Name: Jaira Biane Maculada	**Date Performed:** 09/28/23
Course/Section:CPE232/ CPE31S6	**Date Submitted:** 09/28/23
Instructor: Dr. Jonathan V. Taylar	**Semester and SY:** 1st Sem(2023-2024)

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"
```

```
le  Edit  View  Search  Terminal  Help
 GNU nano 2.9.3                         site.yml

--

hosts: all
become: true
tasks:

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

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

                                [ Wrote 22 lines ]
```

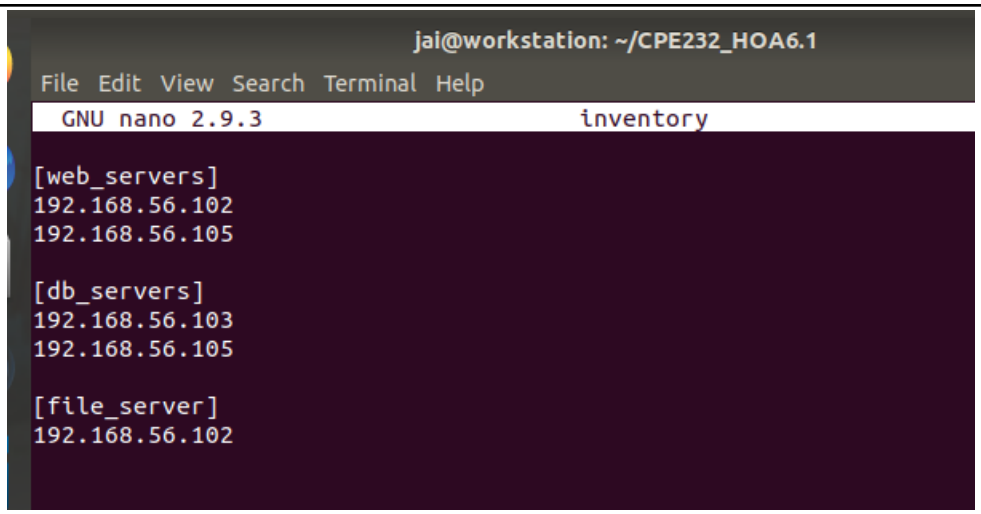
```
jai@workstation:~/CPE232_HOA6.1$ ansible-playbook --ask-become-pass
site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************
************
ok: [192.168.56.102]
ok: [192.168.56.103]
ok: [192.168.56.105]
untu Software
TASK [install apache and php for Ubuntu servers] *******************
************
skipping: [192.168.56.105]
ok: [192.168.56.103]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] *******************
************
skipping: [192.168.56.102]
skipping: [192.168.56.103]
changed: [192.168.56.105]
```

```
TASK [install apache and php for CentOS servers] *******************
************
skipping: [192.168.56.102]
skipping: [192.168.56.103]
changed: [192.168.56.105]

PLAY RECAP *********************************************************
************
192.168.56.102             : ok=2    changed=0    unreachable=0    f
ailed=0    skipped=1    rescued=0    ignored=0
192.168.56.103             : ok=2    changed=0    unreachable=0    f
ailed=0    skipped=1    rescued=0    ignored=0
192.168.56.105             : ok=2    changed=1    unreachable=0    f
ailed=0    skipped=1    rescued=0    ignored=0
```

2. Edit the inventory file. Remove the variables we put in our last activity and group according to the image shown below:

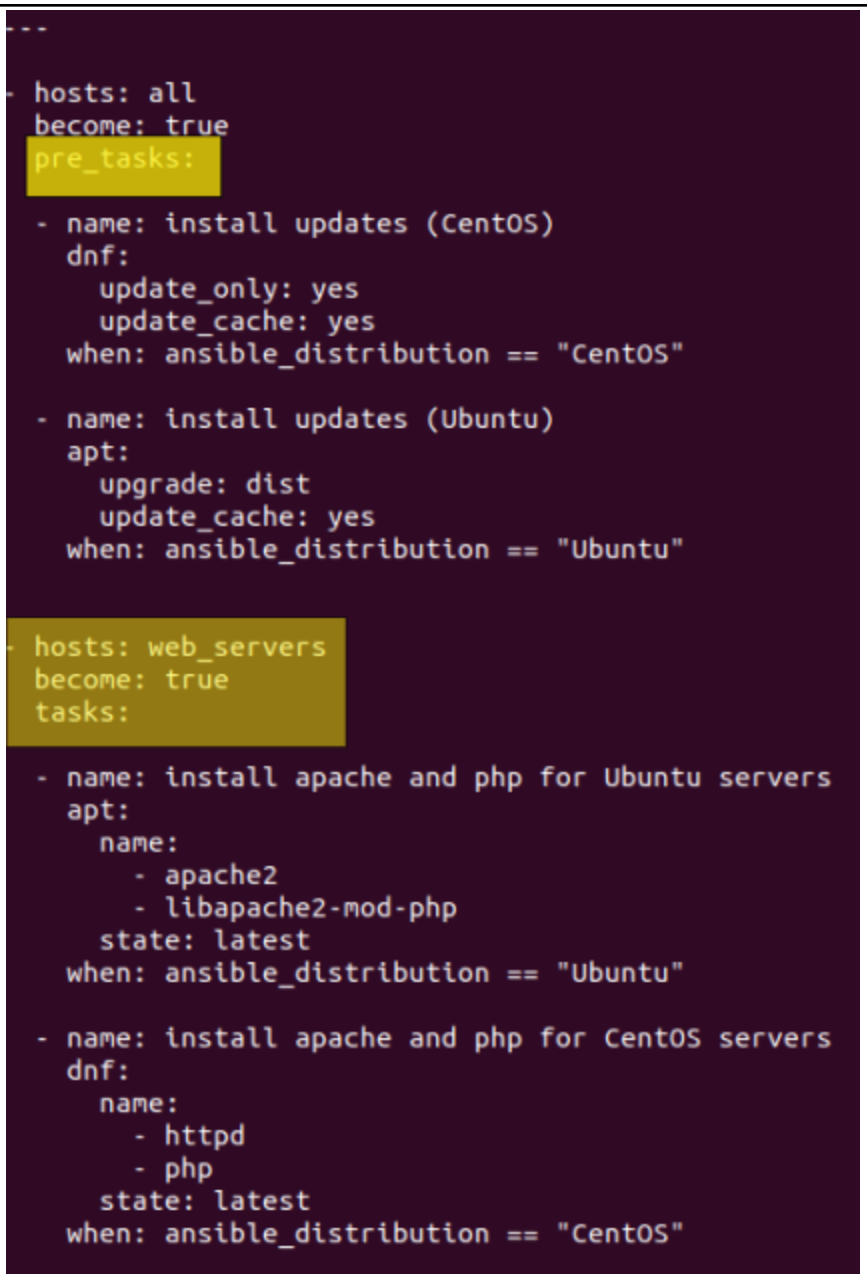
```
[web_servers]
192.168.56.120
192.168.56.121

[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 2.9.3                    site.yml

---

- hosts: all
  become: true
  pre_tasks:

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

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

```
- hosts: web_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"
```

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.

```
jai@workstation:~/CPE232_HOA6.1$ ansible-playbook --ask-become-pass
site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************
************
ok: [192.168.56.102]
ok: [192.168.56.103]
ok: [192.168.56.105]

TASK [install updates (CentOS)] ************************************
************
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.105]

TASK [install updates (Ubuntu)] ************************************
************
skipping: [192.168.56.105]
ok: [192.168.56.102]
ok: [192.168.56.103]
```

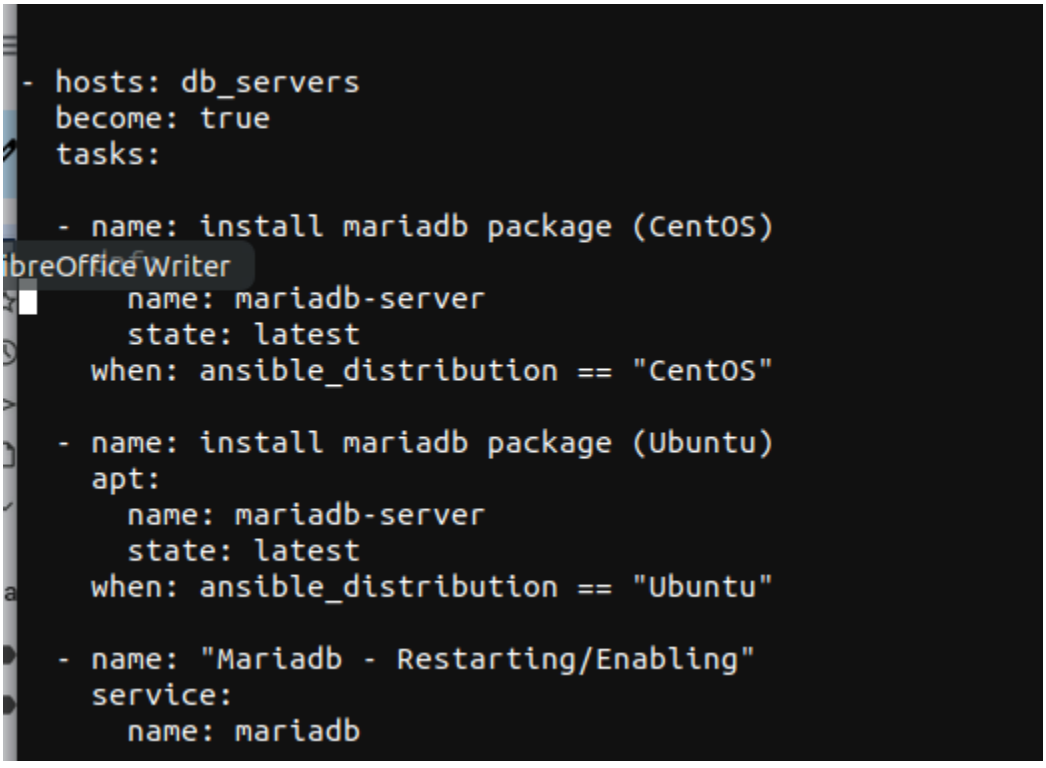
```
PLAY [web_servers] *************************************************
************

TASK [Gathering Facts] *********************************************
************
ok: [192.168.56.102]
ok: [192.168.56.105]

TASK [install apache and php for Ubuntu servers] *******************
************
skipping: [192.168.56.105]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] *******************
************
skipping: [192.168.56.102]
ok: [192.168.56.105]

PLAY RECAP *********************************************************
************
192.168.56.102             : ok=4    changed=0    unreachable=0    f
ailed=0    skipped=2    rescued=0    ignored=0
192.168.56.103             : ok=2    changed=0    unreachable=0    f
ailed=0    skipped=1    rescued=0    ignored=0
192.168.56.105             : ok=4    changed=0    unreachable=0    f
```

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

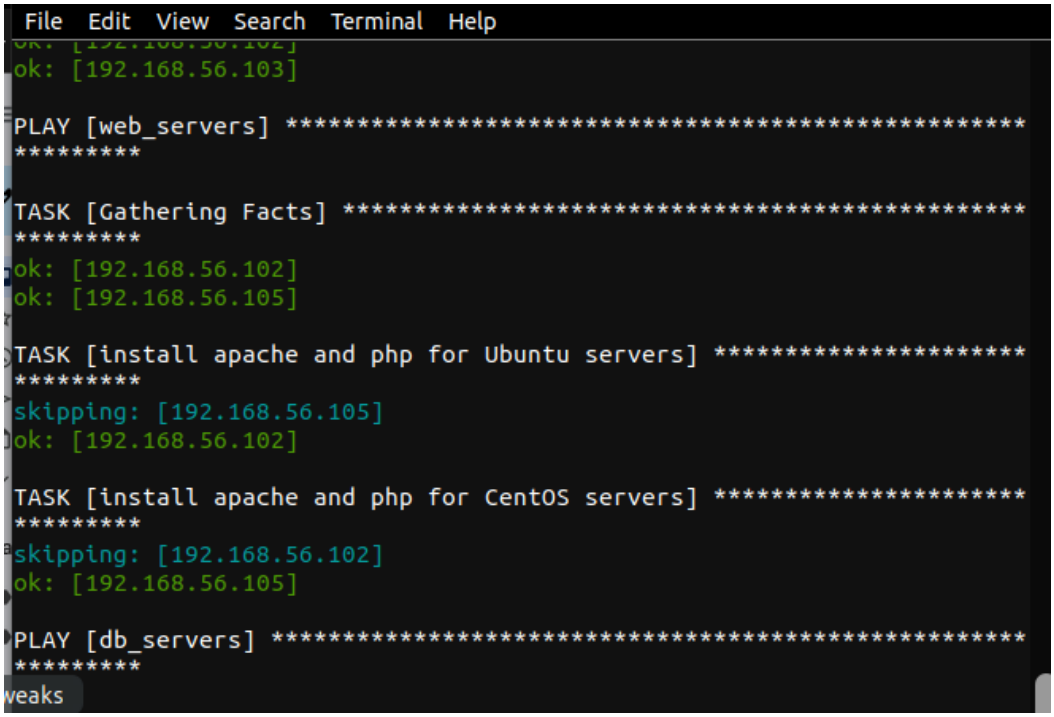
the current *site.yml*. Below is an example: (Note add this at the end of the playbooks from task 1.3.

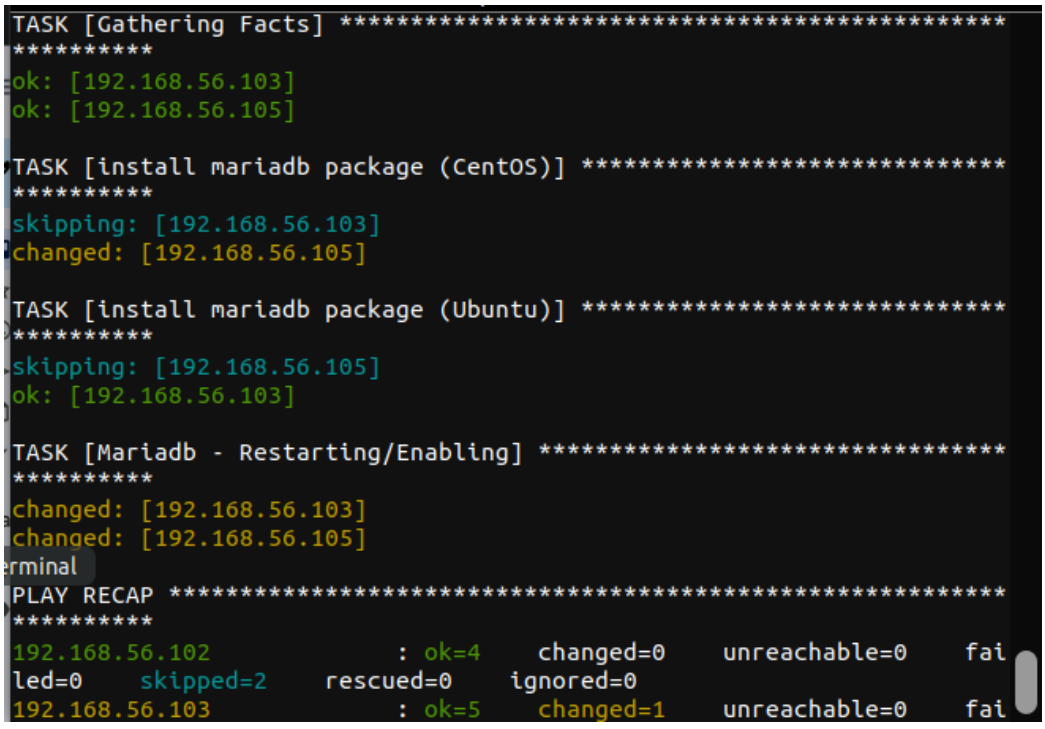
```
- hosts: db_servers
  become: true
  tasks:

  - 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. **It is successful.**





```
PLAY RECAP *********************************************************
**********
192.168.56.102             : ok=4    changed=0    unreachable=0    fai
led=0    skipped=2    rescued=0    ignored=0
192.168.56.103             : ok=5    changed=1    unreachable=0    fai
led=0    skipped=2    rescued=0    ignored=0
192.168.56.105             : ok=7    changed=2    unreachable=0    fai
led=0    skipped=3    rescued=0    ignored=0
```

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. **It is successful.**

```
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: di
abled)
   Active: active (running) since Thu 2023-09-28 05:56:11 EDT; 15min ago
  Process: 8475 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited, s
atus=0/SUCCESS)
  Process: 8387 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, stat
s=0/SUCCESS)
 Main PID: 8474 (mysqld_safe)
    Tasks: 20
   CGroup: /system.slice/mariadb.service
           ├─8474 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
           └─8639 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --plug..

Sep 28 05:56:09 localhost.localdomain mariadb-prepare-db-dir[8387]: MySQL manual for..
Sep 28 05:56:09 localhost.localdomain mariadb-prepare-db-dir[8387]: Please report an..
Sep 28 05:56:09 localhost.localdomain mariadb-prepare-db-dir[8387]: The latest infor..
Sep 28 05:56:09 localhost.localdomain mariadb-prepare-db-dir[8387]: You can find add..
Sep 28 05:56:09 localhost.localdomain mariadb-prepare-db-dir[8387]: http://dev.mysql..
Sep 28 05:56:09 localhost.localdomain mariadb-prepare-db-dir[8387]: Consider joining..
Sep 28 05:56:09 localhost.localdomain mariadb-prepare-db-dir[8387]: https://mariadb...
Sep 28 05:56:09 localhost.localdomain mysqld_safe[8474]: 230928 05:56:09 mysqld_safe..
Sep 28 05:56:09 localhost.localdomain mysqld_safe[8474]: 230928 05:56:09 mysqld_safe..
Sep 28 05:56:11 localhost.localdomain systemd[1]: Started MariaDB database server.
Hint: Some lines were ellipsized, use -l to show in full.
```

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.

```
File  Edit  View  Search  Terminal  Help
 GNU nano 2.9.3                    site.yml

      enabled: true


- hosts: file_servers
  become: true
  tasks:

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

Run the *site.yml* file and describe the result. **It is successful**

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

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

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

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

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.

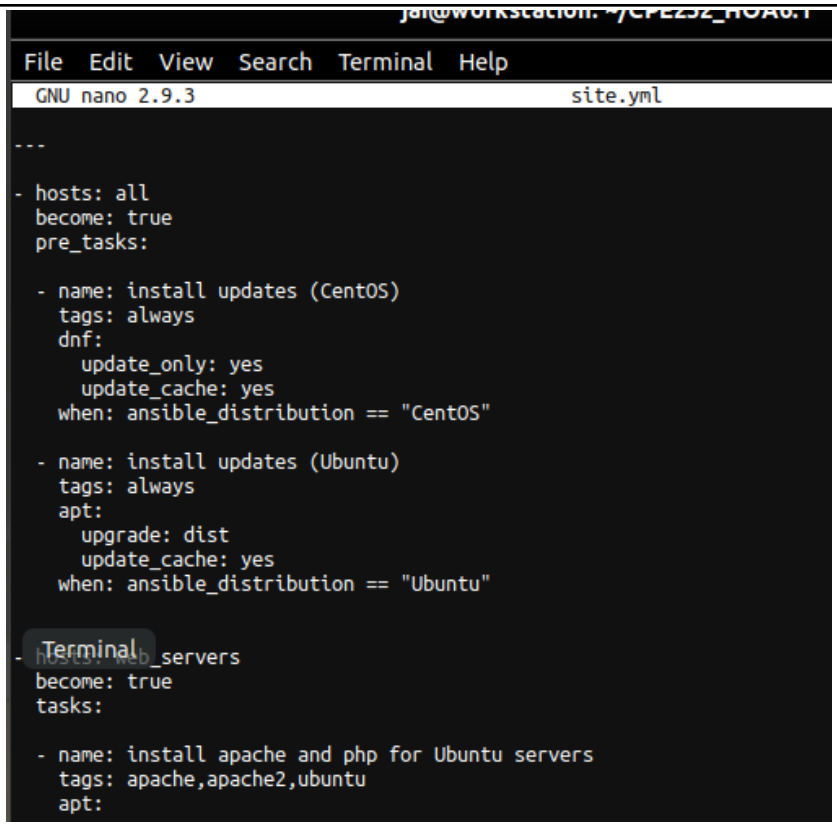
1. Edit the *site.yml* file. Add tags to the playbook. After the name, we can place the tags: *name_of_tag*. This is an arbitrary command, which means you can use any name for a tag.

```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"
```

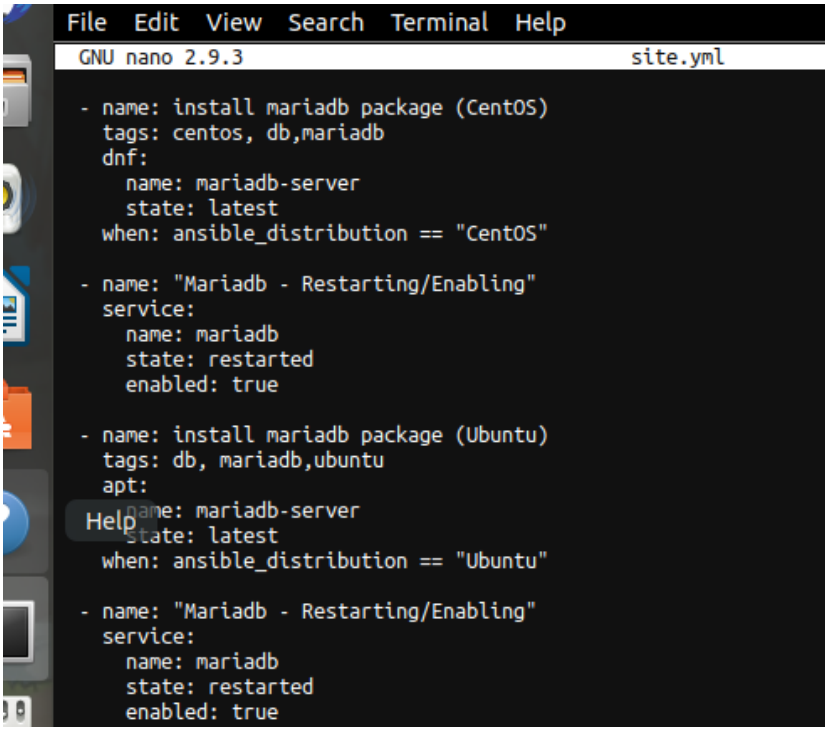
```yaml
- hosts: web_servers
  become: true
  tasks:

  - 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"
```

```
- 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.


File Edit View Search Terminal Help
GNU nano 2.9.3 site.yml

- 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"
Terminal
become: true
tasks:
- name: install apache and php for Ubuntu servers
tags: apache,apache2,ubuntu
apt:


File Edit View Search Terminal Help
GNU nano 2.9.3 site.yml
- 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 package (Ubuntu)
tags: db, mariadb,ubuntu
apt:
Help
when: ansible_distribution == "Ubuntu"
- name: "Mariadb - Restarting/Enabling"
service:
name: mariadb
state: restarted
enabled: true

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

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

Terminal

Run the *site.yml* file and describe the result. It run successfully.

```
jai@workstation:~/CPE232_HOA6.1$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.102]
ok: [192.168.56.103]
ok: [192.168.56.105]

TASK [install updates (CentOS)] ************************************************
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.105]

TASK [install updates (Ubuntu)] ************************************************
skipping: [192.168.56.105]
ok: [192.168.56.103]
ok: [192.168.56.102]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.102]
ok: [192.168.56.105]

TASK [install apache and php for Ubuntu servers] *******************************
skipping: [192.168.56.105]
ok: [192.168.56.102]

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

Help

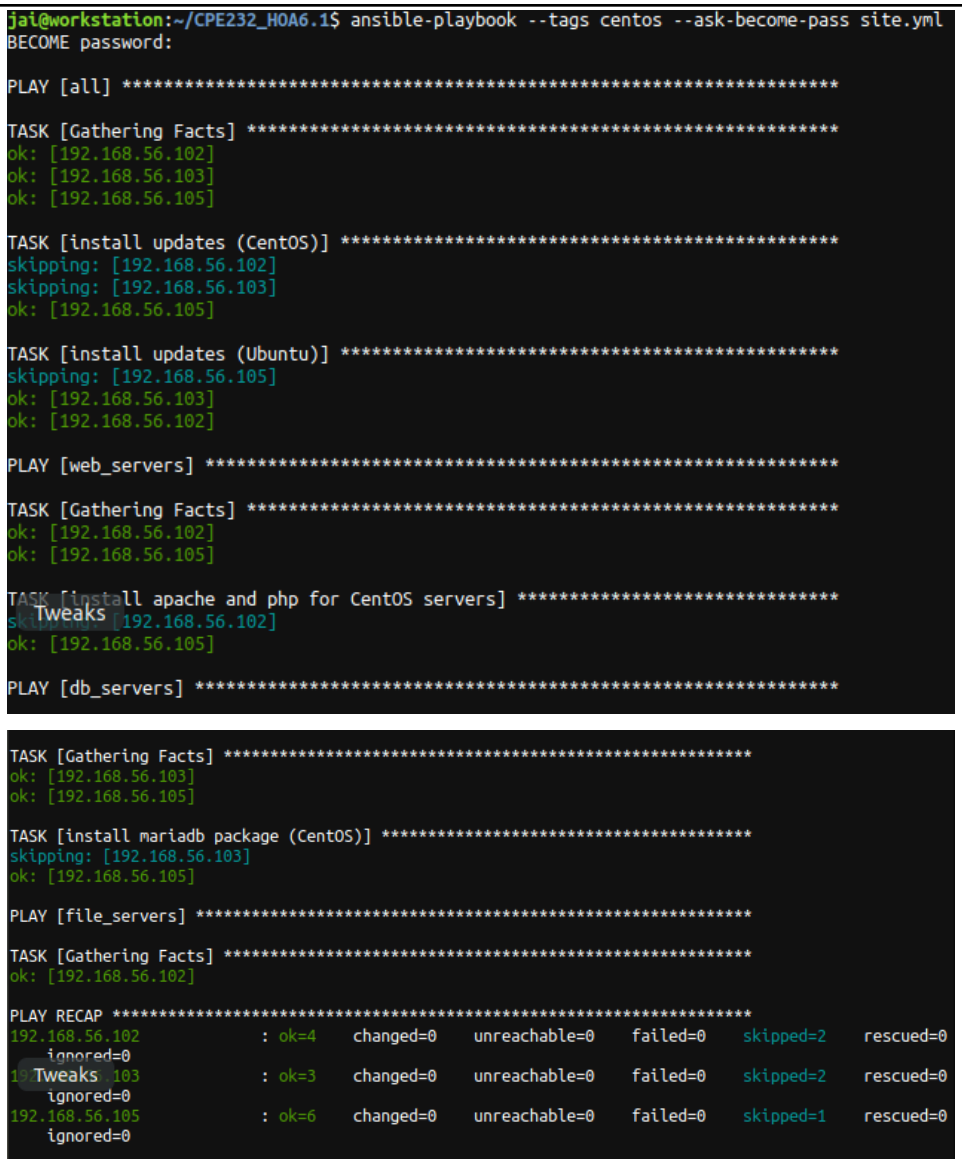
```
TASK [Gathering Facts] *********************************************************
ok: [192.168.56.103]
ok: [192.168.56.105]

TASK [install mariadb package (CentOS)] ****************************************
skipping: [192.168.56.103]
ok: [192.168.56.105]

TASK [install mariadb package (Ubuntu)] ****************************************
skipping: [192.168.56.105]
ok: [192.168.56.103]

TASK [Mariadb - Restarting/Enabling] *******************************************
changed: [192.168.56.105]
changed: [192.168.56.103]

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

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

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

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

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

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

```
jai@workstation:~/CPE232_HOA6.1$ 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]
```

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



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

```
jai@workstation:~/CPE232_HOA6.1$ ansible-playbook --tags db --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.102]
ok: [192.168.56.103]
ok: [192.168.56.105]

TASK [install updates (CentOS)] ************************************************
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.105]

TASK [install updates (Ubuntu)] ************************************************
skipping: [192.168.56.105]
ok: [192.168.56.103]
ok: [192.168.56.102]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.102]
ok: [192.168.56.105]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.103]
ok: [192.168.56.105]

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

```
ok: [192.168.56.103]

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

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

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

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

```
jai@workstation:~/CPE232_HOA6.1$ ansible-playbook --tags apache --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.102]
ok: [192.168.56.103]
ok: [192.168.56.105]

TASK [install updates (CentOS)] ************************************************************
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.105]

TASK [install updates (Ubuntu)] ************************************************************
skipping: [192.168.56.105]
ok: [192.168.56.103]
ok: [192.168.56.102]

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.102]
ok: [192.168.56.105]

TASK [install apache and php for Ubuntu servers] *******************************************
skipping: [192.168.56.105]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] *******************************************
skipping: [192.168.56.102]
ok: [192.168.56.105]
```

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.103]
ok: [192.168.56.105]

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

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

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

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

```
jai@workstation:~/CPE232_HOA6.1$ ansible-playbook --tags "apache,db" --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.102]
ok: [192.168.56.103]
ok: [192.168.56.105]

TASK [install updates (CentOS)] ************************************************************************
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.105]

TASK [install updates (Ubuntu)] ************************************************************************
skipping: [192.168.56.105]
ok: [192.168.56.103]
ok: [192.168.56.102]

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.102]
ok: [192.168.56.105]

TASK [install apache and php for Ubuntu servers] *******************************************************
skipping: [192.168.56.105]
ok: [192.168.56.102]

TASK [                                nd php for CentOS servers] ***************************************
skipping: [192.168.56.102]
ok: [192.168.56.105]
```

Show Applications

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.103]
ok: [192.168.56.105]

TASK [install mariadb package (CentOS)] ****************************************************************
skipping: [192.168.56.103]
ok: [192.168.56.105]

TASK [install mariadb package (Ubuntu)] ****************************************************************
skipping: [192.168.56.105]
ok: [192.168.56.103]

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

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

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

Task 3: Managing Services

1. Edit the file site.yml and add a play that will automatically start the httpd on CentOS server.

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

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

Figure 3.1.1

Make sure to save the file and exit.

```
- 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"

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

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

TASK [Gathering Facts] ********************************************
*
ok: [192.168.56.102]
ok: [192.168.56.105]

TASK [install apache and php for Ubuntu servers] ******************
*
skipping: [192.168.56.105]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] ******************
*
skipping: [192.168.56.102]
ok: [192.168.56.105]

TASK [start httpd (CentOS)] ***************************************
*
skipping: [192.168.56.102]
changed: [192.168.56.105]
```

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

```
- 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 package (Ubuntu)
    tags: db, mariadb,ubuntu
    apt:
      name: mariadb-server
```

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

TASK [Gathering Facts] **********************************************************************
ok: [192.168.56.103]
ok: [192.168.56.105]

TASK [install mariadb package (CentOS)] *****************************************************
skipping: [192.168.56.103]
ok: [192.168.56.105]

TASK [Mariadb - Restarting/Enabling] ********************************************************
changed: [192.168.56.103]
changed: [192.168.56.105]

TASK [install mariadb package (Ubuntu)] *****************************************************
skipping: [192.168.56.105]
ok: [192.168.56.103]

TASK [Mariadb - Restarting/Enabling] ********************************************************
changed: [192.168.56.103]
changed: [192.168.56.105]

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

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

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

PLAY RECAP **********************************************************************************
192.168.56.102             : ok=6    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    igno
192.168.56.103             : ok=6    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    igno
192.168.56.105             : ok=9    changed=2    unreachable=0    failed=0    skipped=3    rescued=0    igno
```

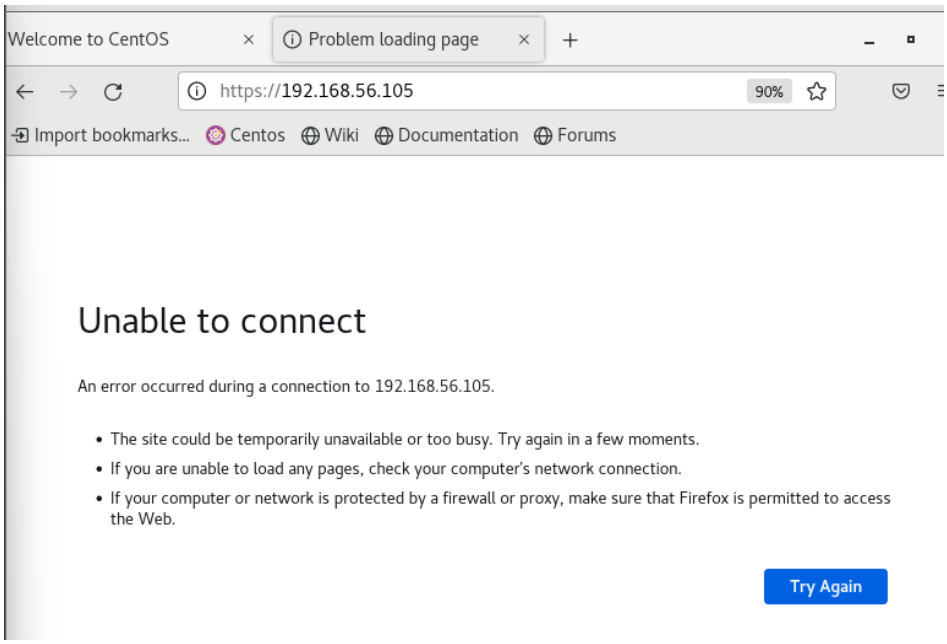
Tweaks

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

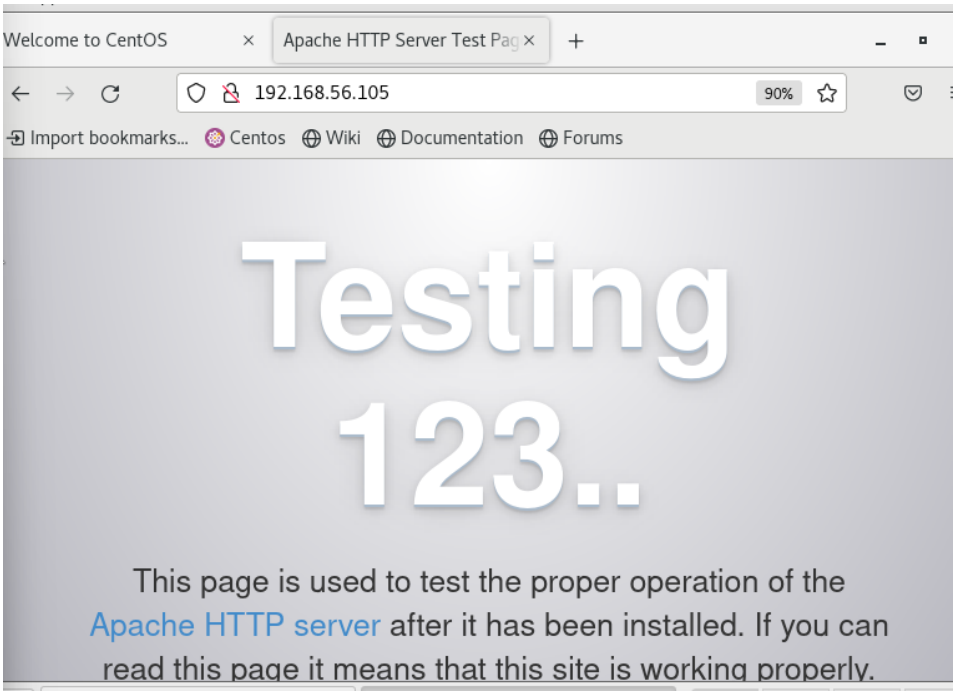
2. To test it, before you run the saved playbook, go to the CentOS server and stop the currently running httpd using the command *sudo systemctl stop httpd*. When prompted, enter the sudo password. After that, open the browser and

enter the CentOS server's IP address. You should not be getting a display because we stopped the httpd service already.

```
jai@localhost ~]$ sudo systemctl stop httpd
sudo] password for jai:
```



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.

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

```
TASK [start httpd (CentOS)] ****************************************************
*
skipping: [192.168.56.102]
changed: [192.168.56.105]
```

```
PLAY RECAP *********************************************************************
* Terminal
192.168.56.102             : ok=6    changed=0    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
192.168.56.103             : ok=6    changed=2    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.105             : ok=9    changed=3    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
```

link: https://github.com/jaebieeee/CPE232_HOA6.1.git

Reflections:

Answer the following:

1. What is the importance of putting our remote servers into groups?
   For effective management, it's essential to group distant servers. It improves organization in the first place, making it simpler to find and administer particular servers. In addition, it standardizes access control and monitoring, streamlining security measures. The final benefit is that it makes resource allocation simpler by guaranteeing that servers with comparable functions are given the right resources, improving overall performance and dependability.

2. What is the importance of tags in playbooks?
   For task execution to be selective, tags in Linux playbooks are crucial. They give you the option to execute only certain sections of a playbook rather than the complete script, which helps you save time and money. By classifying tasks, tags improve the readability and upkeep of playbooks and make it easier to update

setups and debug issues. Effective automation and system management depend on this organization and adaptability.

3. Why do think some services need to be managed automatically in playbooks? Playbooks can automate Apache installation, configuration, and upkeep, ensuring that websites function properly. Apache is a web server. Also, automation can manage MySQL setup, user permissions, and database backups, preserving data integrity. Using playbooks to handle SSH setups improves server security by automatically enforcing secure access controls and key management.

CONCLUSION:

After performing this Hands-On Activity, I was able to gain knowledge aboutg tags. To be very honest, it is quite quite instructive, to say the least. With control over what is processed on distant machines, these solutions have proven to be essential for easing server management operations. Streamlined resource allocation, saved time, and made troubleshooting simpler by deliberately identifying particular jobs. The automation prowess of playbooks is evident when setting web servers, databases, or securing SSH access. My newly acquired skill set enables me to ensure that they operate securely and efficiently regardless of the Linux distribution being used.