Name: Maxine Audrey D. Pulao	**Date Performed: October 4, 2022**
Course/Section: CPE31S2	**Date Submitted: October 8, 2022**
Instructor: Dr. Jonathan Taylar	**Semester and SY: 2022-2023**

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.

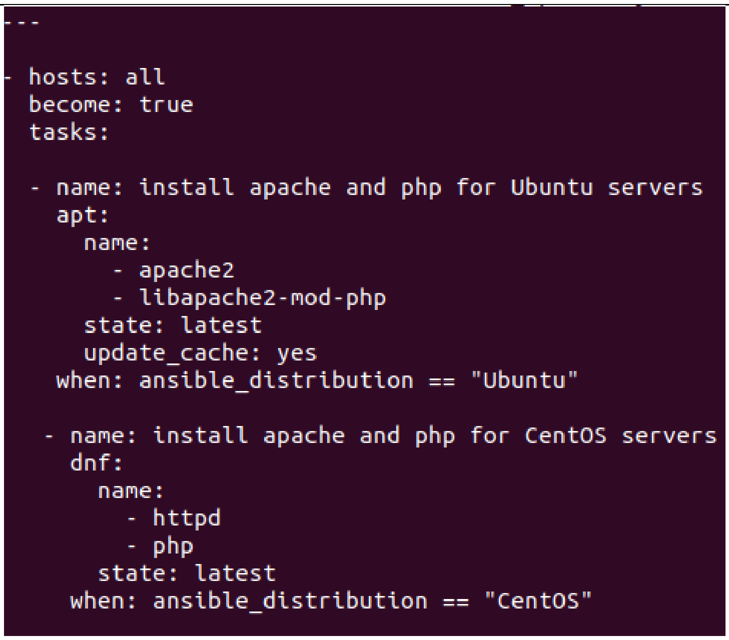
We also try to manage services that do not automatically run using the automations in playbook. For example, when we install web servers or httpd for CentOS, we notice that the service did not start automatically.

Requirement:

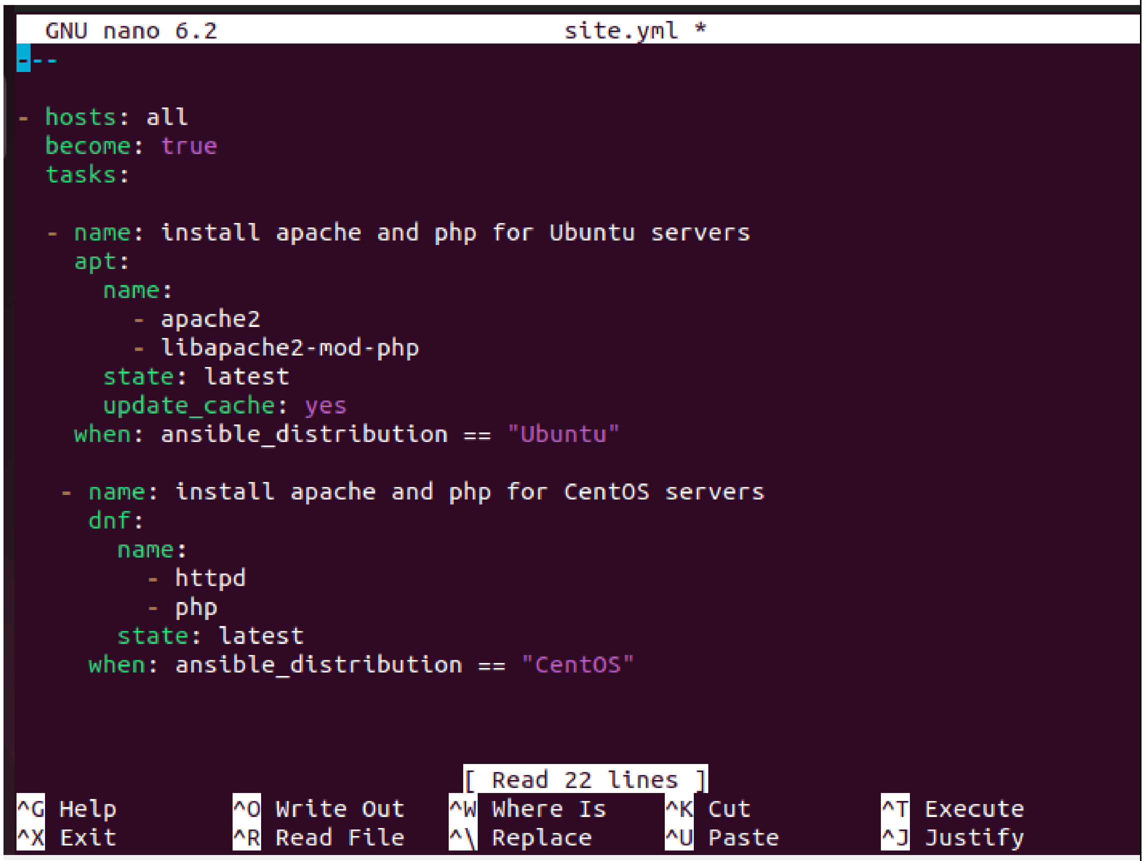
In this activity, you will need to create another Ubuntu VM and name it Server 3. Likewise, you need to activate the second adapter to a host-only adapter after the installations. Take note of the IP address of the Server 3. Make sure to use the command *ssh-copy-id* to copy the public key to Server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

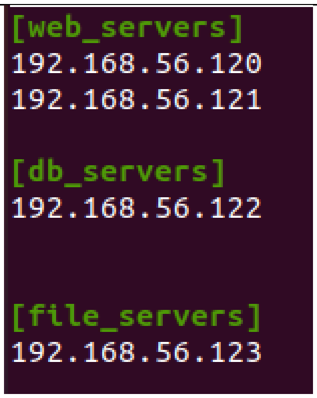


```
---

- 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"
```



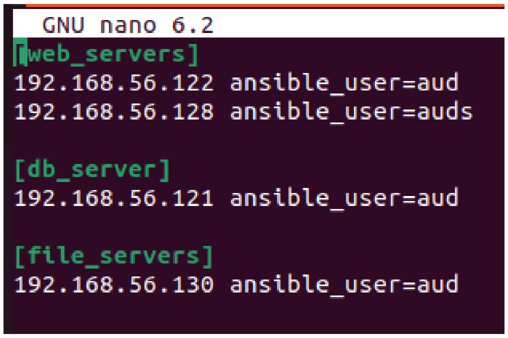
```
  GNU nano 6.2                       site.yml *
---

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


                               [ Read 22 lines ]
^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify
```

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
```

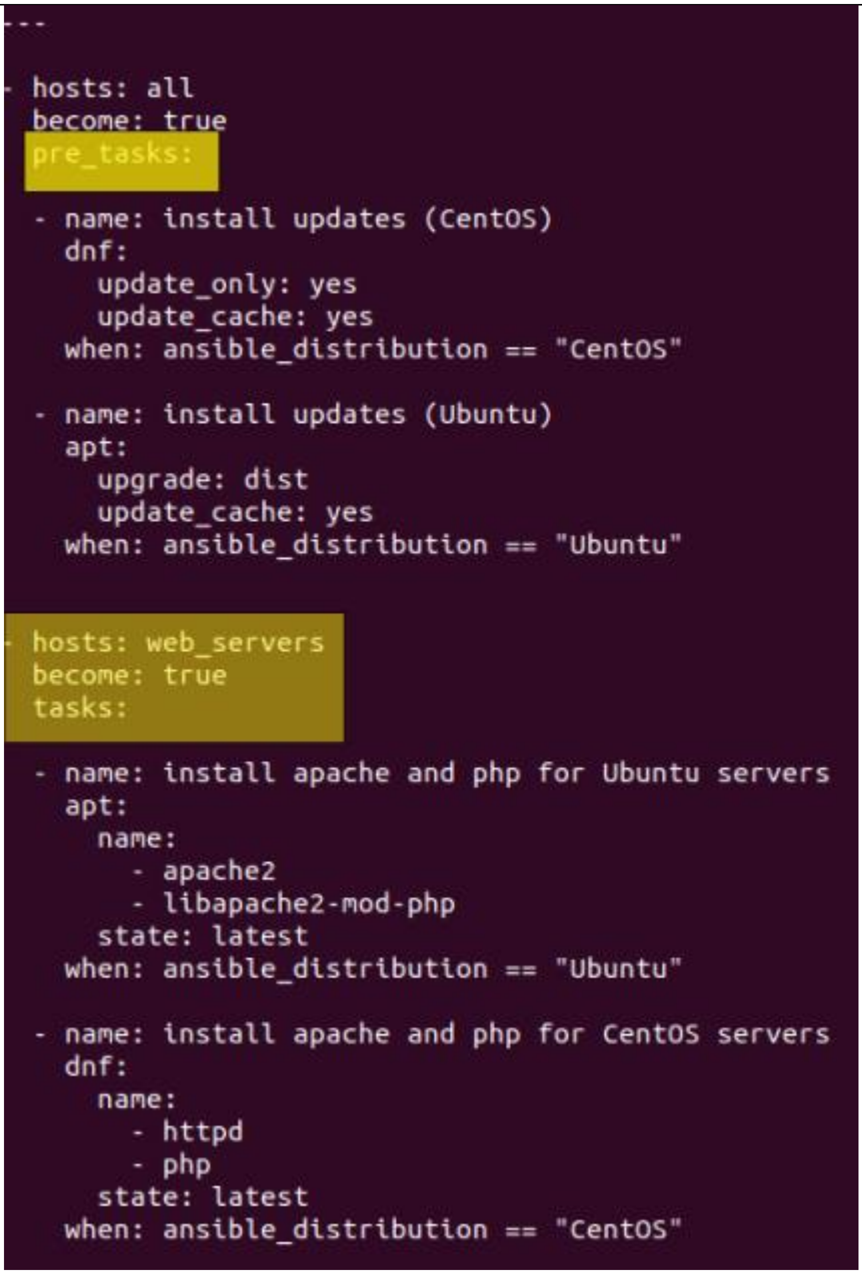
Make sure to save the file and exit.



```
  GNU nano 6.2
[web_servers]
192.168.56.122 ansible_user=aud
192.168.56.128 ansible_user=auds

[db_server]
192.168.56.121 ansible_user=aud

[file_servers]
192.168.56.130 ansible_user=aud
```

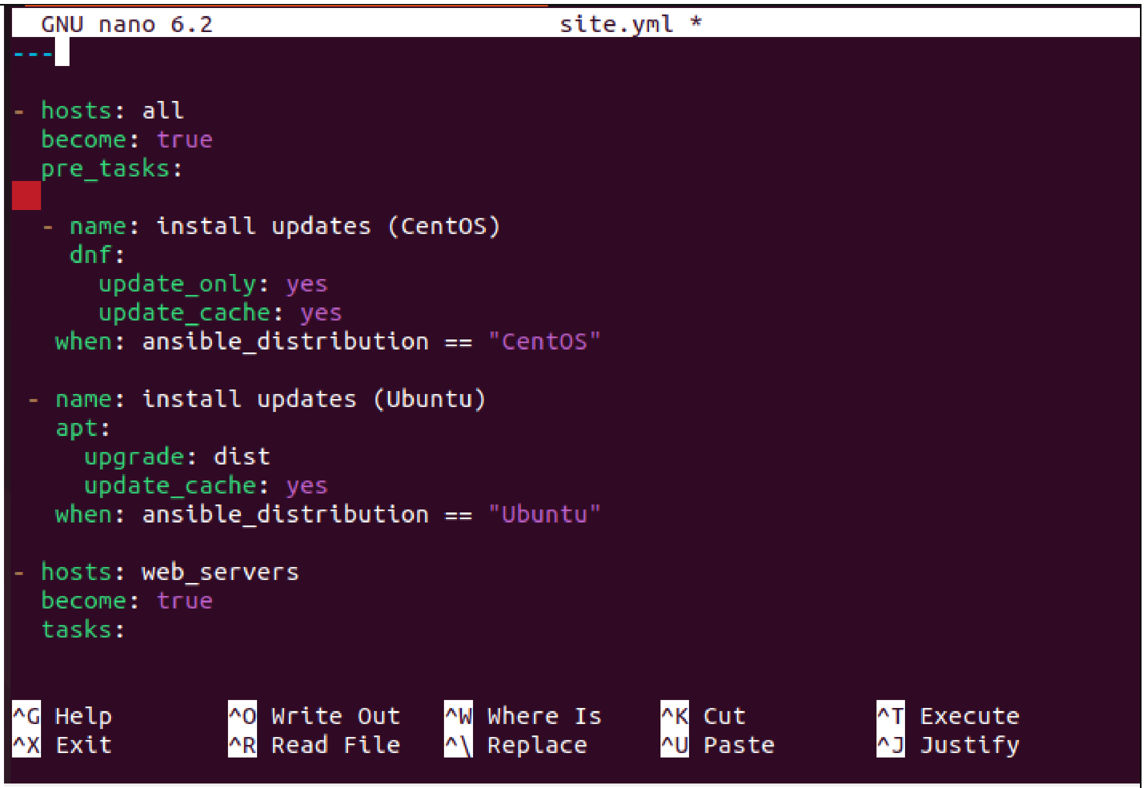
Right now, we have created groups in our inventory file and put each server in its own group. In other cases, you can have a server be a member of multiple groups, for example you have a test server that is also a web server.

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

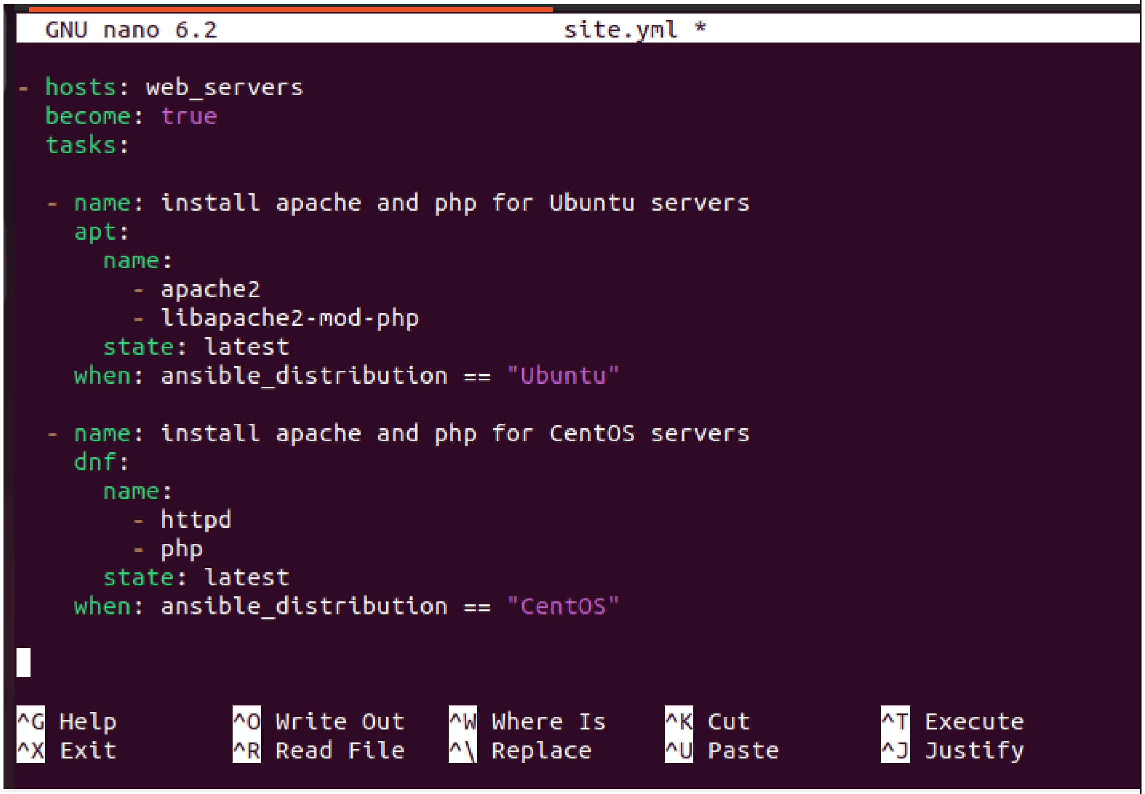


```
---

- 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.

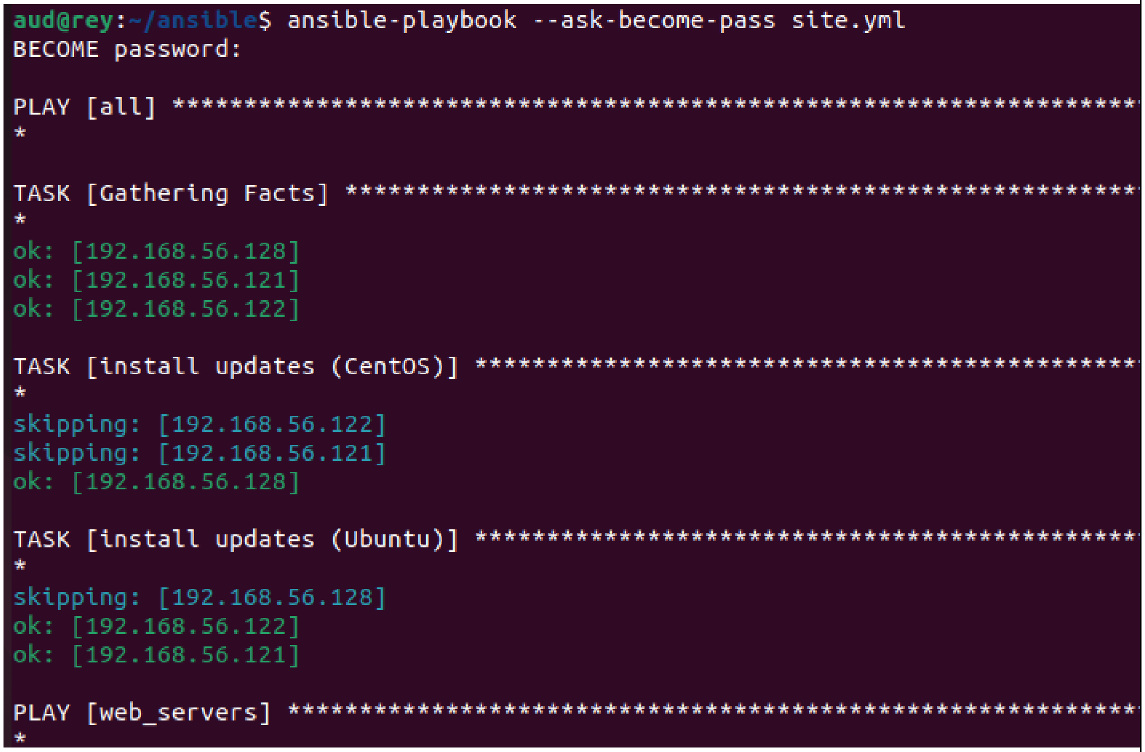


```
---

- 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:
```

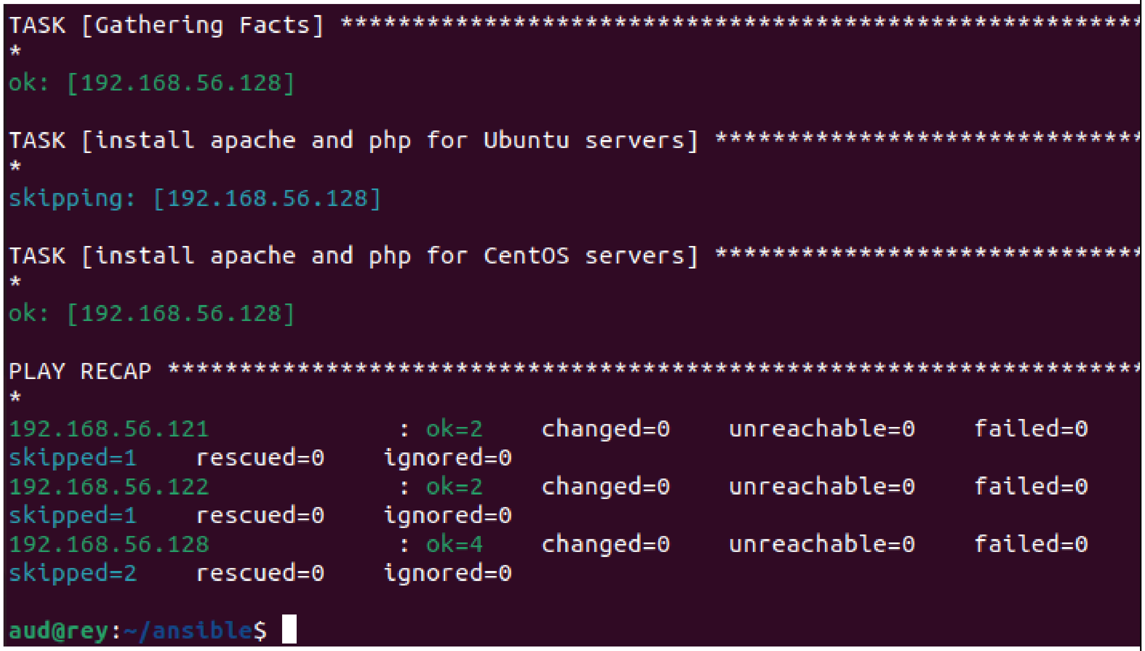


```
- hosts: web_servers
  become: true
  tasks:

  - 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.



```
aud@rey:~/ansible$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *******************************************************
*
ok: [192.168.56.128]
ok: [192.168.56.121]
ok: [192.168.56.122]

TASK [install updates (CentOS)] **********************************************
*
skipping: [192.168.56.122]
skipping: [192.168.56.121]
ok: [192.168.56.128]

TASK [install updates (Ubuntu)] **********************************************
*
skipping: [192.168.56.128]
ok: [192.168.56.122]
ok: [192.168.56.121]

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

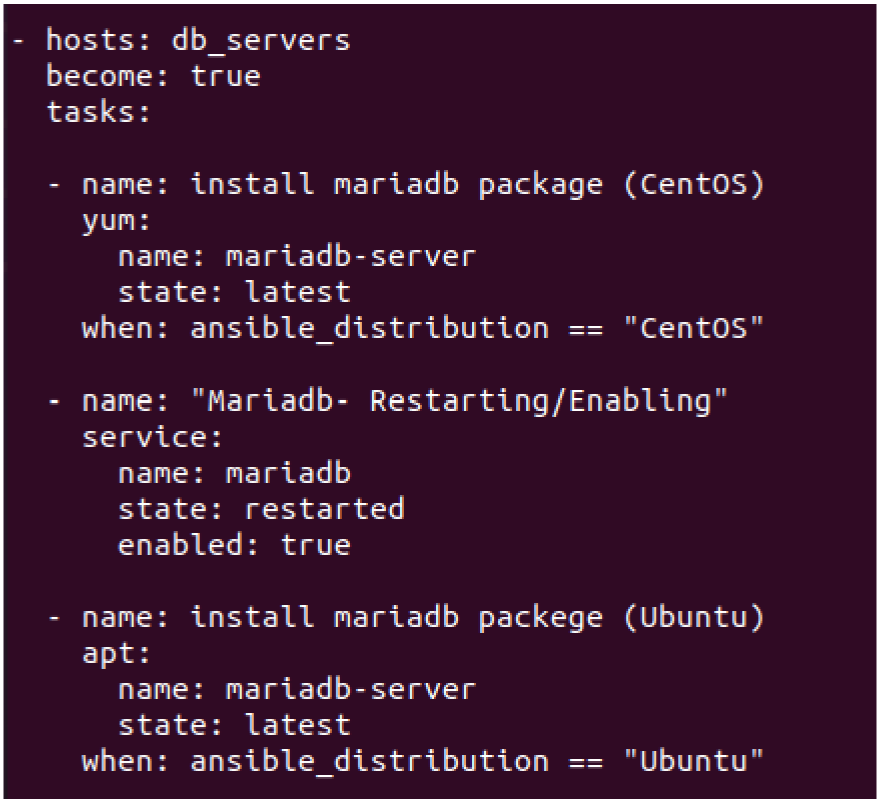


```
TASK [Gathering Facts] *******************************************************
*
ok: [192.168.56.128]

TASK [install apache and php for Ubuntu servers] *****************************
*
skipping: [192.168.56.128]

TASK [install apache and php for CentOS servers] *****************************
*
ok: [192.168.56.128]

PLAY RECAP *******************************************************************
*
192.168.56.121             : ok=2    changed=0    unreachable=0    failed=0    
skipped=1    rescued=0    ignored=0
192.168.56.122             : ok=2    changed=0    unreachable=0    failed=0    
skipped=1    rescued=0    ignored=0
192.168.56.128             : ok=4    changed=0    unreachable=0    failed=0    
skipped=2    rescued=0    ignored=0

aud@rey:~/ansible$ 
```

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

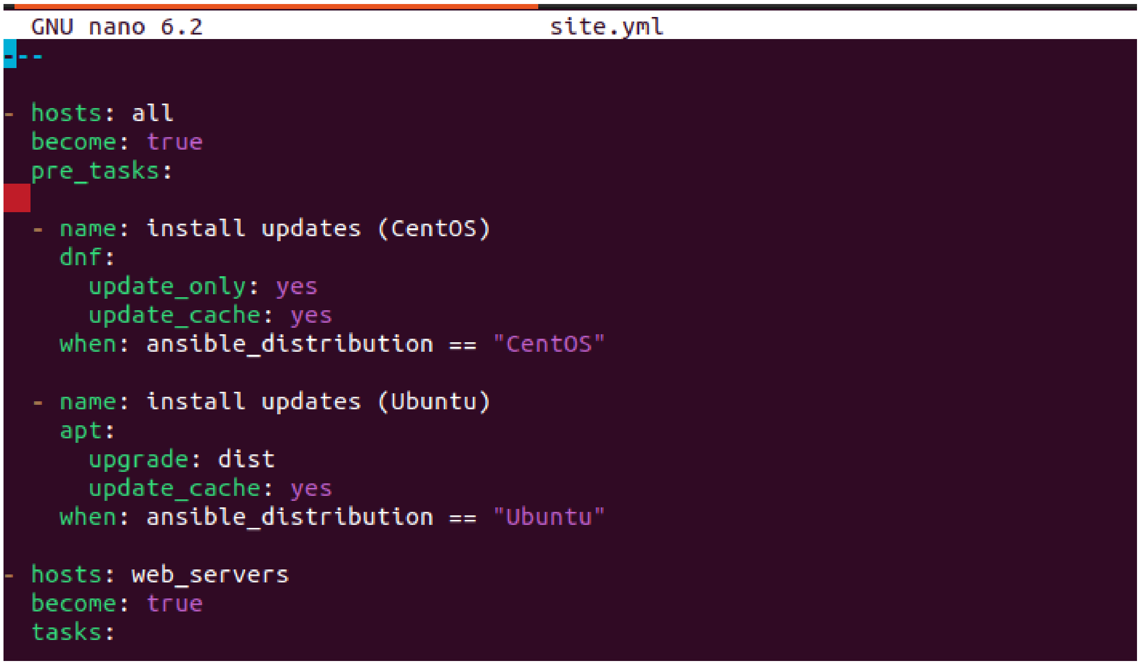
- After running the file, the PLAY segment appears right away when the code is encoded properly. However, the TASK segment takes awhile to appear especially if it involves updating and installing.

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

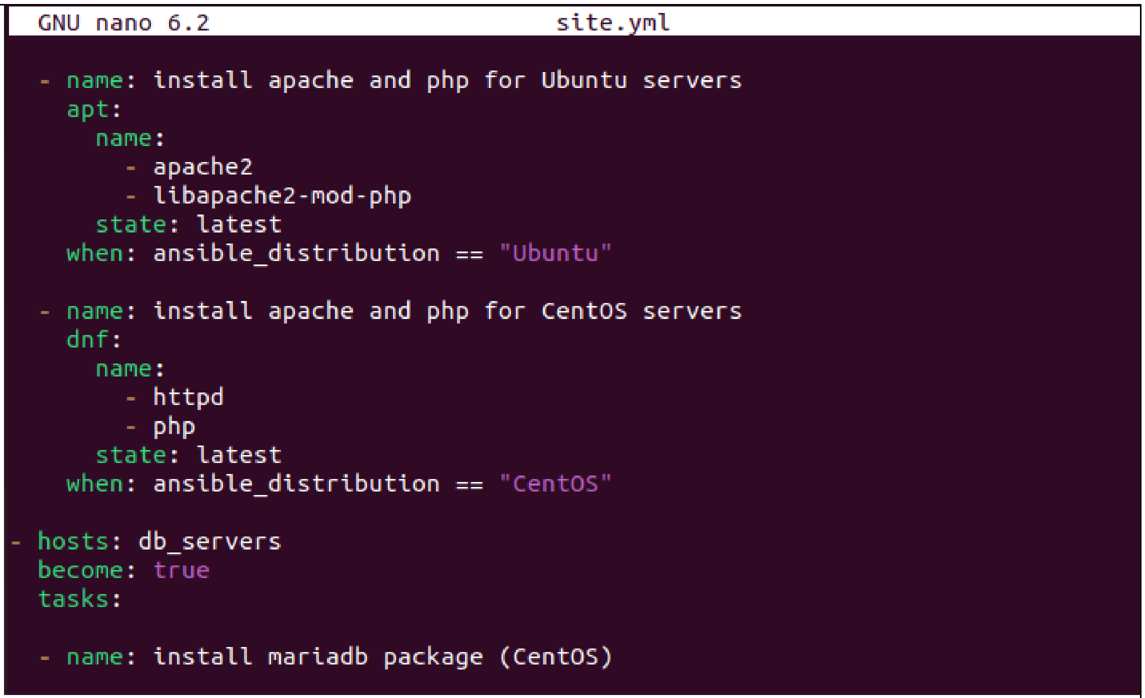


```
- hosts: db_servers
  become: true
  tasks:

  - name: install mariadb package (CentOS)
    yum:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "CentOS"

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

  - name: install mariadb packege (Ubuntu)
    apt:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "Ubuntu"
```

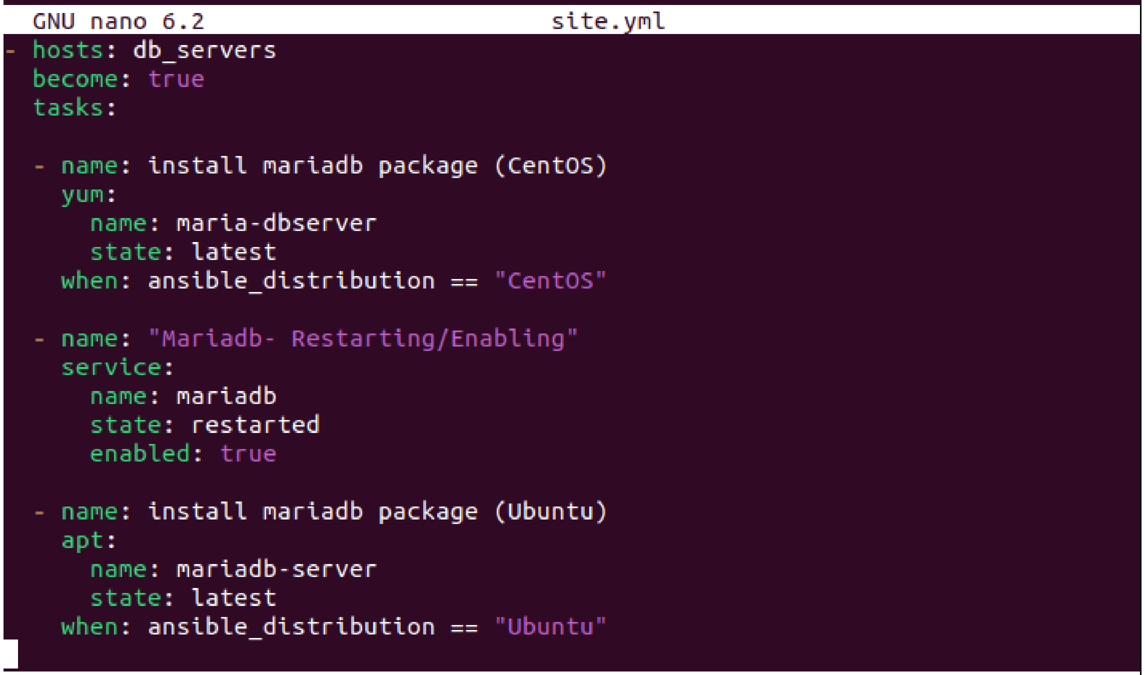
Make sure to save the file and exit.



```
  GNU nano 6.2                              site.yml
---

- 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:
```

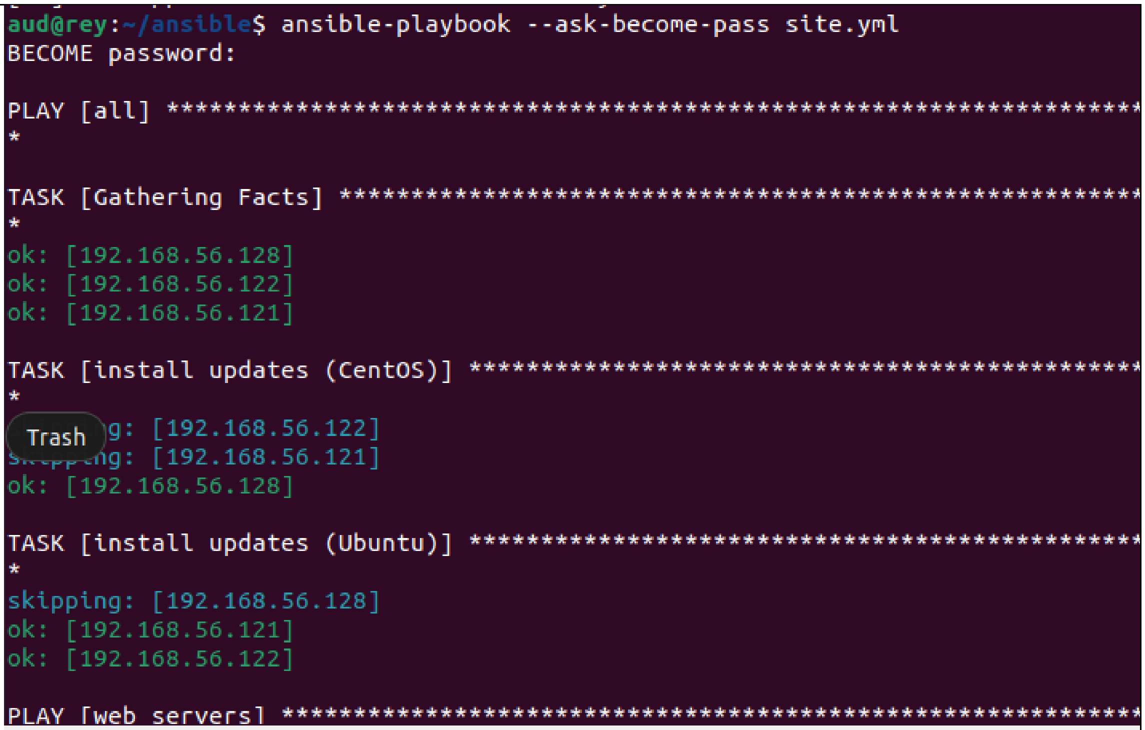


```
  GNU nano 6.2                                site.yml

  - 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"

- hosts: db_servers
  become: true
  tasks:

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

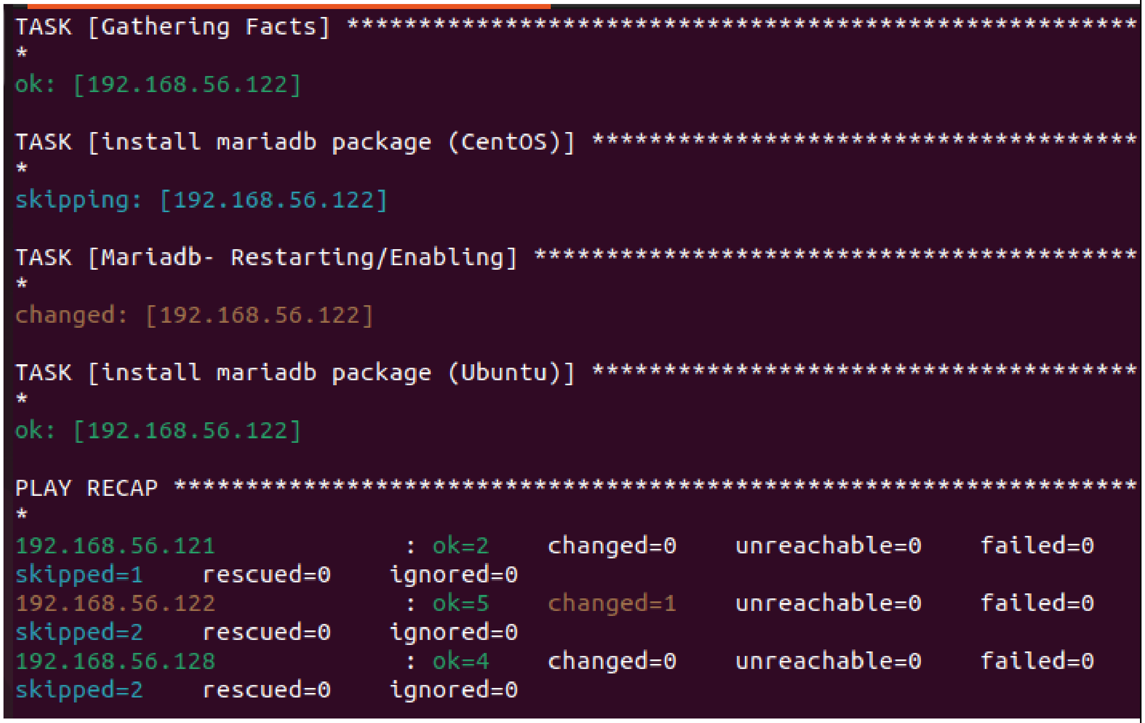


```
  GNU nano 6.2                                site.yml
- hosts: db_servers
  become: true
  tasks:

  - name: install mariadb package (CentOS)
    yum:
      name: maria-dbserver
      state: latest
    when: ansible_distribution == "CentOS"

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

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

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

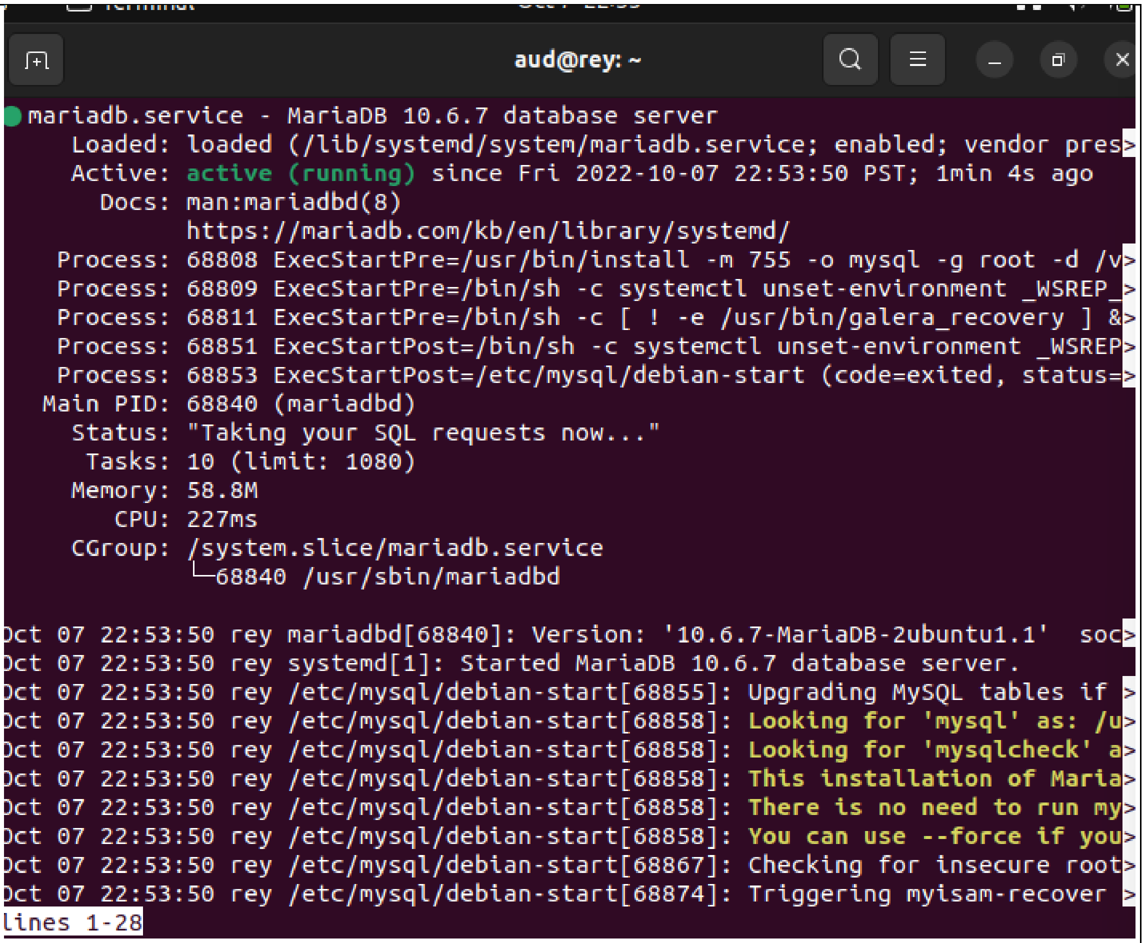


```
aud@rey:~/ansible$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *******************************************************
*
ok: [192.168.56.128]
ok: [192.168.56.122]
ok: [192.168.56.121]

TASK [install updates (CentOS)] **********************************************
*
skipping: [192.168.56.122]
skipping: [192.168.56.121]
ok: [192.168.56.128]

TASK [install updates (Ubuntu)] **********************************************
*
skipping: [192.168.56.128]
ok: [192.168.56.121]
ok: [192.168.56.122]

PLAY [web servers] ***********************************************************
```



```
TASK [Gathering Facts] *******************************************************
*
ok: [192.168.56.122]

TASK [install mariadb package (CentOS)] **************************************
*
skipping: [192.168.56.122]

TASK [Mariadb- Restarting/Enabling] ******************************************
*
changed: [192.168.56.122]

TASK [install mariadb package (Ubuntu)] **************************************
*
ok: [192.168.56.122]

PLAY RECAP *******************************************************************
*
192.168.56.121             : ok=2    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.122             : ok=5    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.128             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
```

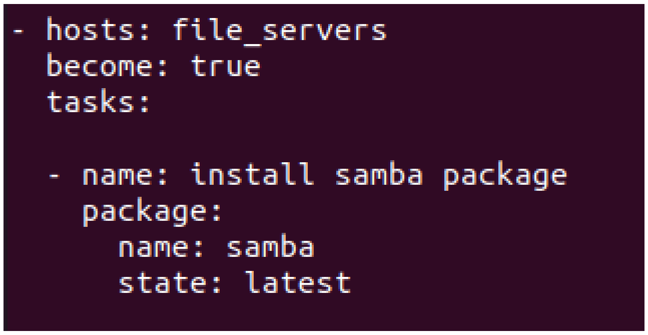
5. Go to the remote server (Ubuntu) terminal that belongs to the db_servers group and check the status for mariadb installation using the command: *systemctl status mariadb.* Do this on the CentOS server also.



```
● mariadb.service - MariaDB 10.6.7 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor pres>
     Active: active (running) since Fri 2022-10-07 22:53:50 PST; 1min 4s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 68808 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /v>
    Process: 68809 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_>
    Process: 68811 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] &>
    Process: 68851 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP>
    Process: 68853 ExecStartPost=/etc/mysql/debian-start (code=exited, status=>
   Main PID: 68840 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 10 (limit: 1080)
     Memory: 58.8M
        CPU: 227ms
     CGroup: /system.slice/mariadb.service
             └─68840 /usr/sbin/mariadbd

Oct 07 22:53:50 rey mariadbd[68840]: Version: '10.6.7-MariaDB-2ubuntu1.1'  soc>
Oct 07 22:53:50 rey systemd[1]: Started MariaDB 10.6.7 database server.
Oct 07 22:53:50 rey /etc/mysql/debian-start[68855]: Upgrading MySQL tables if >
Oct 07 22:53:50 rey /etc/mysql/debian-start[68858]: Looking for 'mysql' as: /u>
Oct 07 22:53:50 rey /etc/mysql/debian-start[68858]: Looking for 'mysqlcheck' a>
Oct 07 22:53:50 rey /etc/mysql/debian-start[68858]: This installation of Maria>
Oct 07 22:53:50 rey /etc/mysql/debian-start[68858]: There is no need to run my>
Oct 07 22:53:50 rey /etc/mysql/debian-start[68858]: You can use --force if you>
Oct 07 22:53:50 rey /etc/mysql/debian-start[68867]: Checking for insecure root>
Oct 07 22:53:50 rey /etc/mysql/debian-start[68874]: Triggering myisam-recover >
lines 1-28
```

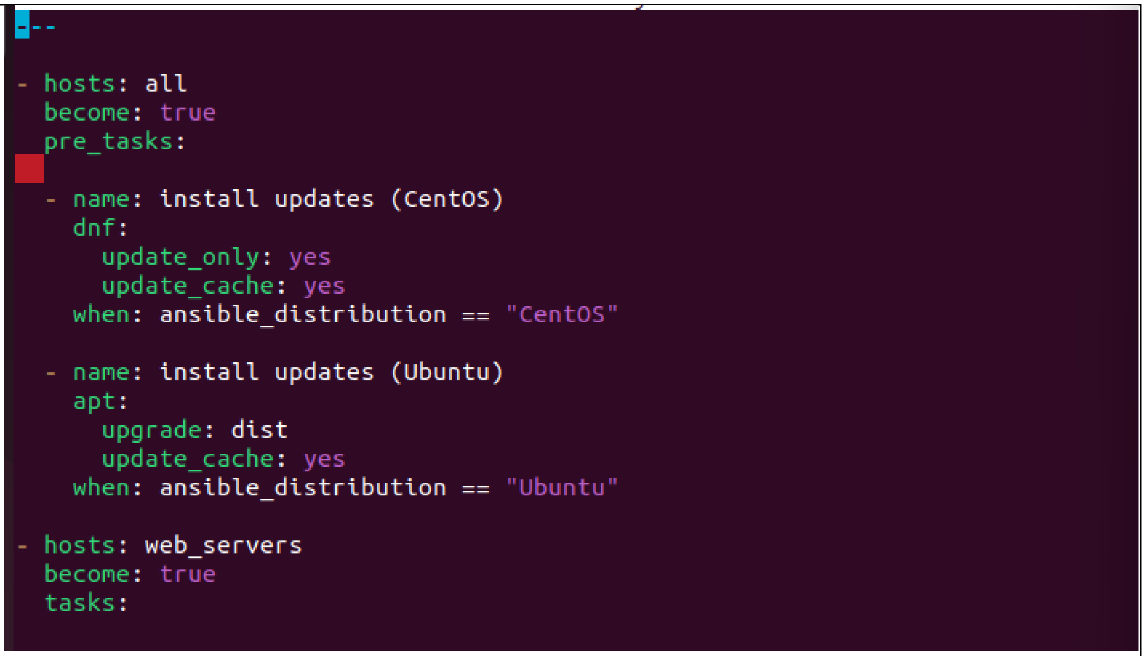
Describe the output.

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.

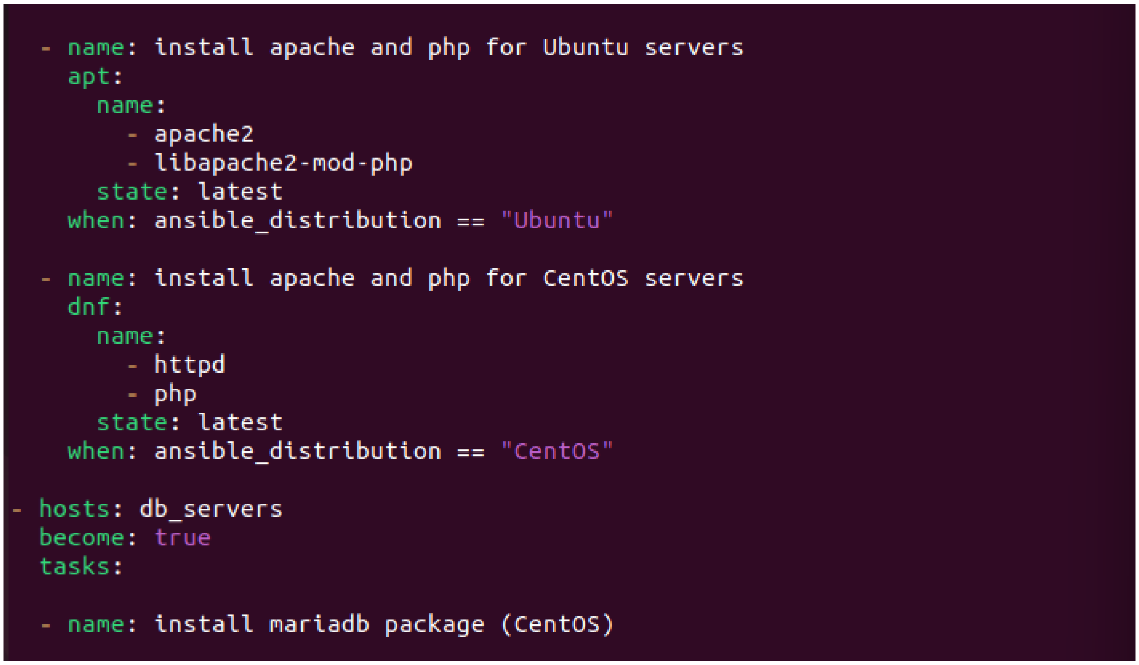


```
- hosts: file_servers
  become: true
  tasks:

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

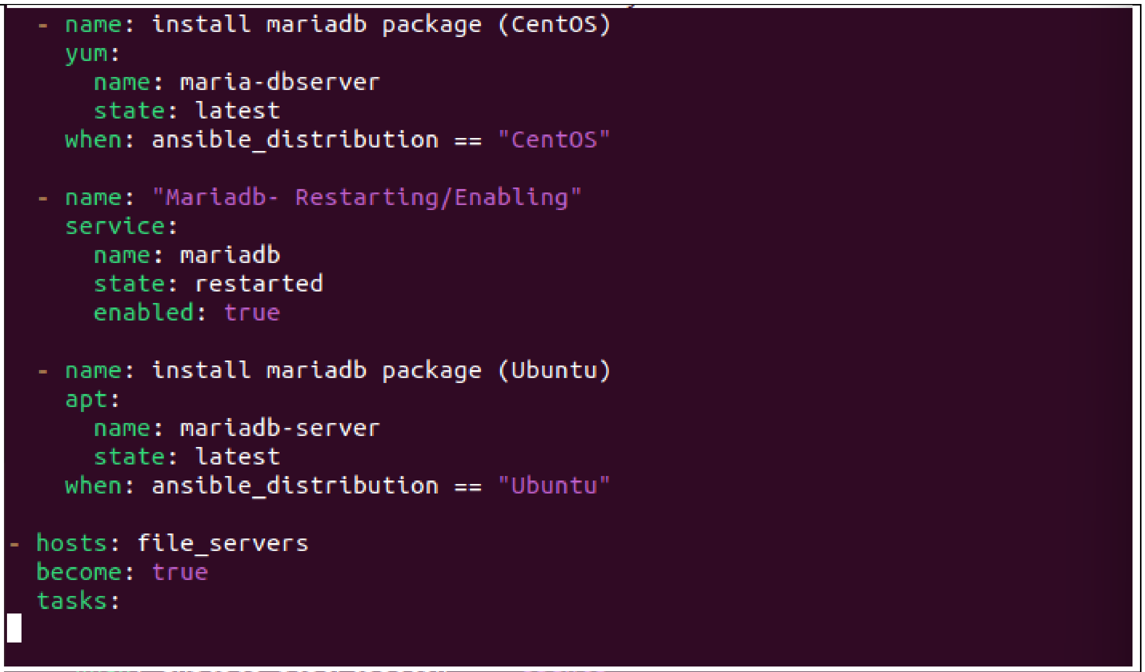
Make sure to save the file and exit.



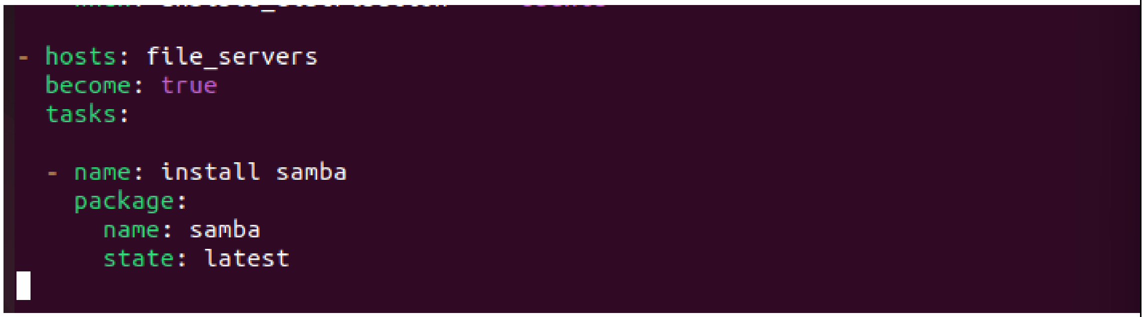
```
---

- 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"

- hosts: db_servers
  become: true
  tasks:

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

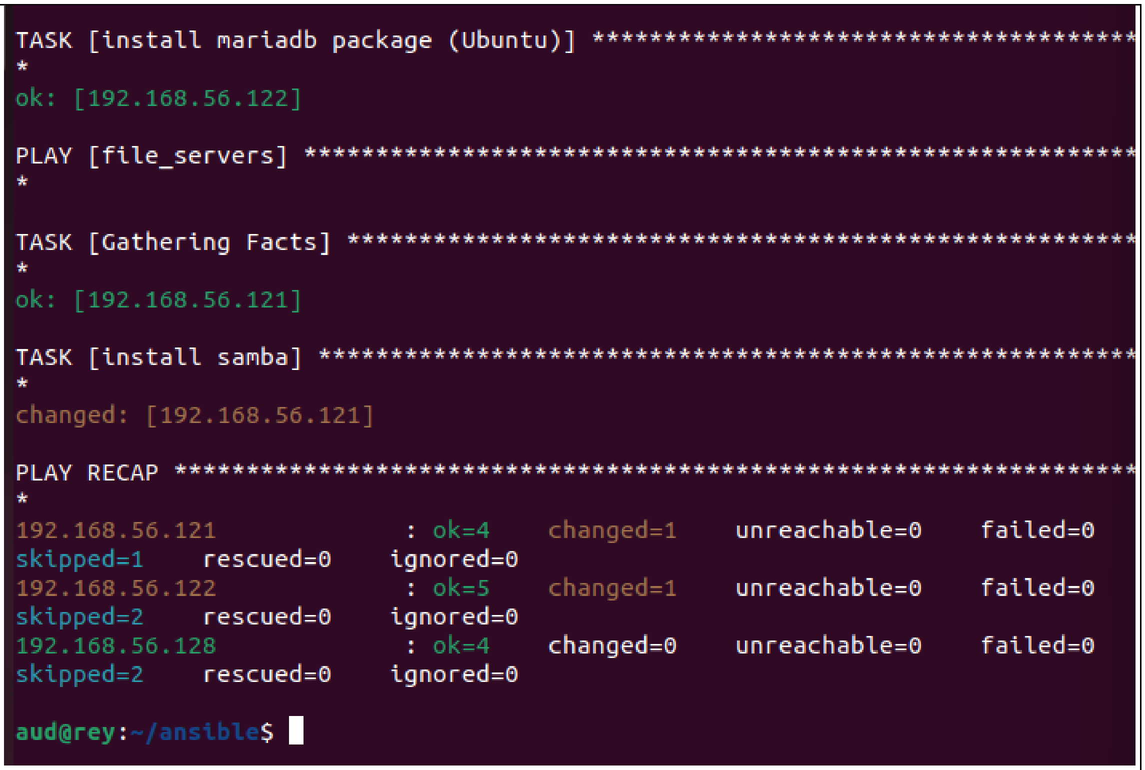


```
  - name: install mariadb package (CentOS)
    yum:
      name: maria-dbserver
      state: latest
    when: ansible_distribution == "CentOS"

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

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

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



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

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

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



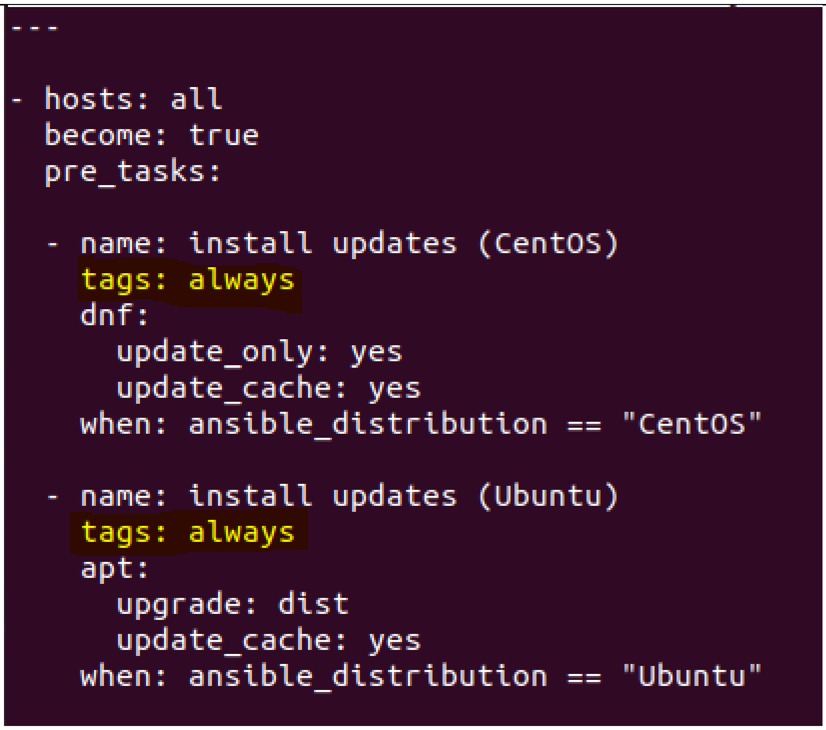
```
TASK [install mariadb package (Ubuntu)] ***************************************
*
ok: [192.168.56.122]

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

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

TASK [install samba] **********************************************************
*
changed: [192.168.56.121]

PLAY RECAP ********************************************************************
*
192.168.56.121             : ok=4    changed=1    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.122             : ok=5    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.128             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0

aud@rey:~/ansible$
```

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.

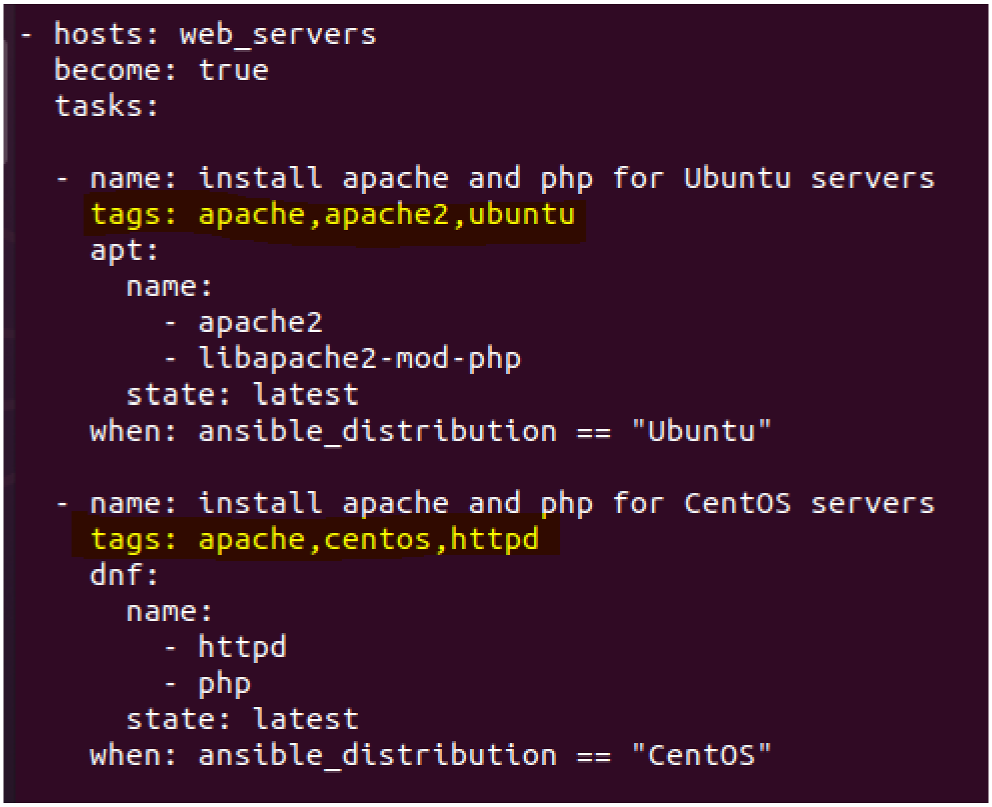
Task 2: Using Tags in running playbooks

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.

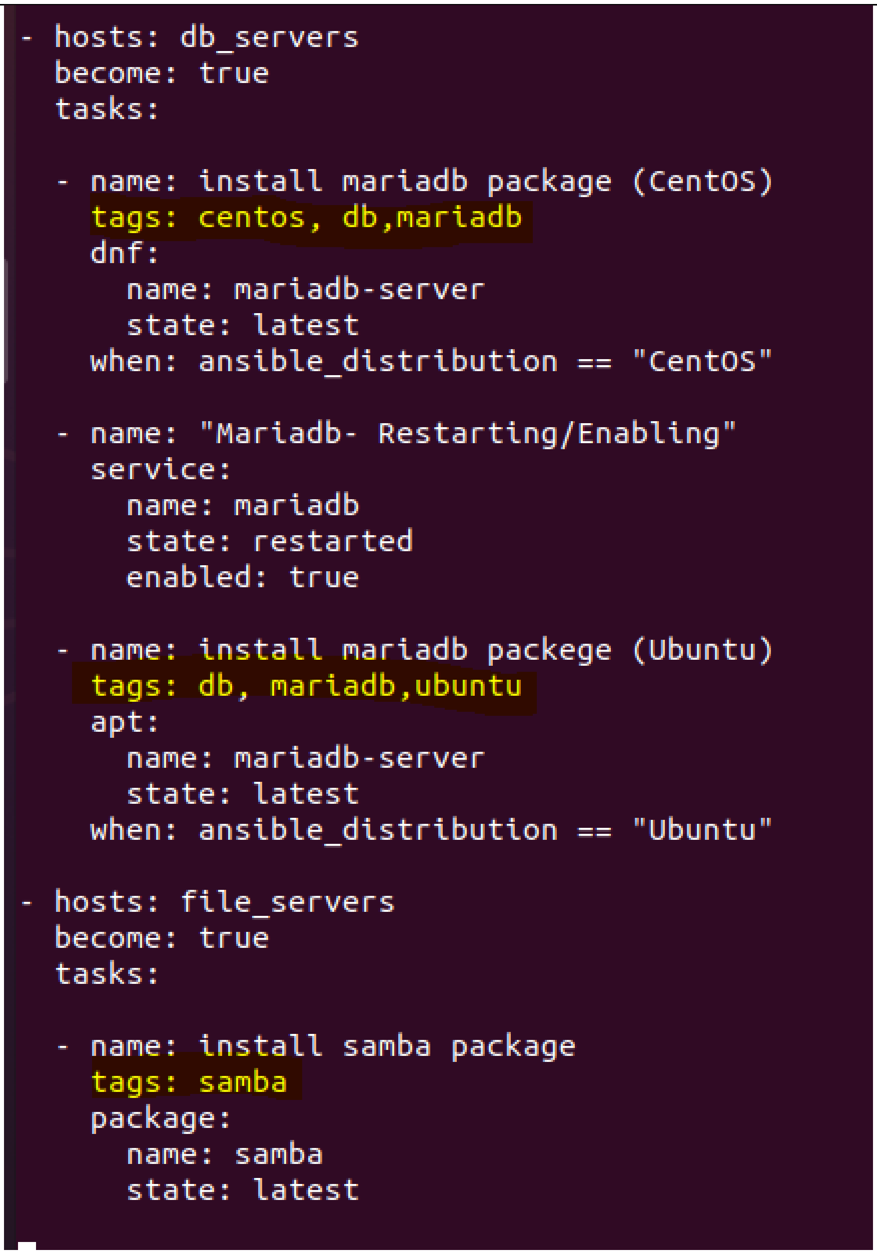
1. Edit the *site.yml* file. Add tags to the playbook. After the name, we can place the tags: *name_of_tag*. This is an arbitrary command, which means you can use any name for a tag.



```
---

- 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"
```

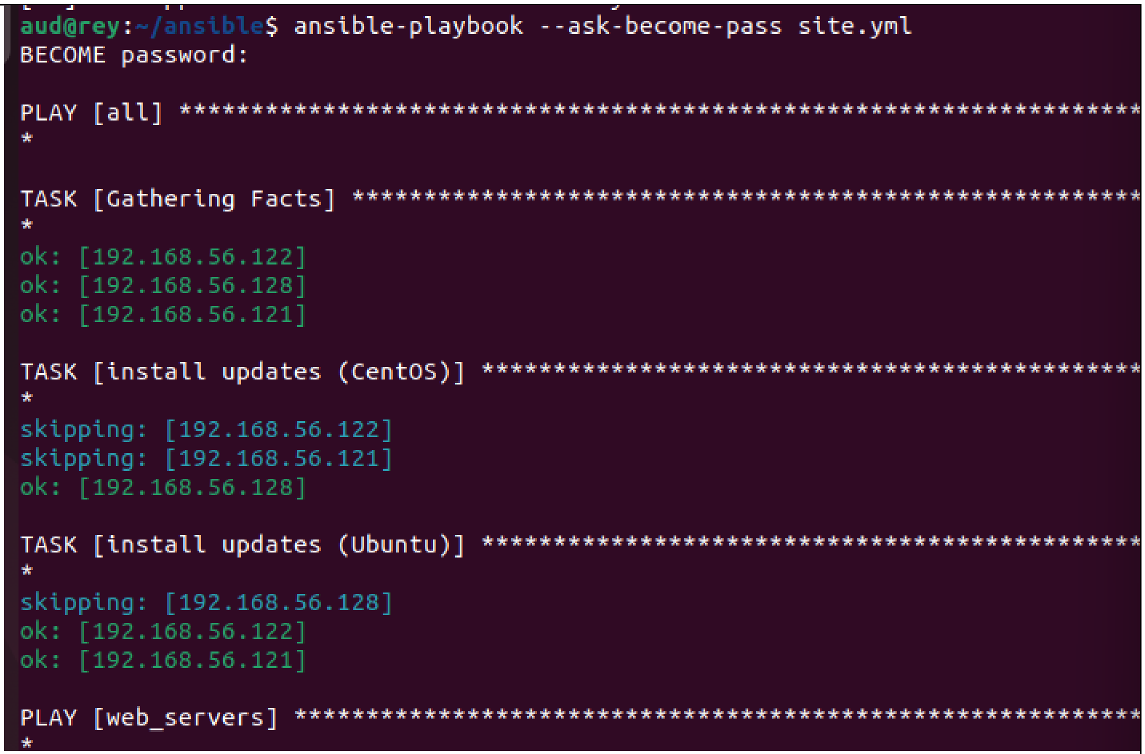


```
- hosts: web_servers
  become: true
  tasks:

  - 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"
```

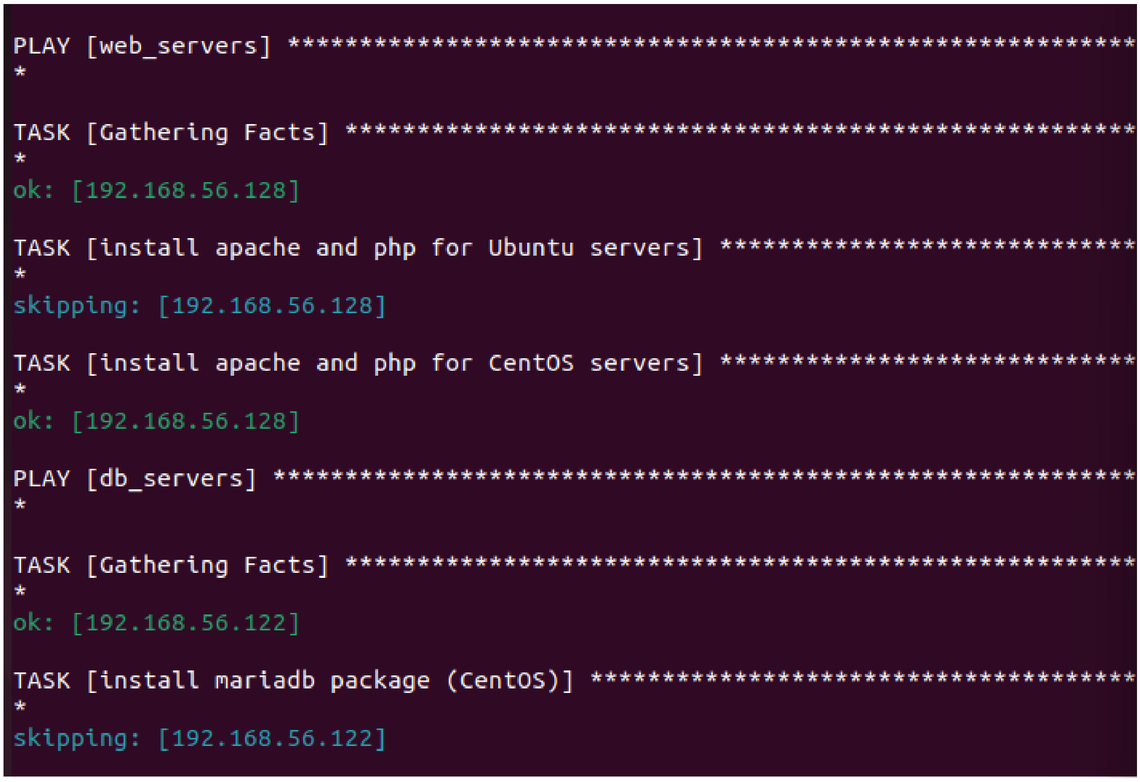


```
- hosts: db_servers
  become: true
  tasks:

  - 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
```

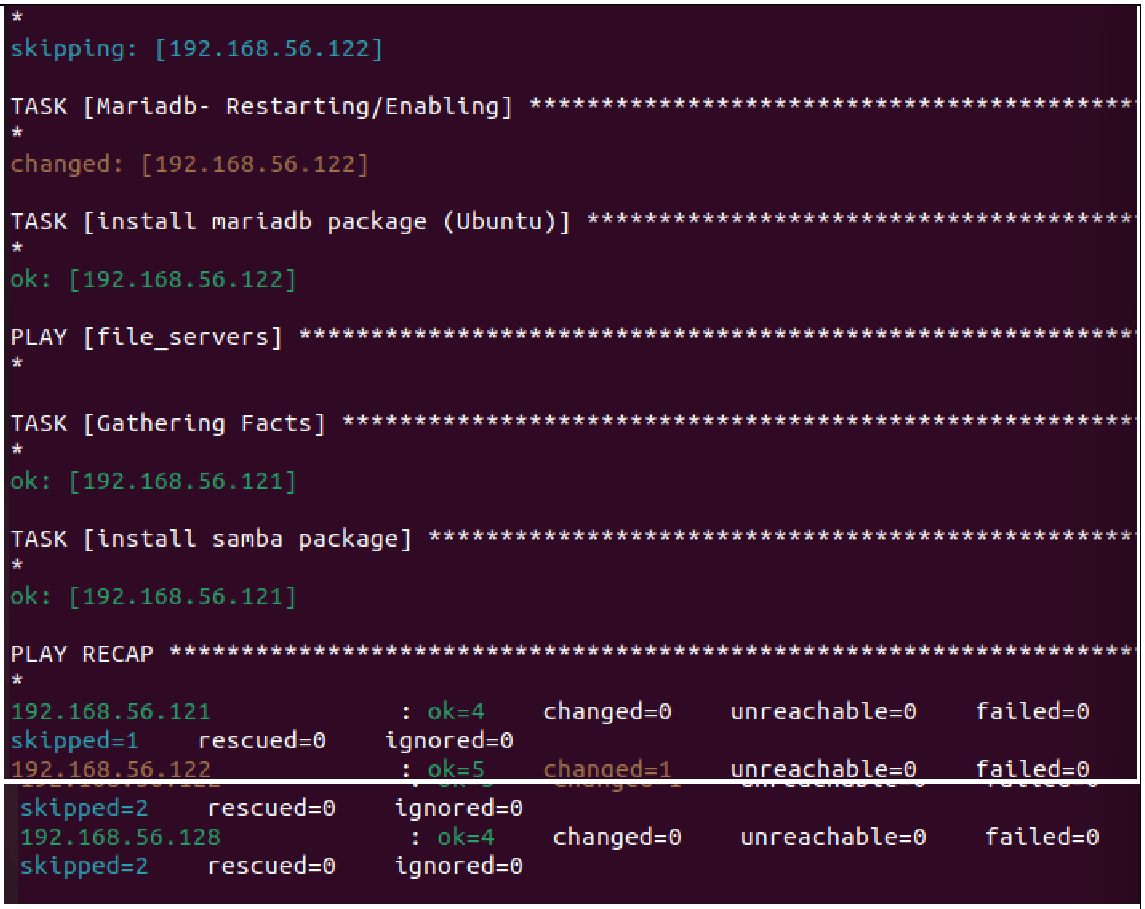
Make sure to save the file and exit.
Run the *site.yml* file and describe the result.



```
aud@rey:~/ansible$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *******************************************************
*
ok: [192.168.56.122]
ok: [192.168.56.128]
ok: [192.168.56.121]

TASK [install updates (CentOS)] **********************************************
*
skipping: [192.168.56.122]
skipping: [192.168.56.121]
ok: [192.168.56.128]

TASK [install updates (Ubuntu)] **********************************************
*
skipping: [192.168.56.128]
ok: [192.168.56.122]
ok: [192.168.56.121]

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



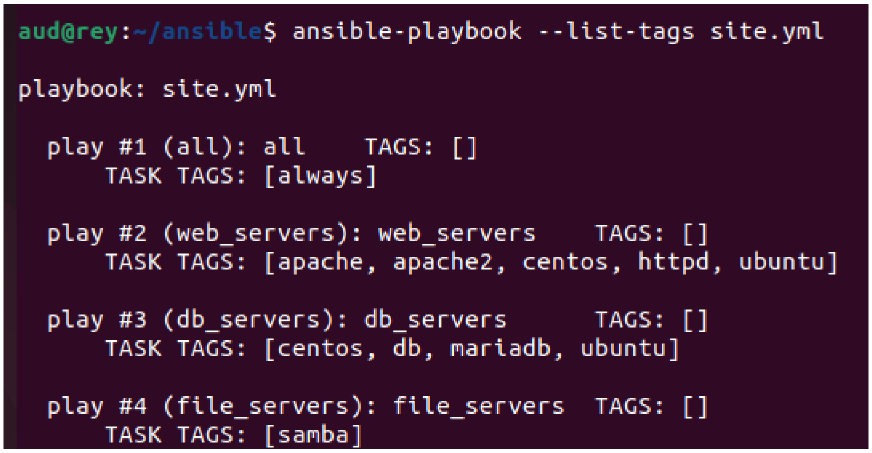
```
PLAY [web_servers] ***********************************************************
*

TASK [Gathering Facts] *******************************************************
*
ok: [192.168.56.128]

TASK [install apache and php for Ubuntu servers] *****************************
*
skipping: [192.168.56.128]

TASK [install apache and php for CentOS servers] *****************************
*
ok: [192.168.56.128]

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

TASK [Gathering Facts] *******************************************************
*
ok: [192.168.56.122]

TASK [install mariadb package (CentOS)] **************************************
*
skipping: [192.168.56.122]
```



```
*
skipping: [192.168.56.122]

TASK [Mariadb- Restarting/Enabling] ******************************************
*
changed: [192.168.56.122]

TASK [install mariadb package (Ubuntu)] **************************************
*
ok: [192.168.56.122]

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

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

TASK [install samba package] *************************************************
*
ok: [192.168.56.121]

PLAY RECAP *******************************************************************
*
192.168.56.121             : ok=4    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.122             : ok=5    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.128             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
```

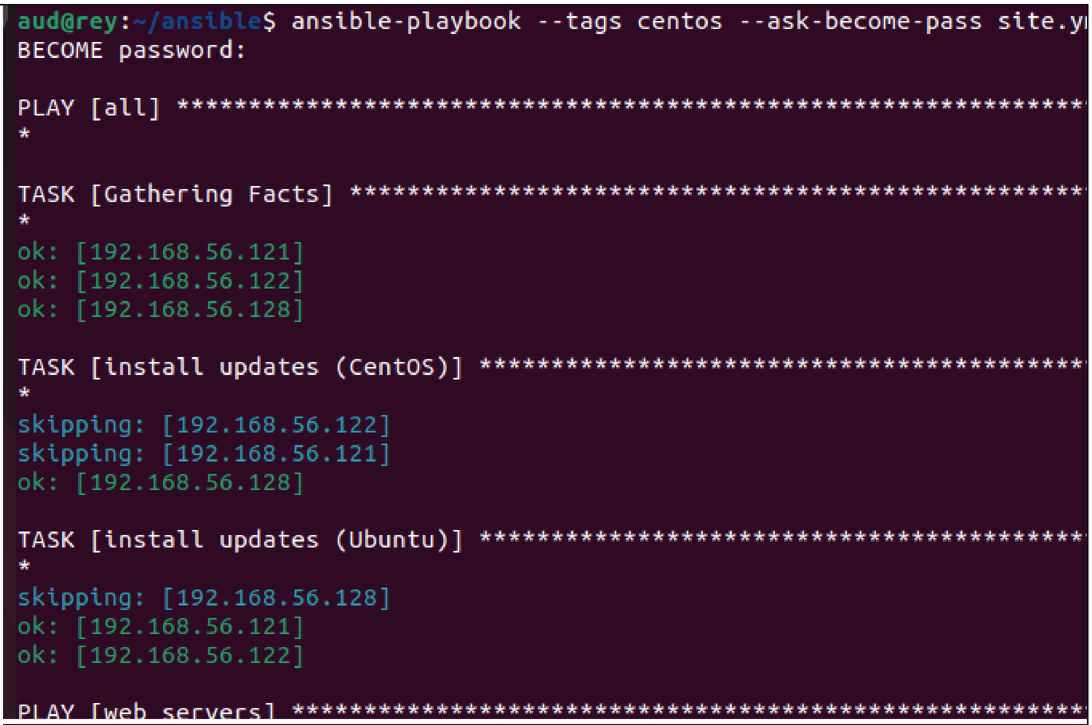
2. On the local machine, try to issue the following commands and describe each result:

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

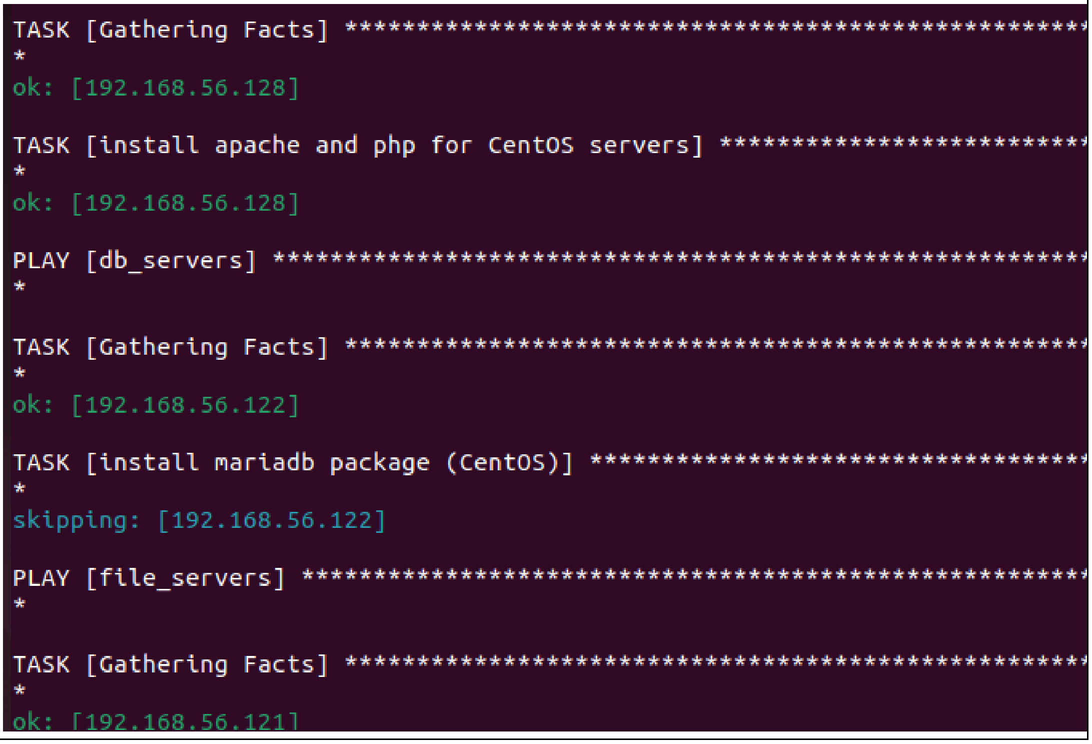


```
aud@rey:~/ansible$ ansible-playbook --list-tags site.yml

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]
```

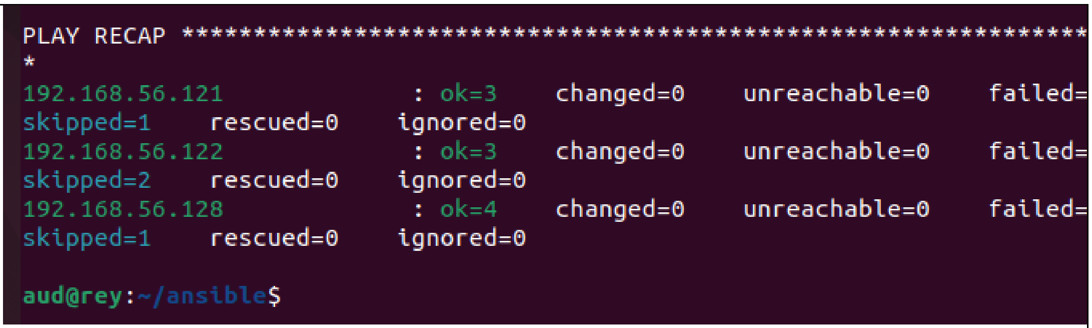
2.2 *ansible-playbook --tags centos --ask-become-pass site.yml*



```
aud@rey:~/ansible$ ansible-playbook --tags centos --ask-become-pass site.y
BECOME password:

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

TASK [Gathering Facts] ***************************************************
*
ok: [192.168.56.121]
ok: [192.168.56.122]
ok: [192.168.56.128]

TASK [install updates (CentOS)] ******************************************
*
skipping: [192.168.56.122]
skipping: [192.168.56.121]
ok: [192.168.56.128]

TASK [install updates (Ubuntu)] ******************************************
*
skipping: [192.168.56.128]
ok: [192.168.56.121]
ok: [192.168.56.122]

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

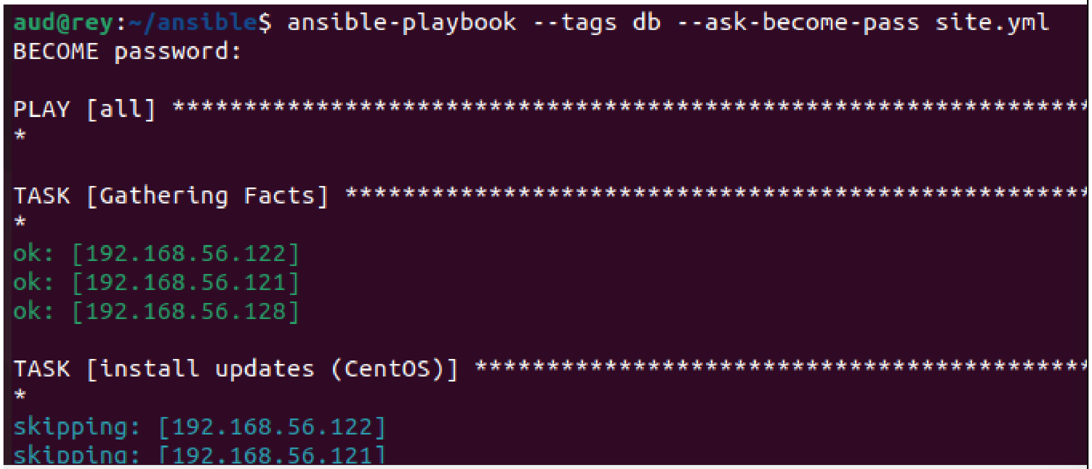


```
TASK [Gathering Facts] ***************************************************
*
ok: [192.168.56.128]

TASK [install apache and php for CentOS servers] *************************
*
ok: [192.168.56.128]

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

TASK [Gathering Facts] ***************************************************
*
ok: [192.168.56.122]

TASK [install mariadb package (CentOS)] **********************************
*
skipping: [192.168.56.122]

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

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

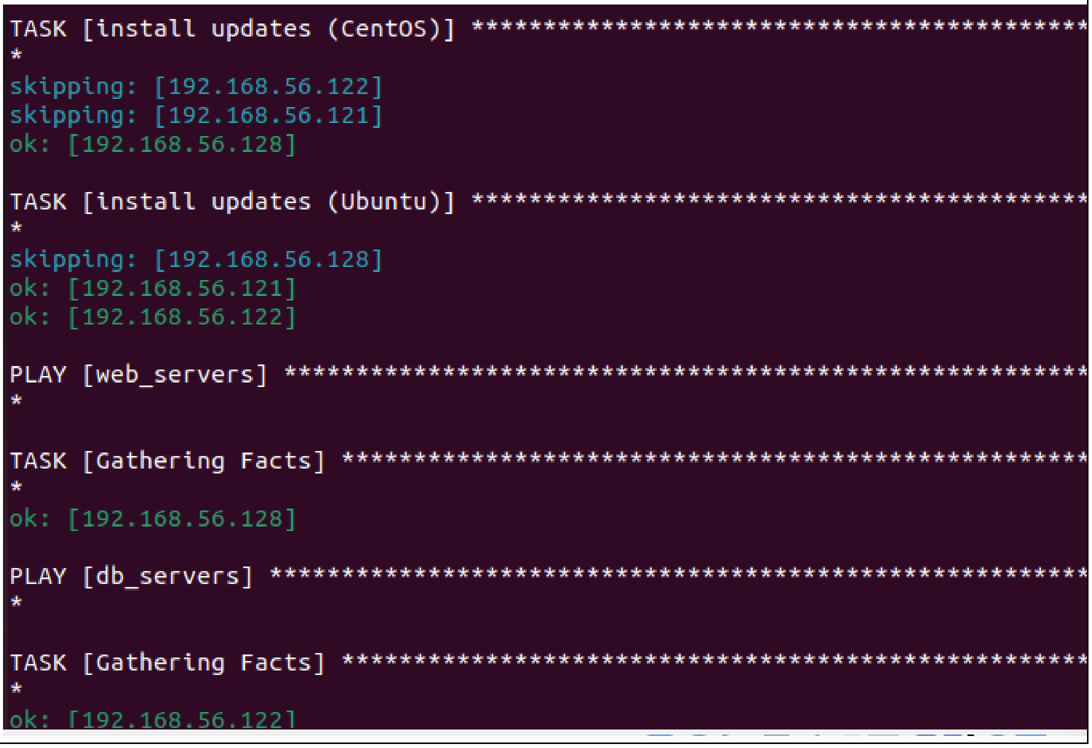


```
PLAY RECAP ***********************************************************************
*
192.168.56.121             : ok=3    changed=0    unreachable=0    failed=
skipped=1     rescued=0    ignored=0
192.168.56.122             : ok=3    changed=0    unreachable=0    failed=
skipped=2     rescued=0    ignored=0
192.168.56.128             : ok=4    changed=0    unreachable=0    failed=
skipped=1     rescued=0    ignored=0

aud@rey:~/ansible$
```

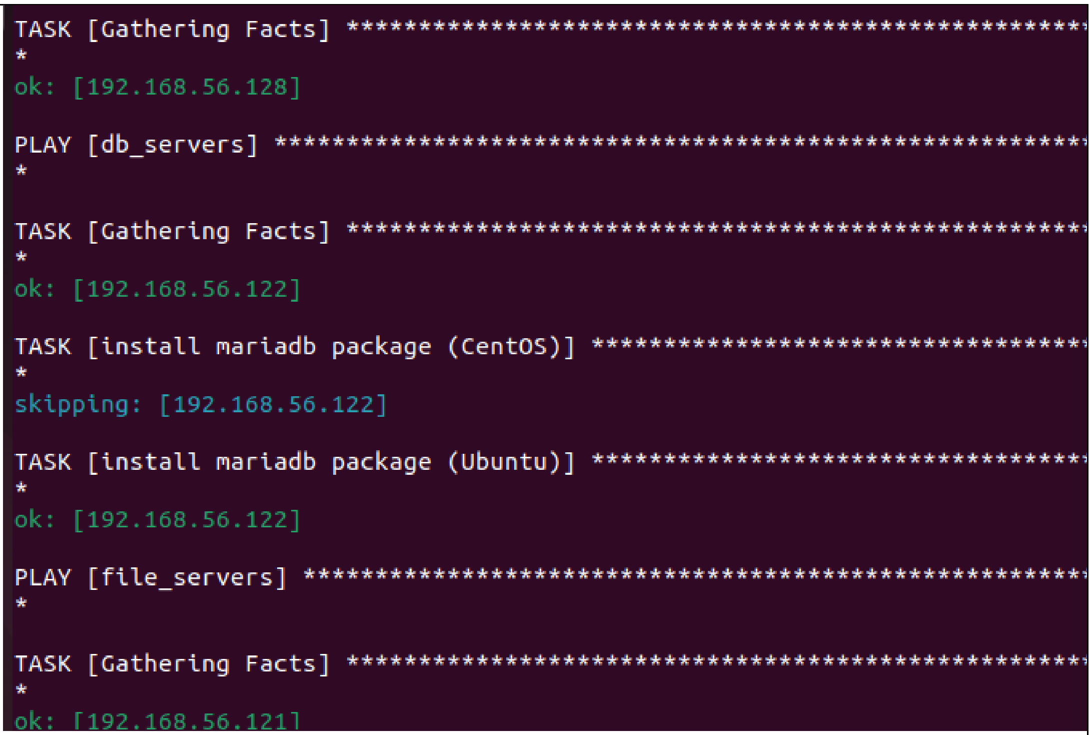
2.3 *ansible-playbook --tags db --ask-become-pass site.yml*



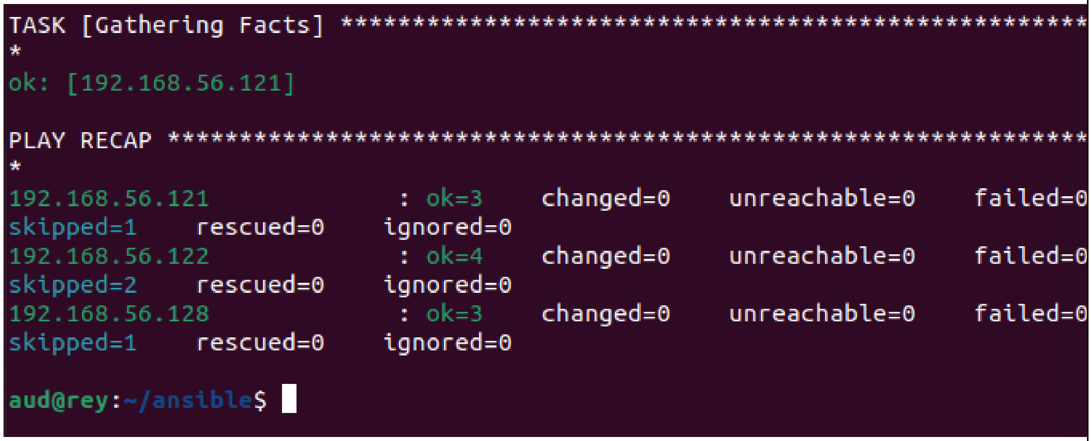
```
aud@rey:~/ansible$ ansible-playbook --tags db --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] ***********************************************************
*
ok: [192.168.56.122]
ok: [192.168.56.121]
ok: [192.168.56.128]

TASK [install updates (CentOS)] **************************************************
*
skipping: [192.168.56.122]
skipping: [192.168.56.121]
```



```
TASK [install updates (CentOS)] **************************************************
*
skipping: [192.168.56.122]
skipping: [192.168.56.121]
ok: [192.168.56.128]

TASK [install updates (Ubuntu)] **************************************************
*
skipping: [192.168.56.128]
ok: [192.168.56.121]
ok: [192.168.56.122]

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

TASK [Gathering Facts] ***********************************************************
*
ok: [192.168.56.128]

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

TASK [Gathering Facts] ***********************************************************
*
ok: [192.168.56.122]
```

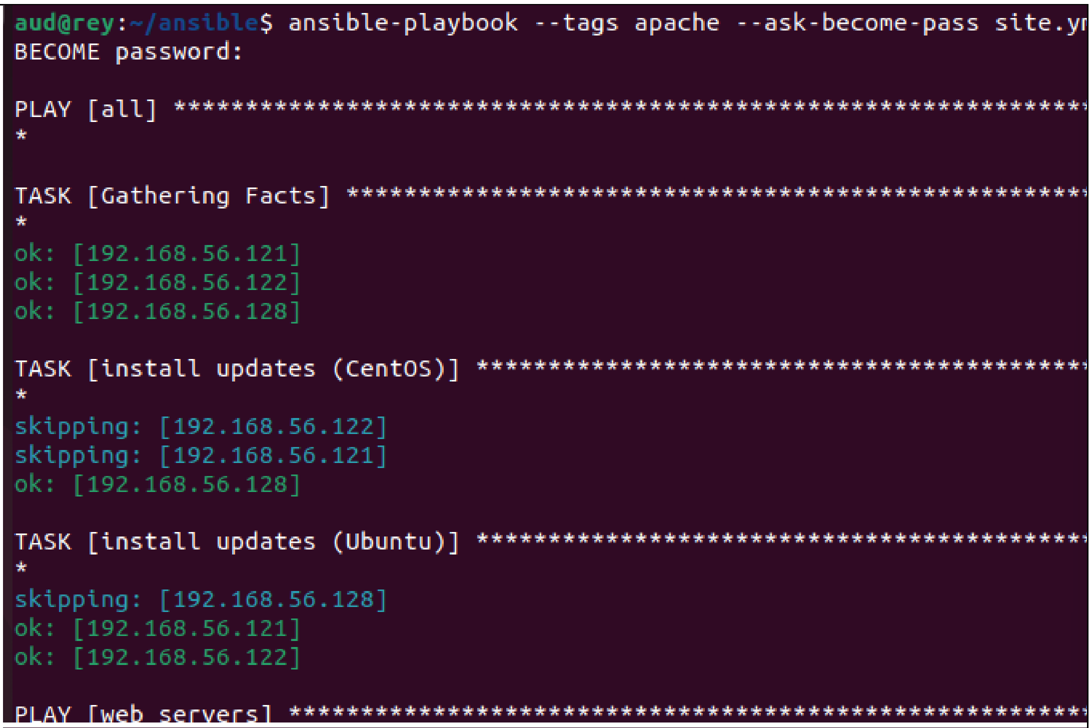


```
TASK [Gathering Facts] ***************************************************
*
ok: [192.168.56.128]

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

TASK [Gathering Facts] ***************************************************
*
ok: [192.168.56.122]

TASK [install mariadb package (CentOS)] **********************************
*
skipping: [192.168.56.122]

TASK [install mariadb package (Ubuntu)] **********************************
*
ok: [192.168.56.122]

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

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

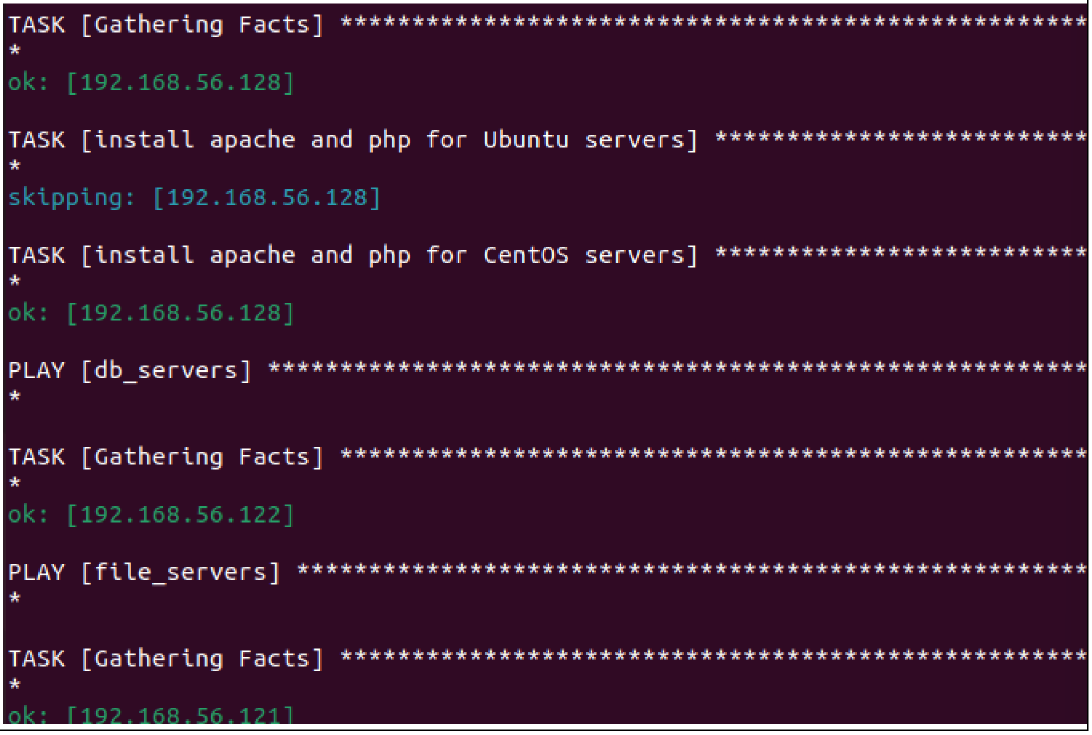


```
TASK [Gathering Facts] ***************************************************
*
ok: [192.168.56.121]

PLAY RECAP ***************************************************************
*
192.168.56.121             : ok=3    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.122             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.128             : ok=3    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0

aud@rey:~/ansible$ 
```

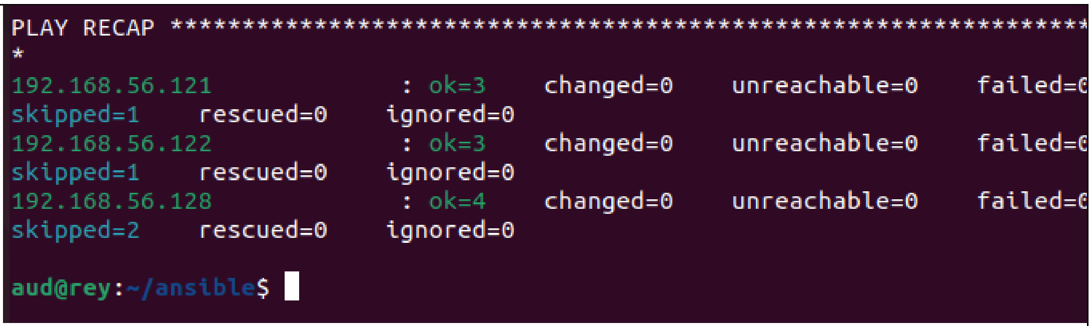
2.4 *ansible-playbook --tags apache --ask-become-pass site.yml*



```
aud@rey:~/ansible$ ansible-playbook --tags apache --ask-become-pass site.ym
BECOME password:

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

TASK [Gathering Facts] *******************************************************
*
ok: [192.168.56.121]
ok: [192.168.56.122]
ok: [192.168.56.128]

TASK [install updates (CentOS)] **********************************************
*
skipping: [192.168.56.122]
skipping: [192.168.56.121]
ok: [192.168.56.128]

TASK [install updates (Ubuntu)] **********************************************
*
skipping: [192.168.56.128]
ok: [192.168.56.121]
ok: [192.168.56.122]

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

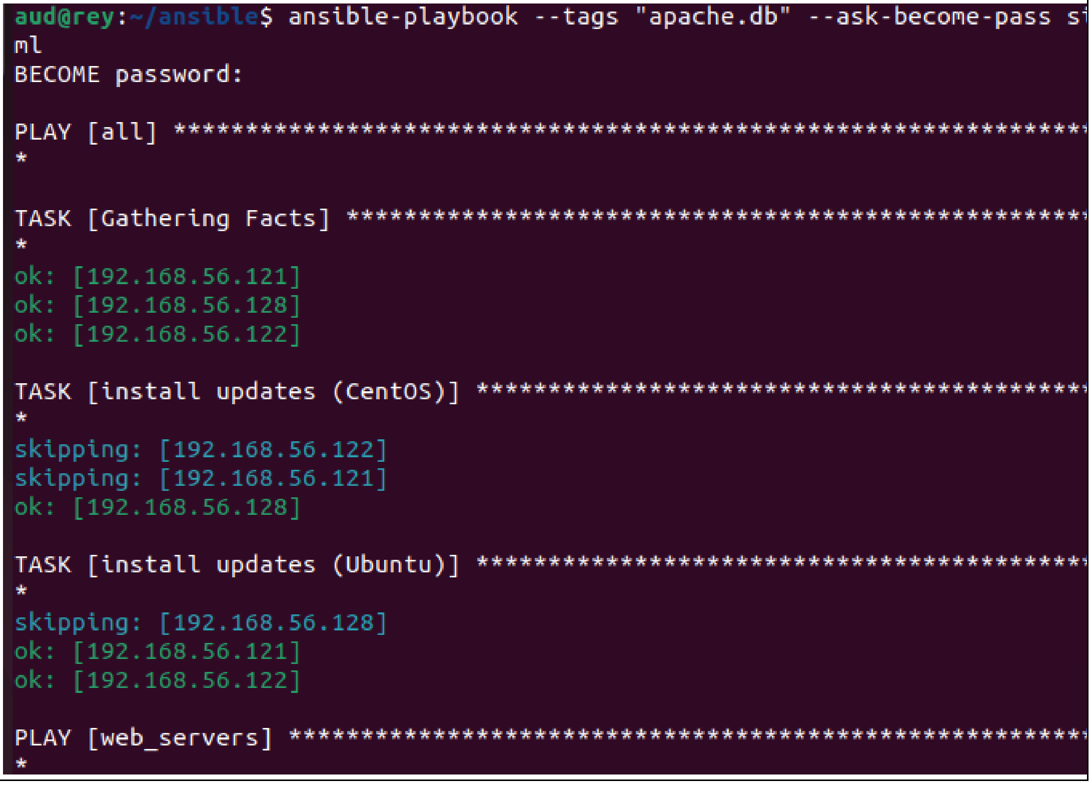


```
TASK [Gathering Facts] *******************************************************
*
ok: [192.168.56.128]

TASK [install apache and php for Ubuntu servers] *****************************
*
skipping: [192.168.56.128]

TASK [install apache and php for CentOS servers] *****************************
*
ok: [192.168.56.128]

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

TASK [Gathering Facts] *******************************************************
*
ok: [192.168.56.122]

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

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

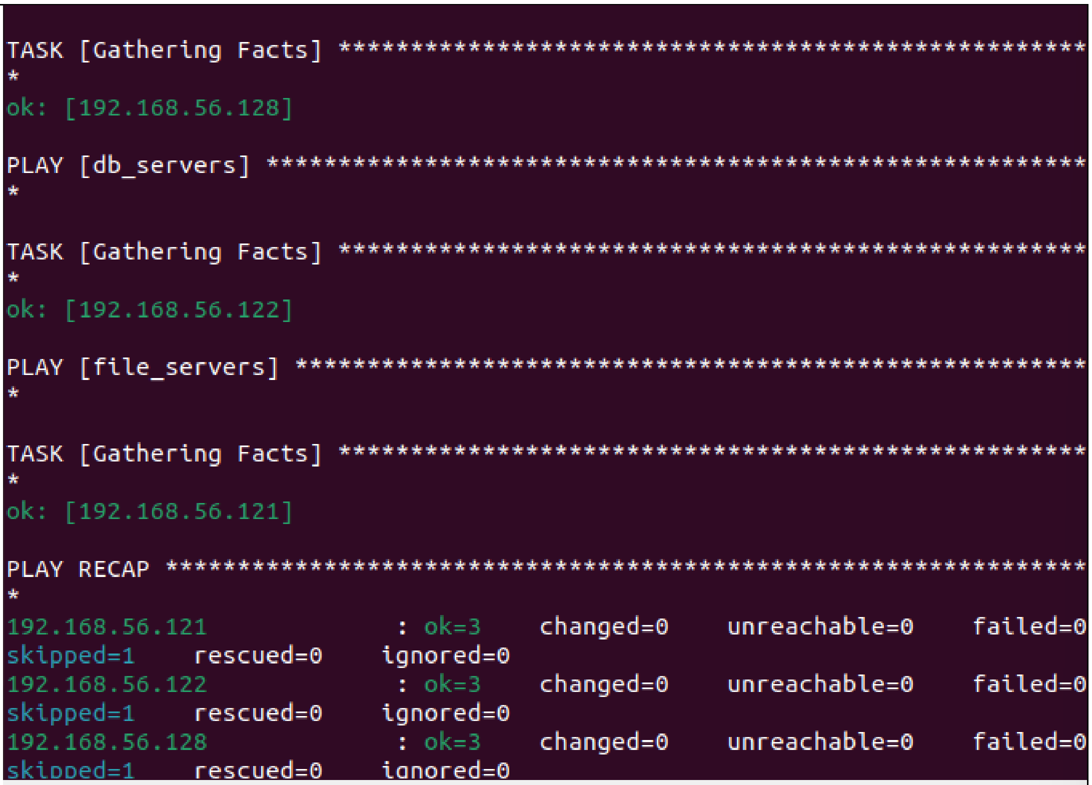


```
PLAY RECAP ****************************************************************
*
192.168.56.121             : ok=3    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.122             : ok=3    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.128             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0

aud@rey:~/ansible$ 
```

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



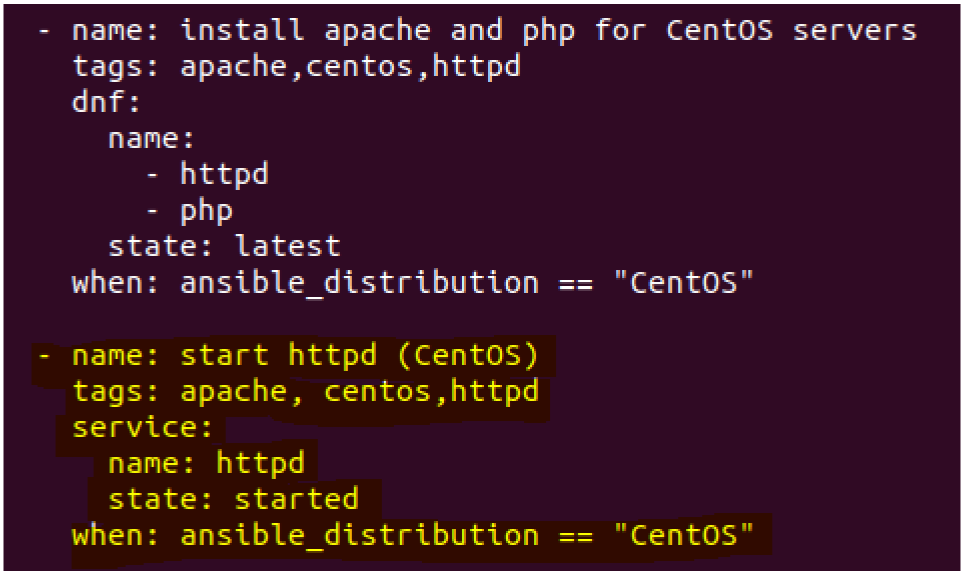
```
aud@rey:~/ansible$ ansible-playbook --tags "apache.db" --ask-become-pass si
ml
BECOME password:

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

TASK [Gathering Facts] ****************************************************
*
ok: [192.168.56.121]
ok: [192.168.56.128]
ok: [192.168.56.122]

TASK [install updates (CentOS)] *******************************************
*
skipping: [192.168.56.122]
skipping: [192.168.56.121]
ok: [192.168.56.128]

TASK [install updates (Ubuntu)] *******************************************
*
skipping: [192.168.56.128]
ok: [192.168.56.121]
ok: [192.168.56.122]

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



```
TASK [Gathering Facts] *****************************************************
*
ok: [192.168.56.128]

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

TASK [Gathering Facts] *****************************************************
*
ok: [192.168.56.122]

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

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

PLAY RECAP *****************************************************************
*
192.168.56.121             : ok=3    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.122             : ok=3    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.128             : ok=3    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
```

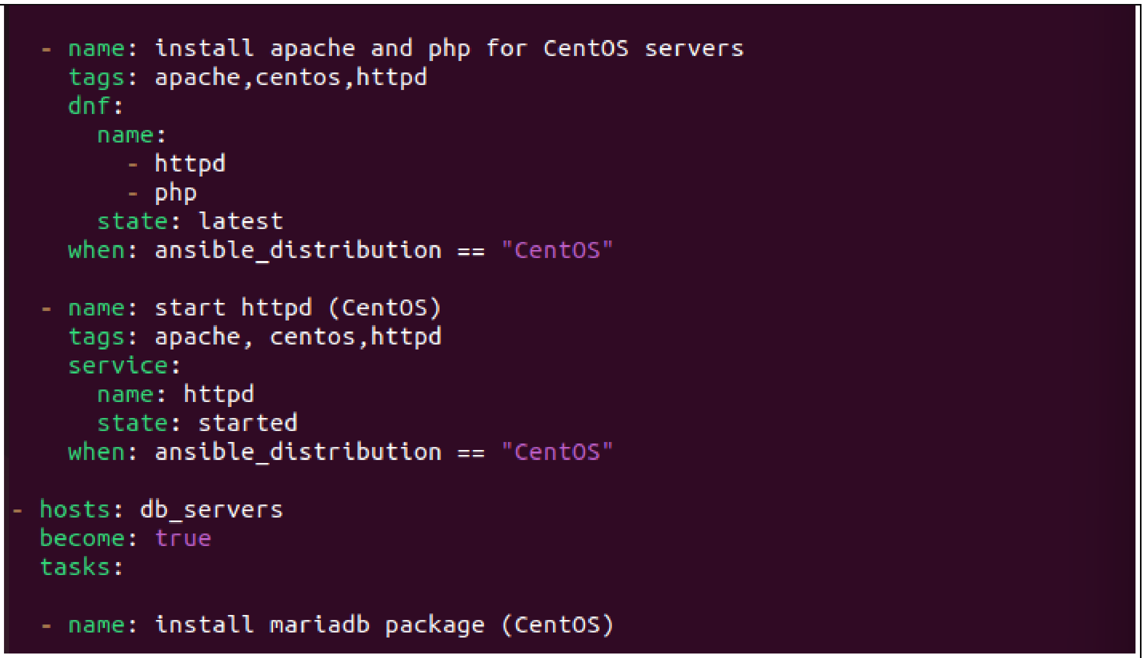
Task 3: Managing Services

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



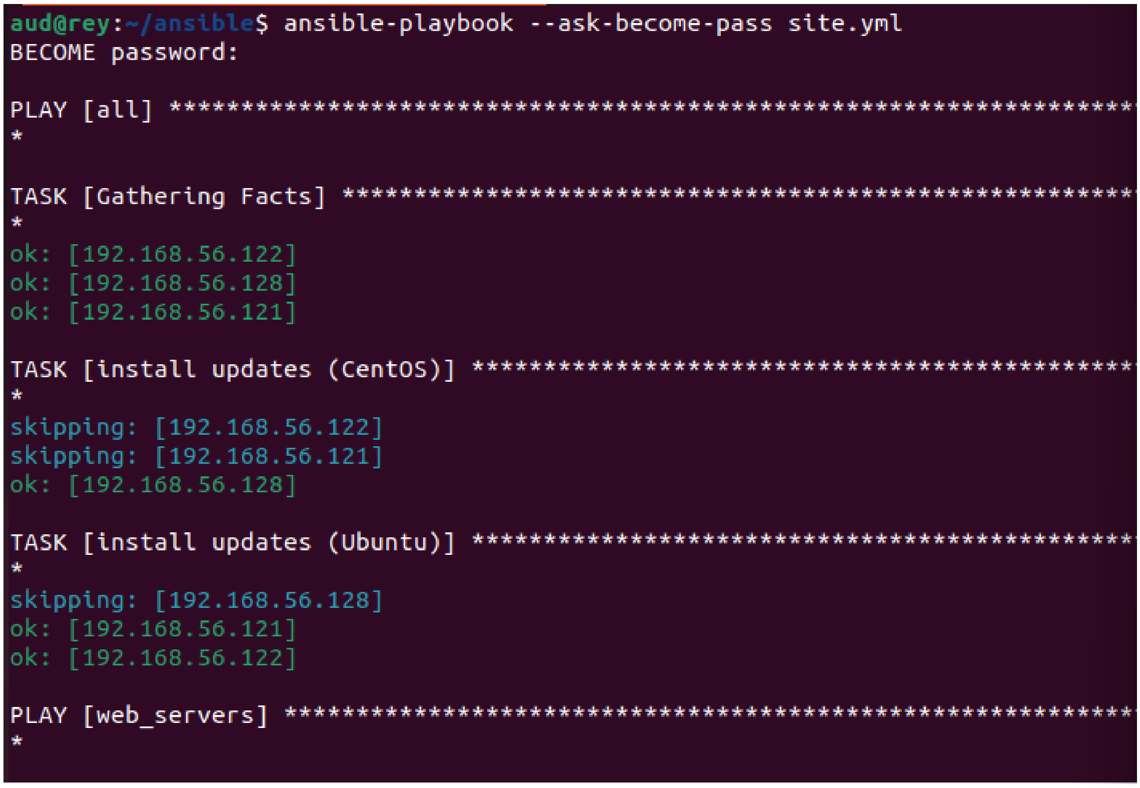
```
- name: install apache and php for CentOS servers
  tags: apache,centos,httpd
  dnf:
    name:
      - httpd
      - php
    state: latest
  when: ansible_distribution == "CentOS"

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

Figure 3.1.1

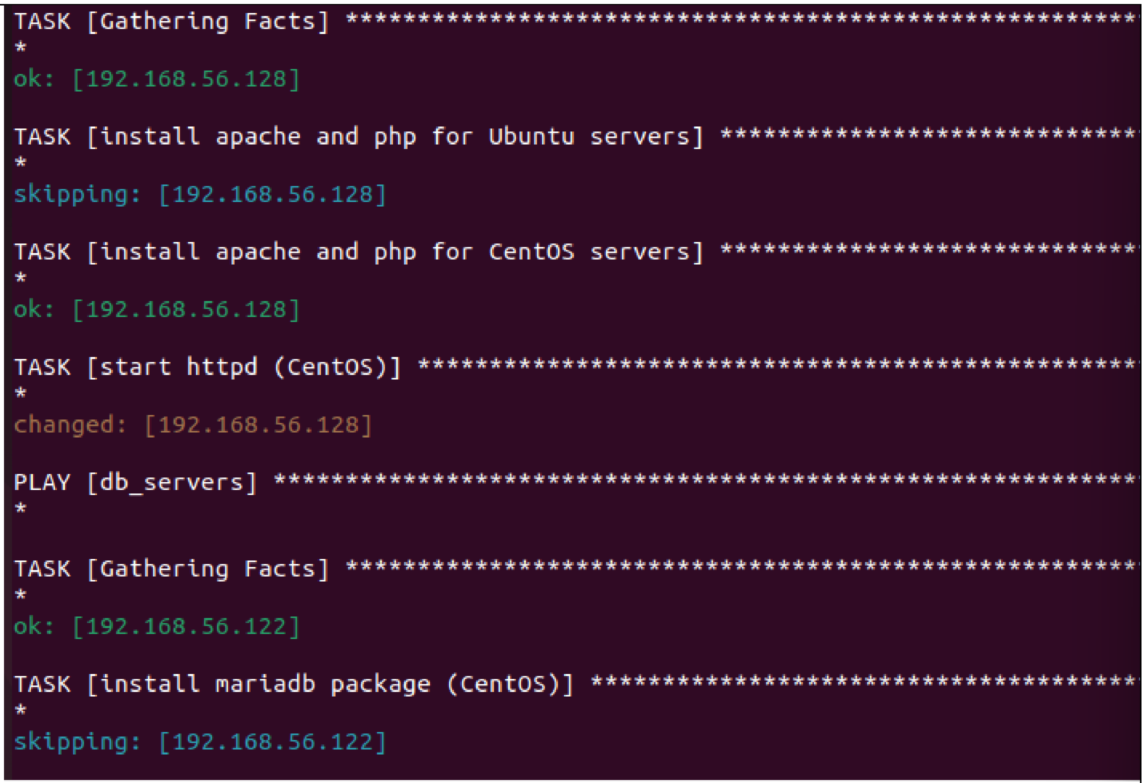


```
  - name: install apache and php for CentOS servers
    tags: apache,centos,httpd
    dnf:
      name:
        - httpd
        - php
      state: latest
    when: ansible_distribution == "CentOS"

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

- hosts: db_servers
  become: true
  tasks:

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

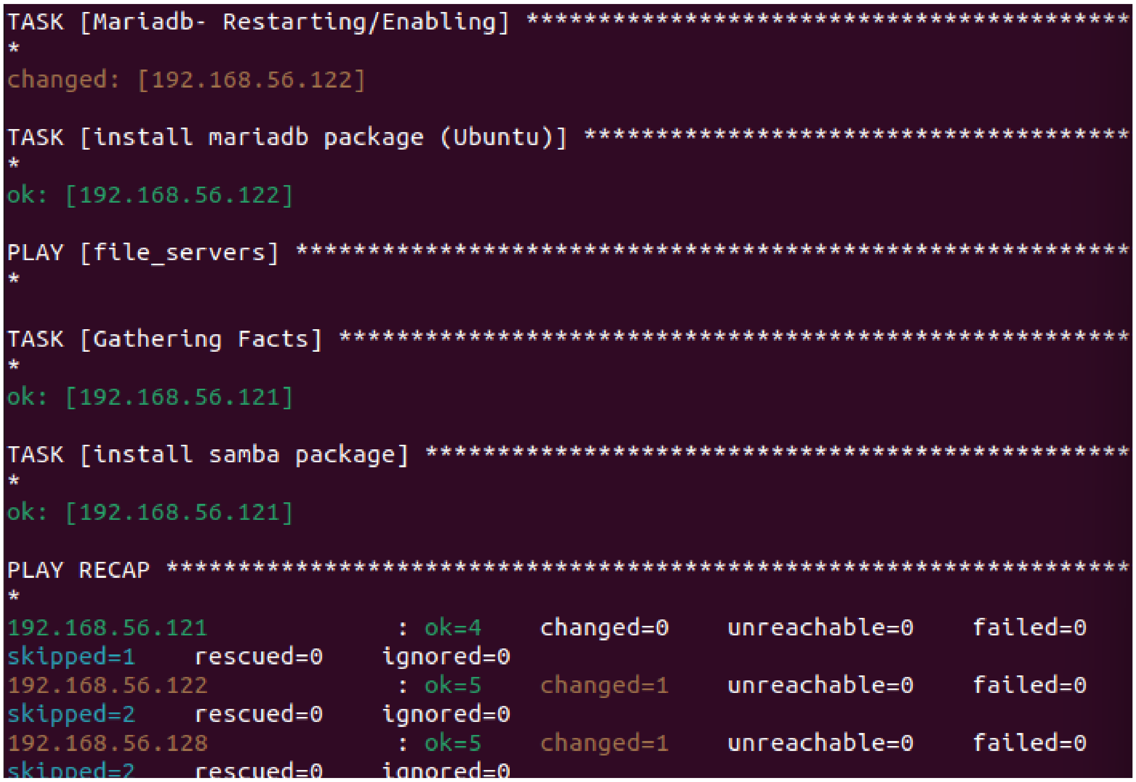
Make sure to save the file and exit.



```
aud@rey:~/ansible$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *******************************************************
*
ok: [192.168.56.122]
ok: [192.168.56.128]
ok: [192.168.56.121]

TASK [install updates (CentOS)] **********************************************
*
skipping: [192.168.56.122]
skipping: [192.168.56.121]
ok: [192.168.56.128]

TASK [install updates (Ubuntu)] **********************************************
*
skipping: [192.168.56.128]
ok: [192.168.56.121]
ok: [192.168.56.122]

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

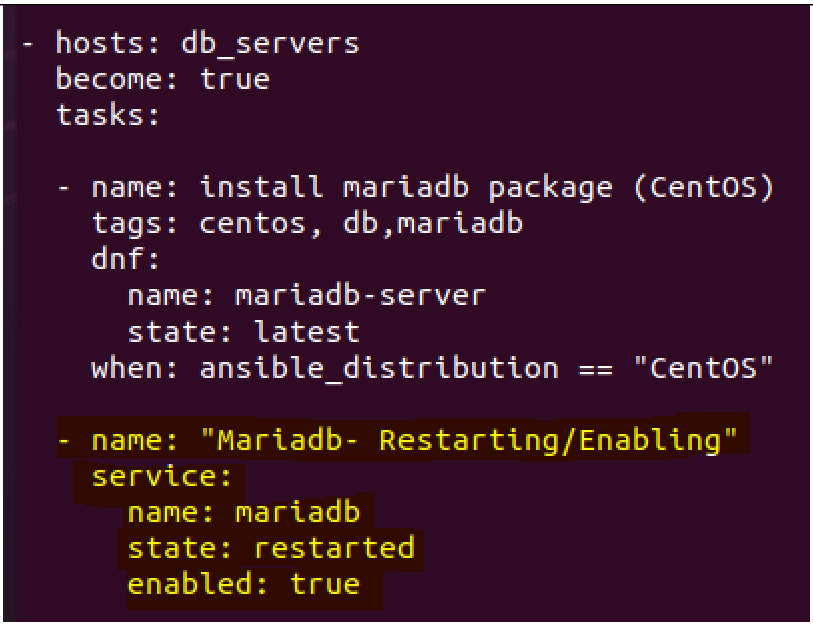


```
TASK [Gathering Facts] *******************************************************
*
ok: [192.168.56.128]

TASK [install apache and php for Ubuntu servers] *****************************
*
skipping: [192.168.56.128]

TASK [install apache and php for CentOS servers] *****************************
*
ok: [192.168.56.128]

TASK [start httpd (CentOS)] **************************************************
*
changed: [192.168.56.128]

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

TASK [Gathering Facts] *******************************************************
*
ok: [192.168.56.122]

TASK [install mariadb package (CentOS)] **************************************
*
skipping: [192.168.56.122]
```



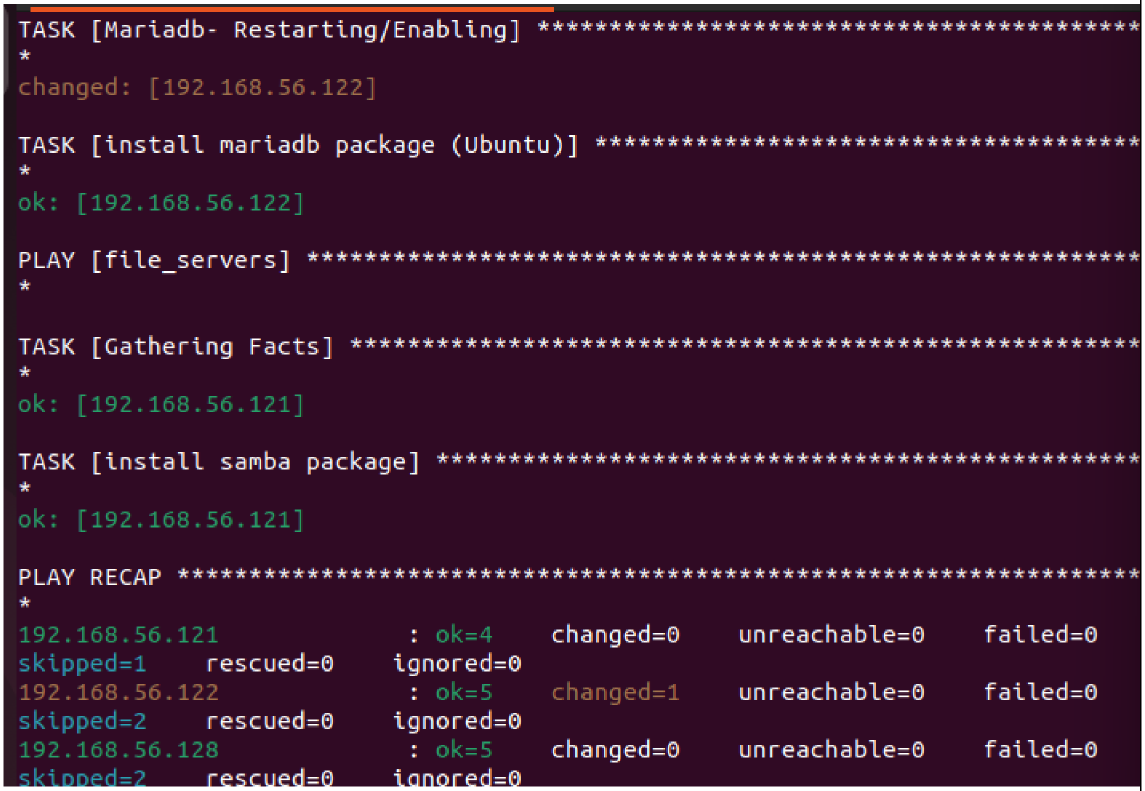
```
TASK [Mariadb- Restarting/Enabling] ******************************************
*
changed: [192.168.56.122]

TASK [install mariadb package (Ubuntu)] **************************************
*
ok: [192.168.56.122]

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

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

TASK [install samba package] *************************************************
*
ok: [192.168.56.121]

PLAY RECAP *******************************************************************
*
192.168.56.121             : ok=4    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.122             : ok=5    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.128             : ok=5    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
```

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

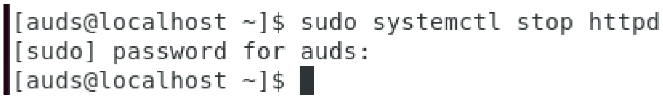


```
TASK [Mariadb- Restarting/Enabling] *******************************************
*
changed: [192.168.56.122]

TASK [install mariadb package (Ubuntu)] ***************************************
*
ok: [192.168.56.122]

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

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

TASK [install samba package] **************************************************
*
ok: [192.168.56.121]

PLAY RECAP ********************************************************************
*
192.168.56.121             : ok=4    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.122             : ok=5    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.128             : ok=5    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
```

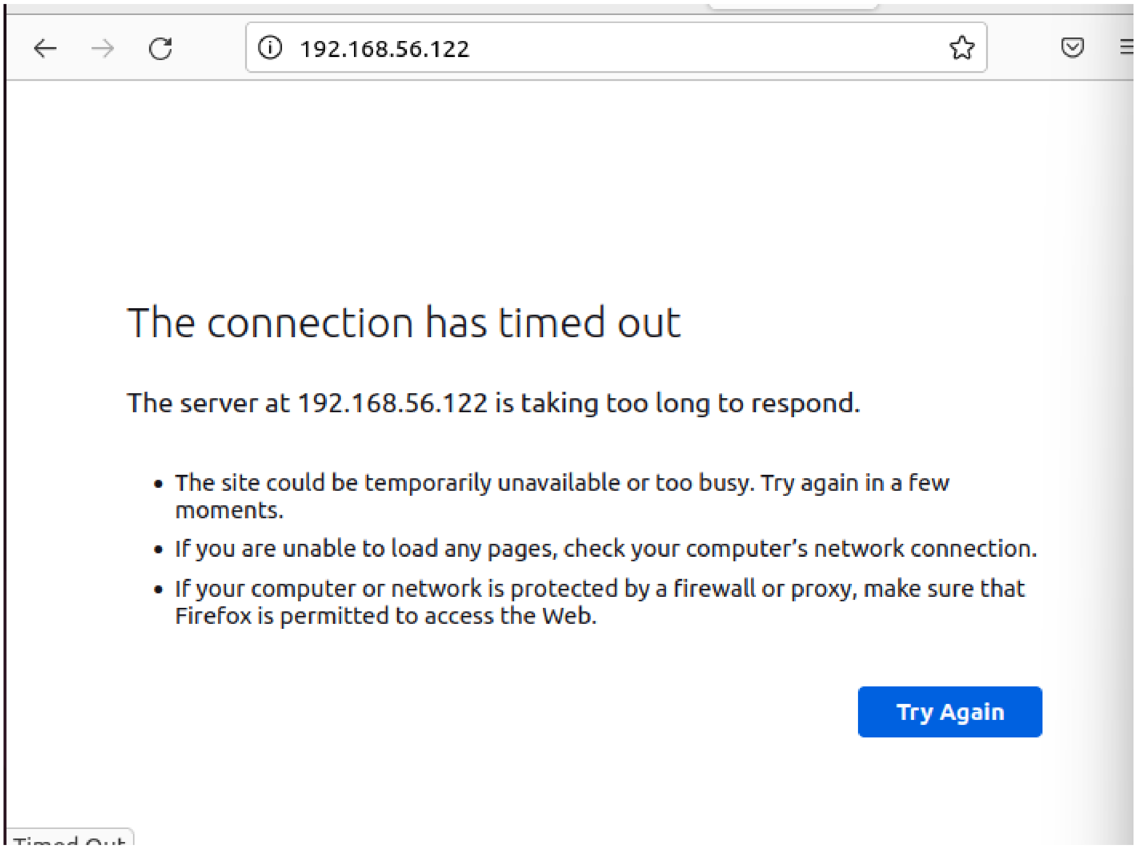
This is because in CentOS, installed packages' services are not run automatically. Thus, we need to create the module to run it automatically.

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.



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

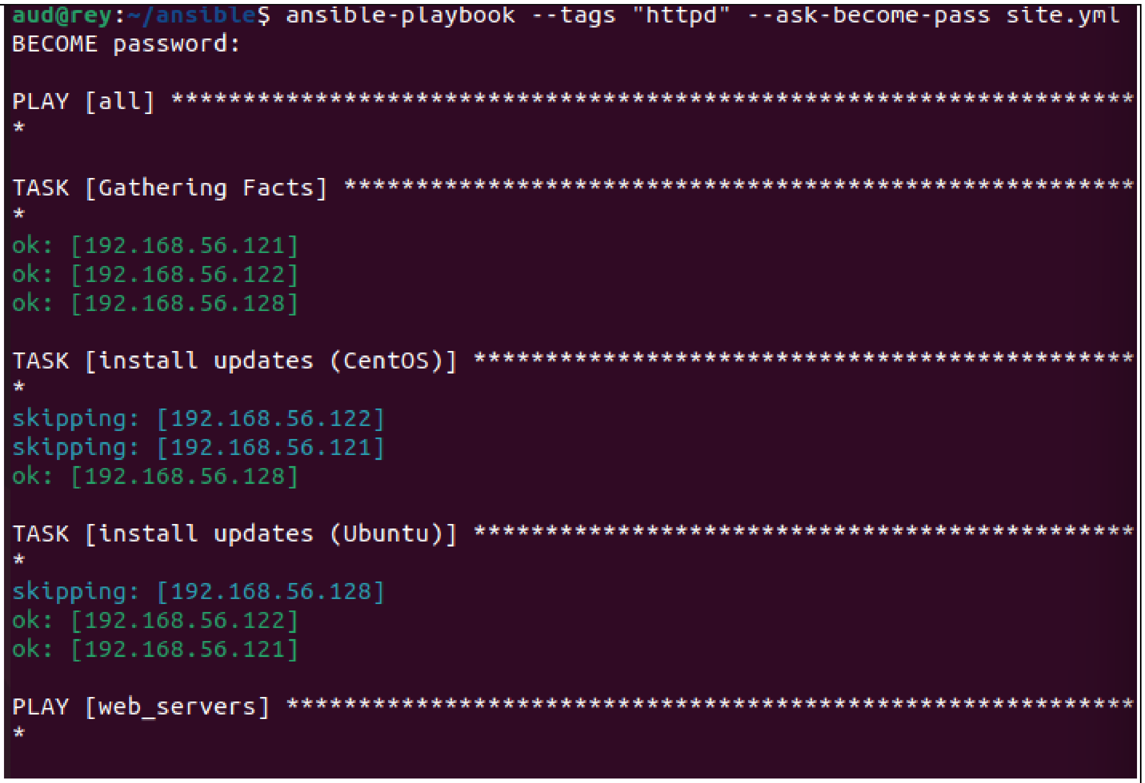


The connection has timed out

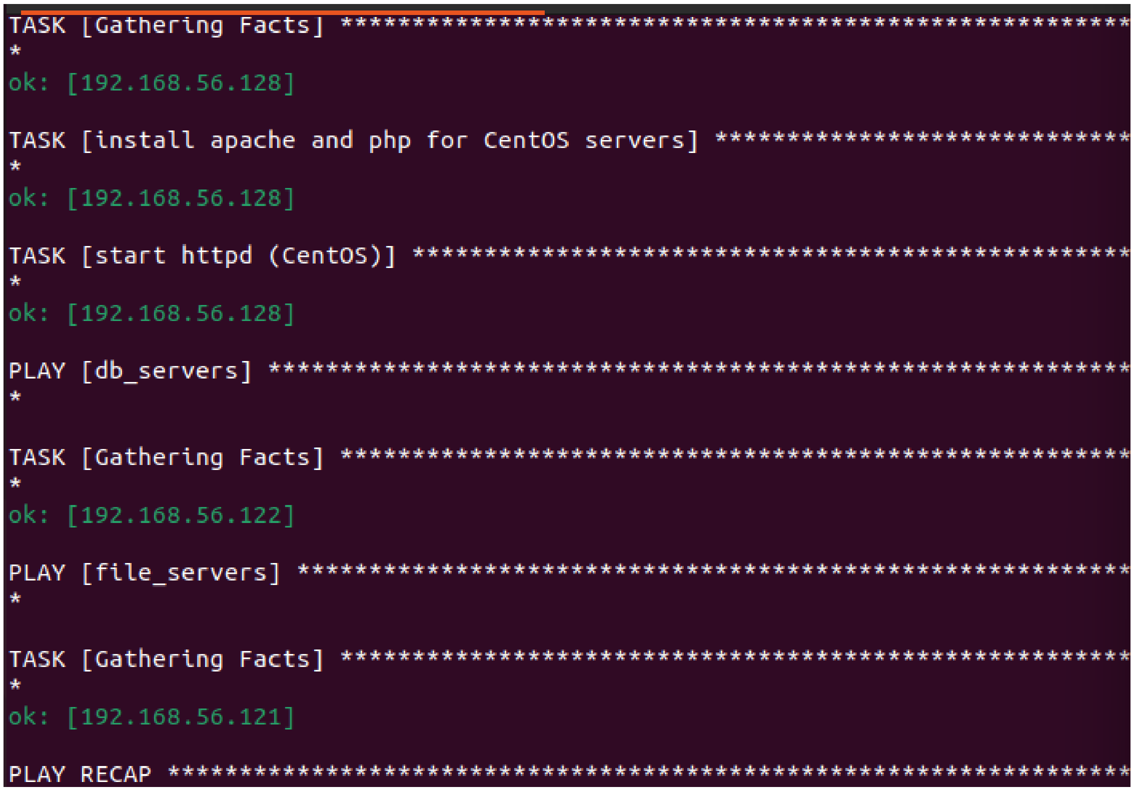
The server at 192.168.56.122 is taking too long to respond.

- 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.

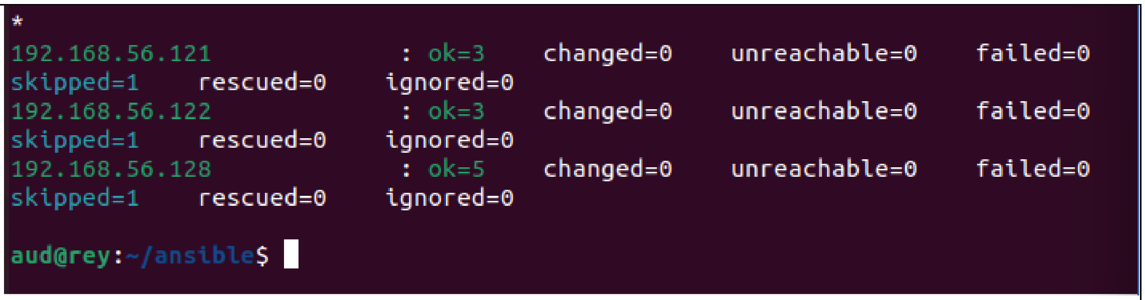
3. Go to the local machine and this time, run the *site.yml* file. Then after running the file, go again to the CentOS server and enter its IP address on the browser. Describe the result.



```
aud@rey:~/ansible$ ansible-playbook --tags "httpd" --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *******************************************************
*
ok: [192.168.56.121]
ok: [192.168.56.122]
ok: [192.168.56.128]

TASK [install updates (CentOS)] **********************************************
*
skipping: [192.168.56.122]
skipping: [192.168.56.121]
ok: [192.168.56.128]

TASK [install updates (Ubuntu)] **********************************************
*
skipping: [192.168.56.128]
ok: [192.168.56.122]
ok: [192.168.56.121]

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

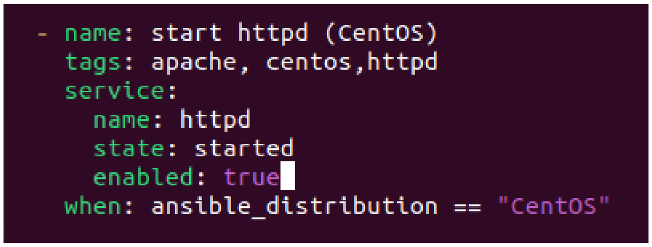


```
TASK [Gathering Facts] *******************************************************
*
ok: [192.168.56.128]

TASK [install apache and php for CentOS servers] *****************************
*
ok: [192.168.56.128]

TASK [start httpd (CentOS)] **************************************************
*
ok: [192.168.56.128]

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

TASK [Gathering Facts] *******************************************************
*
ok: [192.168.56.122]

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

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

PLAY RECAP *******************************************************************
```

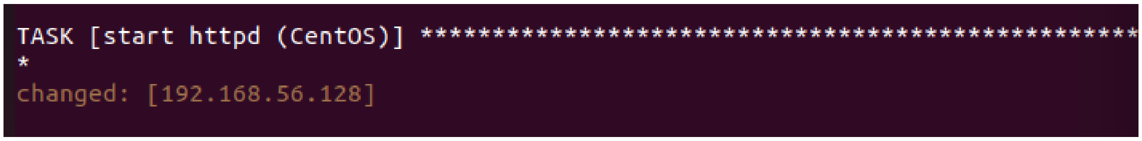


```
*
192.168.56.121             : ok=3    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.122             : ok=3    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.128             : ok=5    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0

aud@rey:~/ansible$ 
```

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)] ****************************************************
*
changed: [192.168.56.128]
```

Reflections:

Answer the following:

1. What is the importance of putting our remote servers into groups?
   It arranges how the duties given to various servers are distributed. This approach allows us to control a task that each of the servers is capable of carrying out. In this task, I noticed that in the part there are three groups in the inventory file that each indicate a separate server function. It exists a server for a database, webpage, and file transfer. The hosts variable calls for the groups inside of the site.yml file. Only the specified server(s) in the group are used to execute the tasks that are issued at this point. This means that when we categorize the servers, we are doing it based on how they operate and how they compare to their neighbors' servers.
2. What is the importance of tags in playbooks?
   As long as the tags are specific to the target job, a playbook can specify and execute a particular task in a server whether it is in another group or not. We assigned many tags to each task in task 2. These tags are comparable to tags used in other tasks. All of the tasks with the tags supplied in the command will be executed when we run the "ansible-playbook -tags "tag name" site.yml" command. By separating multiple tags with a comma and placing them inside quote marks, Ansible supports executing multiple tags.
3. Why do think some services need to be managed automatically in playbooks?
   I do consider that some services require automatic administration. For that reason, some services must be started when the computer is updated or upgraded, or

because the computer did not already have that package. Additionally, this will guarantee that the service has been effectively launched. As seen in the activity's final section, we attempted to stop the service httpd and restart it using the Ansible playbook with a set job to start the CentOS httpd server. In order to have the service start automatically when the playbook is executed, we also include "enable: true" in the task.

HONOR PLEDGE:

"I affirm that I will not give or receive unauthorized help on this activity and that all will be my own."