Name: Bueno, Joshua C.	**Date Performed: 10/03/25**
Course/Section: CPE32S2	**Date Submitted: 10/03/25**
Instructor: Engr. Robin Valenzuela	**Semester and SY: 1st sem 2025-2026**

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:

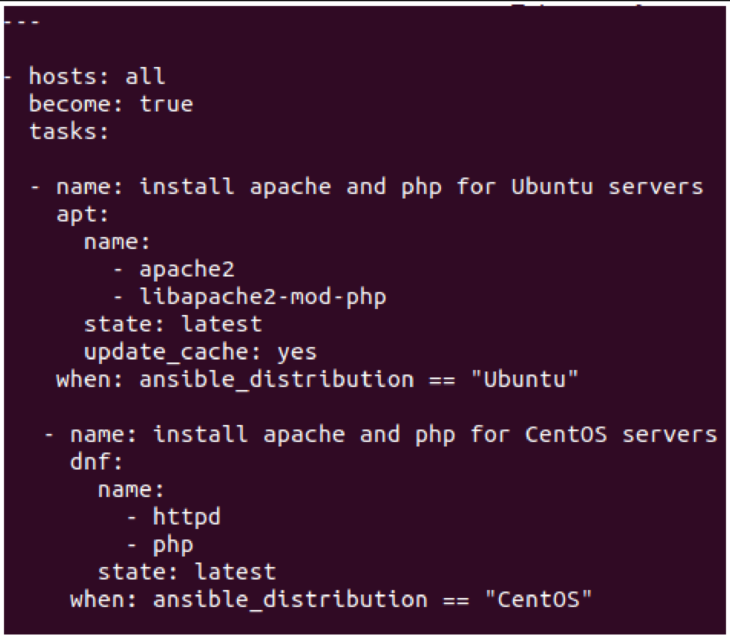
In this activity, we try to individualize hosts. For example, we don't want apache on all our servers, or maybe only one of our servers is a web server, or maybe we have different servers like database or file servers running different things on different categories of servers and that is what we are 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.

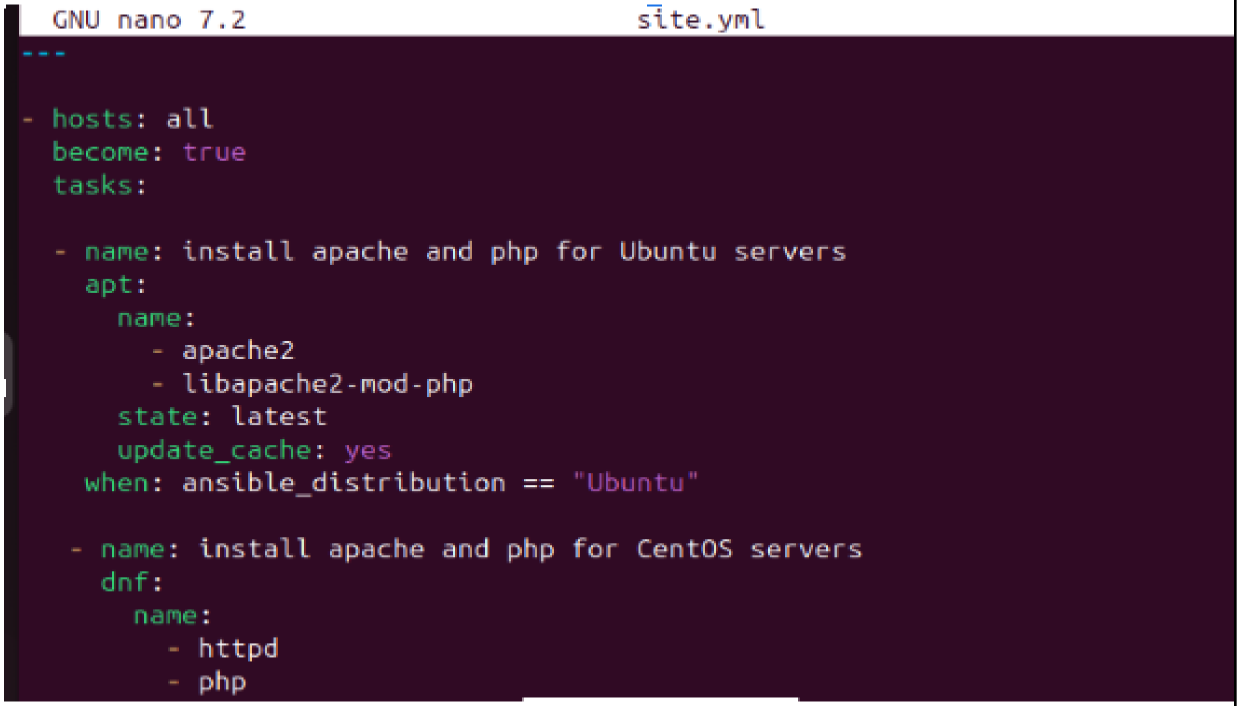
Requirement:
In this activity, you will need to create another Ubuntu VM and name it Server 3. Likewise, you need to activate the second adapter to a host-only adapter after the installations. Take note of the IP address of the Server 3. Make sure to use the command *ssh-copy-id* to copy the public key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

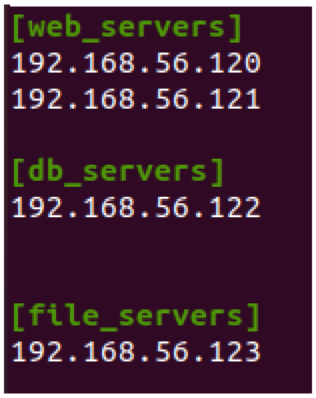


```
---

- 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"
```

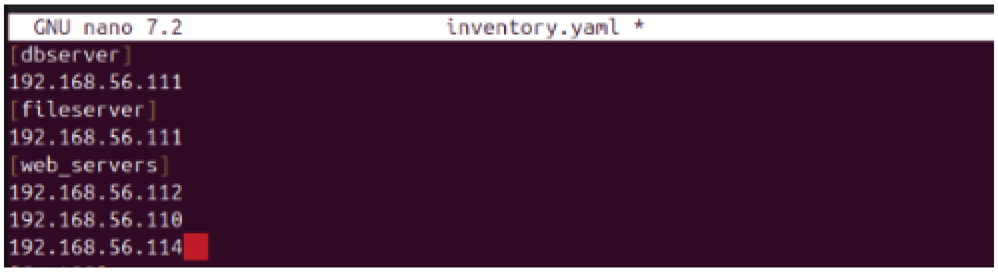


```
  GNU nano 7.2                                 site.yml
---

- 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
```

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
```

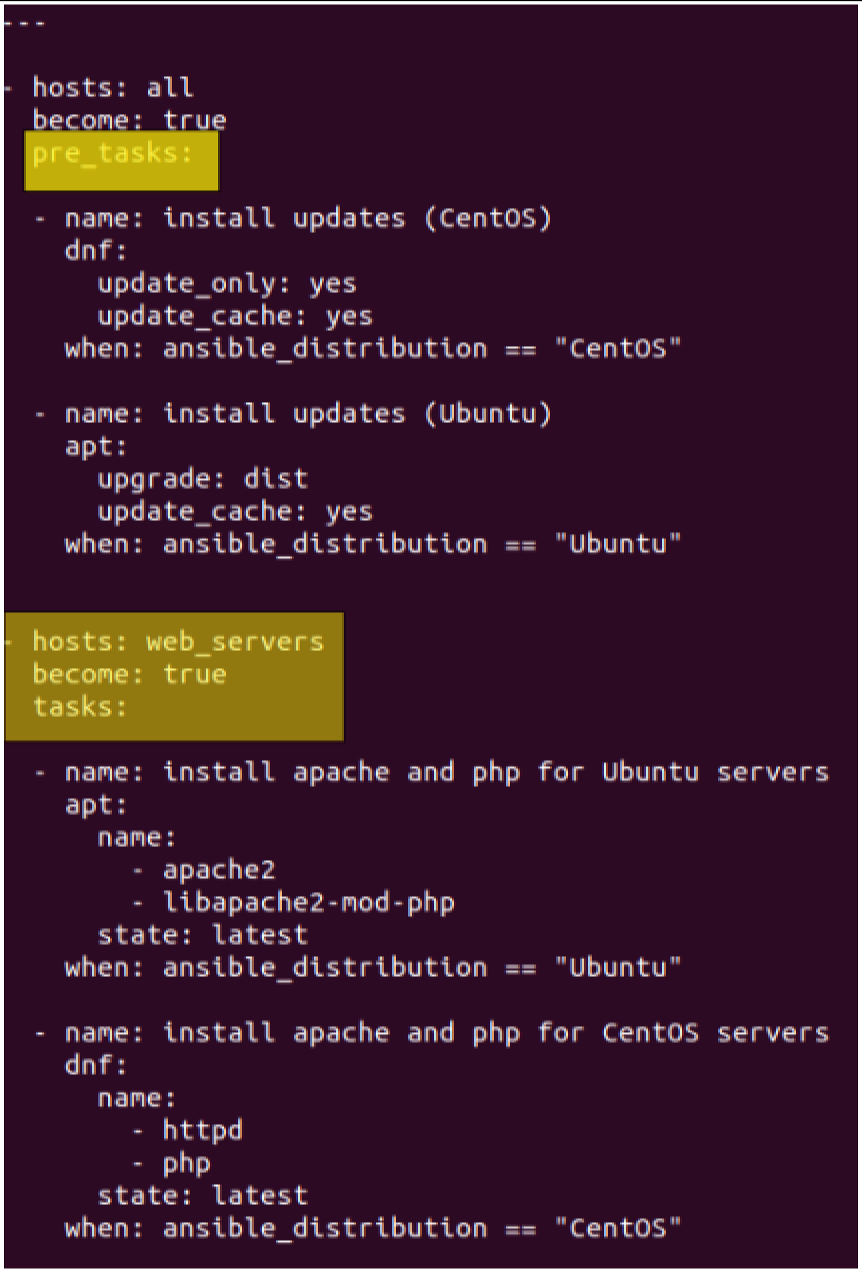


```
  GNU nano 7.2                        inventory.yaml *
[dbserver]
192.168.56.111
[fileserver]
192.168.56.111
[web_servers]
192.168.56.112
192.168.56.110
192.168.56.114
```

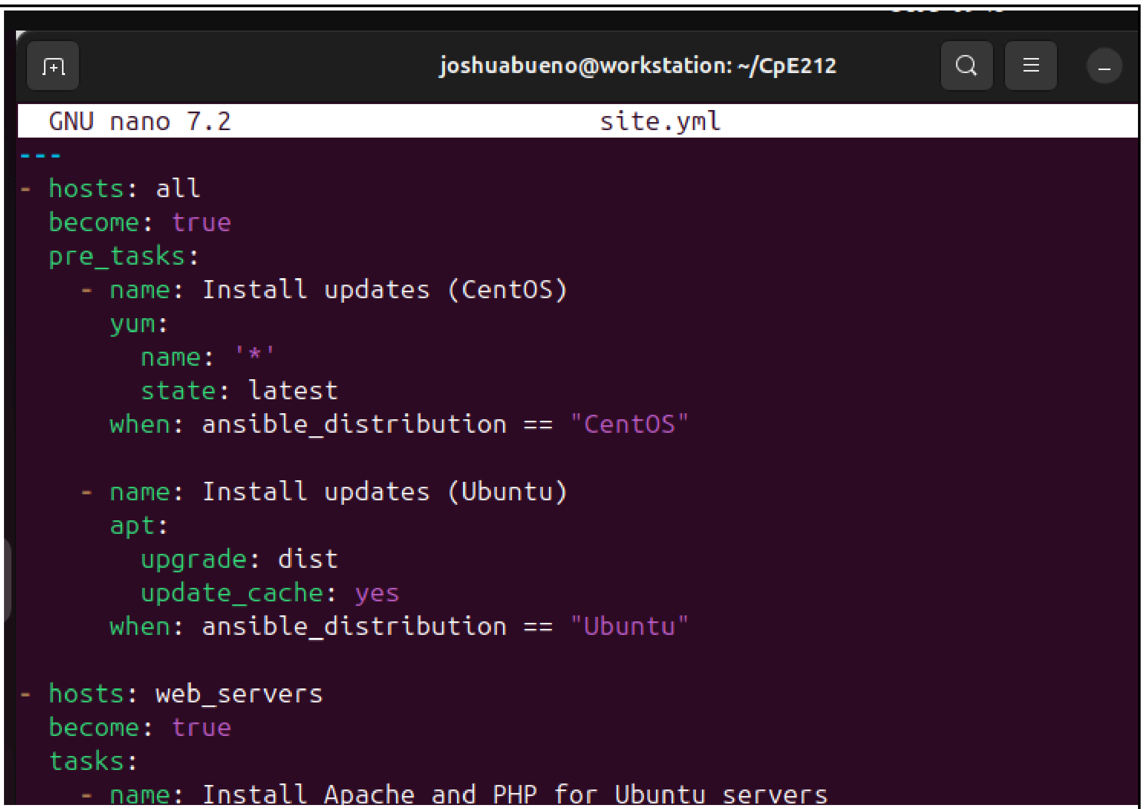
Make sure to save the file and exit.

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"
```

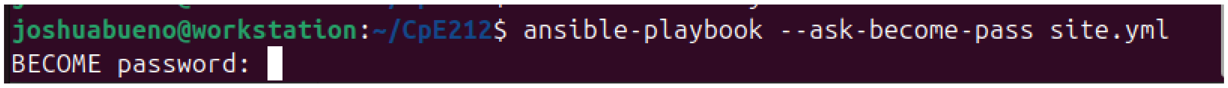

joshuabueno@workstation: ~/CpE212

```
  GNU nano 7.2                          site.yml
---
- hosts: all
  become: true
  pre_tasks:
    - name: Install updates (CentOS)
      yum:
        name: '*'
        state: latest
      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
```

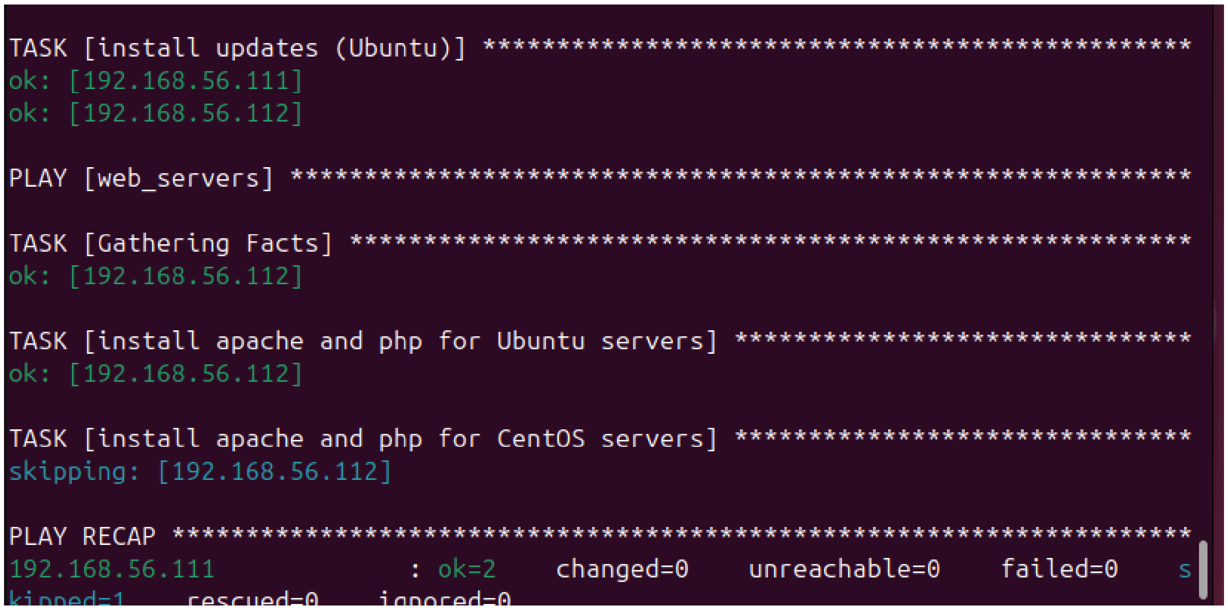
Make sure to save the file and exit.

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.

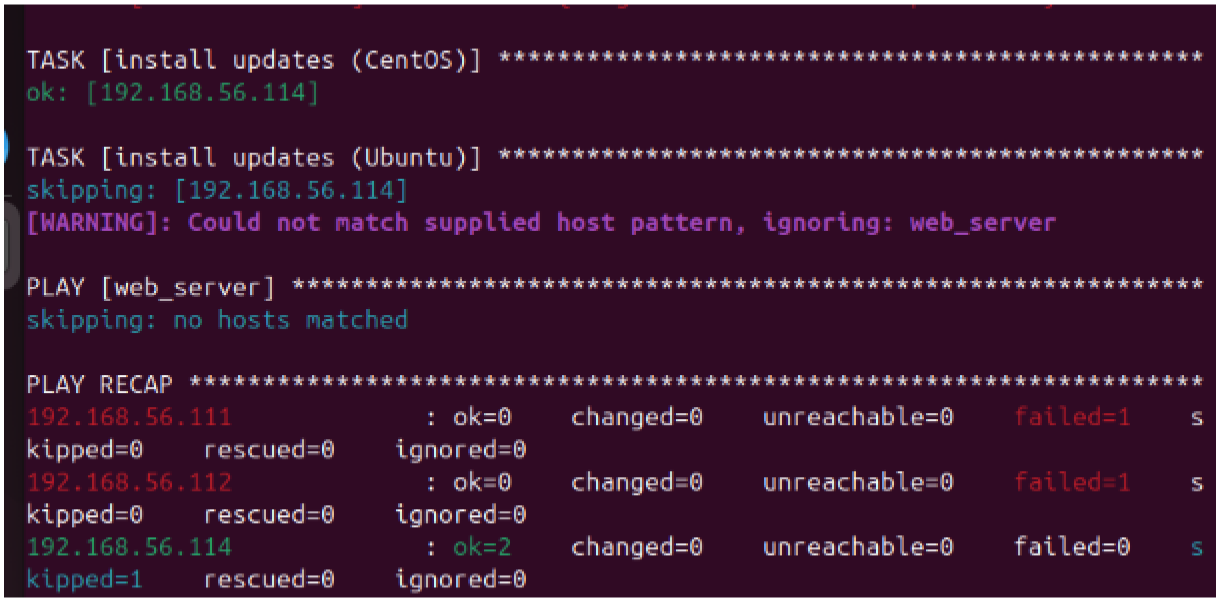
Run the *site.yml* file and describe the result.



```
joshuabueno@workstation:~/CpE212$ ansible-playbook --ask-become-pass site.yml
BECOME password: 
```



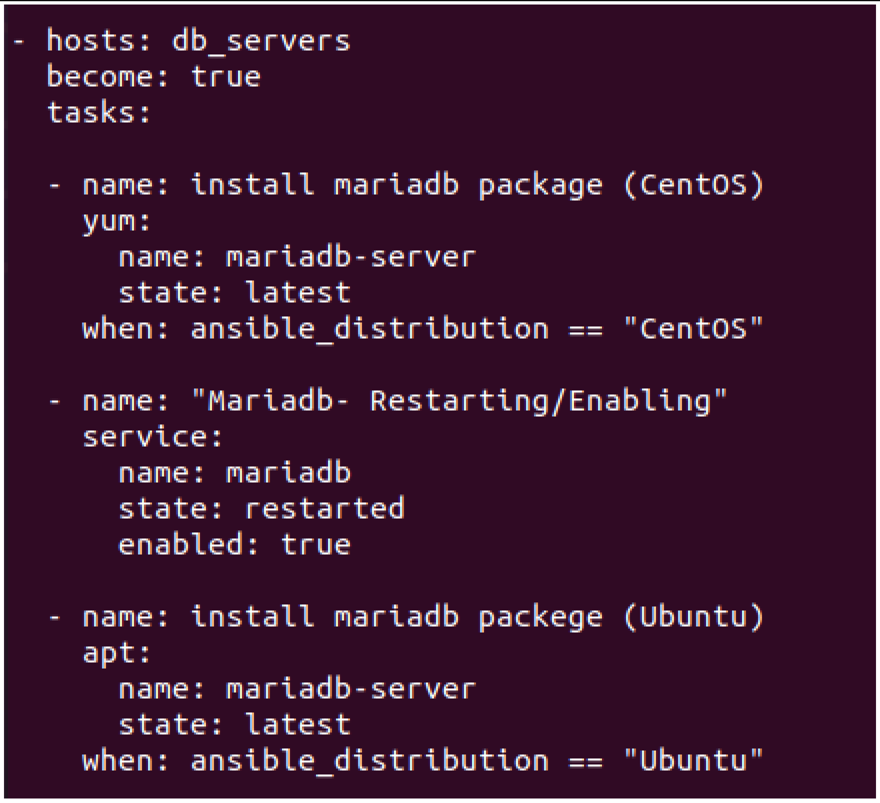
```
TASK [install updates (Ubuntu)] ***********************************************
ok: [192.168.56.111]
ok: [192.168.56.112]

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

TASK [Gathering Facts] ********************************************************
ok: [192.168.56.112]

TASK [install apache and php for Ubuntu servers] ******************************
ok: [192.168.56.112]

TASK [install apache and php for CentOS servers] ******************************
skipping: [192.168.56.112]

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



```
TASK [install updates (CentOS)] ***********************************************
ok: [192.168.56.114]

TASK [install updates (Ubuntu)] ***********************************************
skipping: [192.168.56.114]
[WARNING]: Could not match supplied host pattern, ignoring: web_server

PLAY [web_server] *************************************************************
skipping: no hosts matched

PLAY RECAP ********************************************************************
192.168.56.111             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
192.168.56.112             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
192.168.56.114             : ok=2    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
```

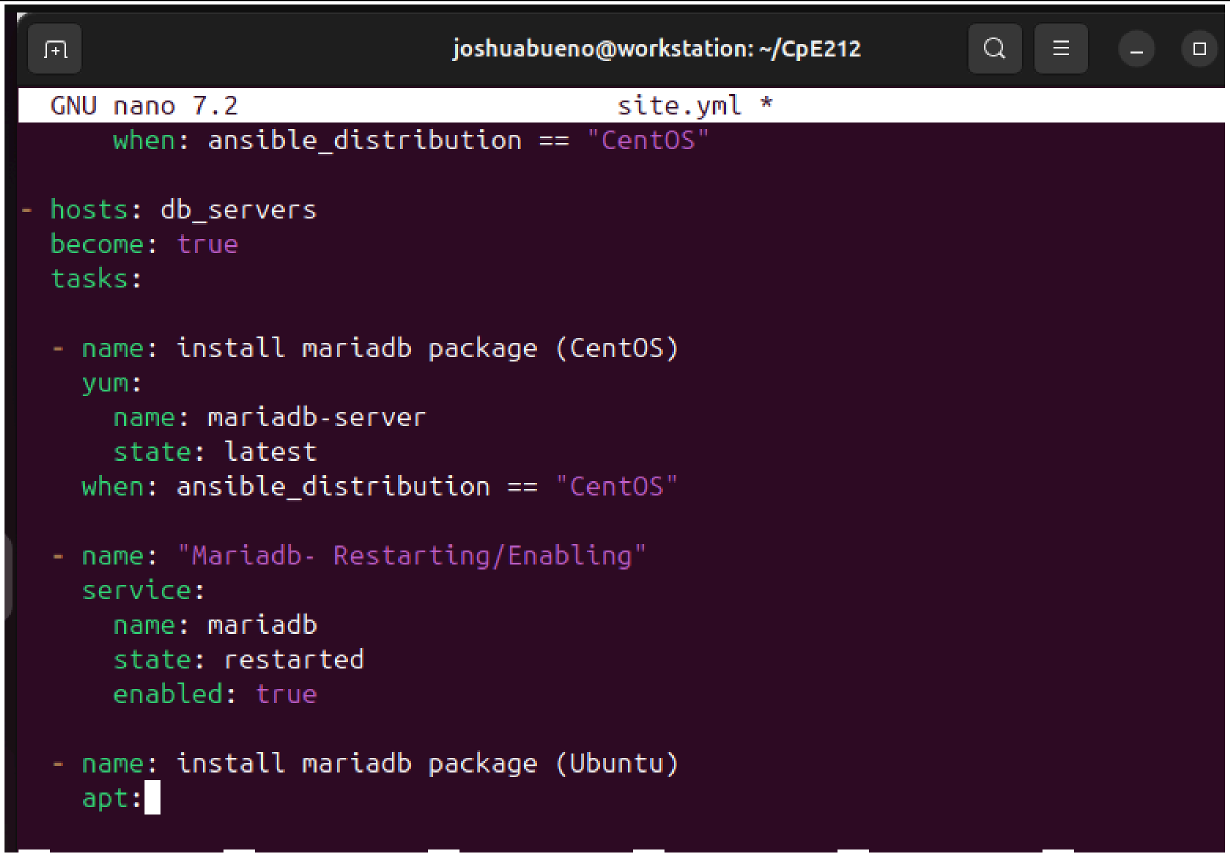
The installation was successful and it update the packages of the Ubuntu, CentOS, and other servers.

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.

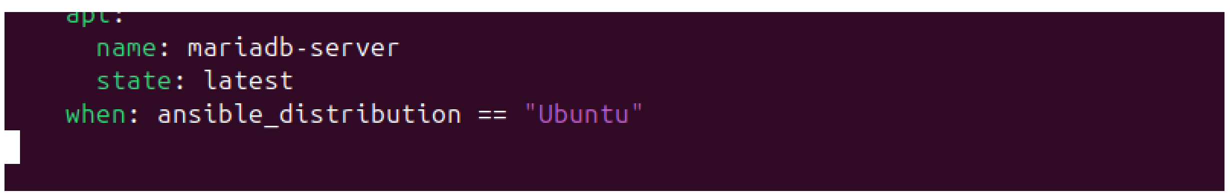


```
- hosts: db_servers
  become: true
  tasks:

  - 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.

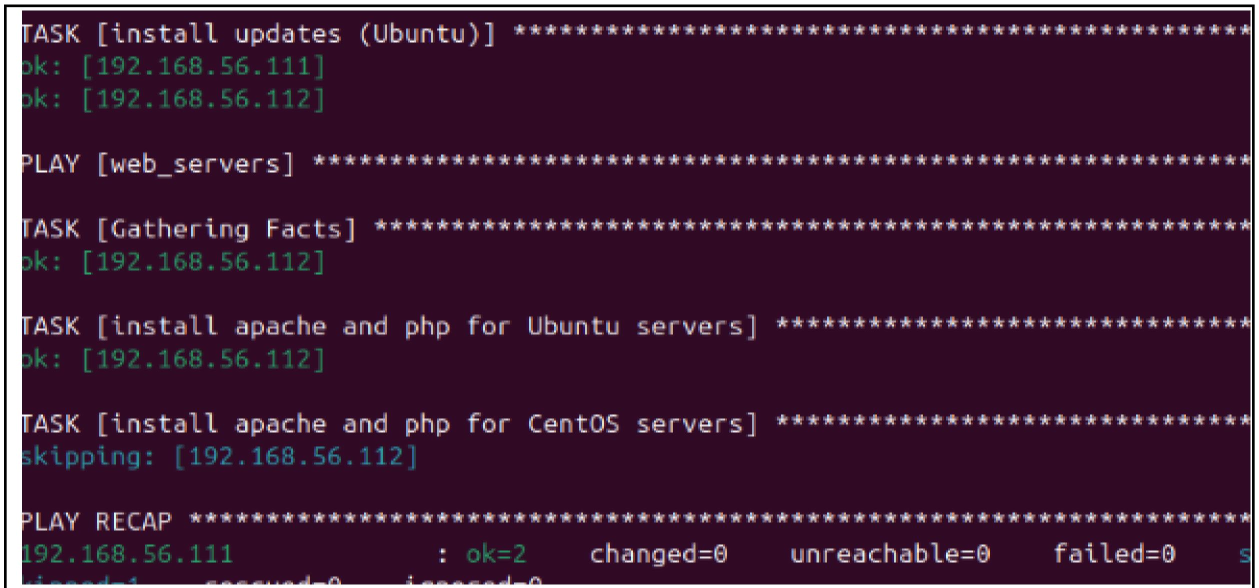


```
  GNU nano 7.2                                   site.yml *
      when: ansible_distribution == "CentOS"

- 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
      enabled: true

  - name: install mariadb package (Ubuntu)
    apt:
```

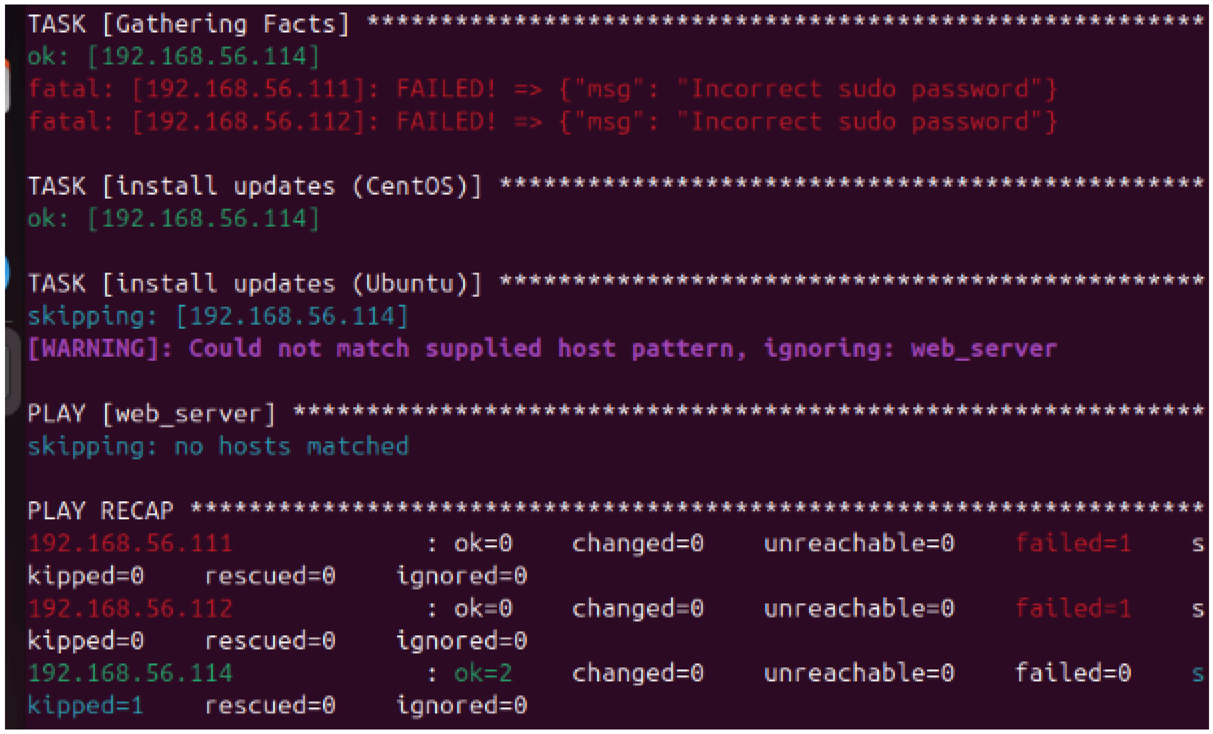


```
    apt:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "Ubuntu"
```

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



```
TASK [install updates (Ubuntu)] ************************************************
ok: [192.168.56.111]
ok: [192.168.56.112]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.112]

TASK [install apache and php for Ubuntu servers] *******************************
ok: [192.168.56.112]

TASK [install apache and php for CentOS servers] *******************************
skipping: [192.168.56.112]

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

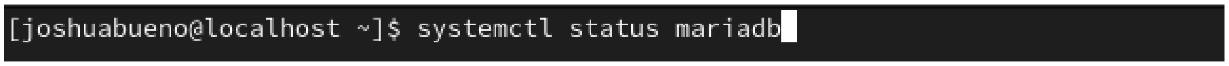


```
TASK [Gathering Facts] *********************************************************
ok: [192.168.56.114]
fatal: [192.168.56.111]: FAILED! => {"msg": "Incorrect sudo password"}
fatal: [192.168.56.112]: FAILED! => {"msg": "Incorrect sudo password"}

TASK [install updates (CentOS)] ************************************************
ok: [192.168.56.114]

TASK [install updates (Ubuntu)] ************************************************
skipping: [192.168.56.114]
[WARNING]: Could not match supplied host pattern, ignoring: web_server

PLAY [web_server] **************************************************************
skipping: no hosts matched

PLAY RECAP *********************************************************************
192.168.56.111             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
192.168.56.112             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
192.168.56.114             : ok=2    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
```

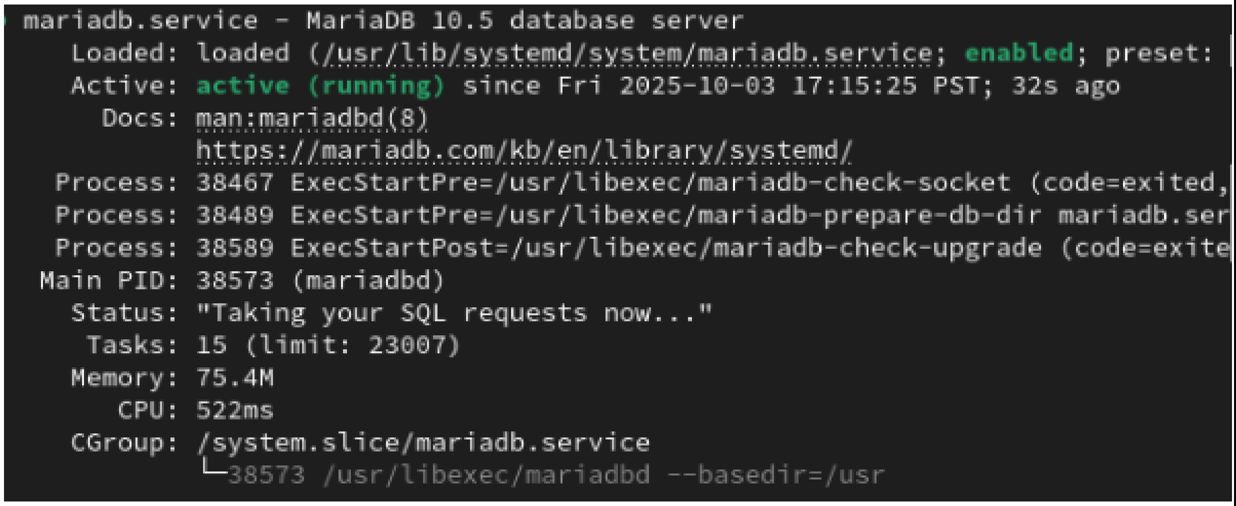
It was successfully installed again and updates the packages and the db_servers.

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.

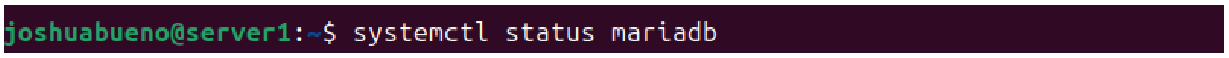
   Describe the output.



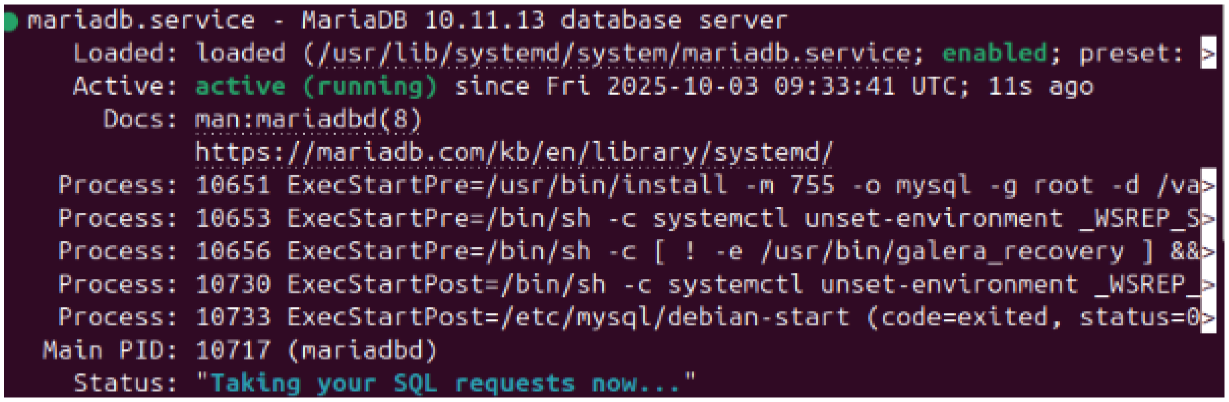
```
[joshuabueno@localhost ~]$ systemctl status mariadb
```



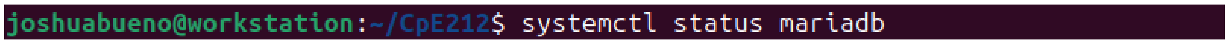
```
 mariadb.service - MariaDB 10.5 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset:
     Active: active (running) since Fri 2025-10-03 17:15:25 PST; 32s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 38467 ExecStartPre=/usr/libexec/mariadb-check-socket (code=exited,
    Process: 38489 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir mariadb.ser
    Process: 38589 ExecStartPost=/usr/libexec/mariadb-check-upgrade (code=exite
   Main PID: 38573 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 15 (limit: 23007)
     Memory: 75.4M
        CPU: 522ms
     CGroup: /system.slice/mariadb.service
             └─38573 /usr/libexec/mariadbd --basedir=/usr
```



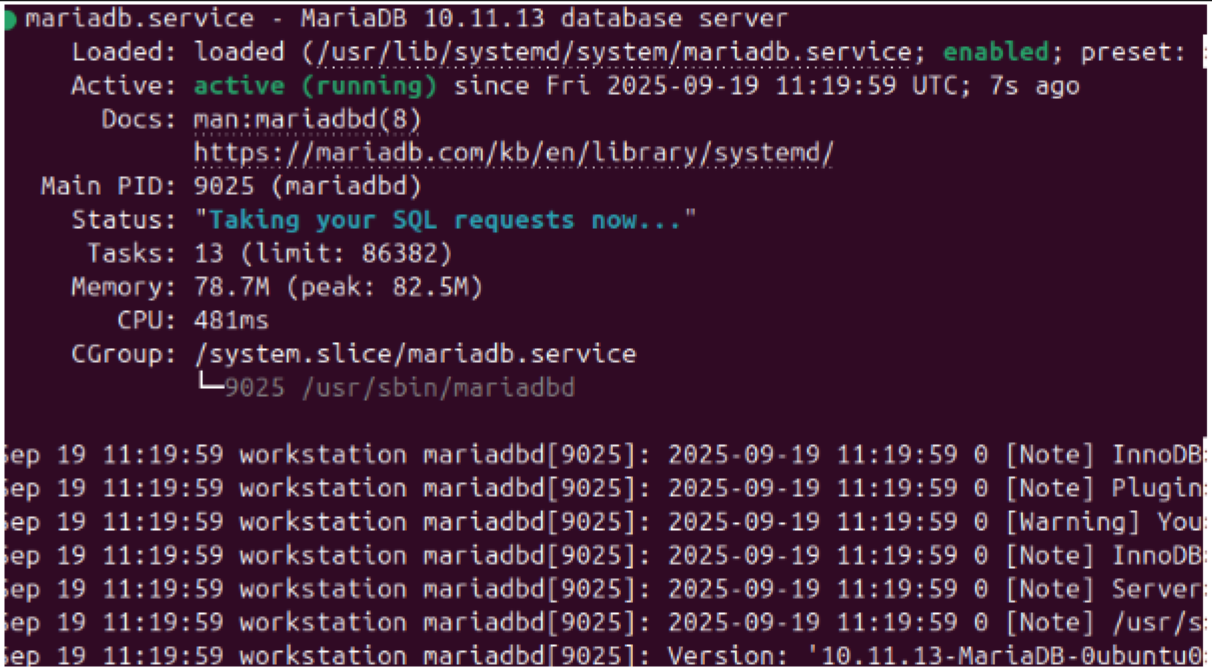
```
joshuabueno@server1:~$ systemctl status mariadb
```



```
● mariadb.service - MariaDB 10.11.13 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: >
     Active: active (running) since Fri 2025-10-03 09:33:41 UTC; 11s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 10651 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /va>
    Process: 10653 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_S>
    Process: 10656 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] &&>
    Process: 10730 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_>
    Process: 10733 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0>
   Main PID: 10717 (mariadbd)
     Status: "Taking your SQL requests now..."
```



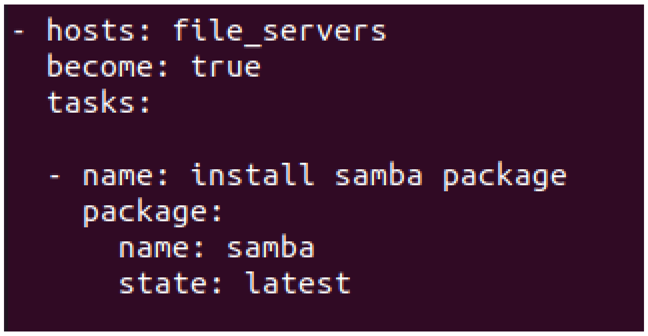
```
joshuabueno@workstation:~/CpE212$ systemctl status mariadb
```



```
● mariadb.service - MariaDB 10.11.13 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset:
     Active: active (running) since Fri 2025-09-19 11:19:59 UTC; 7s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 9025 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 13 (limit: 86382)
     Memory: 78.7M (peak: 82.5M)
        CPU: 481ms
     CGroup: /system.slice/mariadb.service
             └─9025 /usr/sbin/mariadbd

Sep 19 11:19:59 workstation mariadbd[9025]: 2025-09-19 11:19:59 0 [Note] InnoDB
Sep 19 11:19:59 workstation mariadbd[9025]: 2025-09-19 11:19:59 0 [Note] Plugin
Sep 19 11:19:59 workstation mariadbd[9025]: 2025-09-19 11:19:59 0 [Warning] You
Sep 19 11:19:59 workstation mariadbd[9025]: 2025-09-19 11:19:59 0 [Note] InnoDB
Sep 19 11:19:59 workstation mariadbd[9025]: 2025-09-19 11:19:59 0 [Note] Server
Sep 19 11:19:59 workstation mariadbd[9025]: 2025-09-19 11:19:59 0 [Note] /usr/s
Sep 19 11:19:59 workstation mariadbd[9025]: Version: '10.11.13-MariaDB-0ubuntu0
```

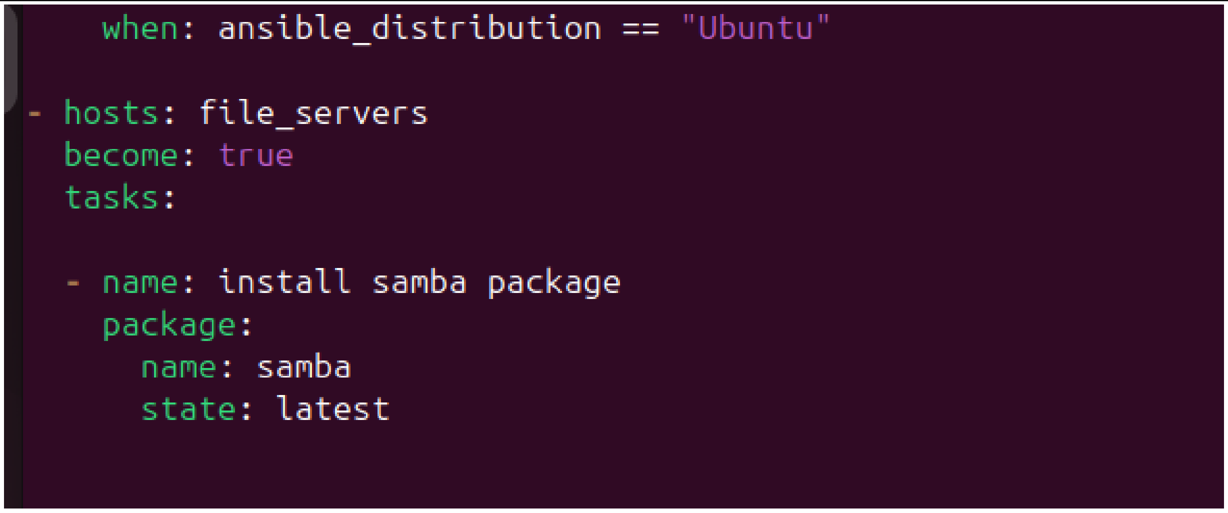
Based on the results. It successfully setup the installation of mariadb in CentOS, Managed Nodes, and the Control Node.

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.



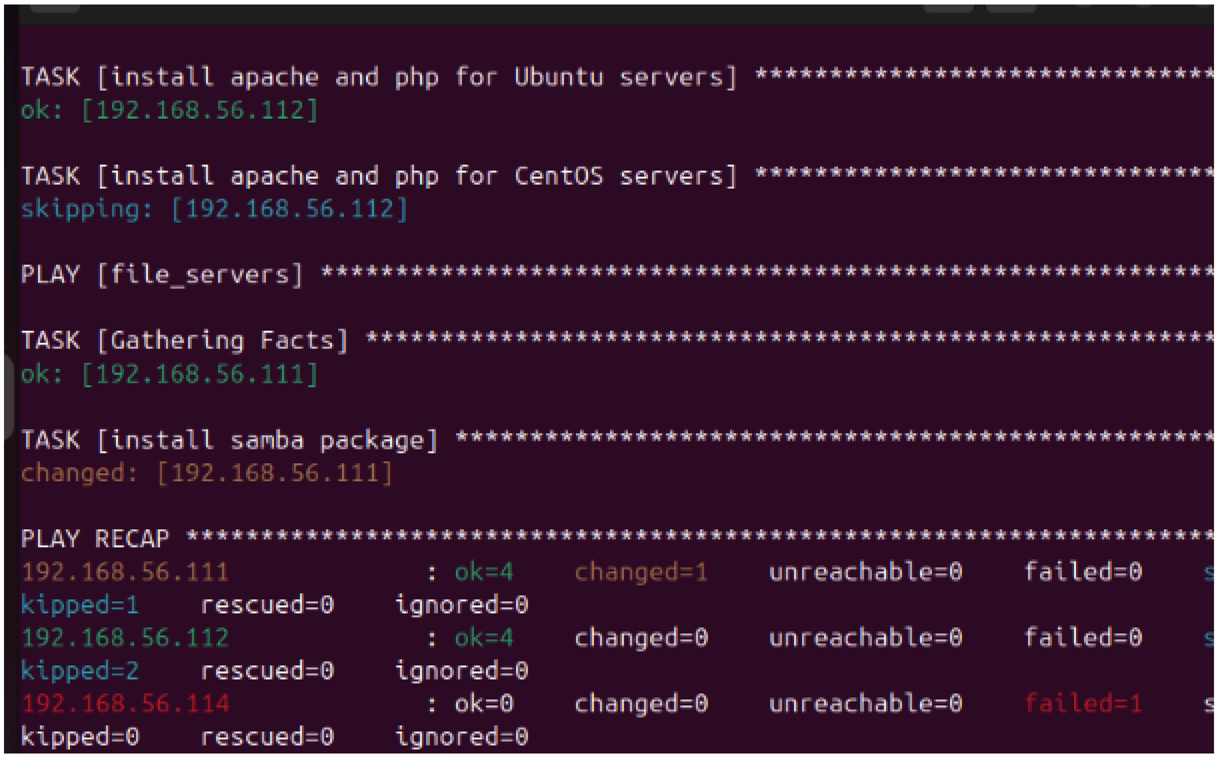
```
- hosts: file_servers
  become: true
  tasks:

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

Make sure to save the file and exit.

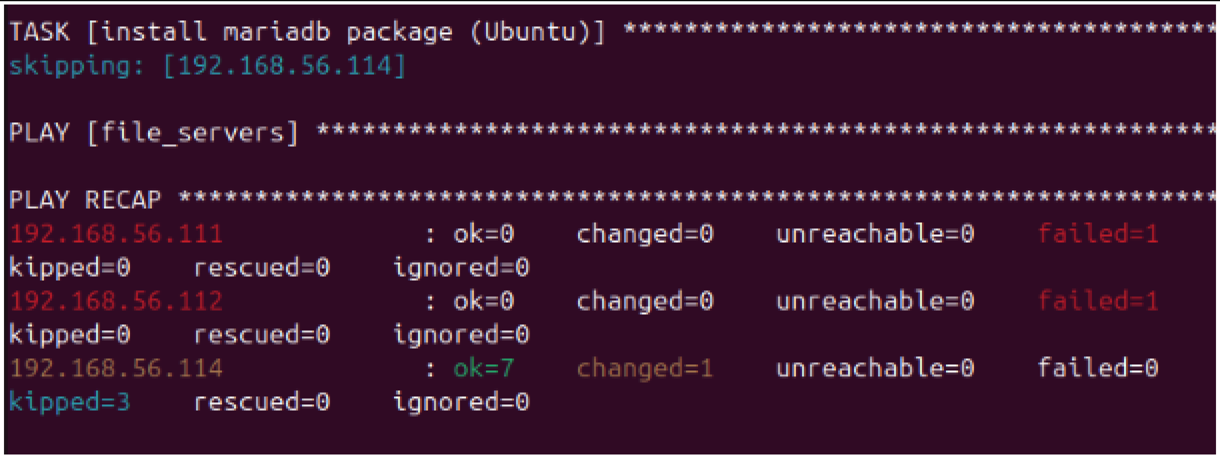


```
    when: ansible_distribution == "Ubuntu"

- hosts: file_servers
  become: true
  tasks:

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

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



```
TASK [install apache and php for Ubuntu servers] ********************************
ok: [192.168.56.112]

TASK [install apache and php for CentOS servers] ********************************
skipping: [192.168.56.112]

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

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

TASK [install samba package] ****************************************************
changed: [192.168.56.111]

PLAY RECAP **********************************************************************
192.168.56.111             : ok=4    changed=1    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.112             : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.114             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
```



```
TASK [install mariadb package (Ubuntu)] ***************************************
skipping: [192.168.56.114]

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

PLAY RECAP ********************************************************************
192.168.56.111             : ok=0    changed=0    unreachable=0    failed=1
kipped=0    rescued=0    ignored=0
192.168.56.112             : ok=0    changed=0    unreachable=0    failed=1
kipped=0    rescued=0    ignored=0
192.168.56.114             : ok=7    changed=1    unreachable=0    failed=0
kipped=3    rescued=0    ignored=0
```

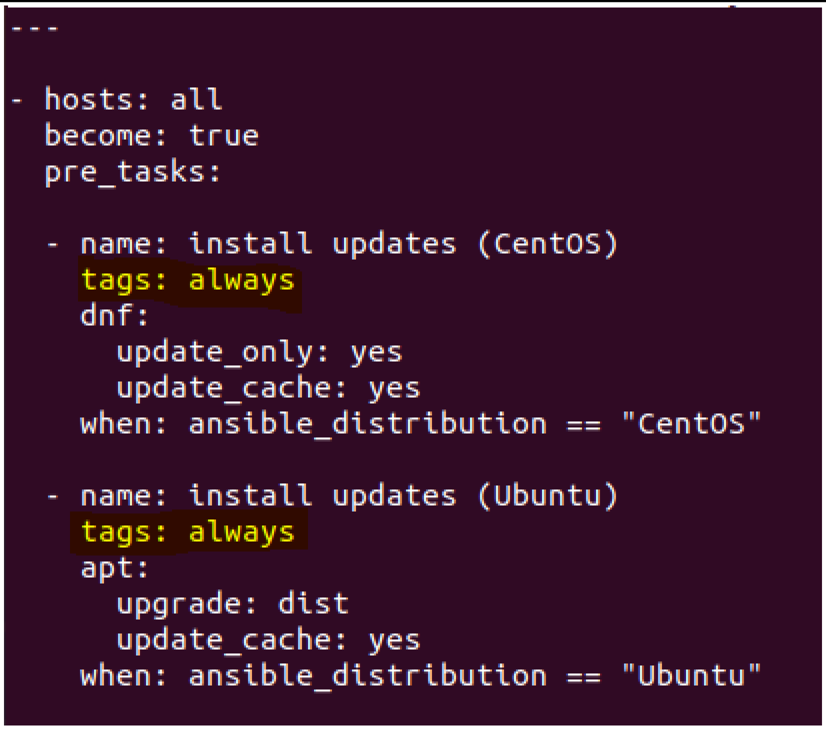
It successfully installed the samba package in file servers.

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.

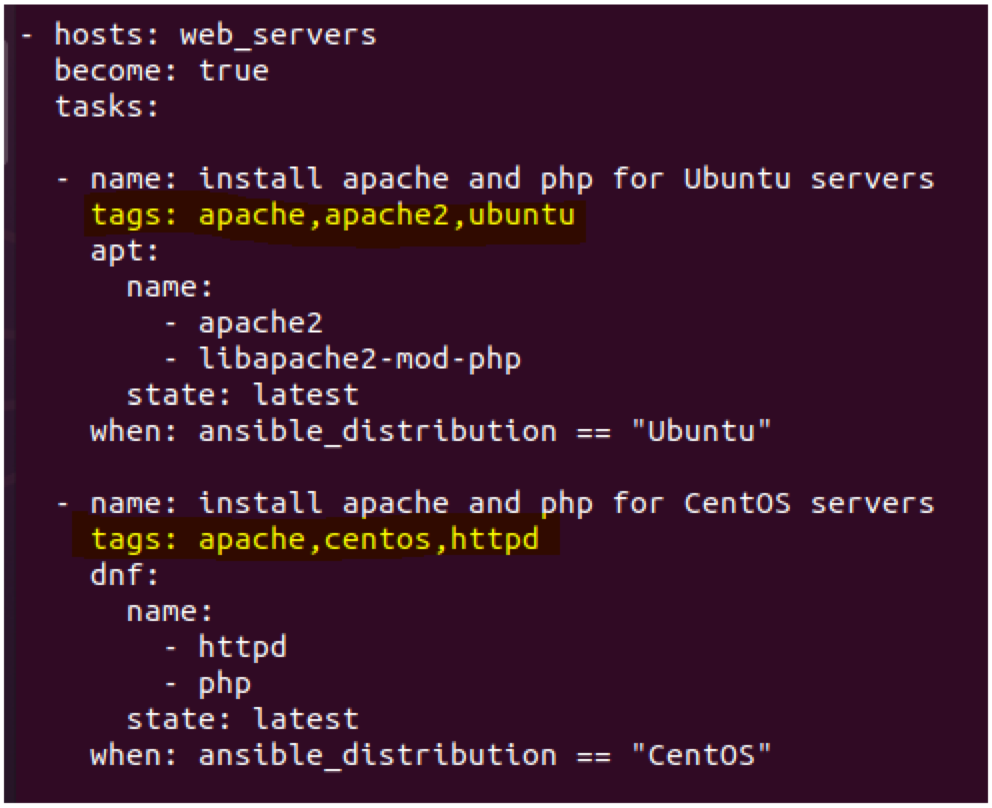
Task 2: Using Tags in running playbooks

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.

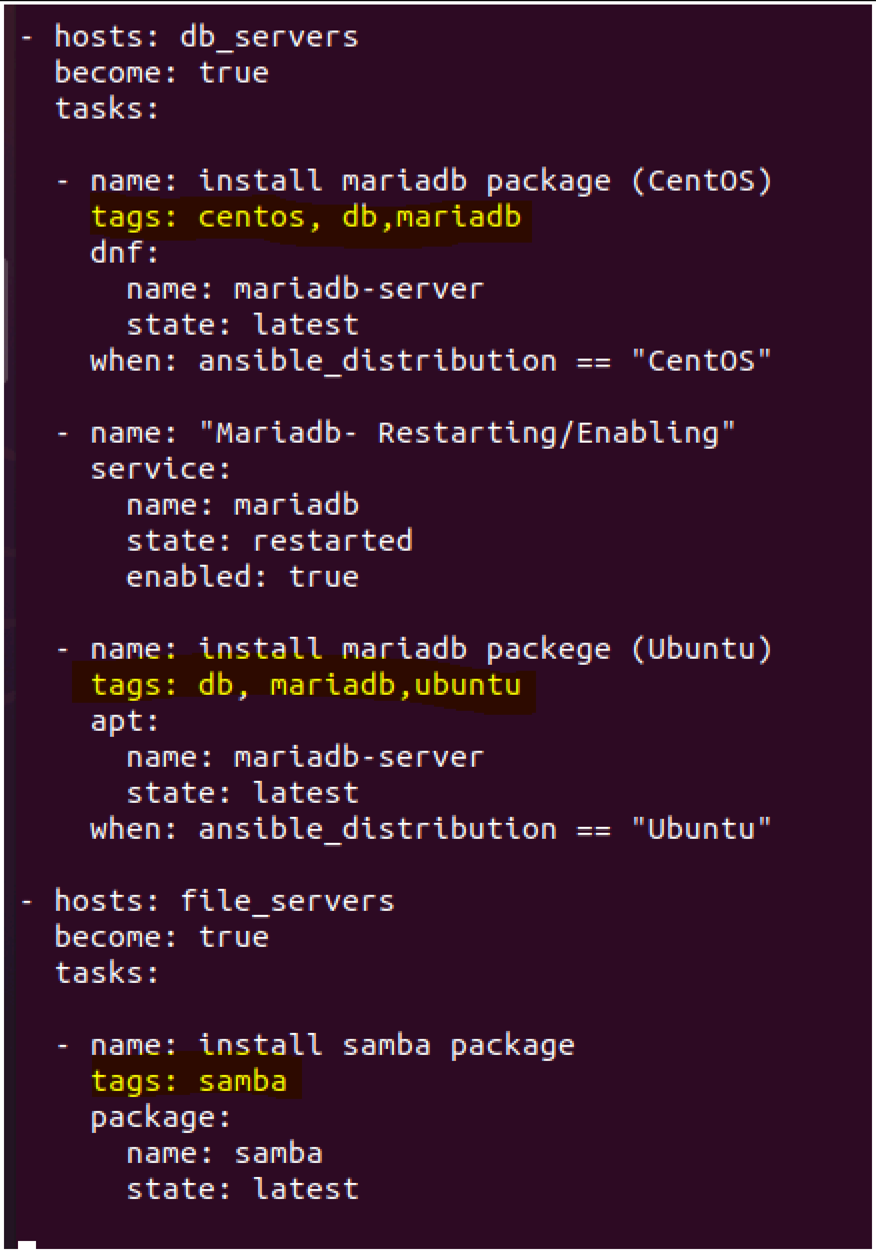
1. Edit the *site.yml* file. Add tags to the playbook. After the name, we can place the tags: *name_of_tag*. This is an arbitrary command, which means you can use any name for a tag.



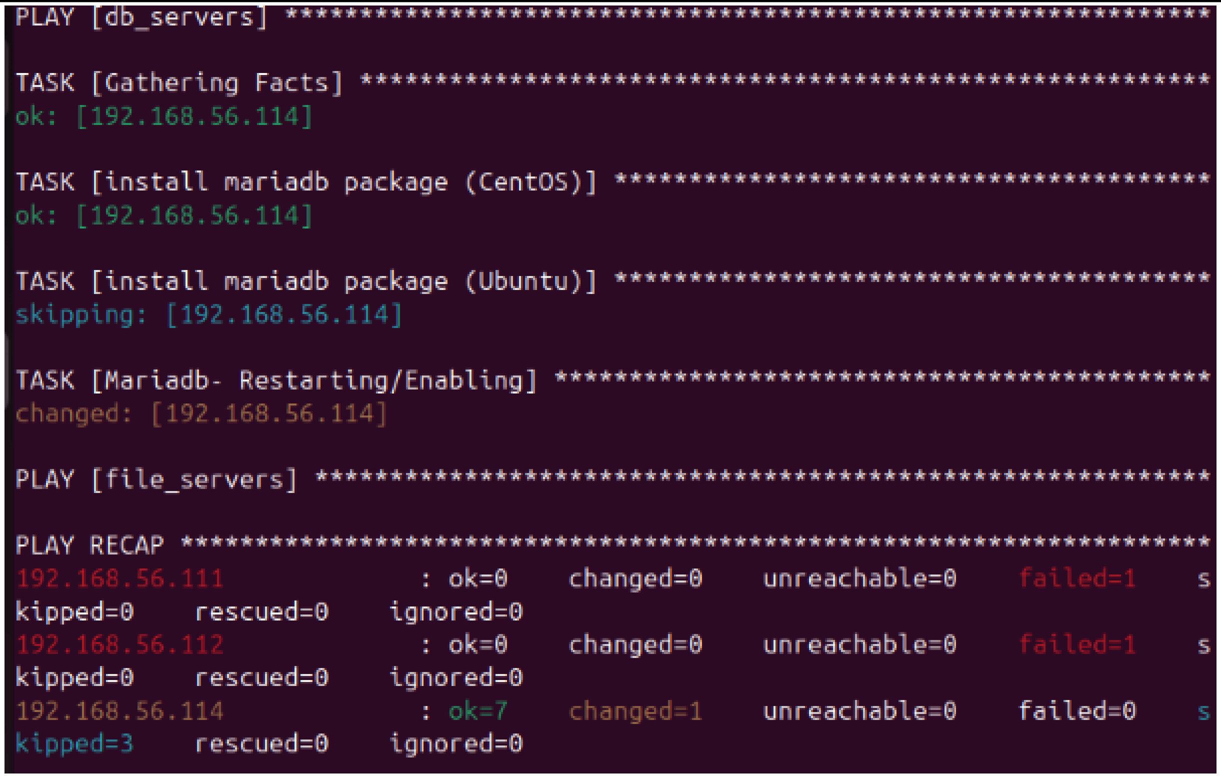
```
---

- 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"
```



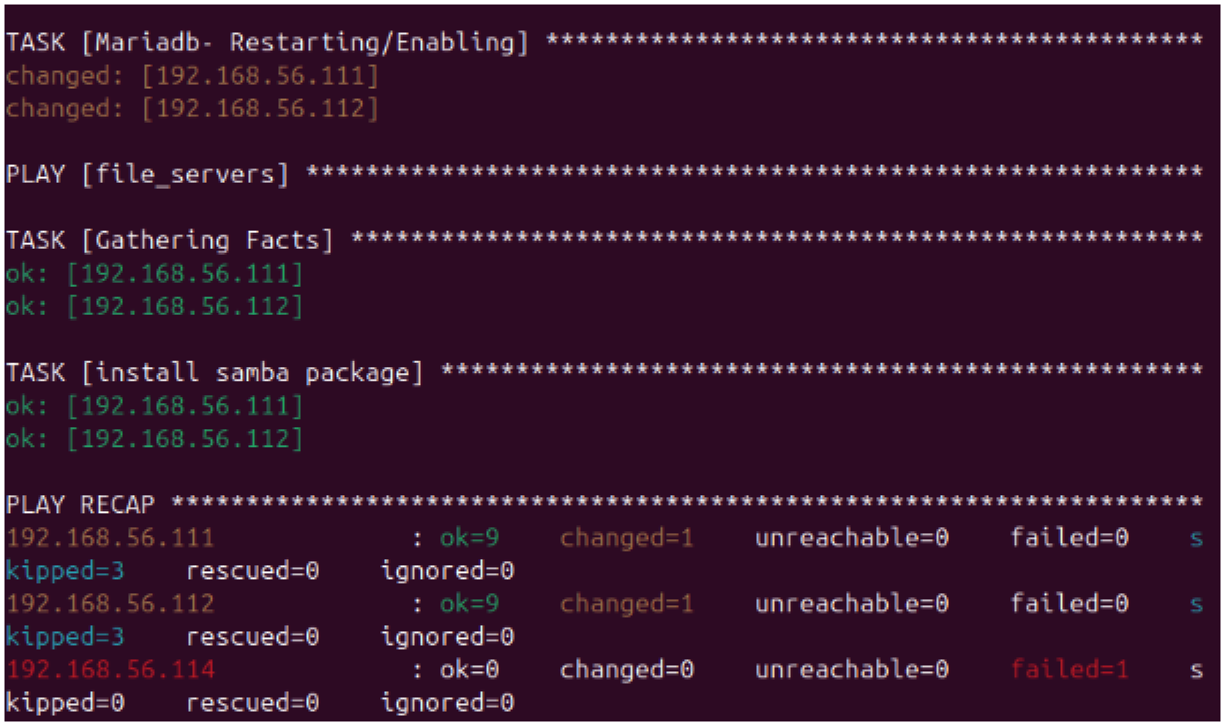
```
- hosts: web_servers
  become: true
  tasks:

  - 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
```



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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.114]

TASK [install mariadb package (CentOS)] ****************************************
ok: [192.168.56.114]

TASK [install mariadb package (Ubuntu)] ****************************************
skipping: [192.168.56.114]

TASK [Mariadb- Restarting/Enabling] ********************************************
changed: [192.168.56.114]

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

PLAY RECAP *********************************************************************
192.168.56.111             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
192.168.56.112             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
192.168.56.114             : ok=7    changed=1    unreachable=0    failed=0    s
kipped=3    rescued=0    ignored=0
```



```
TASK [Mariadb- Restarting/Enabling] ********************************************
changed: [192.168.56.111]
changed: [192.168.56.112]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.111]
ok: [192.168.56.112]

TASK [install samba package] ***************************************************
ok: [192.168.56.111]
ok: [192.168.56.112]

PLAY RECAP *********************************************************************
192.168.56.111             : ok=9    changed=1    unreachable=0    failed=0    s
kipped=3    rescued=0    ignored=0
192.168.56.112             : ok=9    changed=1    unreachable=0    failed=0    s
kipped=3    rescued=0    ignored=0
192.168.56.114             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
```

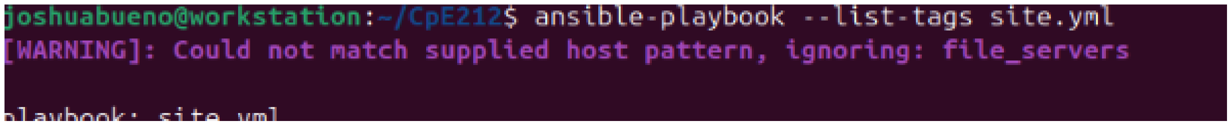
Make sure to save the file and exit.

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

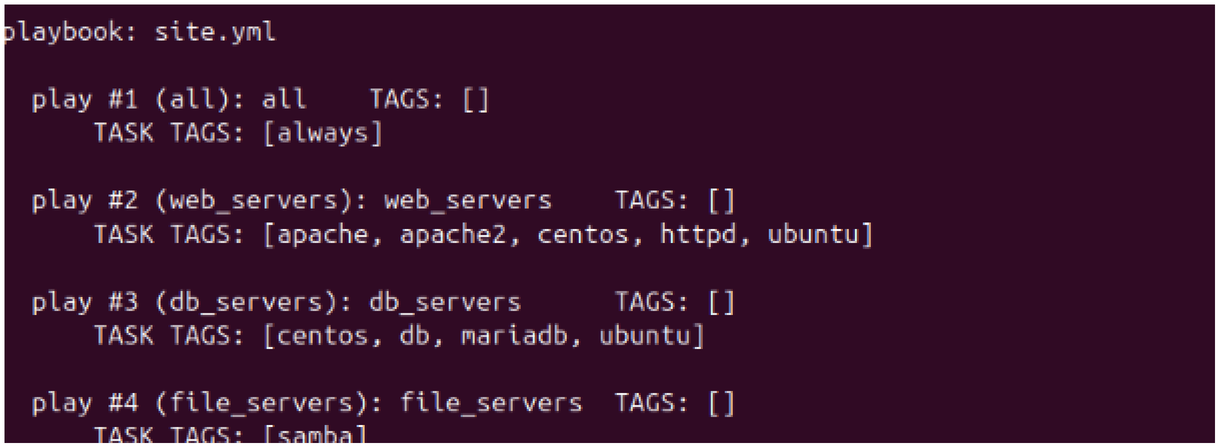
It run or skip specific parts of the playbook by calling the tag name during execution.

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

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

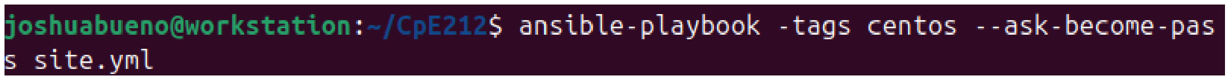


```
joshuabueno@workstation:~/CpE212$ ansible-playbook --list-tags site.yml
[WARNING]: Could not match supplied host pattern, ignoring: file_servers

playbook: site.yml
```

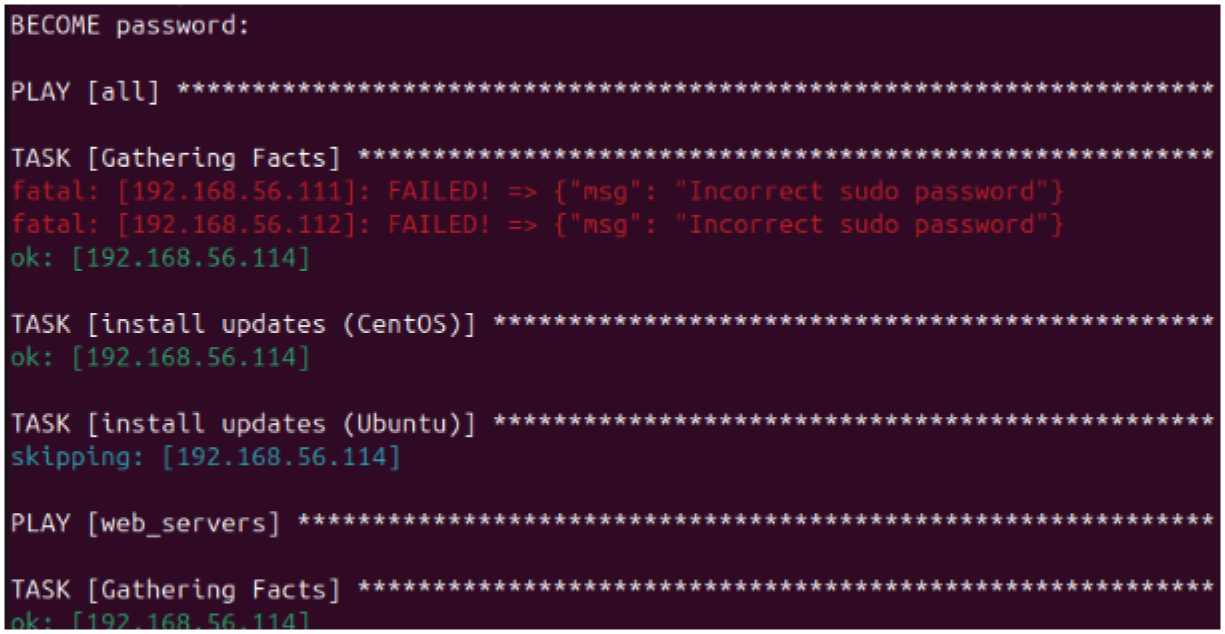


```
playbook: site.yml

  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]
```

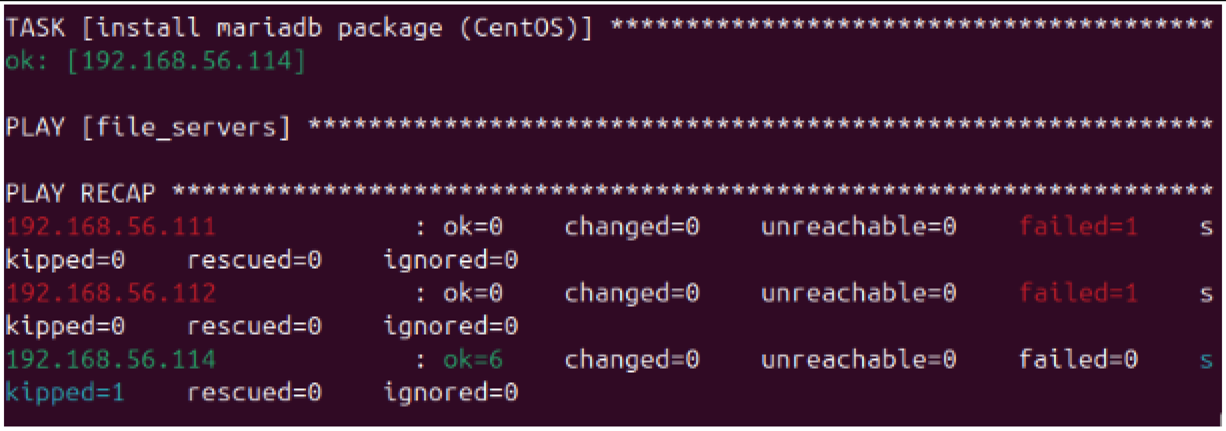
2.2 *ansible-playbook --tags centos --ask-become-pass site.yml*



```
joshuabueno@workstation:~/CpE212$ ansible-playbook -tags centos --ask-become-pas
s site.yml
```

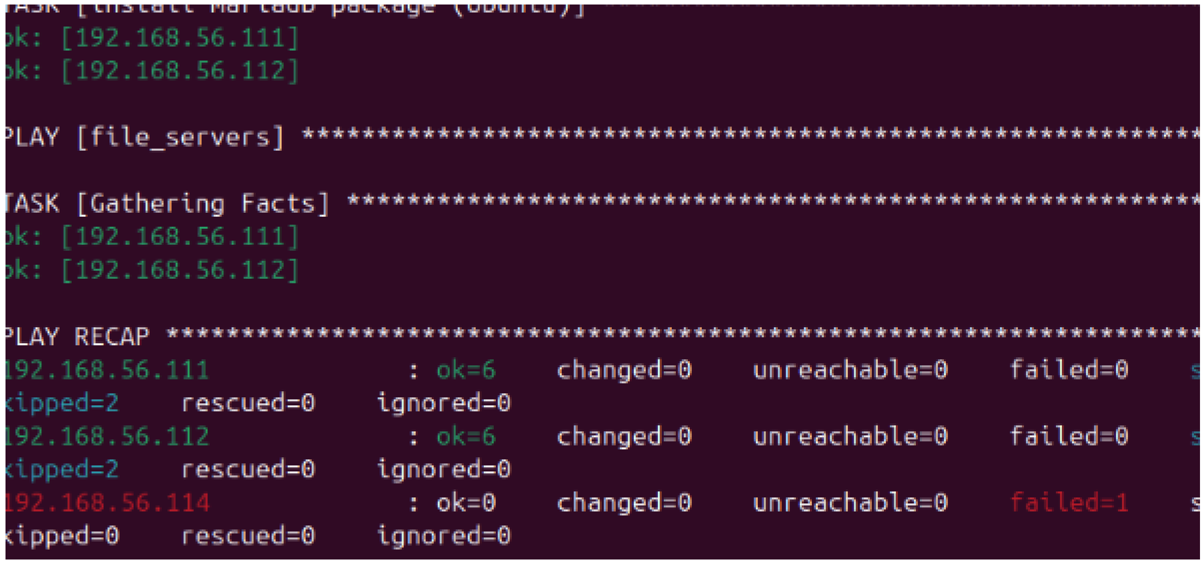


```
BECOME password:

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

TASK [Gathering Facts] ***********************************************************
fatal: [192.168.56.111]: FAILED! => {"msg": "Incorrect sudo password"}
fatal: [192.168.56.112]: FAILED! => {"msg": "Incorrect sudo password"}
ok: [192.168.56.114]

TASK [install updates (CentOS)] **************************************************
ok: [192.168.56.114]

TASK [install updates (Ubuntu)] **************************************************
skipping: [192.168.56.114]

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

TASK [Gathering Facts] ***********************************************************
ok: [192.168.56.114]
```

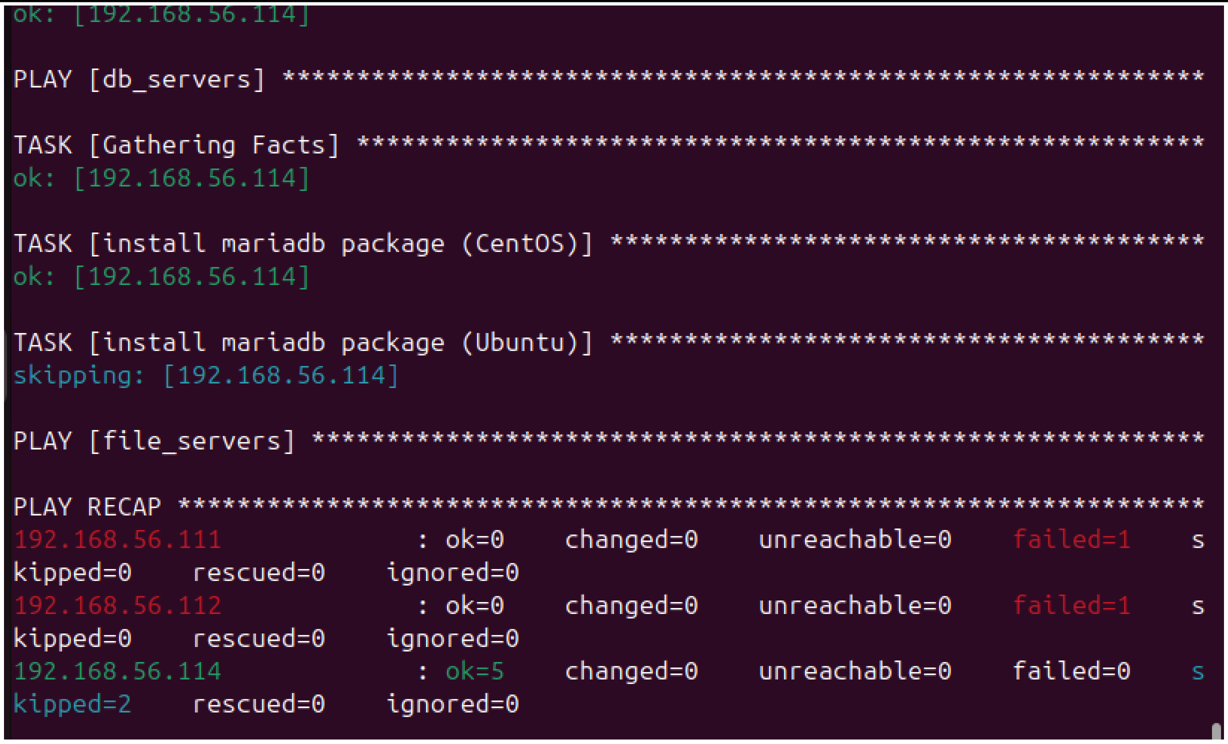


```
TASK [install mariadb package (CentOS)] ****************************************
ok: [192.168.56.114]

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

PLAY RECAP *********************************************************************
192.168.56.111             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
192.168.56.112             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
192.168.56.114             : ok=6    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
```

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

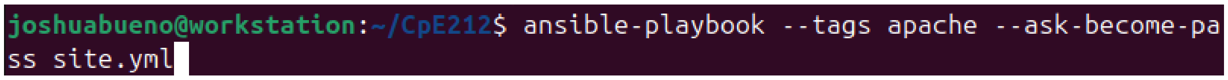


```
ok: [192.168.56.111]
ok: [192.168.56.112]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.111]
ok: [192.168.56.112]

PLAY RECAP *********************************************************************
192.168.56.111             : ok=6    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.112             : ok=6    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.114             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
```

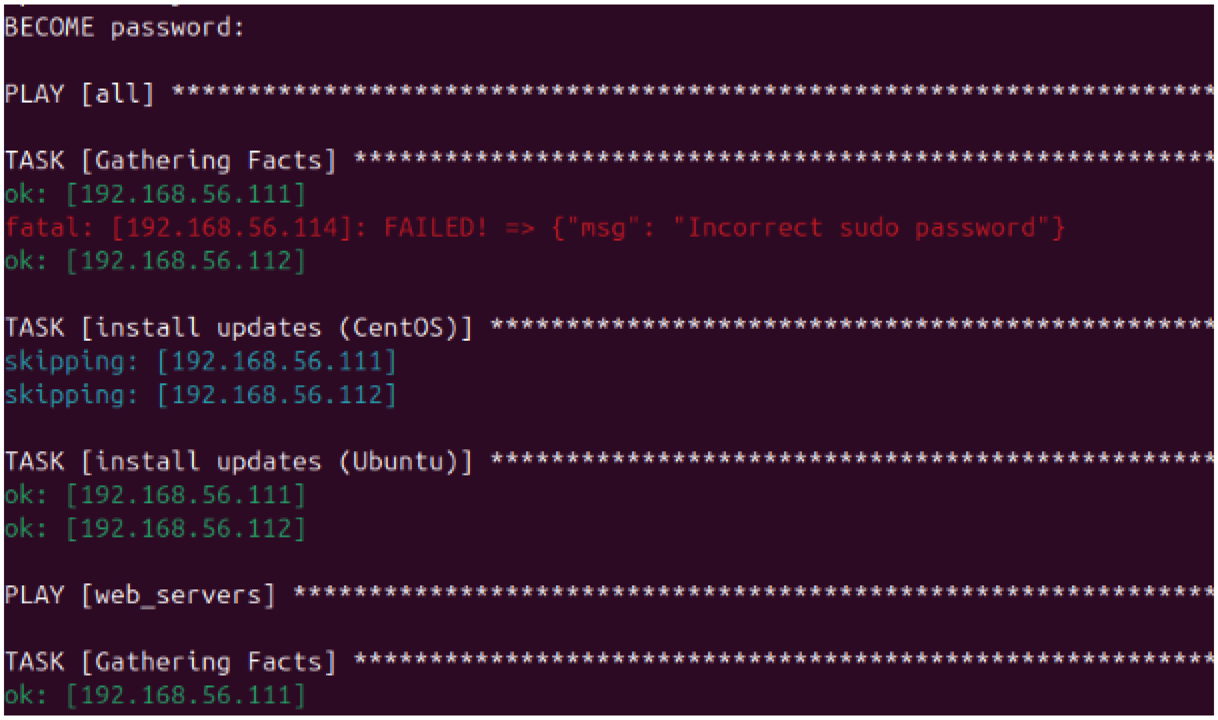


```
ok: [192.168.56.114]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.114]

TASK [install mariadb package (CentOS)] ****************************************
ok: [192.168.56.114]

TASK [install mariadb package (Ubuntu)] ****************************************
skipping: [192.168.56.114]

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

PLAY RECAP *********************************************************************
192.168.56.111             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
192.168.56.112             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
192.168.56.114             : ok=5    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
```

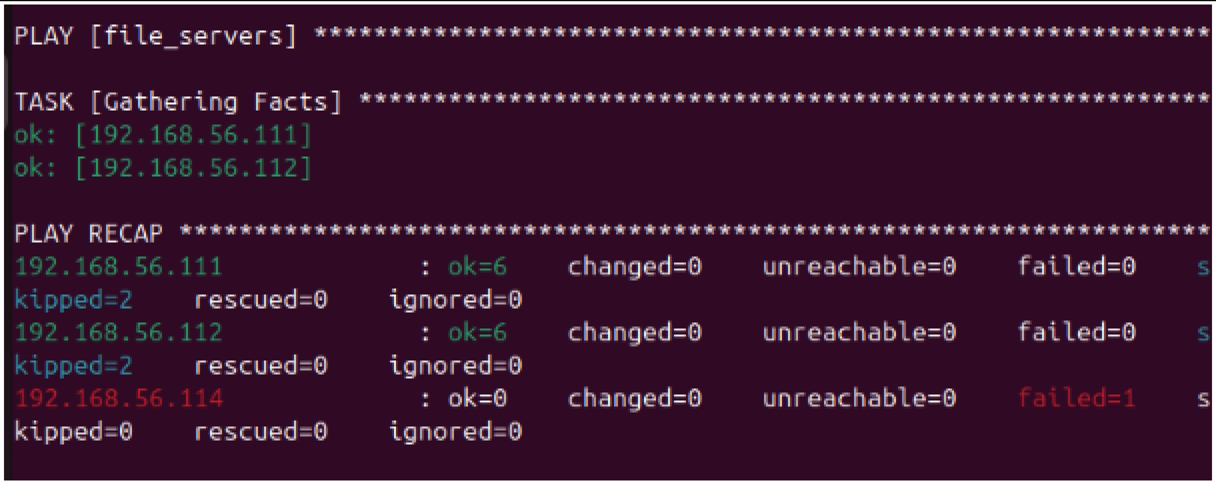
2.4 *ansible-playbook --tags apache --ask-become-pass site.yml*



```
joshuabueno@workstation:~/CpE212$ ansible-playbook --tags apache --ask-become-pa
ss site.yml
```

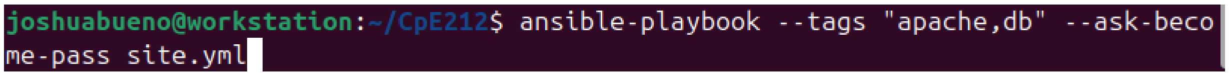


```
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.111]
fatal: [192.168.56.114]: FAILED! => {"msg": "Incorrect sudo password"}
ok: [192.168.56.112]

TASK [install updates (CentOS)] ************************************************
skipping: [192.168.56.111]
skipping: [192.168.56.112]

TASK [install updates (Ubuntu)] ************************************************
ok: [192.168.56.111]
ok: [192.168.56.112]

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

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

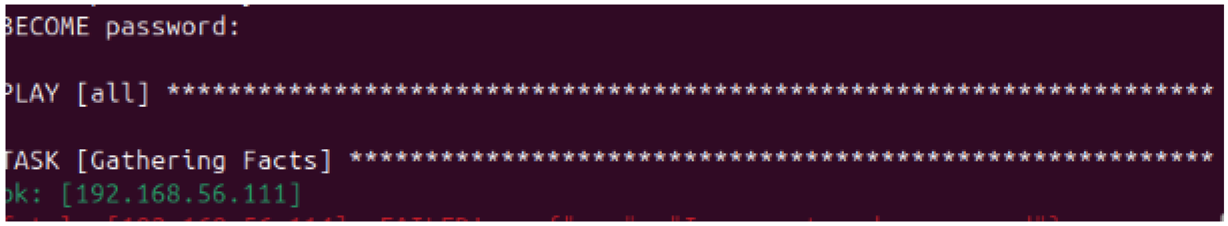


```
PLAY [file_servers] ***********************************************************

TASK [Gathering Facts] ********************************************************
ok: [192.168.56.111]
ok: [192.168.56.112]

PLAY RECAP ********************************************************************
192.168.56.111             : ok=6    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.112             : ok=6    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.114             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
```

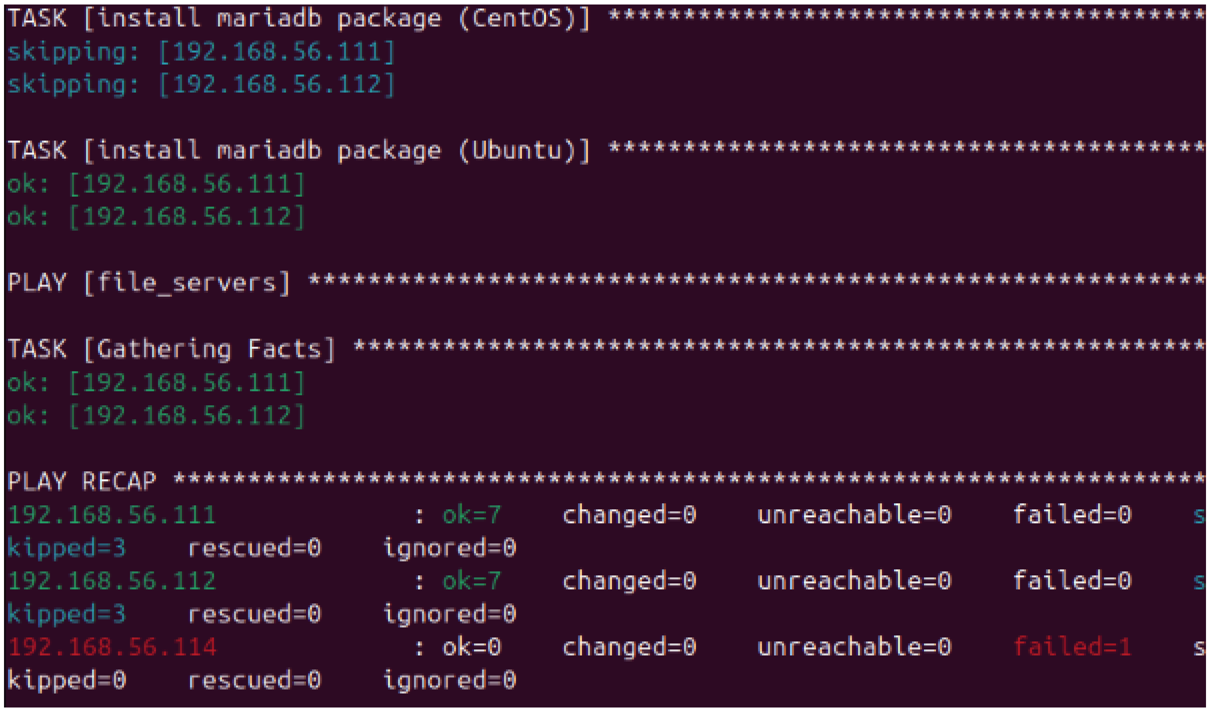
2.5 *ansible-playbook --tags “apache,db” --ask-become-pass site.yml*



```
joshuabueno@workstation:~/CpE212$ ansible-playbook --tags "apache,db" --ask-beco
me-pass site.yml
```



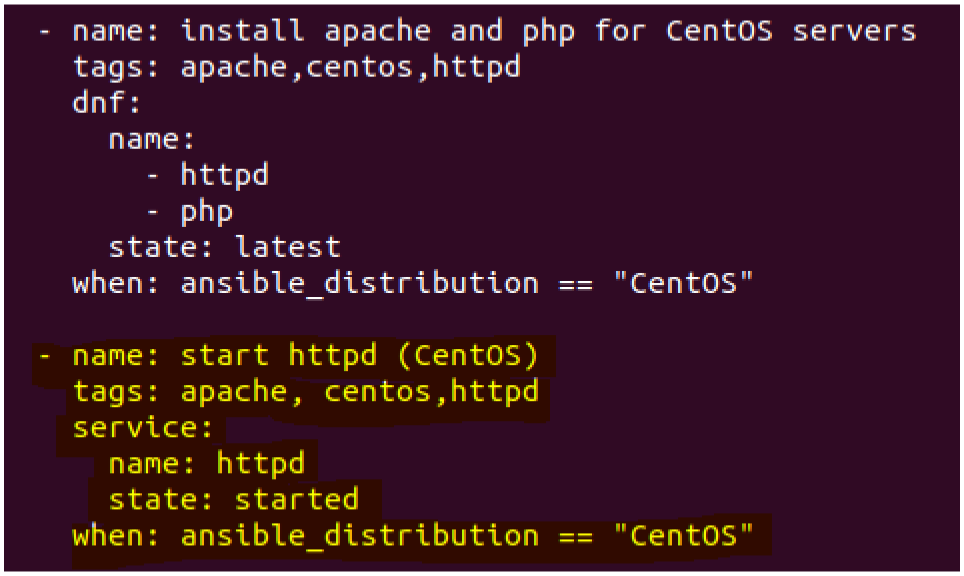
```
BECOME password:

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

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



```
TASK [install mariadb package (CentOS)] ***************************************
skipping: [192.168.56.111]
skipping: [192.168.56.112]

TASK [install mariadb package (Ubuntu)] ***************************************
ok: [192.168.56.111]
ok: [192.168.56.112]

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

TASK [Gathering Facts] ********************************************************
ok: [192.168.56.111]
ok: [192.168.56.112]

PLAY RECAP ********************************************************************
192.168.56.111             : ok=7    changed=0    unreachable=0    failed=0    s
kipped=3    rescued=0    ignored=0
192.168.56.112             : ok=7    changed=0    unreachable=0    failed=0    s
kipped=3    rescued=0    ignored=0
192.168.56.114             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
```

Task 3: Managing Services

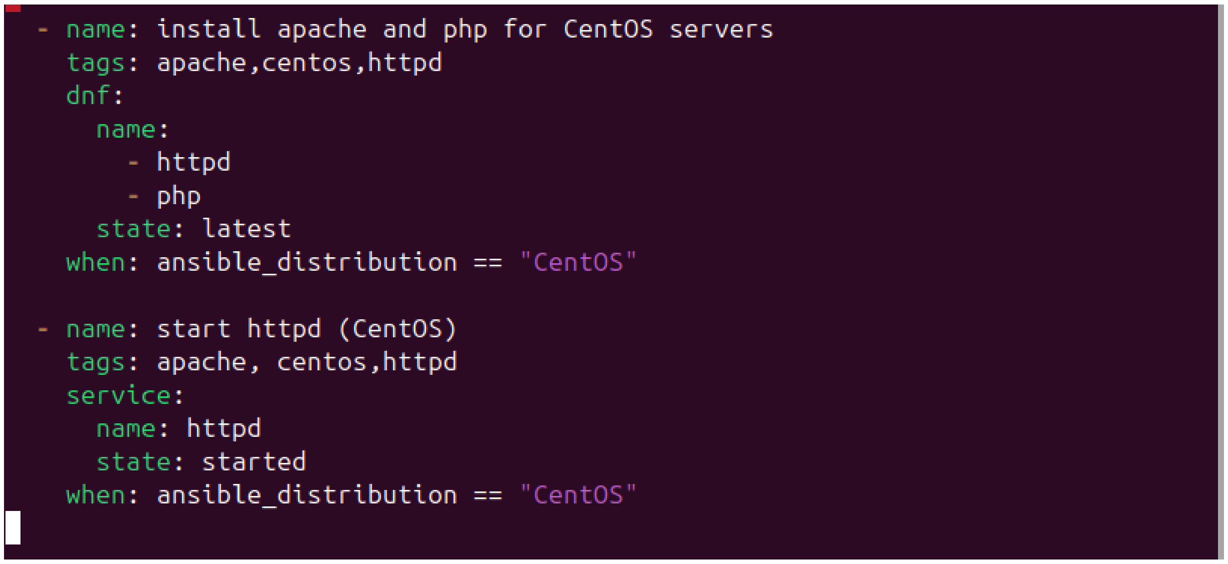
1. Edit the file site.yml and add a play that will automatically start the httpd on CentOS server.



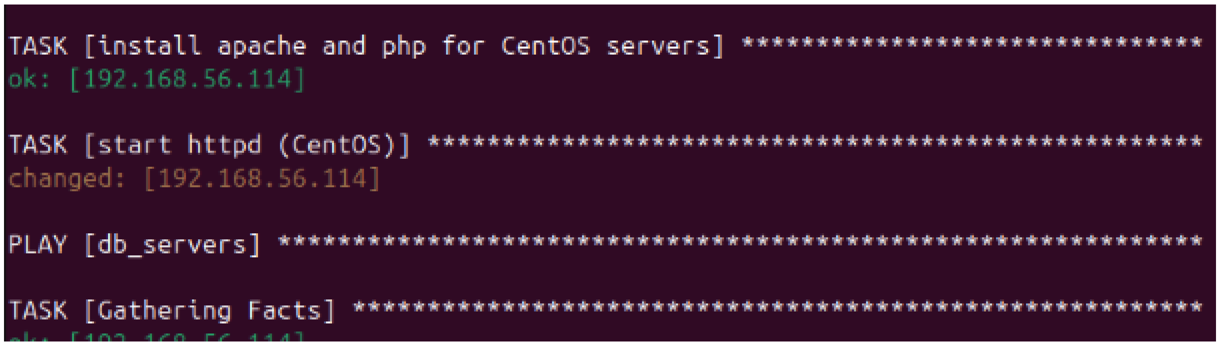
```
- name: install apache and php for CentOS servers
  tags: apache,centos,httpd
  dnf:
    name:
      - httpd
      - php
    state: latest
  when: ansible_distribution == "CentOS"

- 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.

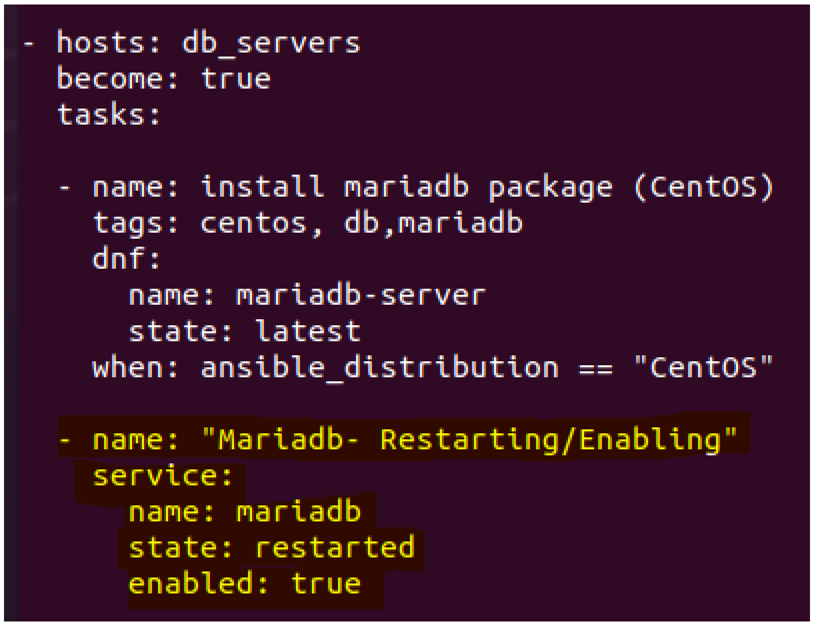


```
- name: install apache and php for CentOS servers
  tags: apache,centos,httpd
  dnf:
    name:
      - httpd
      - php
    state: latest
  when: ansible_distribution == "CentOS"

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



```
TASK [install apache and php for CentOS servers] *********************************
ok: [192.168.56.114]

TASK [start httpd (CentOS)] ******************************************************
changed: [192.168.56.114]

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

TASK [Gathering Facts] ***********************************************************
```

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

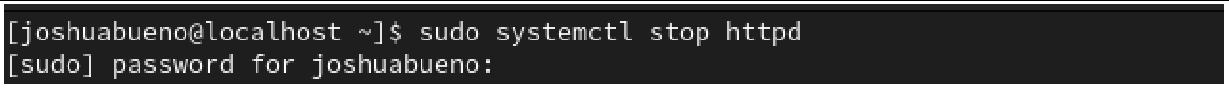


```
- hosts: db_servers
  become: true
  tasks:

  - 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
```

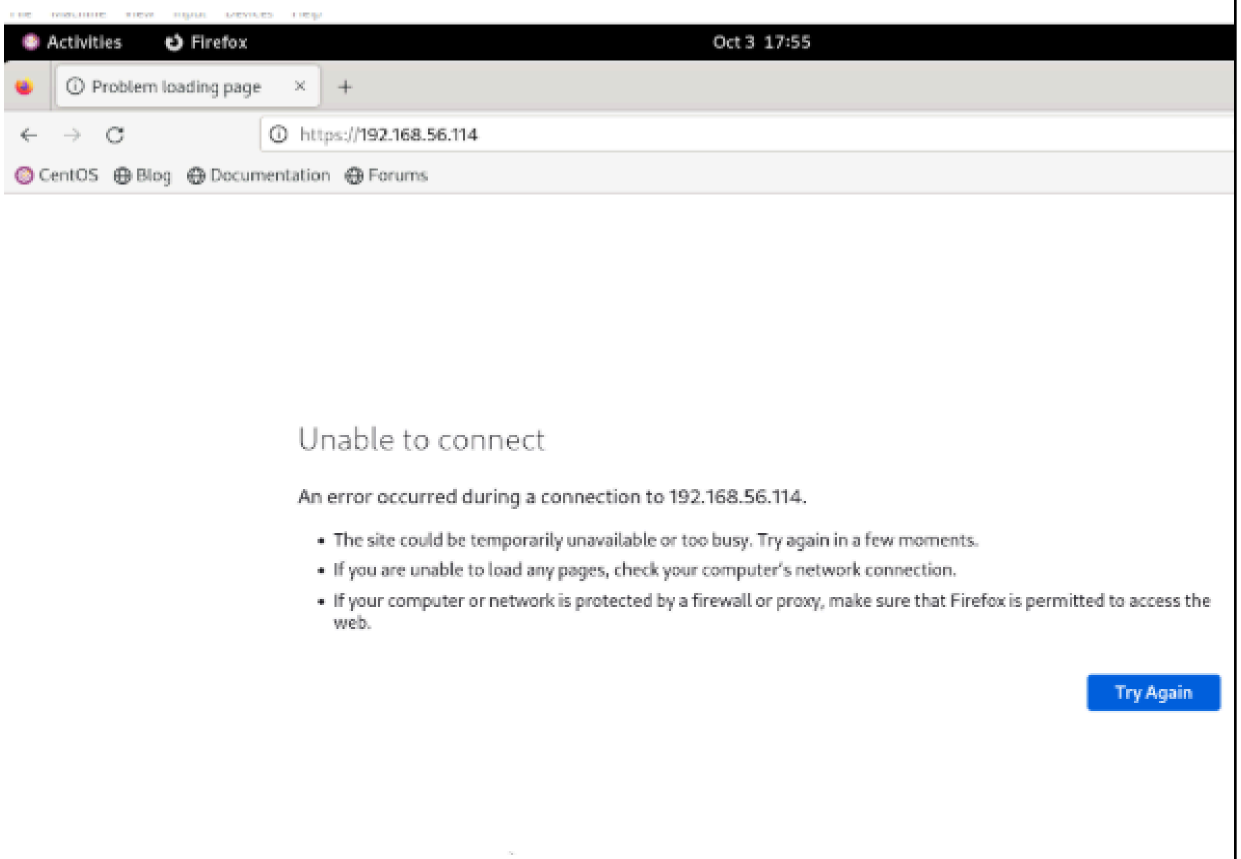
Figure 3.1.2

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.



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

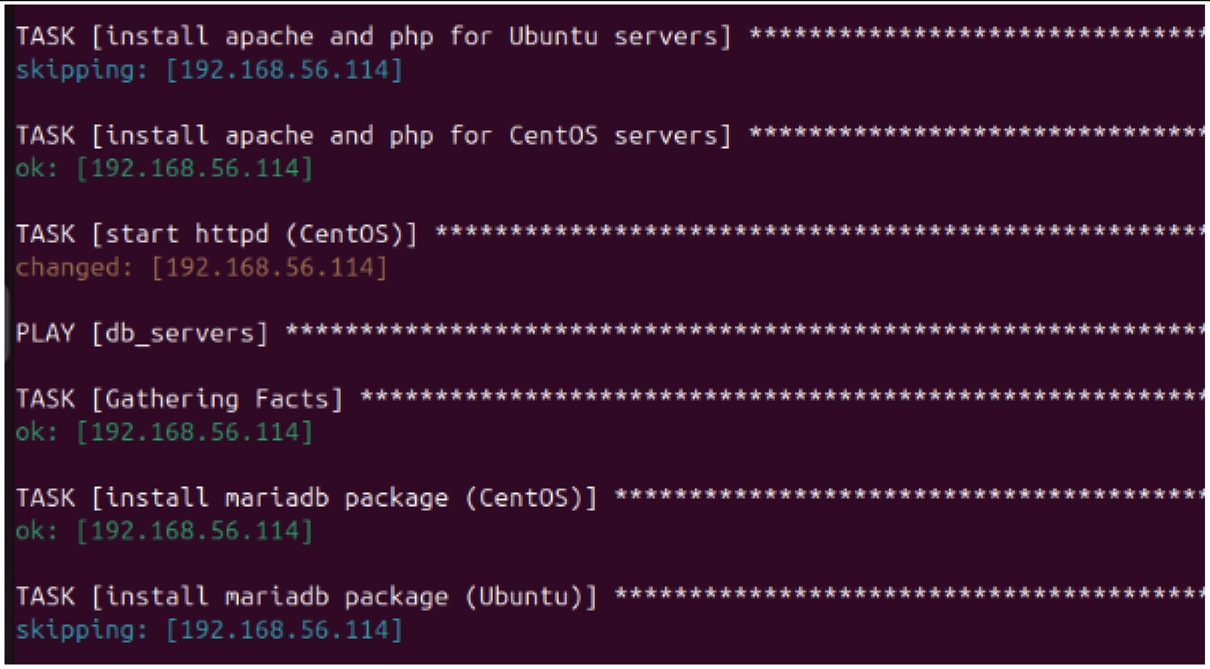


Unable to connect

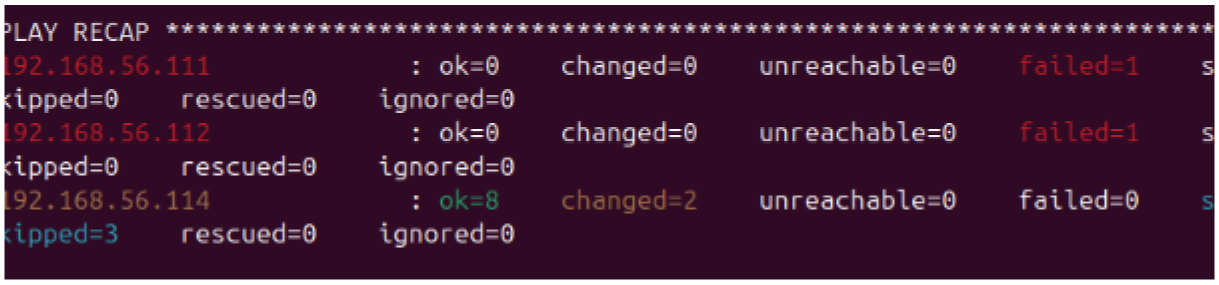
An error occurred during a connection to 192.168.56.114.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the web.

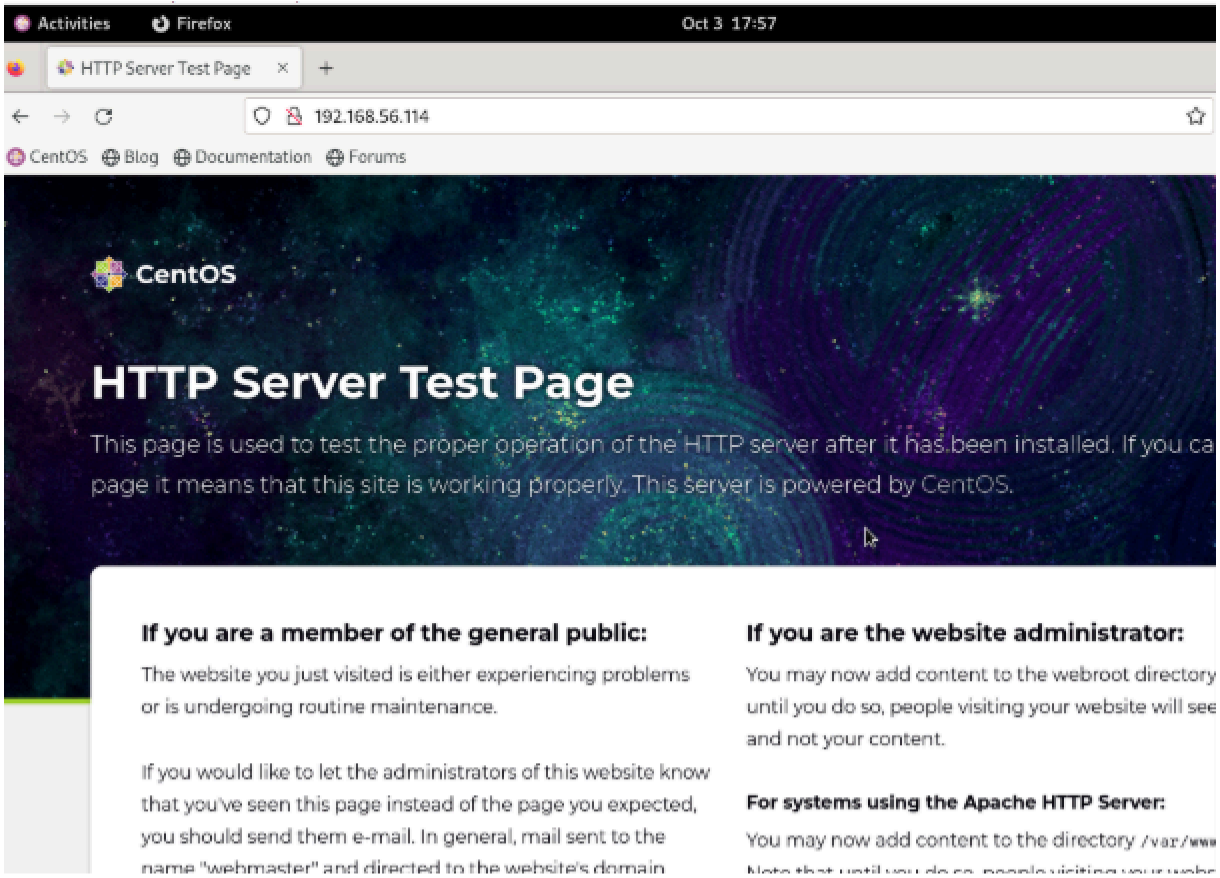
3. Go to the local machine and this time, run the *site.yml* file. Then after running the file, go again to the CentOS server and enter its IP address on the browser. Describe the result.



```
TASK [install apache and php for Ubuntu servers] ********************************
skipping: [192.168.56.114]

TASK [install apache and php for CentOS servers] ********************************
ok: [192.168.56.114]

TASK [start httpd (CentOS)] *****************************************************
changed: [192.168.56.114]

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

TASK [Gathering Facts] **********************************************************
ok: [192.168.56.114]

TASK [install mariadb package (CentOS)] *****************************************
ok: [192.168.56.114]

TASK [install mariadb package (Ubuntu)] *****************************************
skipping: [192.168.56.114]
```



```
PLAY RECAP ***************************************************************************
192.168.56.111             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
192.168.56.112             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
192.168.56.114             : ok=8    changed=2    unreachable=0    failed=0    s
kipped=3    rescued=0    ignored=0
```


Activities   Firefox   Oct 3 17:57

HTTP Server Test Page

192.168.56.114

CentOS Blog Documentation Forums

CentOS

HTTP Server Test Page

This page is used to test the proper operation of the HTTP server after it has been installed. If you ca
page it means that this site is working properly. This server is powered by CentOS.

If you are a member of the general public:

The website you just visited is either experiencing problems or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain

If you are the website administrator:

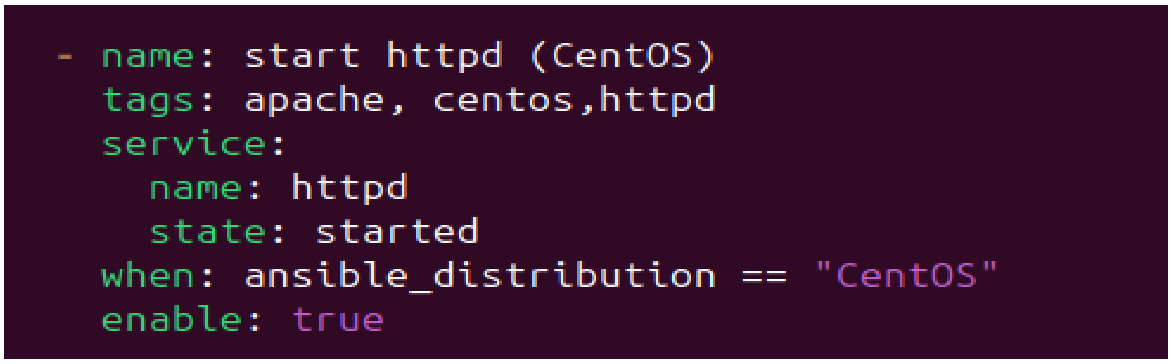
You may now add content to the webroot directory until you do so, people visiting your website will see and not your content.

For systems using the Apache HTTP Server:

You may now add content to the directory /var/www

Based on the result, when i first run the httpd server it is "unable to connect" because I stop it by the command systemctl status stop but after start the httpd and enter the IP address of my CentOS again it finally became successful.

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
  when: ansible_distribution == "CentOS"
  enable: true
```

Reflections:

Answer the following:

1. What is the importance of putting our remote servers into groups?
    - **Putting remote servers into groups is important because it allows administrators to organize and manage multiple servers more efficiently. You can apply specific configurations or tasks to the right set of machines without repeating code**
    -
2. What is the importance of tags in playbooks?

    - **Tags in playbooks are important because they give flexibility in running only specific parts of a playbook without executing the whole thing. This is useful for saving time, debugging, and focusing on particular tasks, especially when dealing with large and complex automation workflows.**

3. Why do think some services need to be managed automatically in playbooks?
    - **Some services need to be managed automatically in playbooks because manual management can be error-prone and time-consuming. Automating tasks such as starting, stopping, or restarting services ensures consistency, reliability, and faster response to system changes or failures.**