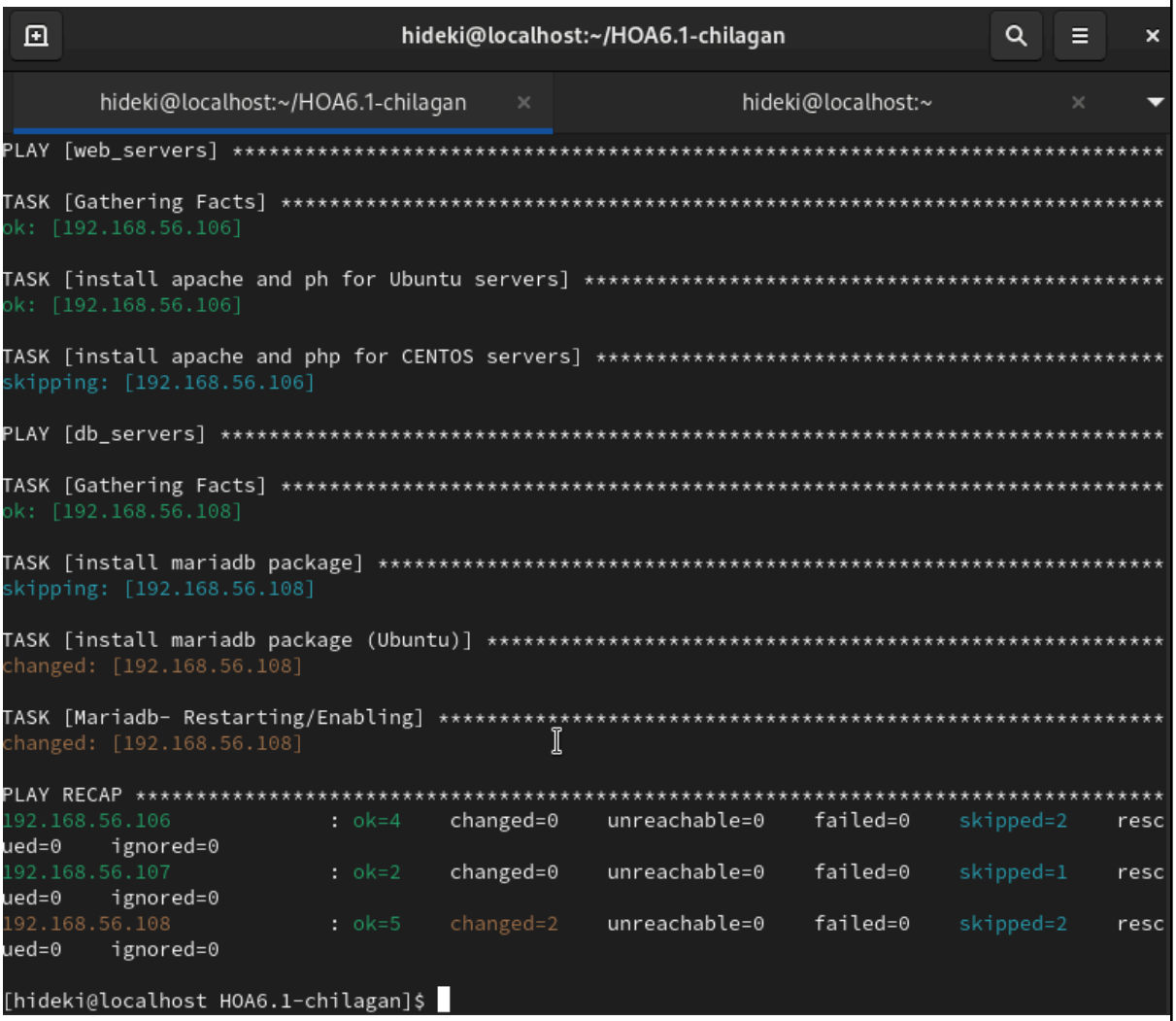
4. Let's try to edit again the *site.yml* file. This time, we are going to add plays targeting the other servers. This time we target the *db_servers* by adding it on the current *site.yml*. Below is an example: (Note add this at the end of the playbooks from task 1.3.

```
- 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.



Since I don't have mariadb in my 3rd Ubuntu server, it was installed through the use of playbook and was changed to enabled right after installing, for the instructions, I was having error since enabling was the first task rather than trying to install.

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.

```
[hideki@localhost ~]$ systemctl status mariadb
● mariadb.service - MariaDB 10.5 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: disabled)
     Active: active (running) since Tue 2024-10-01 23:52:41 PST; 38s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 10015 ExecStartPre=/usr/libexec/mariadb-check-socket (code=exited, status=0/SUCCESS)
    Process: 10119 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir mariadb.service (code=exited, st
    Process: 13270 ExecStartPost=/usr/libexec/mariadb-check-upgrade (code=exited, status=0/SUCCESS)
   Main PID: 13157 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 17 (limit: 10949)
     Memory: 79.2M
        CPU: 503ms
     CGroup: /system.slice/mariadb.service
             └─13157 /usr/libexec/mariadbd --basedir=/usr

Oct 01 23:52:41 localhost.localdomain mariadb-prepare-db-dir[11997]: The second is mysql@localhost,
Oct 01 23:52:41 localhost.localdomain mariadb-prepare-db-dir[11997]: you need to be the system 'mysq
```

- Since i only have the CentOS ip in the db_servers, the maria db package was only installed inside the CentOS servers

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.

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

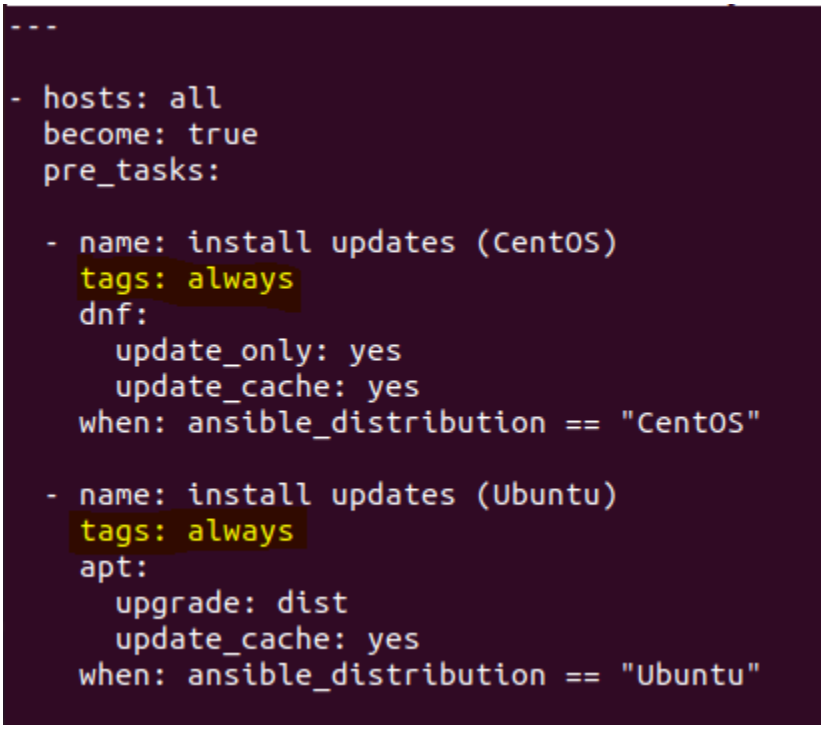
- Samba package was installed in the remote server that was in the group of file_servers which is the 192.168.56.103

The testing of the *file_servers* is beyond the scope of this activity, and as well as our topics and objectives. However, in this activity we were able to show that we can target hosts or servers using grouping in ansible playbooks.

Task 2: Using Tags in running playbooks

In this task, our goal is to add metadata to our plays so that we can only run the plays that we want to run, and not all the plays in our playbook.

1. Edit the *site.yml* file. Add tags to the playbook. After the name, we can place the tags: *name_of_tag*. This is an arbitrary command, which means you can use any name for a tag.



```
---

- hosts: all
  become: true
  pre_tasks:

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

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

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

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

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

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

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

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

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

- hosts: file_servers
  become: true
  tasks:

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

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

```
TASK [install updates (Ubuntu)] ************************************************
ok: [192.168.56.103]
ok: [192.168.56.102]

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

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

TASK [install apache and ph for Ubuntu servers] ********************************
ok: [192.168.56.102]

TASK [install apache and php for CENTOS servers] *******************************
skipping: [192.168.56.102]

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

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

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

- Same output as I was running the ansible playbook command

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

```
hideki@workstation:~/HOA6.1-chilagan$ 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]
```

- Showing list of tags available

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

```
TASK [install updates (Ubuntu)] ************************************************
ok: [192.168.56.107]
ok: [192.168.56.106]

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

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

TASK [install apache and php for CENTOS servers] *******************************
skipping: [192.168.56.106]

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

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.107]
        to retry, use: --limit @/home/hideki/HOA6.1-chilagan/site.retry

PLAY RECAP *********************************************************************
192.168.56.106             : ok=3    changed=0    unreachable=0    failed=0
```

- Task to install samba did not show since there is no centos tag on it.

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

```
TASK [Gathering Facts] *********************************************************
ok: [192.168.56.111]

TASK [install mariadb package in CENTOS] ***************************************
ok: [192.168.56.111]

TASK [install mariadb package in Ubuntu] ***************************************
skipping: [192.168.56.111]

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

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

PLAY RECAP *********************************************************************
192.168.56.106             : ok=3    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.107             : ok=3    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.111             : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
```

- Ony tasks with db tags ran when this command was used

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

- Ony tasks with apache tags ran when this command was used

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

- Ony tasks with apache and db tags ran when this command was used

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"
```

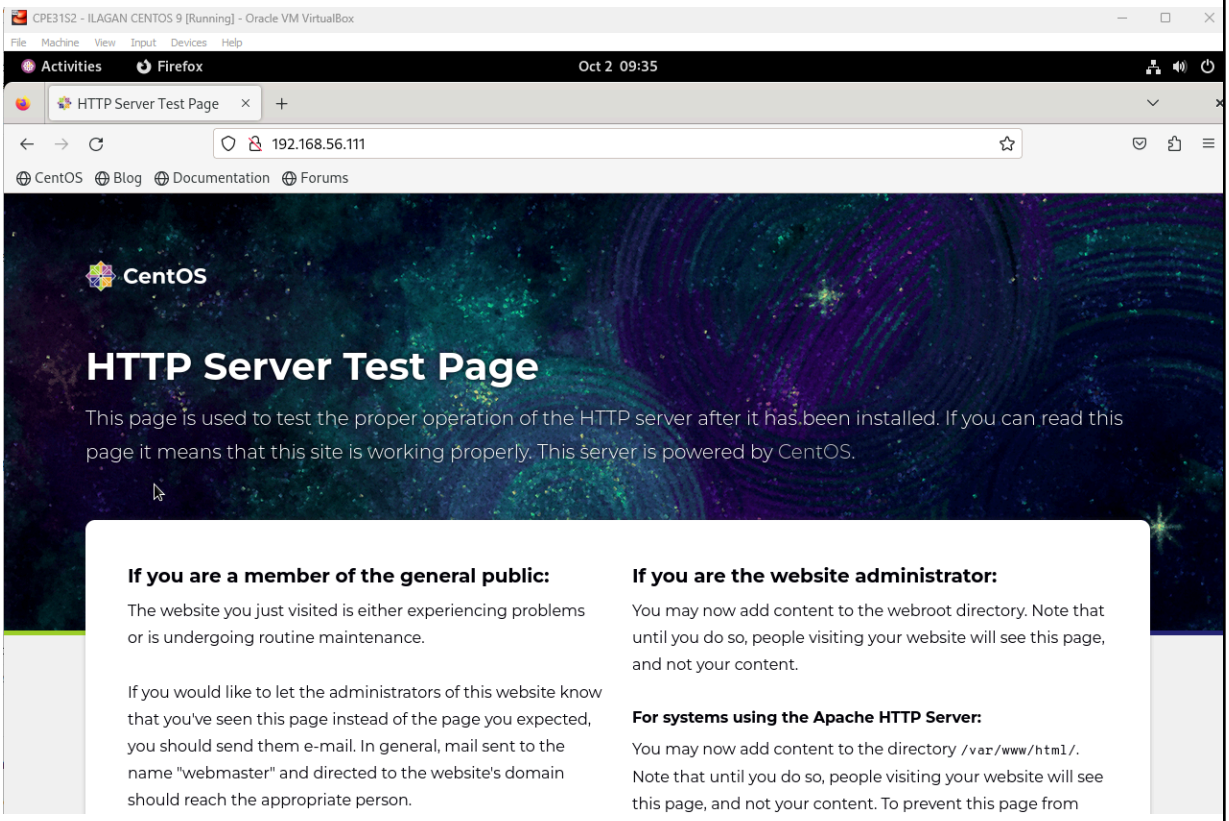
You would also notice from our previous activity that we already created a module that runs a service.

```
- 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.
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.



- The httpd ran properly after it was re-enable through the playbook. Screenshot was provided above that the command worked

Reflections:

Answer the following:

1. What is the importance of putting our remote servers into groups?
   - In order to specify what are the roles for each servers that are currently being administered
2. What is the importance of tags in playbooks?
   - Tags in playbooks provide an efficient way to run the yml, for example, if the administrator only wants to run tasks with tags CentOS. By putting –tags CentOS, this will automatically process playbook commands with only CentOS tags.
3. Why do think some services need to be managed automatically in playbooks?
   - Some services needed to be managed automatically in order for it to start without the need to manually process it by the administrator.