| | |
|---|---|
| **Name:** ROALLOS, Jean Gabriel Vincent G. | **Date Performed:** 09 / 12 / 2025 |
| **Course/Section:** CPE212 - CPE31S2 | **Date Submitted:** 09 / 19 / 2025 |
| **Instructor:** Engr. Robin Valenzuela | **Semester and SY:** 1st, 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**:

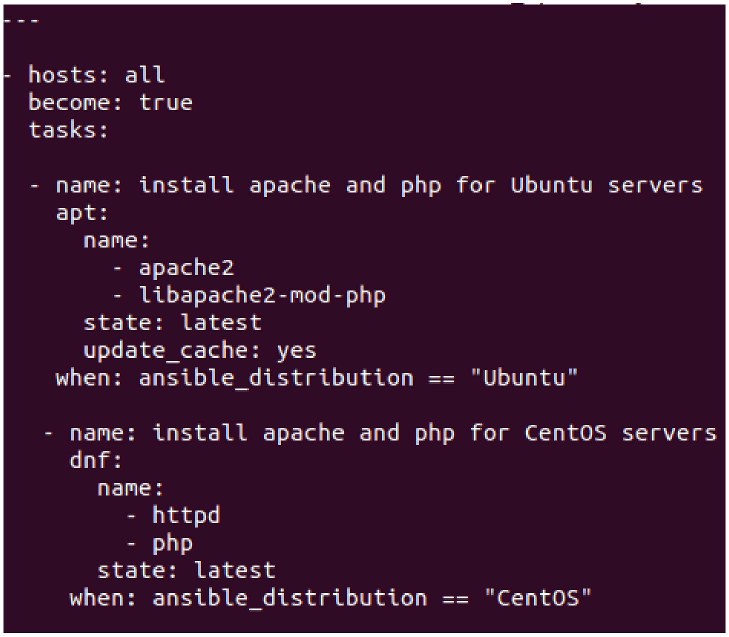
In this activity, we try to individualize hosts. For example, we don’t want apache on all our servers, or maybe only one of our servers is a web server, or maybe we have different servers like database or file servers running different things on different categories of servers and that is what we are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

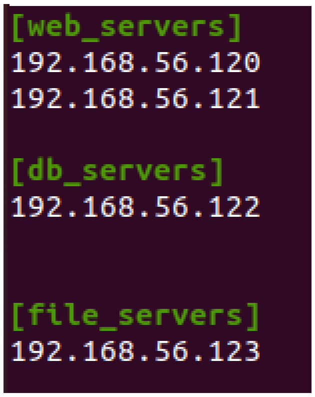
**Task 1: Targeting Specific Nodes**

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.



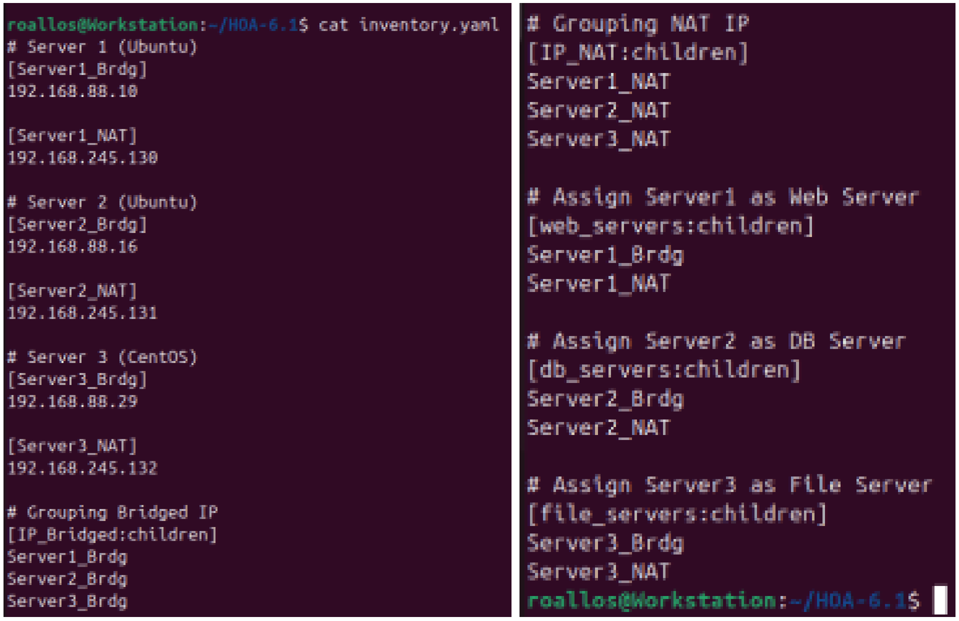
```
---

- 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"
```

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.

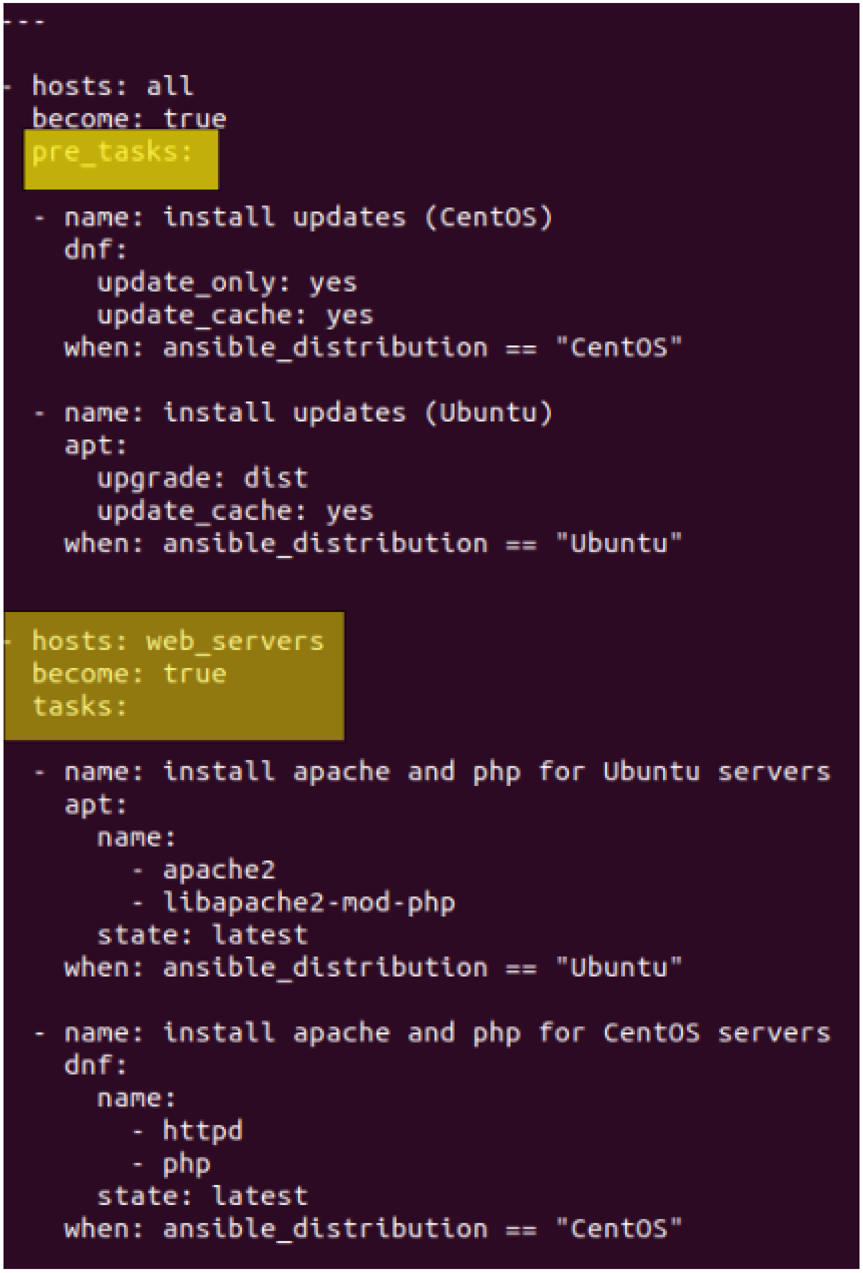


```
roallos@Workstation:~/HOA-6.1$ cat inventory.yaml
# Server 1 (Ubuntu)
[Server1_Brdg]
192.168.88.10

[Server1_NAT]
192.168.245.130

# Server 2 (Ubuntu)
[Server2_Brdg]
192.168.88.16

[Server2_NAT]
192.168.245.131

# Server 3 (CentOS)
[Server3_Brdg]
192.168.88.29

[Server3_NAT]
192.168.245.132

# Grouping Bridged IP
[IP_Bridged:children]
Server1_Brdg
Server2_Brdg
Server3_Brdg
```

```
# Grouping NAT IP
[IP_NAT:children]
Server1_NAT
Server2_NAT
Server3_NAT

# Assign Server1 as Web Server
[web_servers:children]
Server1_Brdg
Server1_NAT

# Assign Server2 as DB Server
[db_servers:children]
Server2_Brdg
Server2_NAT

# Assign Server3 as File Server
[file_servers:children]
Server3_Brdg
Server3_NAT
roallos@Workstation:~/HOA-6.1$
```

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

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

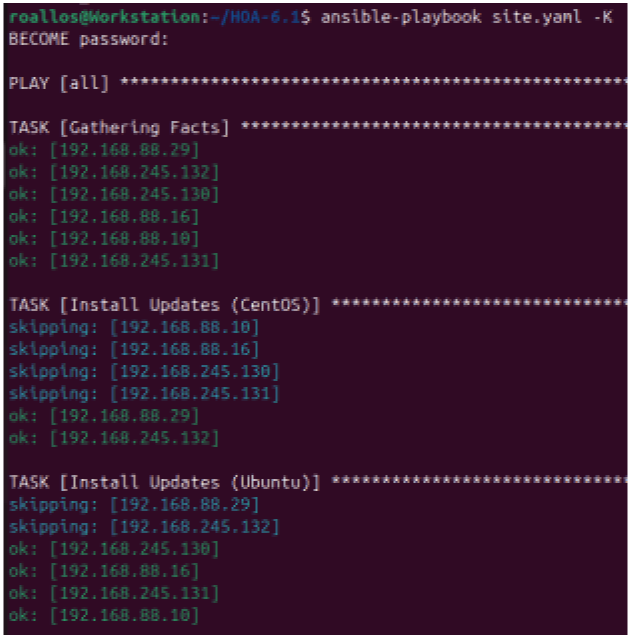


```
---

- hosts: all
  become: true
  pre_tasks:

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

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


- hosts: web_servers
  become: true
  tasks:

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

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

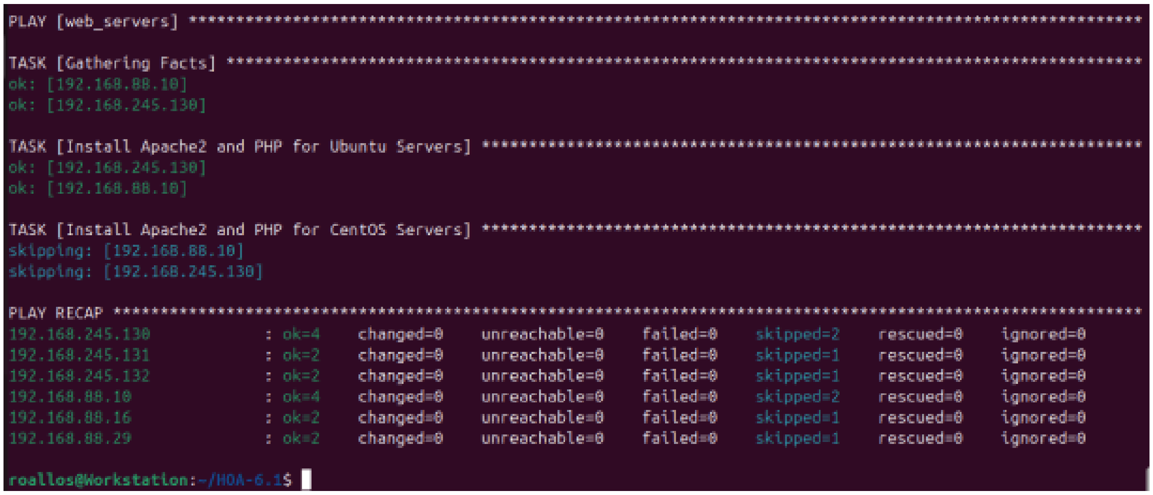
Make sure to save the file and exit.

The *pre-tasks* command tells the ansible to run it before any other thing. In the *pre-tasks*, CentOS will install updates while Ubuntu will upgrade its distribution package. This will run before running the second play, which is targeted at *web_servers*. In the second play, apache and php will be installed on both Ubuntu servers and CentOS servers.

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

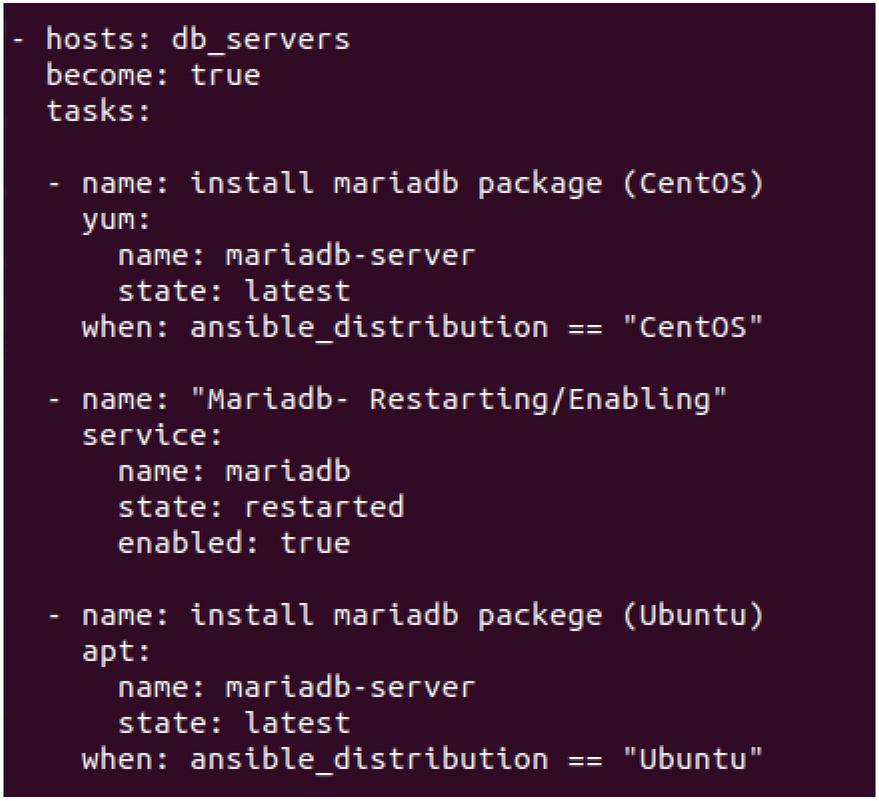


```
roallos@Workstation:~/HOA-6.1$ ansible-playbook site.yaml -K
BECOME password:

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

TASK [Gathering Facts] ***************************************
ok: [192.168.88.29]
ok: [192.168.245.132]
ok: [192.168.245.130]
ok: [192.168.88.16]
ok: [192.168.88.10]
ok: [192.168.245.131]

TASK [Install Updates (CentOS)] ******************************
skipping: [192.168.88.10]
skipping: [192.168.88.16]
skipping: [192.168.245.130]
skipping: [192.168.245.131]
ok: [192.168.88.29]
ok: [192.168.245.132]

TASK [Install Updates (Ubuntu)] ******************************
skipping: [192.168.88.29]
skipping: [192.168.245.132]
ok: [192.168.245.130]
ok: [192.168.88.16]
ok: [192.168.245.131]
ok: [192.168.88.10]
```



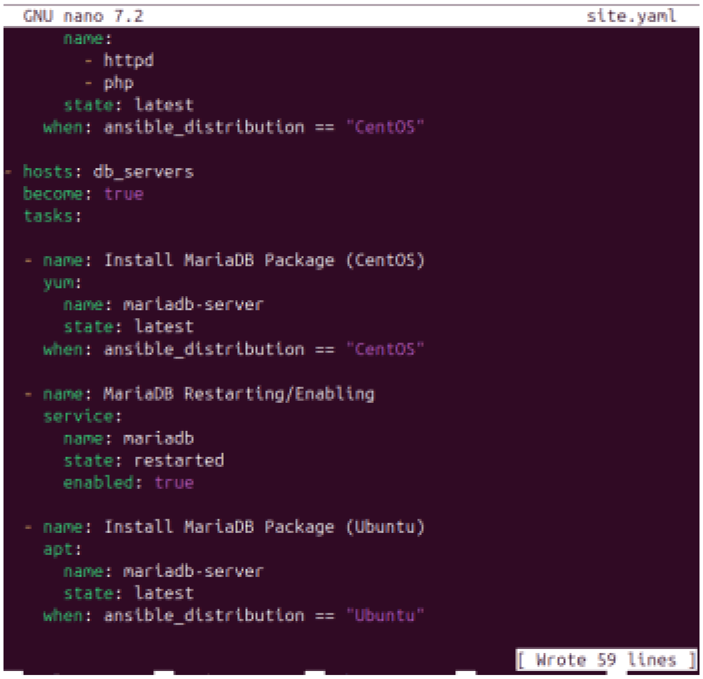
```
PLAY [web_servers] ***************************************************************************************

TASK [Gathering Facts] ***********************************************************************************
ok: [192.168.88.10]
ok: [192.168.245.130]

TASK [Install Apache2 and PHP for Ubuntu Servers] ********************************************************
ok: [192.168.245.130]
ok: [192.168.88.10]

TASK [Install Apache2 and PHP for CentOS Servers] ********************************************************
skipping: [192.168.88.10]
skipping: [192.168.245.130]

PLAY RECAP ***********************************************************************************************
192.168.245.130            : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.245.131            : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.245.132            : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.88.10              : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.88.16              : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.88.29              : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

roallos@Workstation:~/HOA-6.1$ 
```

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.



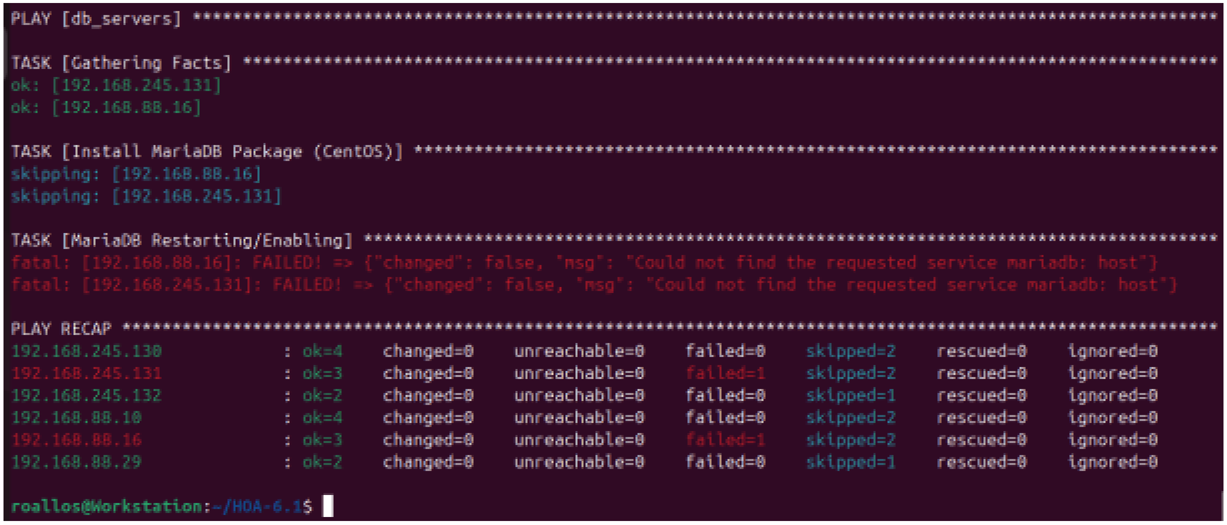
```
- hosts: db_servers
  become: true
  tasks:

  - 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.yaml
      name:
        - httpd
        - php
      state: latest
    when: ansible_distribution == "CentOS"

- hosts: db_servers
  become: true
  tasks:

  - name: Install MariaDB Package (CentOS)
    yum:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "CentOS"

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

  - name: Install MariaDB Package (Ubuntu)
    apt:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "Ubuntu"

                                                   [ Wrote 59 lines ]
```

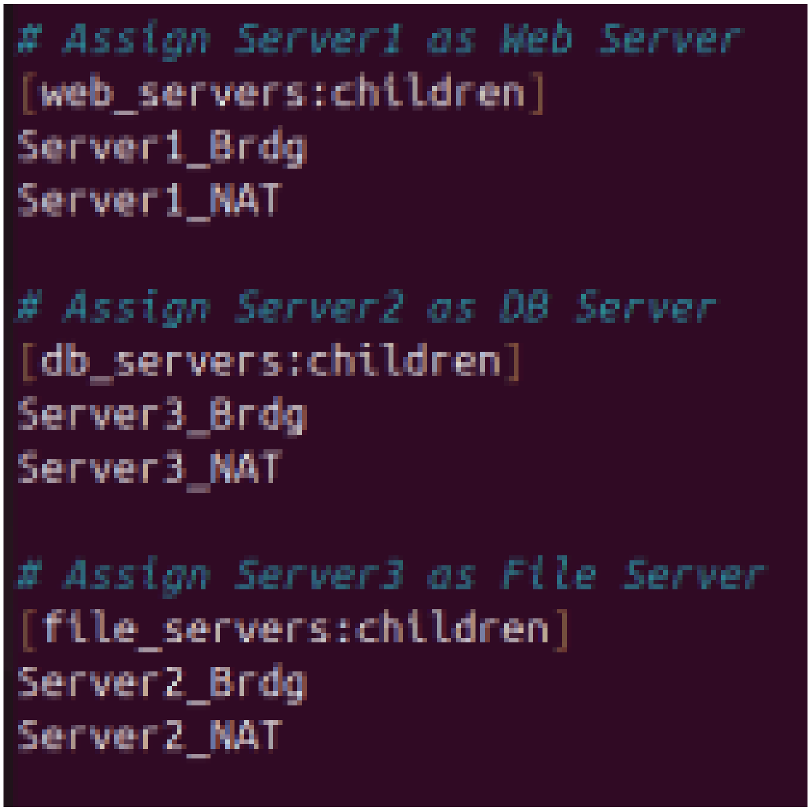
Run the *site.yml* file and describe the result.



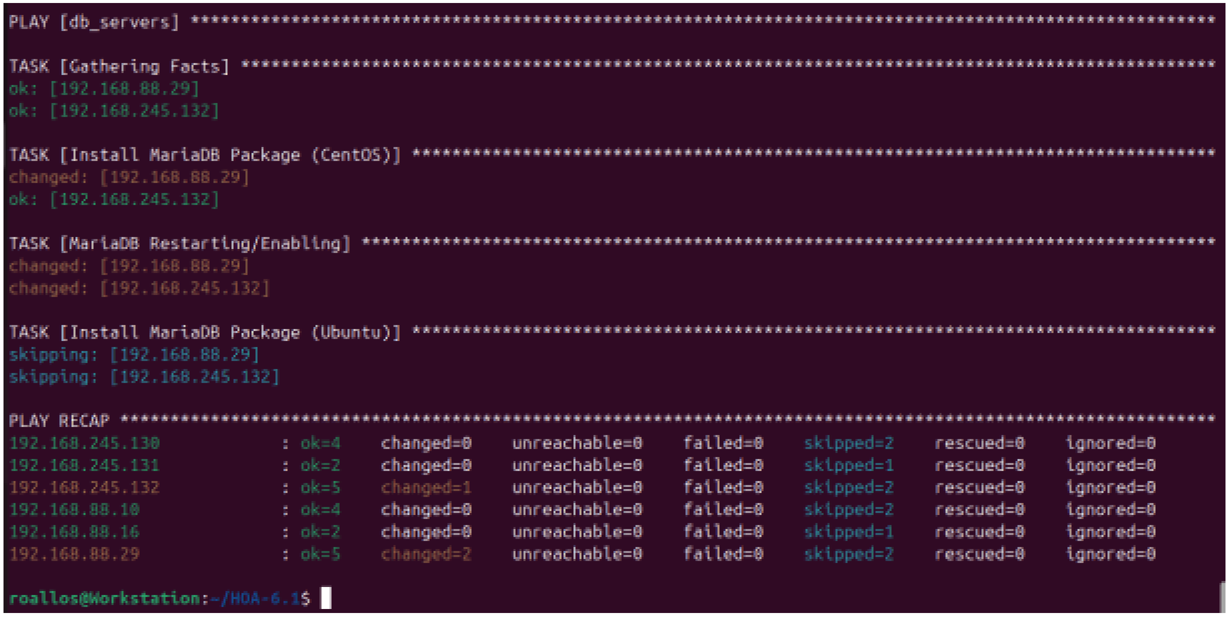
```
PLAY [db_servers] ********************************************************************************************************

TASK [Gathering Facts] ***************************************************************************************************
ok: [192.168.245.131]
ok: [192.168.88.16]

TASK [Install MariaDB Package (CentOS)] **********************************************************************************
skipping: [192.168.88.16]
skipping: [192.168.245.131]

TASK [MariaDB Restarting/Enabling] ***************************************************************************************
fatal: [192.168.88.16]: FAILED! => {"changed": false, "msg": "Could not find the requested service mariadb: host"}
fatal: [192.168.245.131]: FAILED! => {"changed": false, "msg": "Could not find the requested service mariadb: host"}

PLAY RECAP ***************************************************************************************************************
192.168.245.130            : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.245.131            : ok=3    changed=0    unreachable=0    failed=1    skipped=2    rescued=0    ignored=0
192.168.245.132            : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.88.10              : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.88.16              : ok=3    changed=0    unreachable=0    failed=1    skipped=2    rescued=0    ignored=0
192.168.88.29              : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

roallos@Workstation:~/HOA-6.1$
```

***This is to be expected because the managed node under the "db_servers" group is an Ubuntu machine, not CentOS. Since the 'mariadb-server' package was not installed in CentOS, the "MariaDB Restarting/Enabling" play would lead to an error. Due to this error, the play for MariaDB installation for Ubuntu was not executed.***

***I have then swapped the managed node under the "db_servers" group to an Ubuntu machine, below is the updated inventory file and the playbook output.***



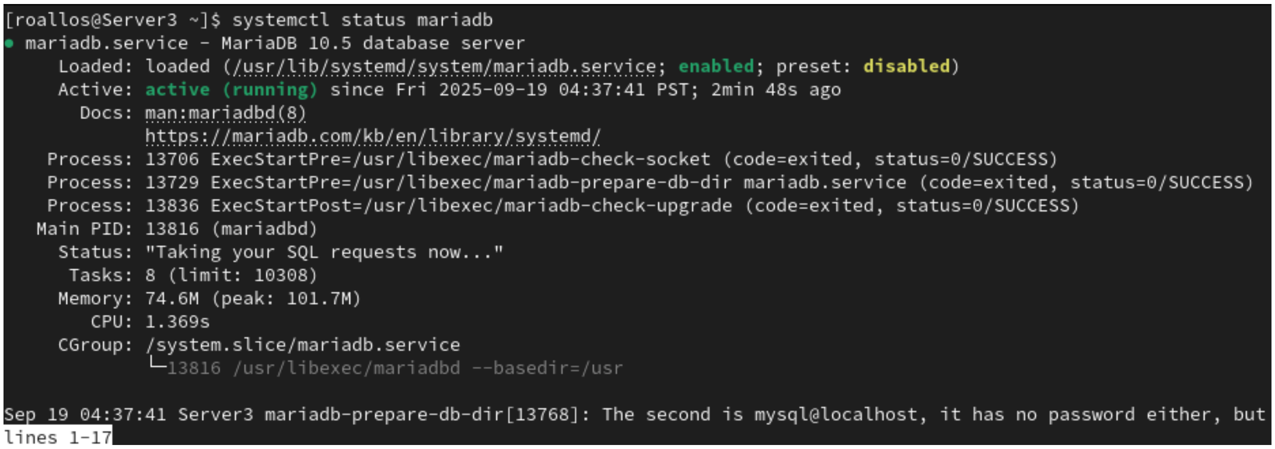
```
# Assign Server1 as Web Server
[web_servers:children]
Server1_Brdg
Server1_NAT

# Assign Server2 as DB Server
[db_servers:children]
Server3_Brdg
Server3_NAT

# Assign Server3 as File Server
[file_servers:children]
Server2_Brdg
Server2_NAT
```



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

TASK [Gathering Facts] *******************************************************************************
ok: [192.168.88.29]
ok: [192.168.245.132]

TASK [Install MariaDB Package (CentOS)] **************************************************************
changed: [192.168.88.29]
ok: [192.168.245.132]

TASK [MariaDB Restarting/Enabling] *******************************************************************
changed: [192.168.88.29]
changed: [192.168.245.132]

TASK [Install MariaDB Package (Ubuntu)] **************************************************************
skipping: [192.168.88.29]
skipping: [192.168.245.132]

PLAY RECAP *******************************************************************************************
192.168.245.130            : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.245.131            : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.245.132            : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.88.10              : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.88.16              : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.88.29              : ok=5    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0

roallos@Workstation:~/HOA-6.1$ 
```

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.

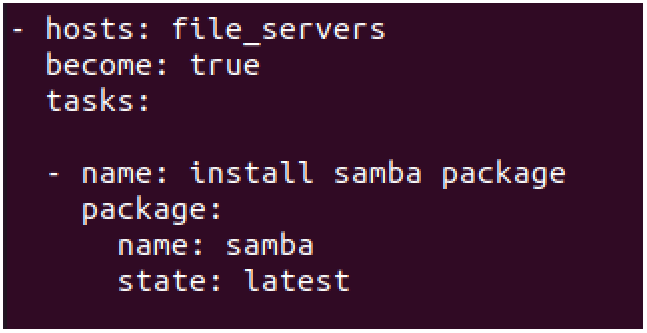
***Since I have only placed a CentOS machine in the "db_servers" group, I would only show a screenshot of the status of MariaDB in that machine.***



```
[roallos@Server3 ~]$ systemctl status mariadb
● mariadb.service - MariaDB 10.5 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: disabled)
     Active: active (running) since Fri 2025-09-19 04:37:41 PST; 2min 48s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 13706 ExecStartPre=/usr/libexec/mariadb-check-socket (code=exited, status=0/SUCCESS)
    Process: 13729 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir mariadb.service (code=exited, status=0/SUCCESS)
    Process: 13836 ExecStartPost=/usr/libexec/mariadb-check-upgrade (code=exited, status=0/SUCCESS)
   Main PID: 13816 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 8 (limit: 10308)
     Memory: 74.6M (peak: 101.7M)
        CPU: 1.369s
     CGroup: /system.slice/mariadb.service
             └─13816 /usr/libexec/mariadbd --basedir=/usr

Sep 19 04:37:41 Server3 mariadb-prepare-db-dir[13768]: The second is mysql@localhost, it has no password either, but
lines 1-17
```

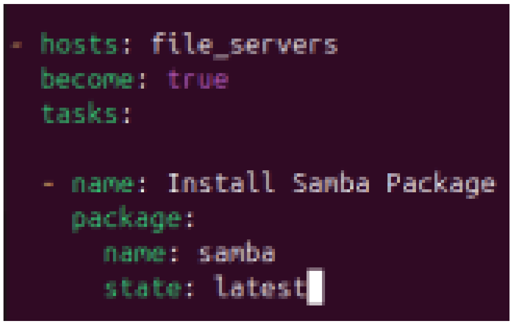
Describe the output.

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.



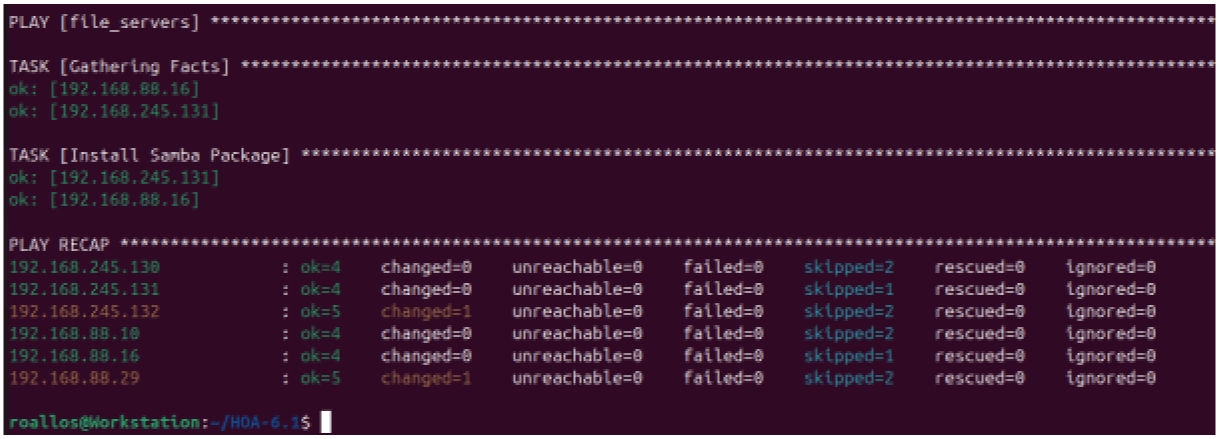
```
- hosts: file_servers
  become: true
  tasks:

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

Make sure to save the file and exit.



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

  - name: Install Samba Package
    package:
      name: samba
      state: latest
```

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



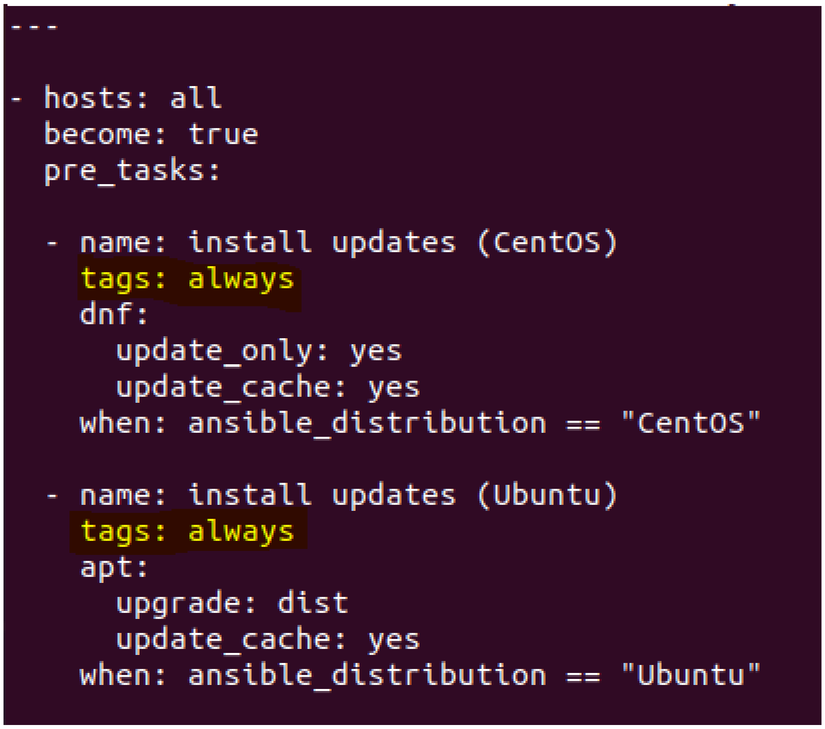
```
PLAY [file_servers] **************************************************************************************

TASK [Gathering Facts] ***********************************************************************************
ok: [192.168.88.16]
ok: [192.168.245.131]

TASK [Install Samba Package] *****************************************************************************
ok: [192.168.245.131]
ok: [192.168.88.16]

PLAY RECAP ***********************************************************************************************
192.168.245.130            : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.245.131            : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.245.132            : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.88.10              : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.88.16              : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.88.29              : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0

roallos@Workstation:~/HOA-6.1$ 
```

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.

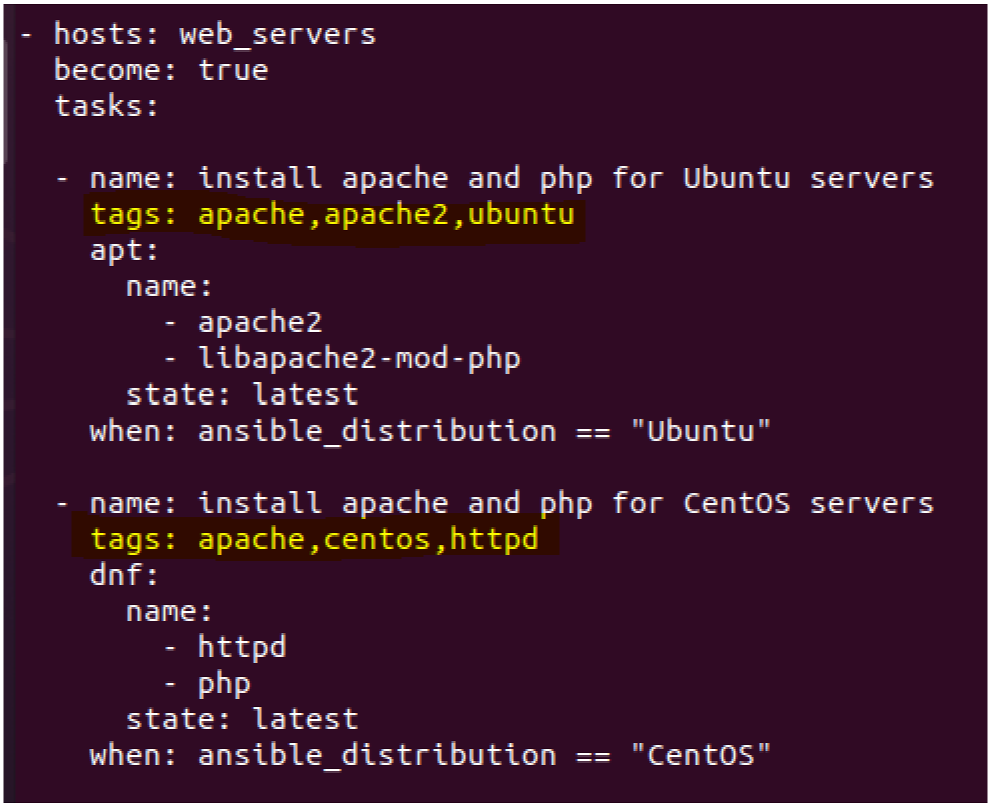
**Task 2: Using Tags in running playbooks**

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.

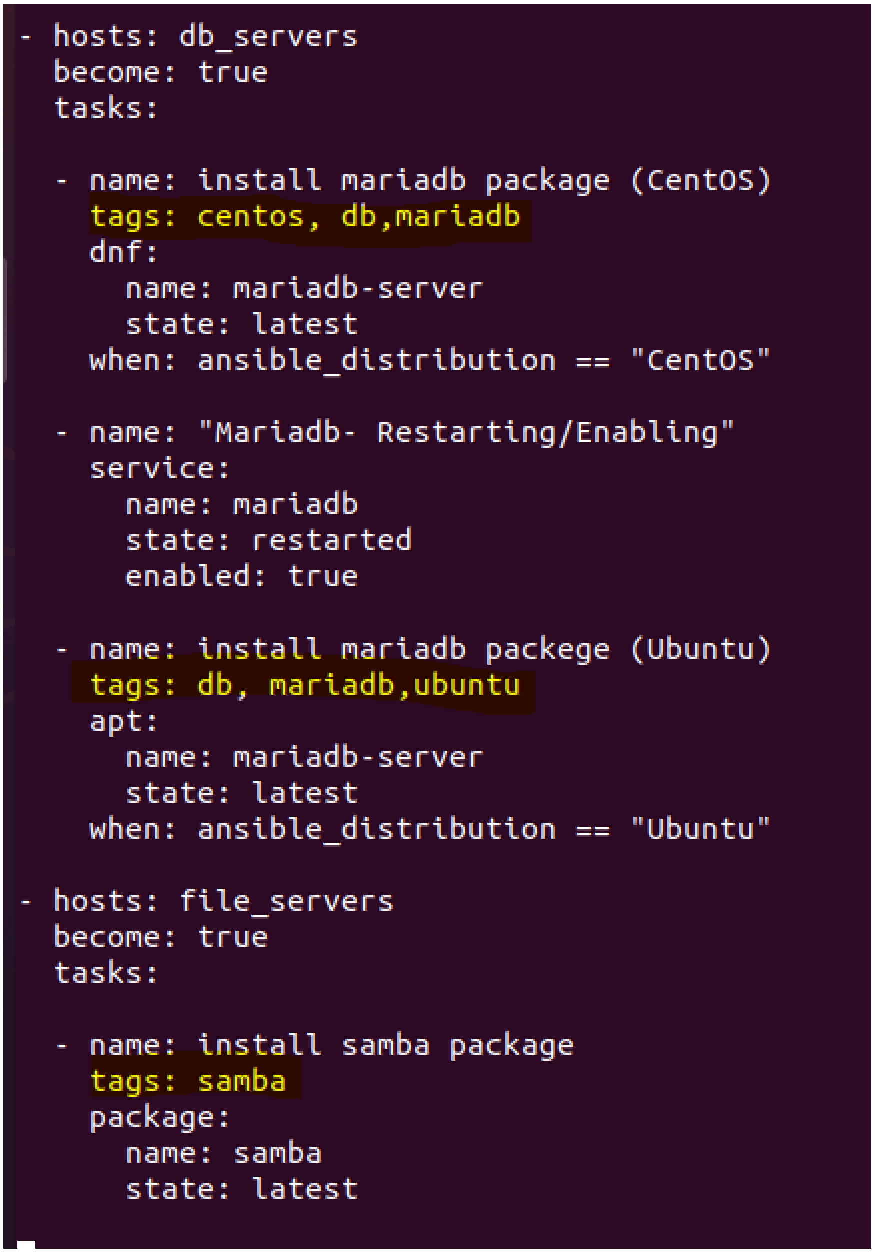
1. Edit the *site.yml* file. Add tags to the playbook. After the name, we can place the tags: *name_of_tag*. This is an arbitrary command, which means you can use any name for a tag.



```
---

- 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"
```

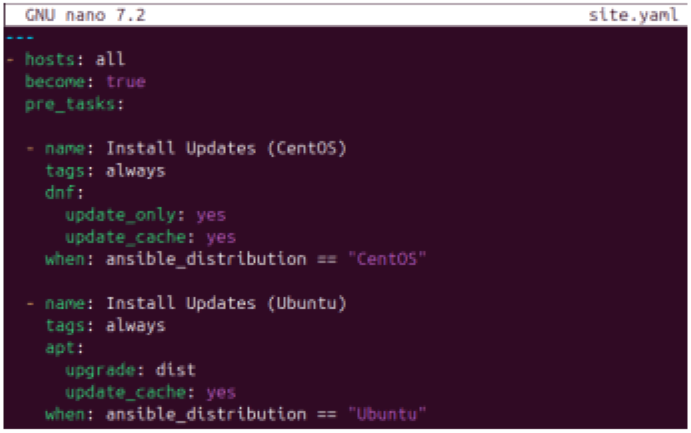


```
- hosts: web_servers
  become: true
  tasks:

  - 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"
```

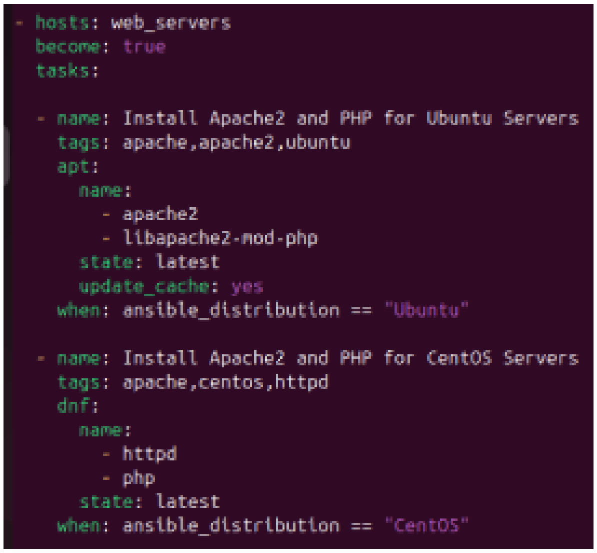


```
- hosts: db_servers
  become: true
  tasks:

  - 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
```

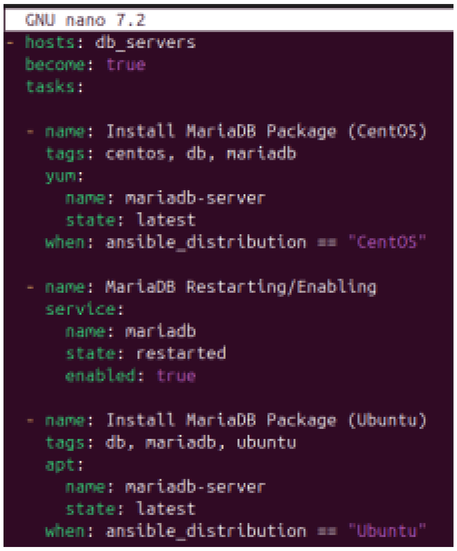
Make sure to save the file and exit.



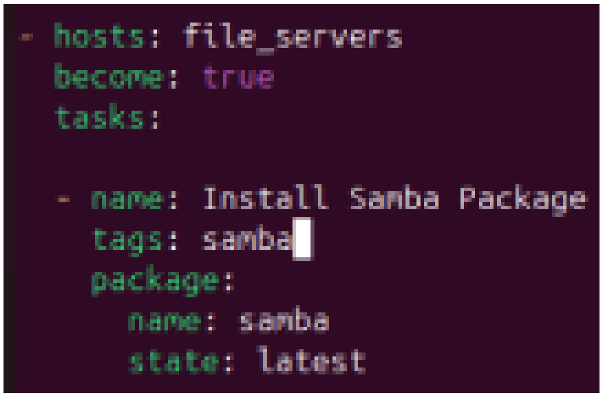
```
GNU nano 7.2                                              site.yaml
---
- hosts: all
  become: true
  pre_tasks:

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

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



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

  - name: Install Apache2 and PHP for Ubuntu Servers
    tags: apache,apache2,ubuntu
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: Install Apache2 and PHP for CentOS Servers
    tags: apache,centos,httpd
    dnf:
      name:
        - httpd
        - php
      state: latest
    when: ansible_distribution == "CentOS"
```



```
GNU nano 7.2
- hosts: db_servers
  become: true
  tasks:

  - name: Install MariaDB Package (CentOS)
    tags: centos, db, mariadb
    yum:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "CentOS"

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

  - name: Install MariaDB Package (Ubuntu)
    tags: db, mariadb, ubuntu
    apt:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "Ubuntu"
```

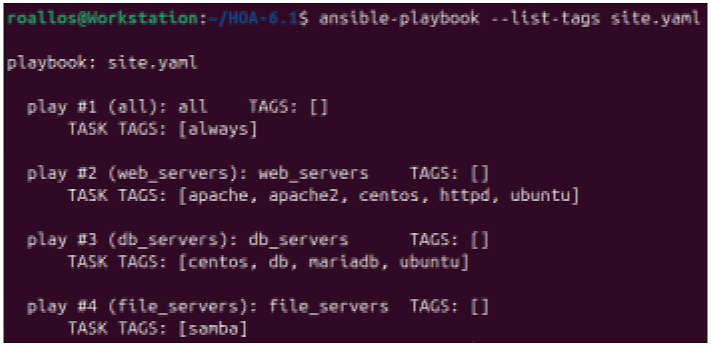


```
- hosts: file_servers
  become: true
  tasks:

  - name: Install Samba Package
    tags: samba
    package:
      name: samba
      state: latest
```

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

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

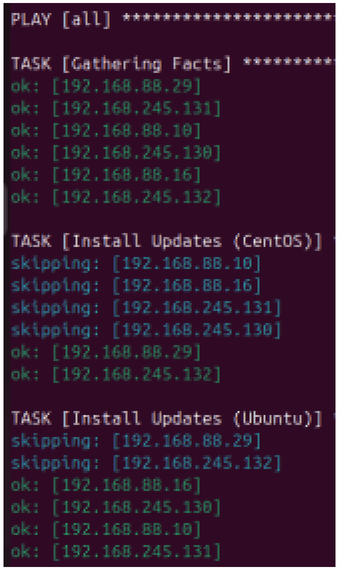
   2.1 *ansible-playbook --list-tags site.yml*



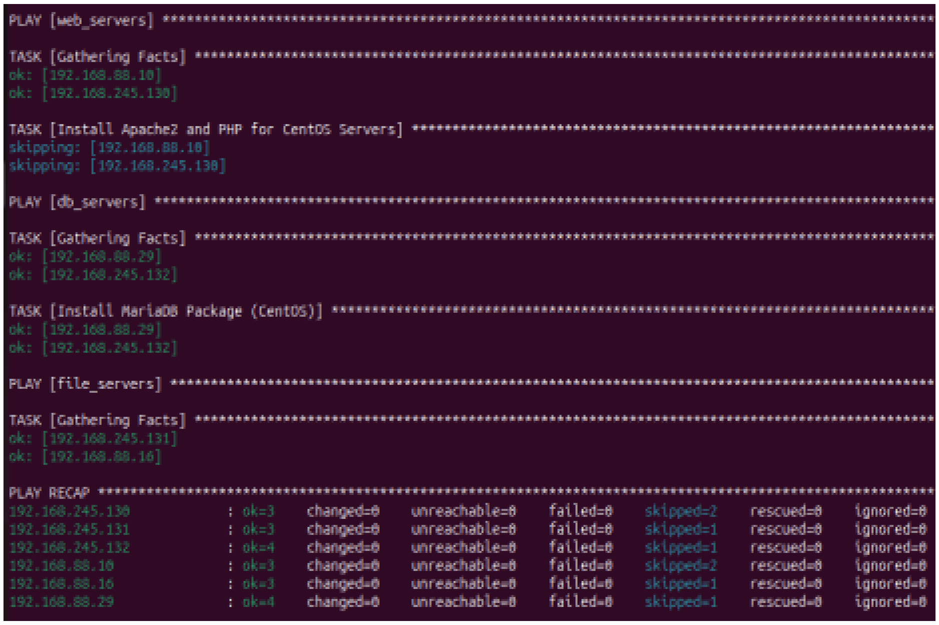
```
roallos@Workstation:~/HOA-6.1$ ansible-playbook --list-tags site.yaml

playbook: site.yaml

  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]
```

***This command lists all plays within given playbook and listing all tags associated or assigned to each play.***

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



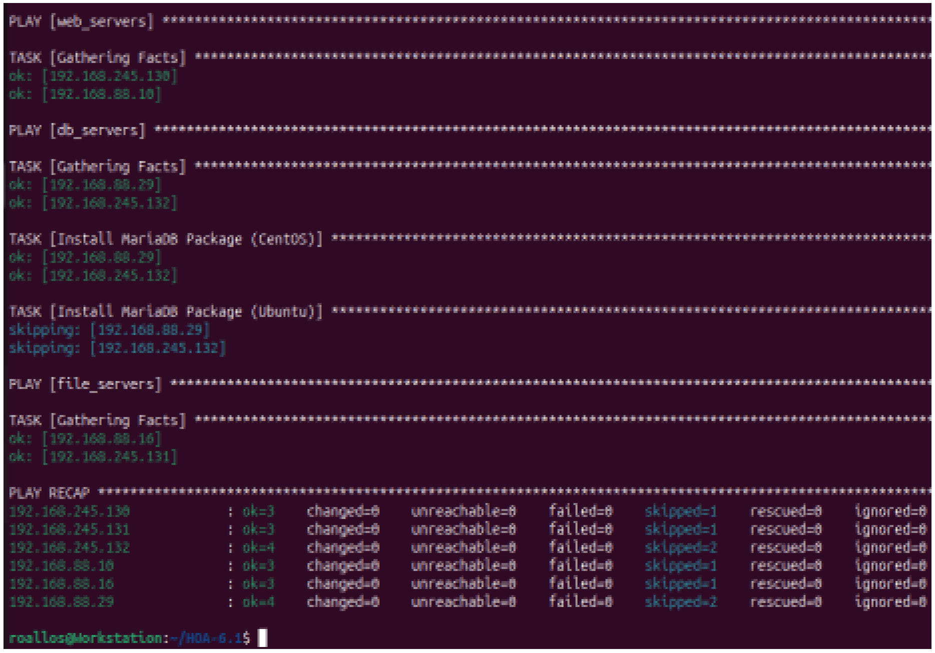
```
PLAY [all] ***********************

TASK [Gathering Facts] *********
ok: [192.168.88.29]
ok: [192.168.245.131]
ok: [192.168.88.10]
ok: [192.168.245.130]
ok: [192.168.88.16]
ok: [192.168.245.132]

TASK [Install Updates (CentOS)] *
skipping: [192.168.88.10]
skipping: [192.168.88.16]
skipping: [192.168.245.131]
skipping: [192.168.245.130]
ok: [192.168.88.29]
ok: [192.168.245.132]

TASK [Install Updates (Ubuntu)] *
skipping: [192.168.88.29]
skipping: [192.168.245.132]
ok: [192.168.88.16]
ok: [192.168.245.130]
ok: [192.168.88.10]
ok: [192.168.245.131]
```



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

TASK [Gathering Facts] ************************************************************************************
ok: [192.168.88.10]
ok: [192.168.245.130]

TASK [Install Apache2 and PHP for CentOS Servers] *********************************************************
skipping: [192.168.88.10]
skipping: [192.168.245.130]

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

TASK [Gathering Facts] ************************************************************************************
ok: [192.168.88.29]
ok: [192.168.245.132]

TASK [Install MariaDB Package (CentOS)] *******************************************************************
ok: [192.168.88.29]
ok: [192.168.245.132]

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

TASK [Gathering Facts] ************************************************************************************
ok: [192.168.245.131]
ok: [192.168.88.16]

PLAY RECAP ************************************************************************************************
192.168.245.130            : ok=3    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.245.131            : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.245.132            : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.88.10              : ok=3    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.88.16              : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.88.29              : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

***This command runs the whole playbook while only fully running plays with the “centos” tag, meaning that operations associated to the tag will only run (Install Apache2 and PHP and Install MariaDB). Noticeably, plays that do not have the specified tag only proceeds to conduct the “Gathering Facts” play, which is guaranteed in each run of a play.***

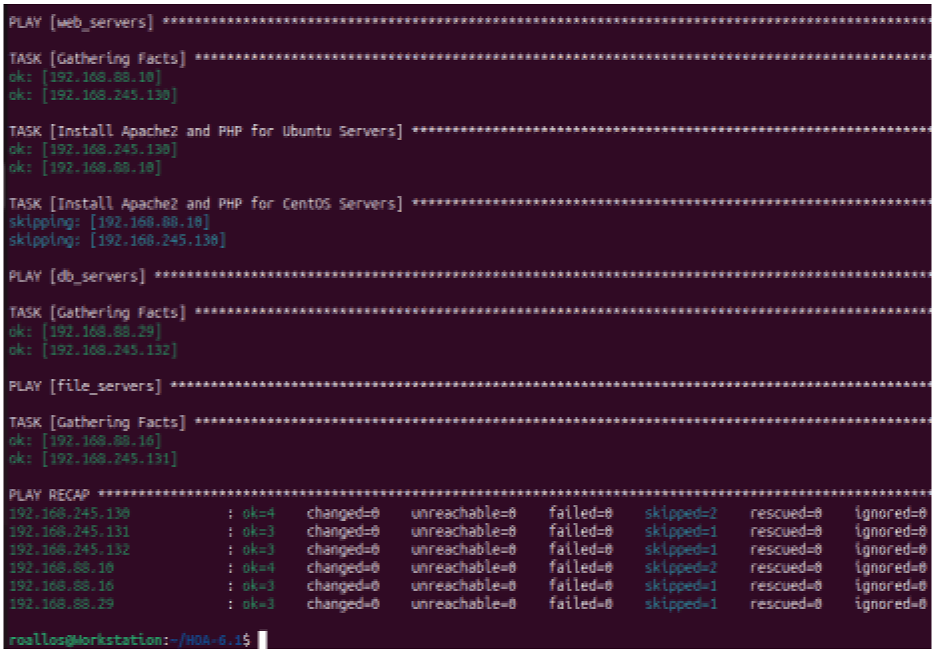
2.3 *ansible-playbook --tags db --ask-become-pass site.yml*



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

TASK [Gathering Facts] ********************************************************************************
ok: [192.168.245.130]
ok: [192.168.88.10]

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

TASK [Gathering Facts] ********************************************************************************
ok: [192.168.88.29]
ok: [192.168.245.132]

TASK [Install MariaDB Package (CentOS)] ***************************************************************
ok: [192.168.88.29]
ok: [192.168.245.132]

TASK [Install MariaDB Package (Ubuntu)] ***************************************************************
skipping: [192.168.88.29]
skipping: [192.168.245.132]

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

TASK [Gathering Facts] ********************************************************************************
ok: [192.168.88.16]
ok: [192.168.245.131]

PLAY RECAP ********************************************************************************************
192.168.245.130            : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.245.131            : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.245.132            : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.88.10              : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.88.16              : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.88.29              : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0

roallos@Workstation:~/HOA-6.1$ 
```

***The 'db' tag in the playbook is located under the the 3rd play, which is targeted for managed nodes within the "db_servers" group. As shown, only plays/operations tagged with the 'db' tag would be fully run.***

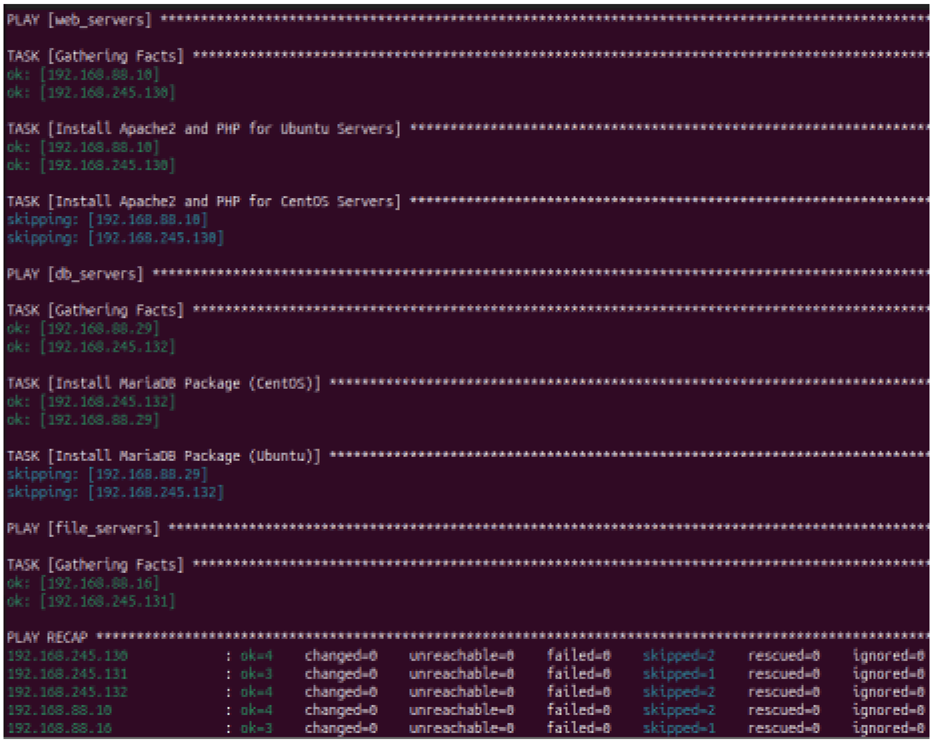
2.4 *ansible-playbook --tags apache --ask-become-pass site.yml*



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

TASK [Gathering Facts] ********************************************************************************
ok: [192.168.88.10]
ok: [192.168.245.130]

TASK [Install Apache2 and PHP for Ubuntu Servers] *****************************************************
ok: [192.168.245.130]
ok: [192.168.88.10]

TASK [Install Apache2 and PHP for CentOS Servers] *****************************************************
skipping: [192.168.88.10]
skipping: [192.168.245.130]

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

TASK [Gathering Facts] ********************************************************************************
ok: [192.168.88.29]
ok: [192.168.245.132]

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

TASK [Gathering Facts] ********************************************************************************
ok: [192.168.88.16]
ok: [192.168.245.131]

PLAY RECAP ********************************************************************************************
192.168.245.130            : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.245.131            : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.245.132            : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.88.10              : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.88.16              : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.88.29              : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

roallos@Workstation:~/HOA-6.1$ 
```

***A similar scenario with the ‘db’ tag, the play tagged with the ‘apache’ tag (2nd play) will only run. This would mean that the plays that installs Apache2 and PHP to Ubuntu and CentOS systems will run.***

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

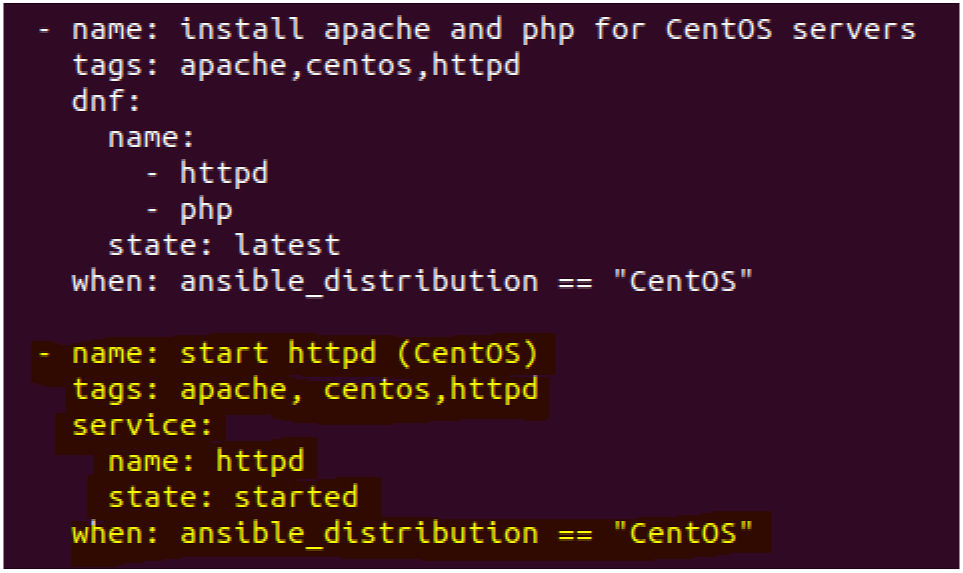


```
PLAY [web_servers] ***************************************************************************************

TASK [Gathering Facts] ***********************************************************************************
ok: [192.168.88.10]
ok: [192.168.245.130]

TASK [Install Apache2 and PHP for Ubuntu Servers] ********************************************************
ok: [192.168.88.10]
ok: [192.168.245.130]

TASK [Install Apache2 and PHP for CentOS Servers] ********************************************************
skipping: [192.168.88.10]
skipping: [192.168.245.130]

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

TASK [Gathering Facts] ***********************************************************************************
ok: [192.168.88.29]
ok: [192.168.245.132]

TASK [Install MariaDB Package (CentOS)] ******************************************************************
ok: [192.168.245.132]
ok: [192.168.88.29]

TASK [Install MariaDB Package (Ubuntu)] ******************************************************************
skipping: [192.168.88.29]
skipping: [192.168.245.132]

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

TASK [Gathering Facts] ***********************************************************************************
ok: [192.168.88.16]
ok: [192.168.245.131]

PLAY RECAP ***********************************************************************************************
192.168.245.130            : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.245.131            : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.245.132            : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.88.10              : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.88.16              : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

***This shows the method of running plays from a playbook by specifying multiple tags that may exist in different plays. In this case, both 2nd (Installs Apache2 and PHP on both types of systems) and 3rd (Installs MariaDB on both types of systems) plays within the “site.yaml” playbook will run.***

**Task 3: Managing Services**

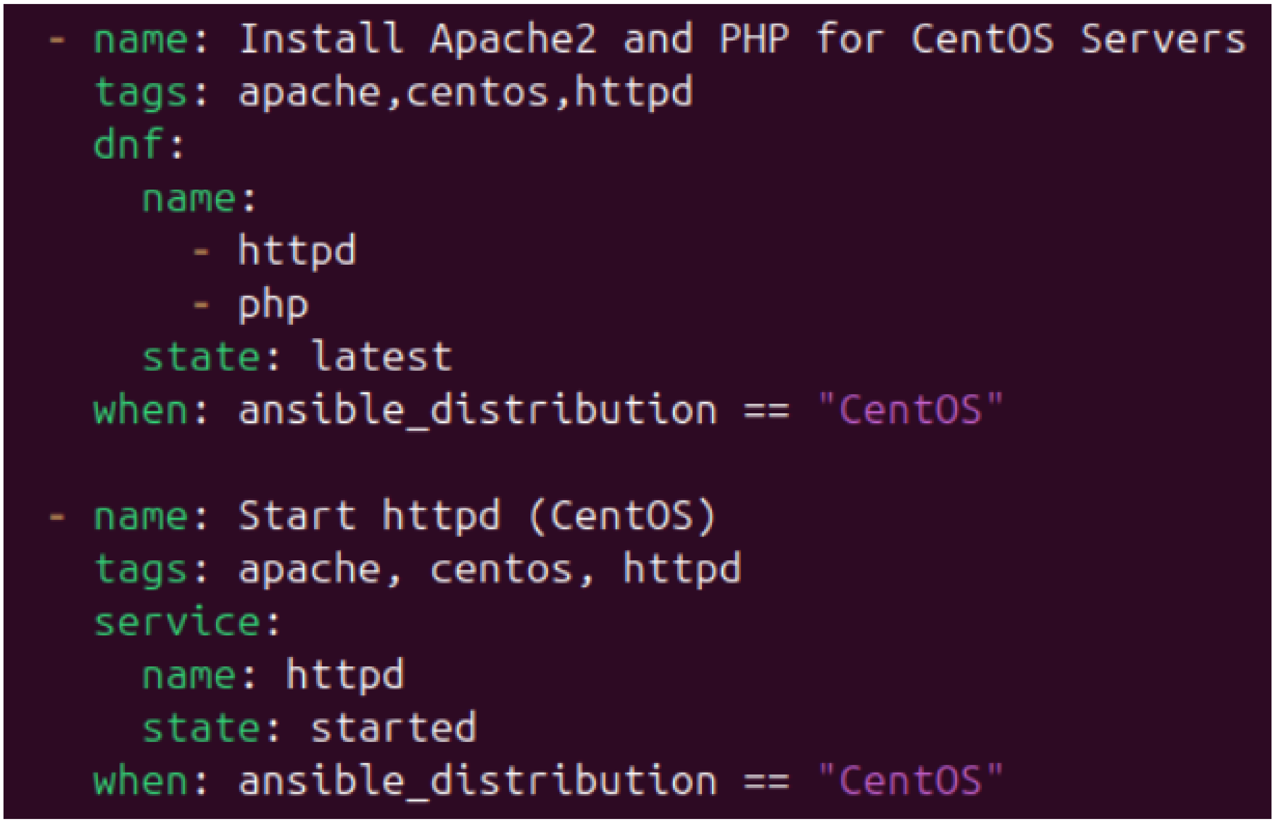
1. Edit the file site.yml and add a play that will automatically start the httpd on CentOS server.



```
- 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"
```

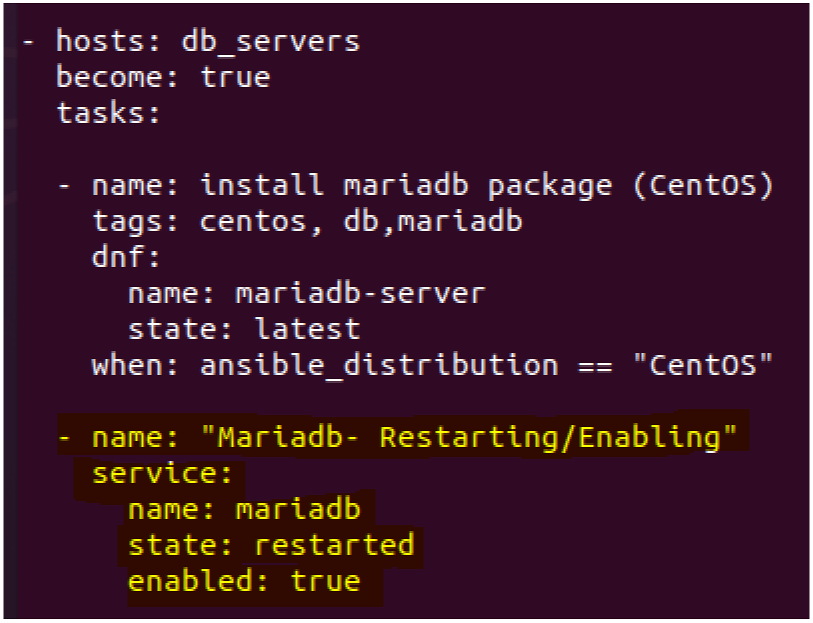
Figure 3.1.1

Make sure to save the file and exit.



```
- name: Install Apache2 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"
```

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

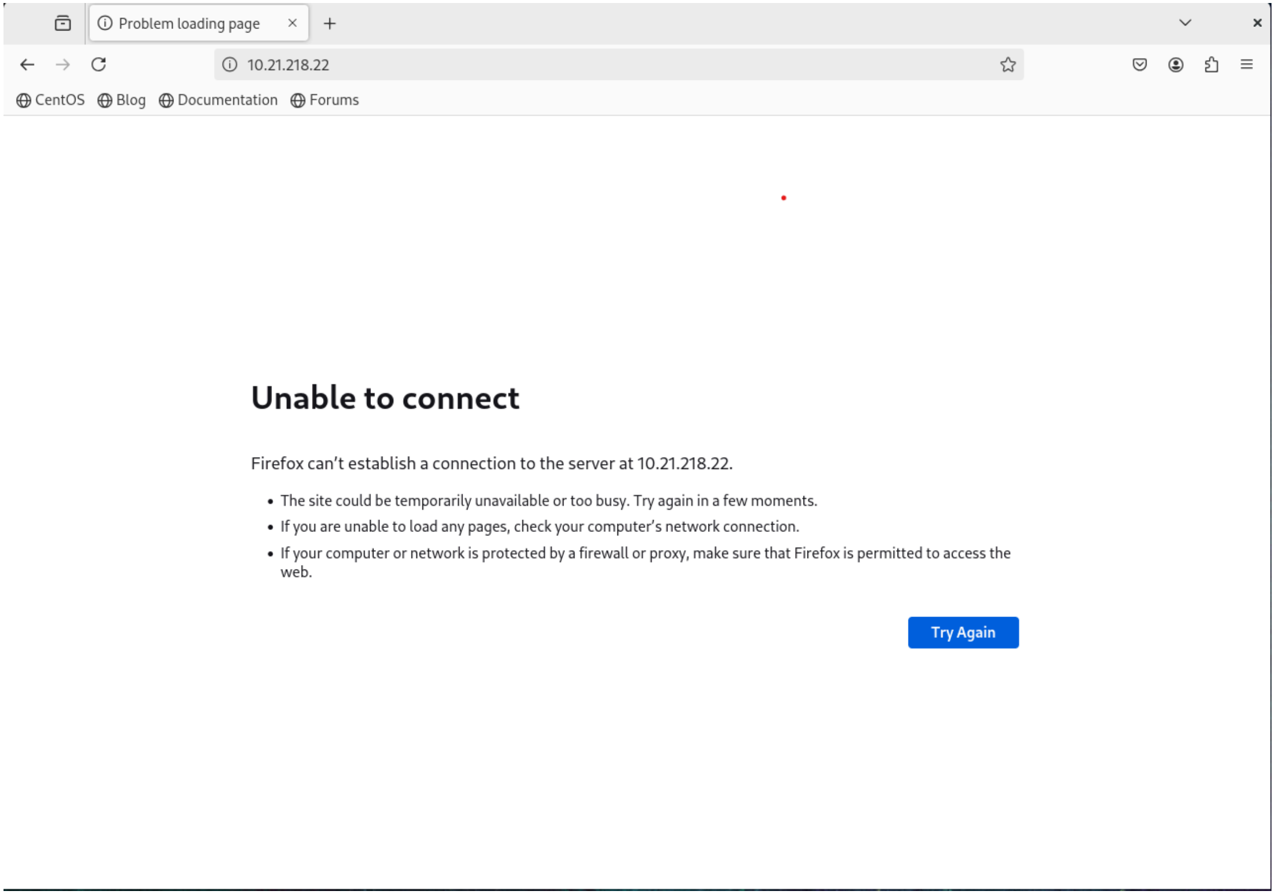


```
- hosts: db_servers
  become: true
  tasks:

  - 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.



Problem loading page

10.21.218.22

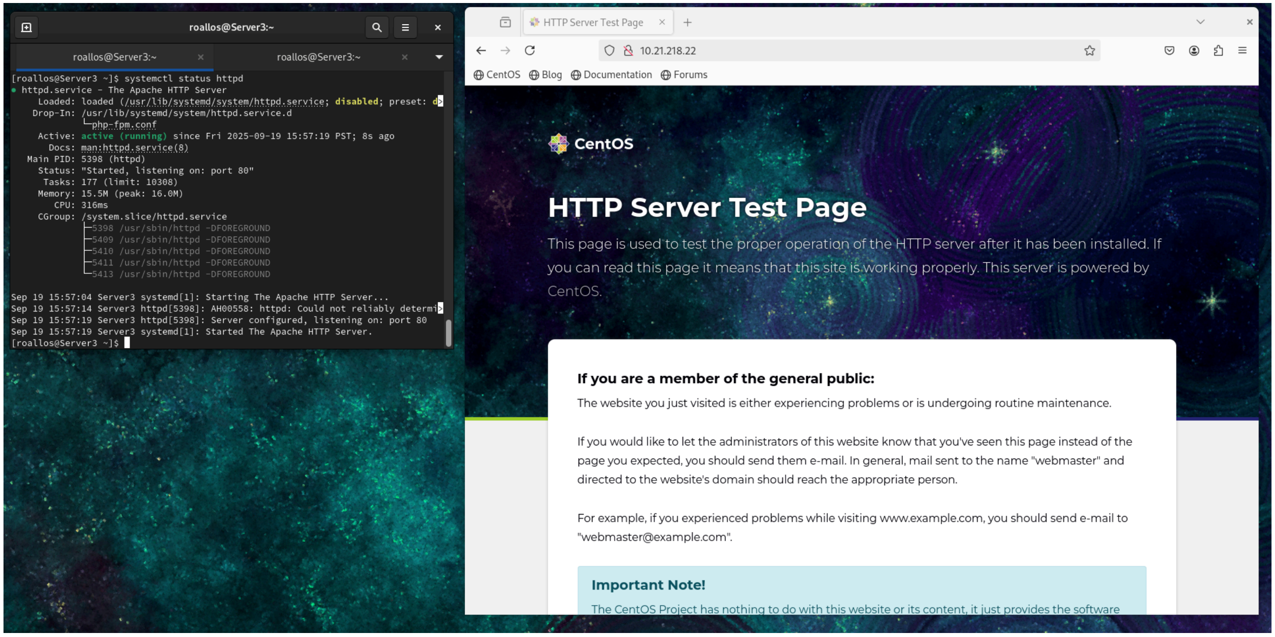
Unable to connect

Firefox can't establish a connection to the server at 10.21.218.22.

- 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.

***The playbook output shows that the installation of MariaDB and the restart/enable play was also successful. Upon checking from the CentOS machine, the restart was successful and I was able to load the HTTP page.***



```
[roallos@Server3 ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: d>
    Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: active (running) since Fri 2025-09-19 15:57:19 PST; 8s ago
       Docs: man:httpd.service(8)
   Main PID: 5398 (httpd)
     Status: "Started, listening on: port 80"
      Tasks: 177 (limit: 10308)
     Memory: 15.5M (peak: 16.0M)
        CPU: 316ms
     CGroup: /system.slice/httpd.service
             ├─5398 /usr/sbin/httpd -DFOREGROUND
             ├─5409 /usr/sbin/httpd -DFOREGROUND
             ├─5410 /usr/sbin/httpd -DFOREGROUND
             ├─5411 /usr/sbin/httpd -DFOREGROUND
             └─5413 /usr/sbin/httpd -DFOREGROUND

Sep 19 15:57:04 Server3 systemd[1]: Starting The Apache HTTP Server...
Sep 19 15:57:14 Server3 httpd[5398]: AH00558: httpd: Could not reliably determi>
Sep 19 15:57:19 Server3 httpd[5398]: Server configured, listening on: port 80
Sep 19 15:57:19 Server3 systemd[1]: Started The Apache HTTP Server.
[roallos@Server3 ~]$ 
```

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 can read this 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 should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

**Important Note!**

The CentOS Project has nothing to do with this website or its content, it just provides the software

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.

**Reflections:**

Answer the following:

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

   ***The practice of classifying and grouping remote servers is to be able to run certain plays onto their system. Having a defined group for each can make administrators to identify the purpose of the host in the network.***

2. What is the importance of tags in playbooks?

   ***Tags in Ansible playbooks classify plays further within a playbook. This lets the administrators run specific plays with ('--tags') or without ('--skip-tags') a specific tag.***

3. Why do think some services need to be managed automatically in playbooks?

   ***As services in certain Linux distros (in this situation, CentOS) do not have automatic initialization, we need to include an Ansible play to be able to interact with these services and enable or start them up, ensuring that it is working in***

***the managed node to be able to conduct the necessary processes that may be vital for the system.***