| | |
|---|---|
| **Name: Jozette Fermin** | **Date Performed: October 2, 2023** |
| **Course/Section: CPE31S4** | **Date Submitted: October 2, 2023** |
| **Instructor: Engr. Jonathan Taylar** | **Semester and SY: 1st sem 2023-2024** |

## Activity 6: Targeting Specific Nodes and Managing Services

1. **Objectives:**

1.1 Individualize hosts

1.2 Apply tags in selecting plays to run

1.3 Managing Services from remote servers using playbooks

2. **Discussion**:

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

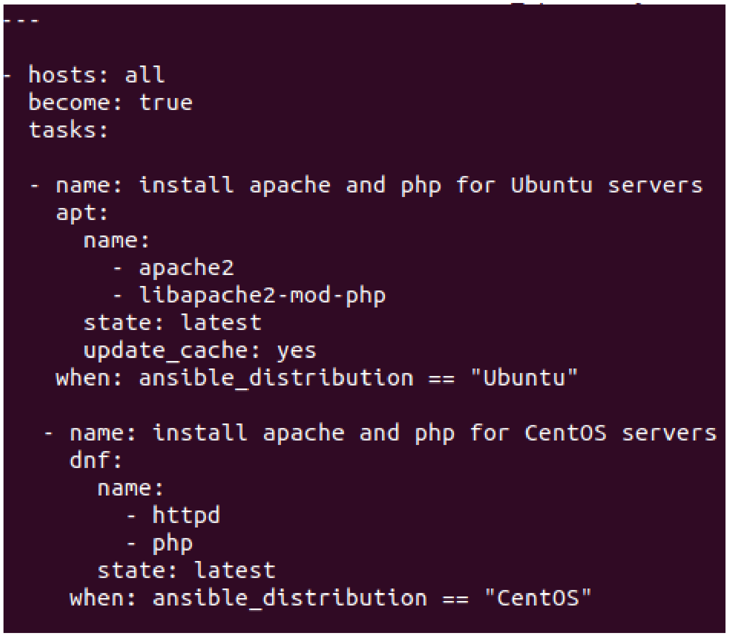
We also try to manage services that do not automatically run using the automations in playbook. For example, when we install web servers or httpd for CentOS, we notice that the service did not start automatically.

**Requirement:**

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

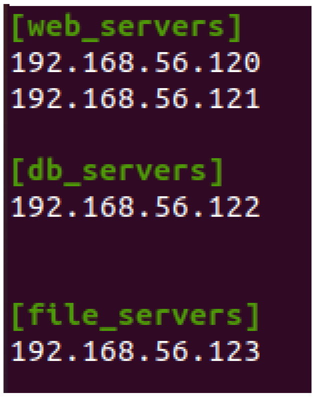
**Task 1: Targeting Specific Nodes**

1. Create a new playbook and named it site.yml. Follow the commands as shown in the image below. Make sure to save the file and exit.



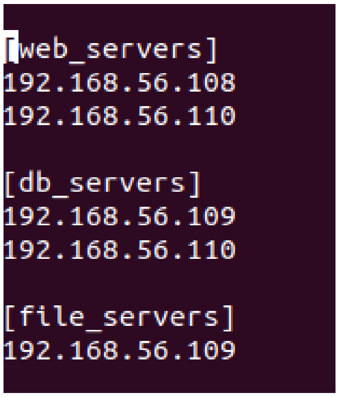
```
---

- hosts: all
  become: true
  tasks:

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

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

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
```

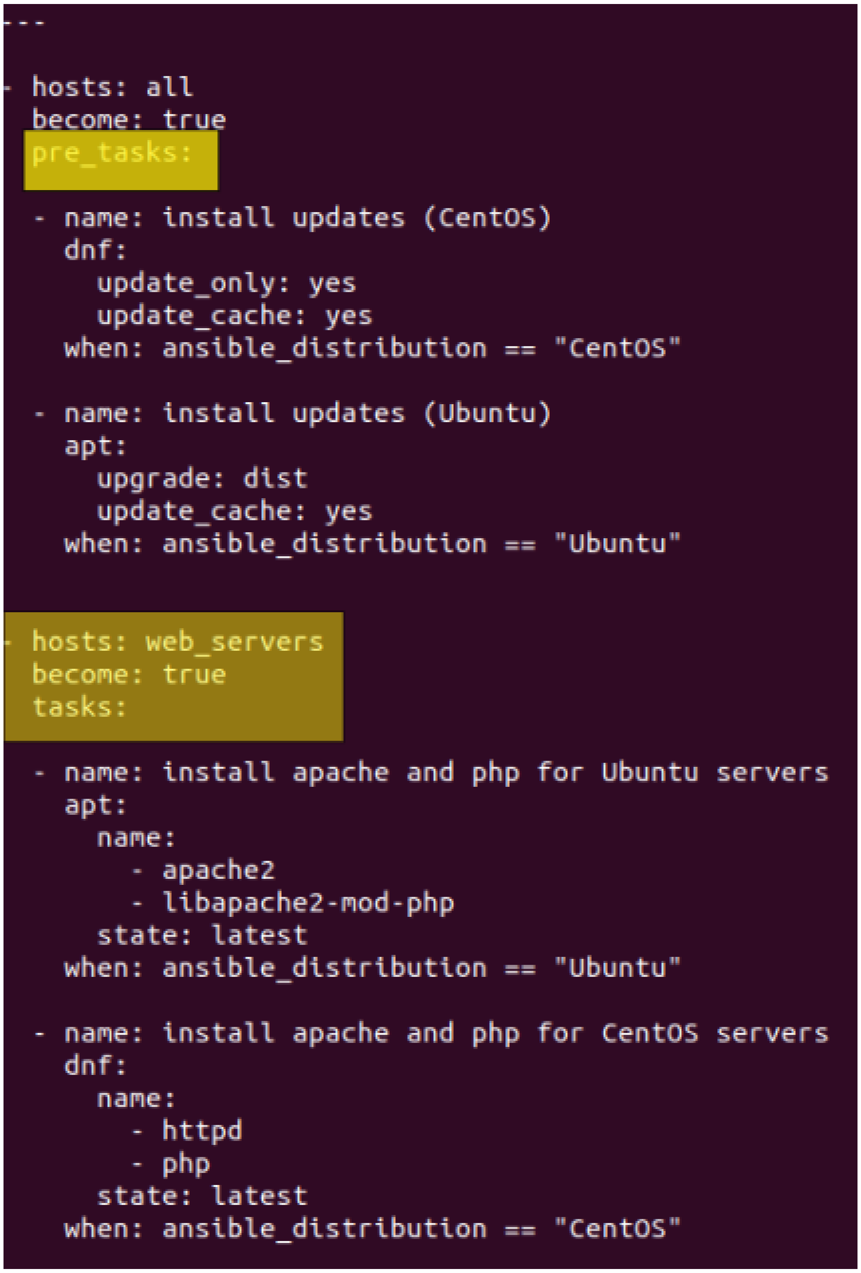
Make sure to save the file and exit.



```
[web_servers]
192.168.56.108
192.168.56.110

[db_servers]
192.168.56.109
192.168.56.110

[file_servers]
192.168.56.109
```

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.

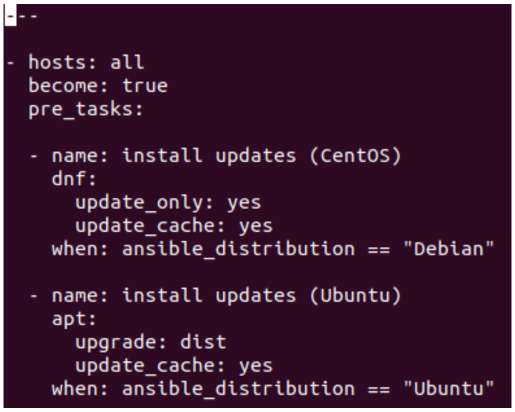
3. Edit the *site.yml* by following the image below:



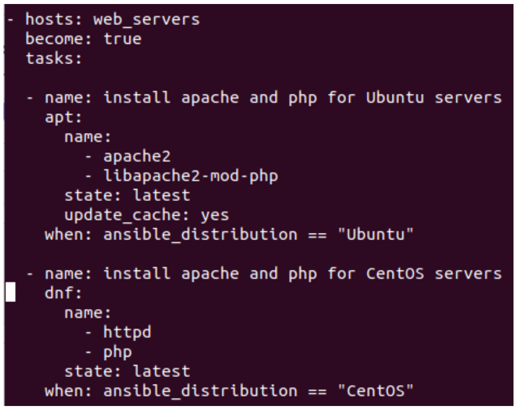
```
---

- 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.

Input

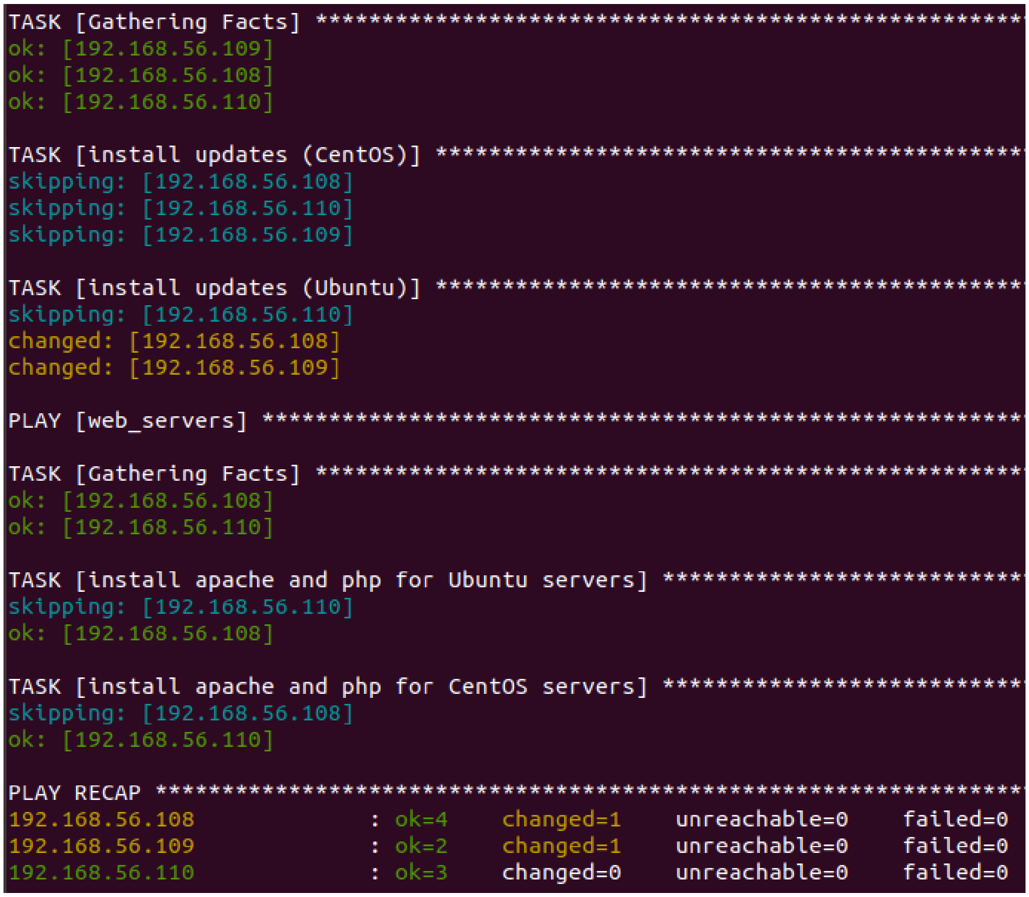


```
---

- hosts: all
  become: true
  pre_tasks:

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

  - 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
      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"
```

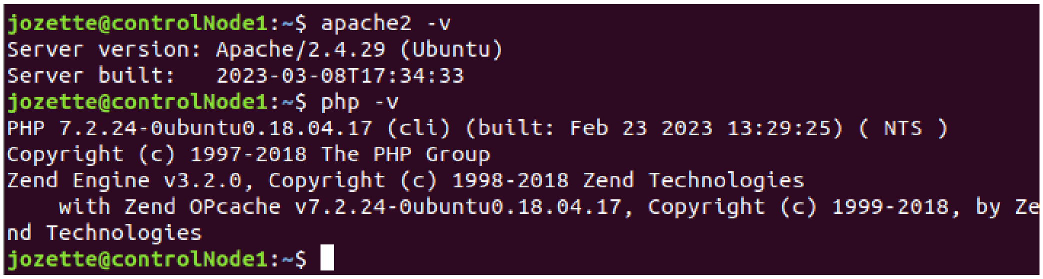
Process



```
TASK [Gathering Facts] *****************************************************
ok: [192.168.56.109]
ok: [192.168.56.108]
ok: [192.168.56.110]

TASK [install updates (CentOS)] ********************************************
skipping: [192.168.56.108]
skipping: [192.168.56.110]
skipping: [192.168.56.109]

TASK [install updates (Ubuntu)] ********************************************
skipping: [192.168.56.110]
changed: [192.168.56.108]
changed: [192.168.56.109]

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

TASK [Gathering Facts] *****************************************************
ok: [192.168.56.108]
ok: [192.168.56.110]

TASK [install apache and php for Ubuntu servers] ***************************
skipping: [192.168.56.110]
ok: [192.168.56.108]

TASK [install apache and php for CentOS servers] ***************************
skipping: [192.168.56.108]
ok: [192.168.56.110]

PLAY RECAP *****************************************************************
192.168.56.108             : ok=4    changed=1    unreachable=0    failed=0
192.168.56.109             : ok=2    changed=1    unreachable=0    failed=0
192.168.56.110             : ok=3    changed=0    unreachable=0    failed=0
```

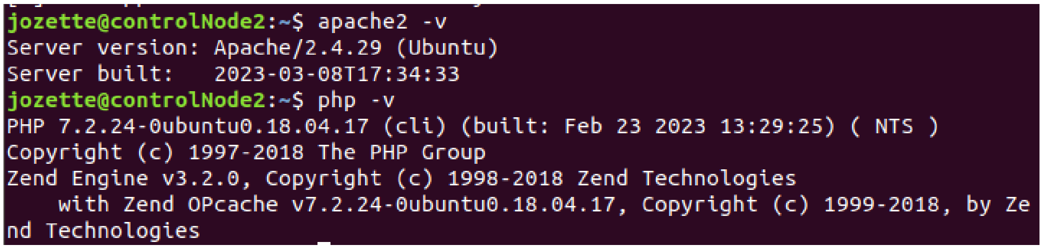
Output

Controlnode1



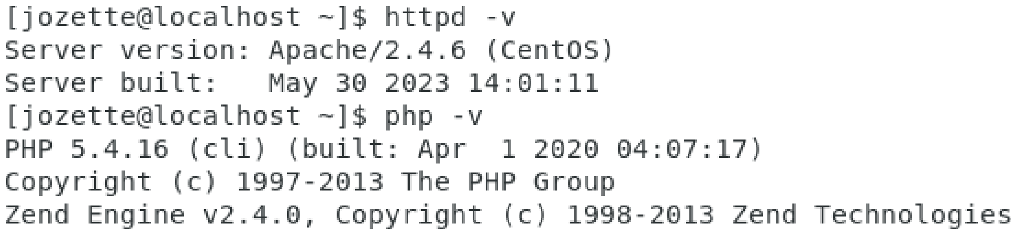
```
jozette@controlNode1:~$ apache2 -v
Server version: Apache/2.4.29 (Ubuntu)
Server built:   2023-03-08T17:34:33
jozette@controlNode1:~$ php -v
PHP 7.2.24-0ubuntu0.18.04.17 (cli) (built: Feb 23 2023 13:29:25) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
    with Zend OPcache v7.2.24-0ubuntu0.18.04.17, Copyright (c) 1999-2018, by Ze
nd Technologies
jozette@controlNode1:~$
```

Controlnode2



```
jozette@controlNode2:~$ apache2 -v
Server version: Apache/2.4.29 (Ubuntu)
Server built:   2023-03-08T17:34:33
jozette@controlNode2:~$ php -v
PHP 7.2.24-0ubuntu0.18.04.17 (cli) (built: Feb 23 2023 13:29:25) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
    with Zend OPcache v7.2.24-0ubuntu0.18.04.17, Copyright (c) 1999-2018, by Ze
nd Technologies
```

CentOS

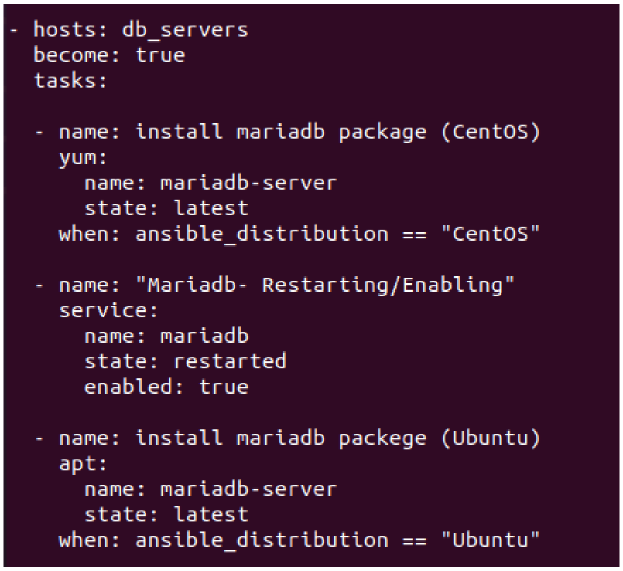


```
[jozette@localhost ~]$ httpd -v
Server version: Apache/2.4.6 (CentOS)
Server built:   May 30 2023 14:01:11
[jozette@localhost ~]$ php -v
PHP 5.4.16 (cli) (built: Apr  1 2020 04:07:17)
Copyright (c) 1997-2013 The PHP Group
Zend Engine v2.4.0, Copyright (c) 1998-2013 Zend Technologies
```

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.

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



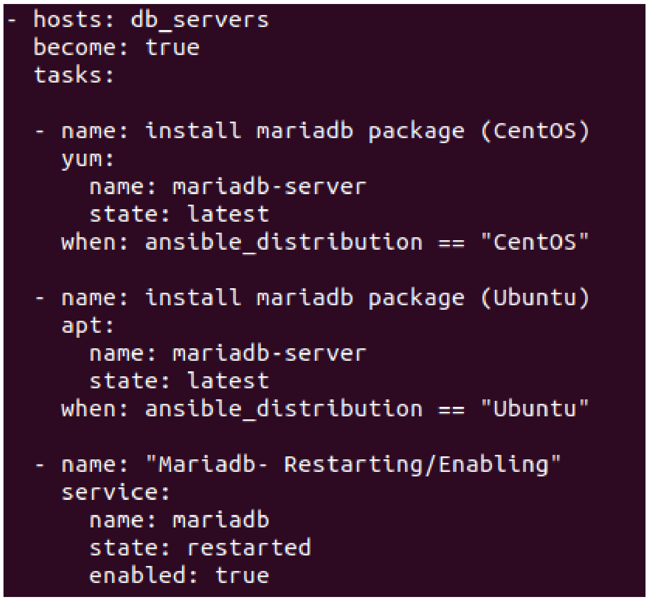
```
- hosts: db_servers
  become: true
  tasks:

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

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

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

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

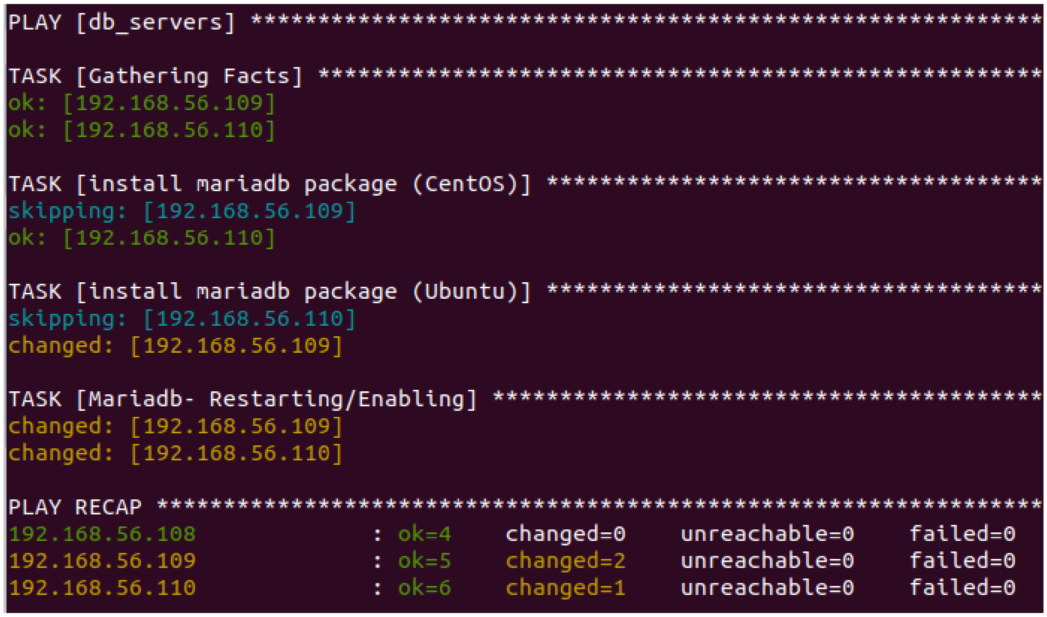
5. Go to the remote server (Ubuntu) terminal that belongs to the db_servers group and check the status for mariadb installation using the command: *systemctl status mariadb.* Do this on the CentOS server also.
Describe the output.

Input

Process



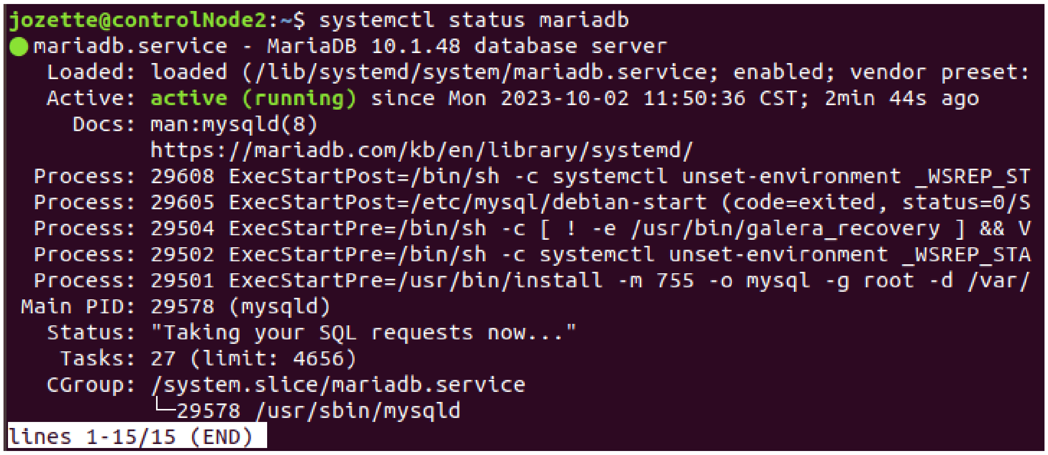
```
- hosts: db_servers
  become: true
  tasks:

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

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

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



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

TASK [Gathering Facts] ******************************************************
ok: [192.168.56.109]
ok: [192.168.56.110]

TASK [install mariadb package (CentOS)] *************************************
skipping: [192.168.56.109]
ok: [192.168.56.110]

TASK [install mariadb package (Ubuntu)] *************************************
skipping: [192.168.56.110]
changed: [192.168.56.109]

TASK [Mariadb- Restarting/Enabling] *****************************************
changed: [192.168.56.109]
changed: [192.168.56.110]

PLAY RECAP ******************************************************************
192.168.56.108             : ok=4    changed=0    unreachable=0    failed=0
192.168.56.109             : ok=5    changed=2    unreachable=0    failed=0
192.168.56.110             : ok=6    changed=1    unreachable=0    failed=0
```

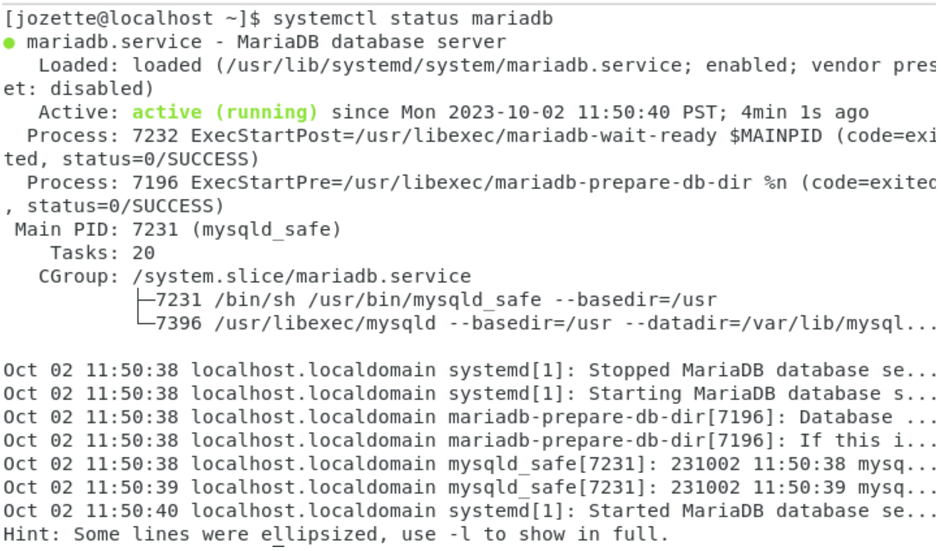
Output

**ControlNode 2**



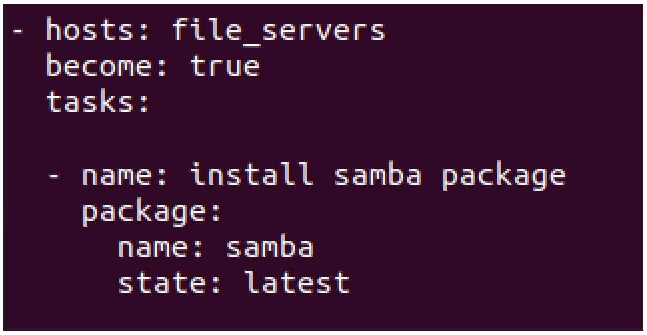
```
jozette@controlNode2:~$ systemctl status mariadb
● mariadb.service - MariaDB 10.1.48 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset:
   Active: active (running) since Mon 2023-10-02 11:50:36 CST; 2min 44s ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
  Process: 29608 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_ST
  Process: 29605 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/S
  Process: 29504 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && V
  Process: 29502 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_STA
  Process: 29501 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/
 Main PID: 29578 (mysqld)
   Status: "Taking your SQL requests now..."
    Tasks: 27 (limit: 4656)
   CGroup: /system.slice/mariadb.service
           └─29578 /usr/sbin/mysqld
lines 1-15/15 (END)
```

**CentOS**



```
[jozette@localhost ~]$ systemctl status mariadb
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor pres
et: disabled)
   Active: active (running) since Mon 2023-10-02 11:50:40 PST; 4min 1s ago
  Process: 7232 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exi
ted, status=0/SUCCESS)
  Process: 7196 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exitec
, status=0/SUCCESS)
 Main PID: 7231 (mysqld_safe)
    Tasks: 20
   CGroup: /system.slice/mariadb.service
           ├─7231 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
           └─7396 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql...

Oct 02 11:50:38 localhost.localdomain systemd[1]: Stopped MariaDB database se...
Oct 02 11:50:38 localhost.localdomain systemd[1]: Starting MariaDB database s...
Oct 02 11:50:38 localhost.localdomain mariadb-prepare-db-dir[7196]: Database ...
Oct 02 11:50:38 localhost.localdomain mariadb-prepare-db-dir[7196]: If this i...
Oct 02 11:50:38 localhost.localdomain mysqld_safe[7231]: 231002 11:50:38 mysq...
Oct 02 11:50:39 localhost.localdomain mysqld_safe[7231]: 231002 11:50:39 mysq...
Oct 02 11:50:40 localhost.localdomain systemd[1]: Started MariaDB database se...
Hint: Some lines were ellipsized, use -l to show in full.
```

6. Edit the *site.yml* again. This time we will append the code to configure installation on the *file_servers* group. We can add the following on our file.

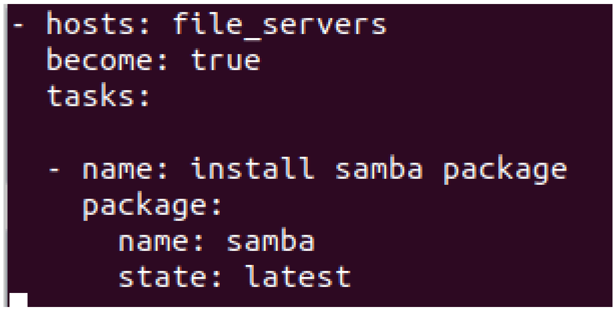


```
- hosts: file_servers
  become: true
  tasks:

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

Make sure to save the file and exit.

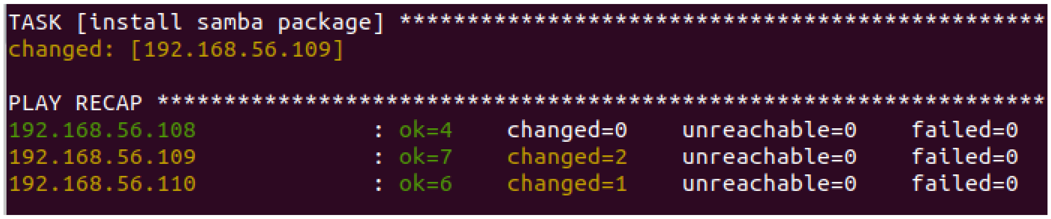
Run the *site.yml* file and describe the result.

Input



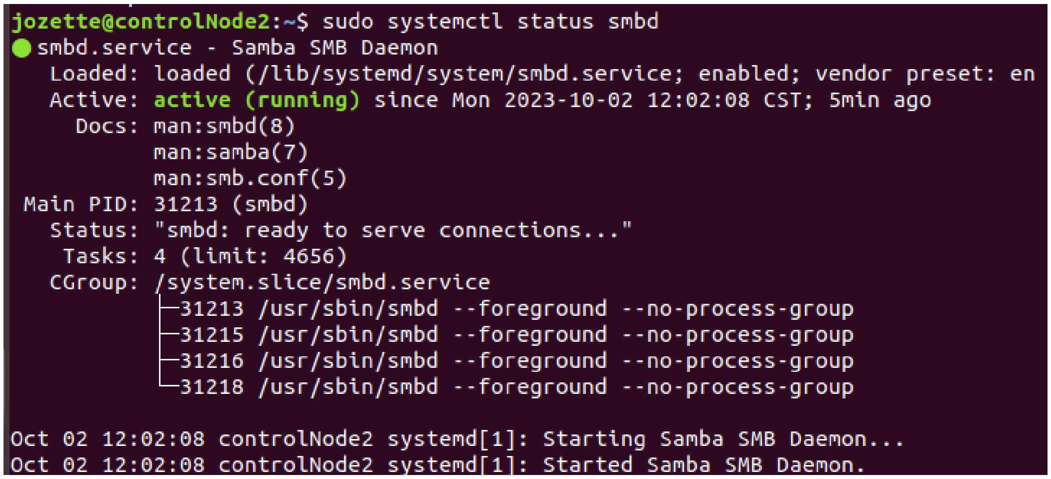
```
- hosts: file_servers
  become: true
  tasks:

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

Process



```
TASK [install samba package] ****************************************************
changed: [192.168.56.109]

PLAY RECAP **********************************************************************
192.168.56.108             : ok=4    changed=0    unreachable=0    failed=0
192.168.56.109             : ok=7    changed=2    unreachable=0    failed=0
192.168.56.110             : ok=6    changed=1    unreachable=0    failed=0
```

Output



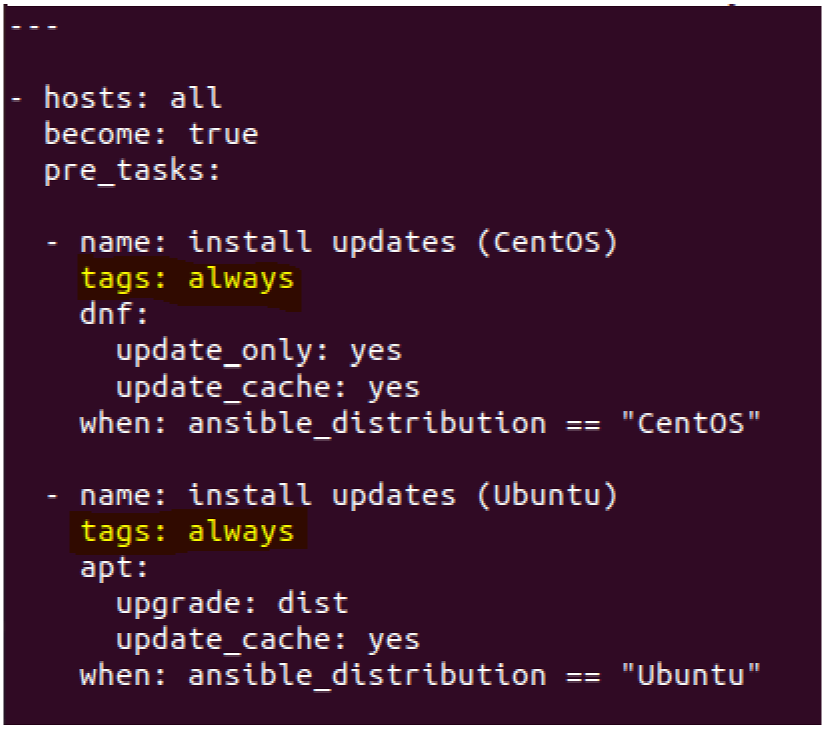
```
jozette@controlNode2:~$ sudo systemctl status smbd
● smbd.service - Samba SMB Daemon
   Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor preset: en
   Active: active (running) since Mon 2023-10-02 12:02:08 CST; 5min ago
     Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
 Main PID: 31213 (smbd)
   Status: "smbd: ready to serve connections..."
    Tasks: 4 (limit: 4656)
   CGroup: /system.slice/smbd.service
           ├─31213 /usr/sbin/smbd --foreground --no-process-group
           ├─31215 /usr/sbin/smbd --foreground --no-process-group
           ├─31216 /usr/sbin/smbd --foreground --no-process-group
           └─31218 /usr/sbin/smbd --foreground --no-process-group

Oct 02 12:02:08 controlNode2 systemd[1]: Starting Samba SMB Daemon...
Oct 02 12:02:08 controlNode2 systemd[1]: Started Samba SMB Daemon.
```

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.

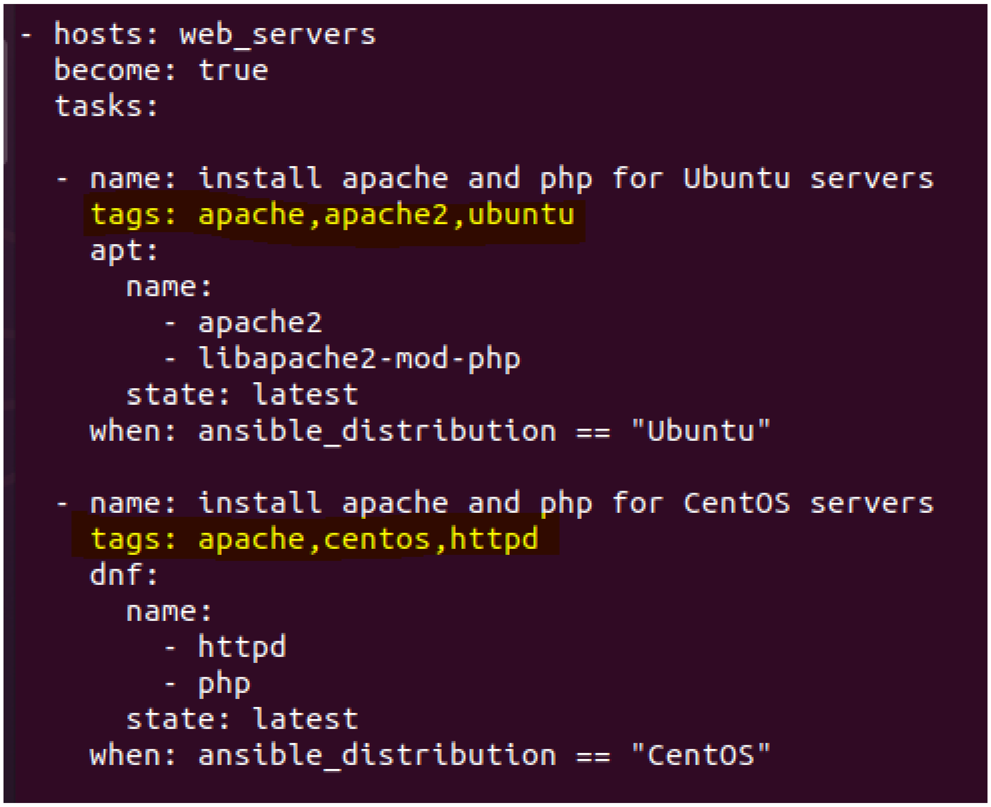
**Task 2: Using Tags in running playbooks**

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.

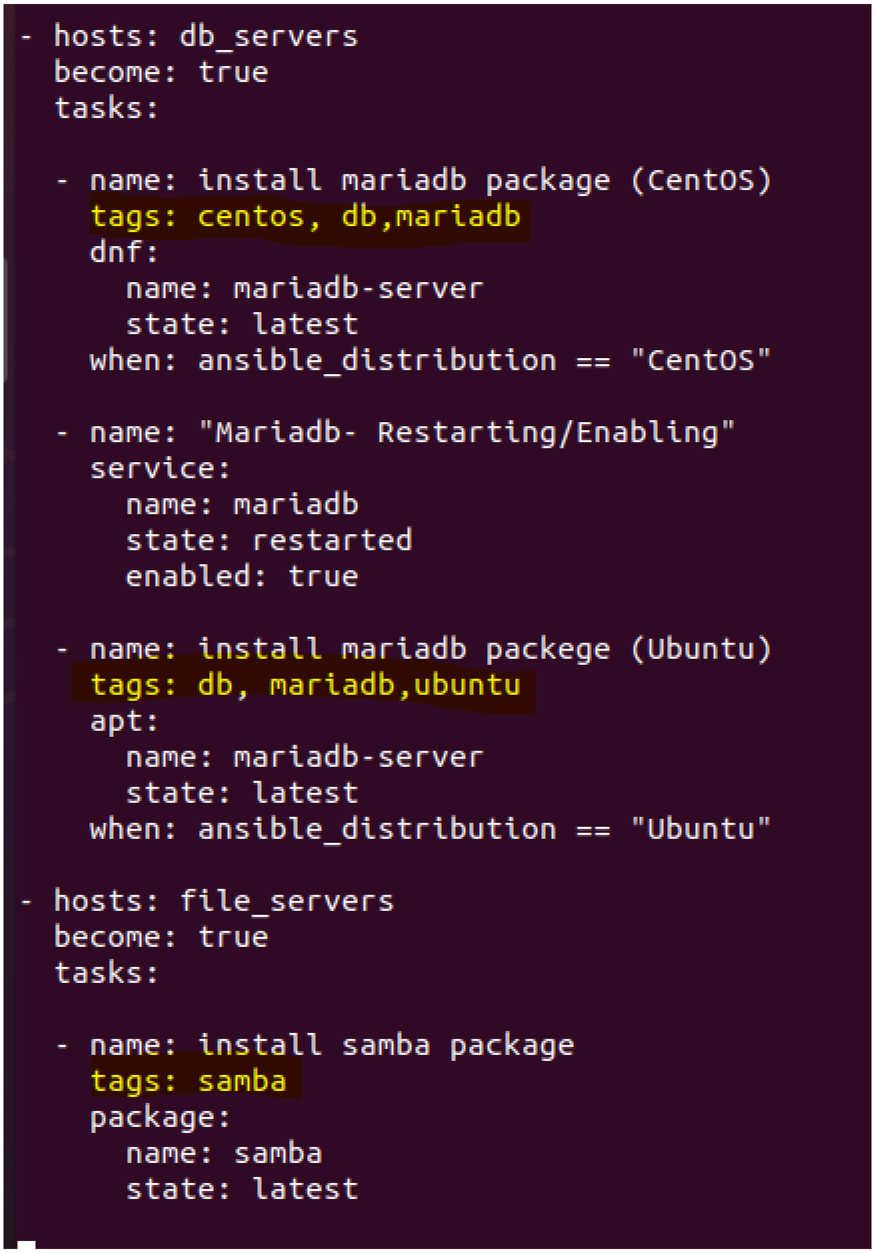
1. Edit the *site.yml* file. Add tags to the playbook. After the name, we can place the tags: *name_of_tag*. This is an arbitrary command, which means you can use any name for a tag.



```
---

- 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"
```



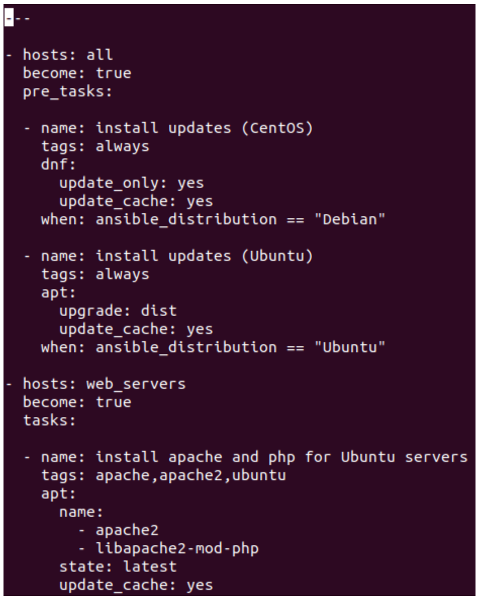
```
- hosts: web_servers
  become: true
  tasks:

  - 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"
```



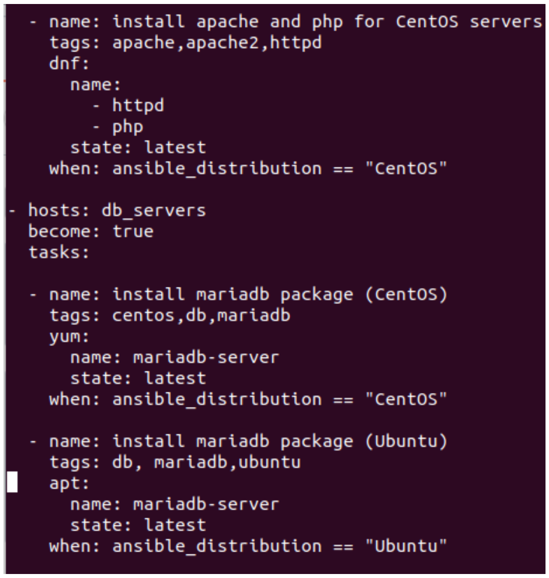
```
- hosts: db_servers
  become: true
  tasks:

  - 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.

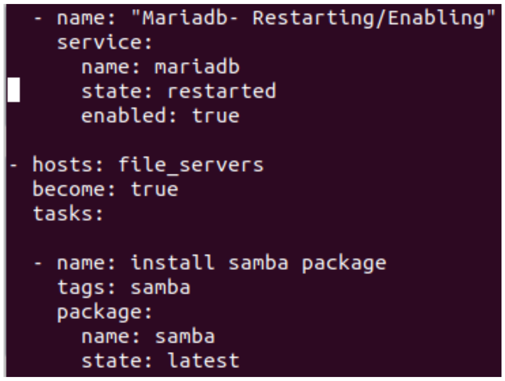
Input



```
---

- hosts: all
  become: true
  pre_tasks:

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

  - 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
      update_cache: yes
```

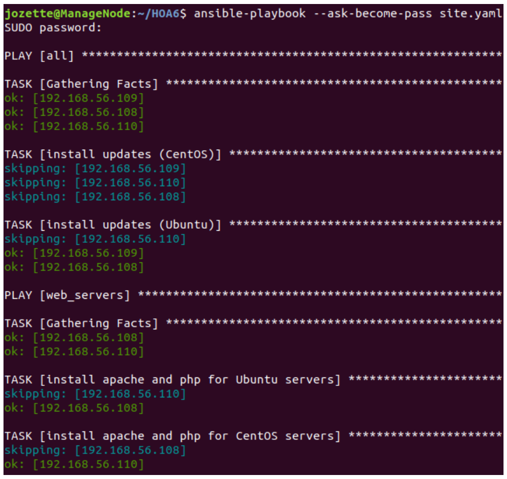


```
  - name: install apache and php for CentOS servers
    tags: apache,apache2,httpd
    dnf:
      name:
        - httpd
        - php
      state: latest
    when: ansible_distribution == "CentOS"

- hosts: db_servers
  become: true
  tasks:

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

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

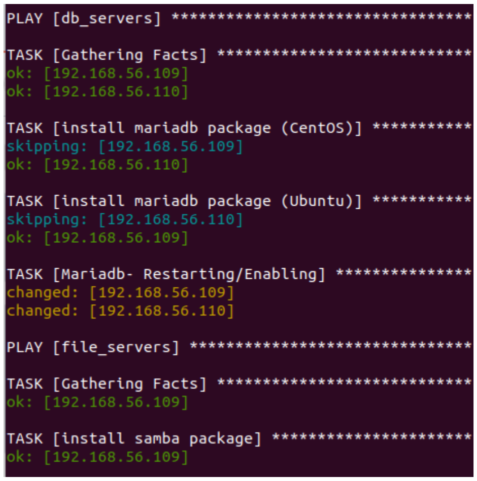


```
  - name: "Mariadb- Restarting/Enabling"
    service:
      name: mariadb
      state: restarted
      enabled: true

- hosts: file_servers
  become: true
  tasks:

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

Process

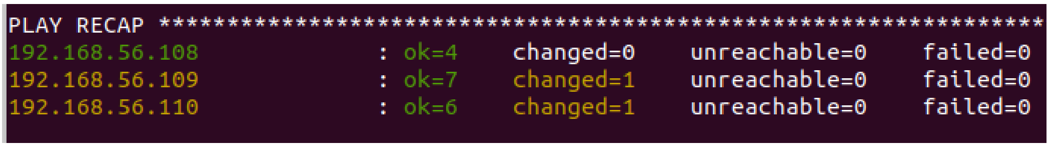


```
jozette@ManageNode:~/HOA6$ ansible-playbook --ask-become-pass site.yaml
SUDO password:

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

TASK [Gathering Facts] ***********************************************************
ok: [192.168.56.109]
ok: [192.168.56.108]
ok: [192.168.56.110]

TASK [install updates (CentOS)] **************************************************
skipping: [192.168.56.109]
skipping: [192.168.56.110]
skipping: [192.168.56.108]

TASK [install updates (Ubuntu)] **************************************************
skipping: [192.168.56.110]
ok: [192.168.56.109]
ok: [192.168.56.108]

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

TASK [Gathering Facts] ***********************************************************
ok: [192.168.56.108]
ok: [192.168.56.110]

TASK [install apache and php for Ubuntu servers] *********************************
skipping: [192.168.56.110]
ok: [192.168.56.108]

TASK [install apache and php for CentOS servers] *********************************
skipping: [192.168.56.108]
ok: [192.168.56.110]
```



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

TASK [Gathering Facts] ******************************
ok: [192.168.56.109]
ok: [192.168.56.110]

TASK [install mariadb package (CentOS)] ***********
skipping: [192.168.56.109]
ok: [192.168.56.110]

TASK [install mariadb package (Ubuntu)] ***********
skipping: [192.168.56.110]
ok: [192.168.56.109]

TASK [Mariadb- Restarting/Enabling] ***************
changed: [192.168.56.109]
changed: [192.168.56.110]

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

TASK [Gathering Facts] ******************************
ok: [192.168.56.109]

TASK [install samba package] ***********************
ok: [192.168.56.109]
```

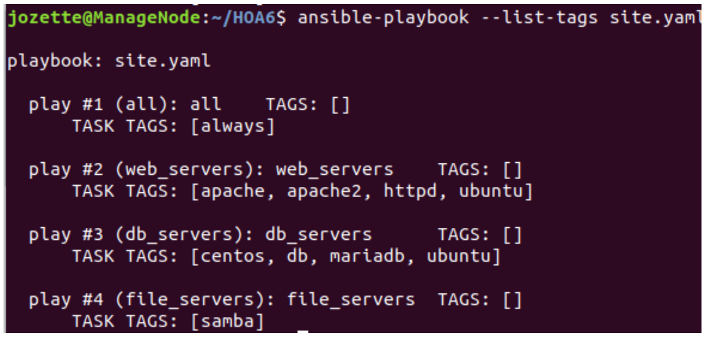
Output



```
PLAY RECAP **************************************************************
192.168.56.108             : ok=4    changed=0    unreachable=0    failed=0
192.168.56.109             : ok=7    changed=1    unreachable=0    failed=0
192.168.56.110             : ok=6    changed=1    unreachable=0    failed=0
```

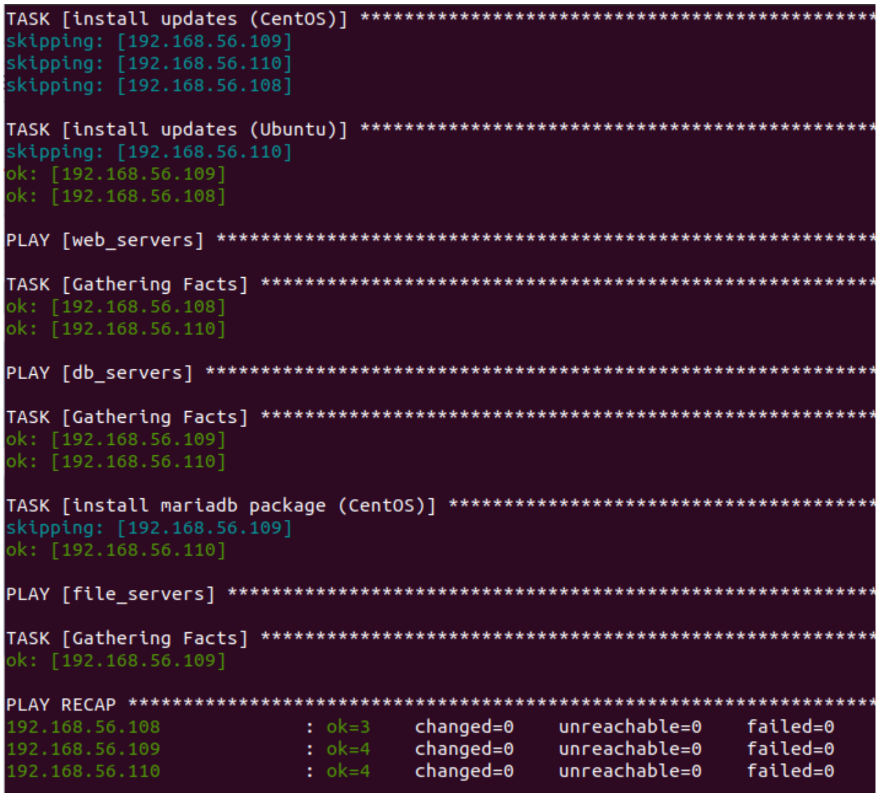
2. On the local machine, try to issue the following commands and describe each result:

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



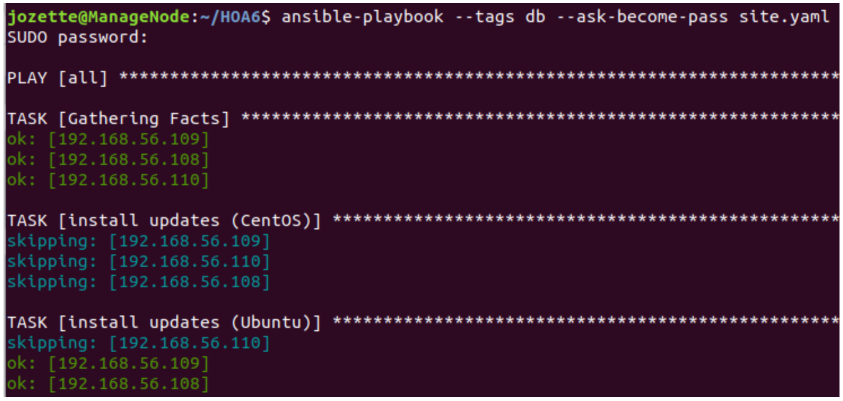
```
jozette@ManageNode:~/HOA6$ ansible-playbook --list-tags site.yaml

playbook: site.yaml

  play #1 (all): all    TAGS: []
      TASK TAGS: [always]

  play #2 (web_servers): web_servers    TAGS: []
      TASK TAGS: [apache, apache2, 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*

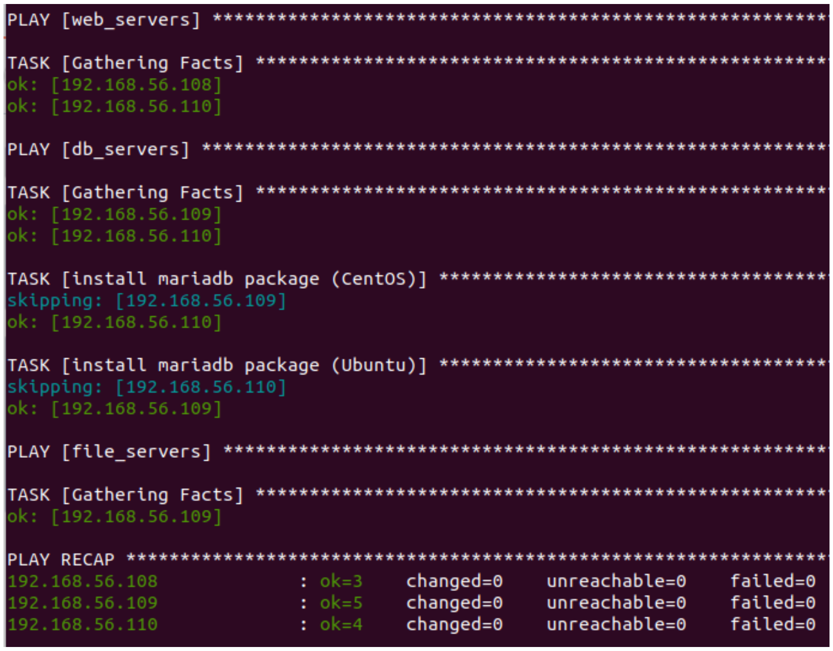


```
TASK [install updates (CentOS)] ***********************************************
skipping: [192.168.56.109]
skipping: [192.168.56.110]
skipping: [192.168.56.108]

TASK [install updates (Ubuntu)] ***********************************************
skipping: [192.168.56.110]
ok: [192.168.56.109]
ok: [192.168.56.108]

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

TASK [Gathering Facts] ********************************************************
ok: [192.168.56.108]
ok: [192.168.56.110]

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

TASK [Gathering Facts] ********************************************************
ok: [192.168.56.109]
ok: [192.168.56.110]

TASK [install mariadb package (CentOS)] ***************************************
skipping: [192.168.56.109]
ok: [192.168.56.110]

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

TASK [Gathering Facts] ********************************************************
ok: [192.168.56.109]

PLAY RECAP ********************************************************************
192.168.56.108             : ok=3    changed=0    unreachable=0    failed=0
192.168.56.109             : ok=4    changed=0    unreachable=0    failed=0
192.168.56.110             : ok=4    changed=0    unreachable=0    failed=0
```

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

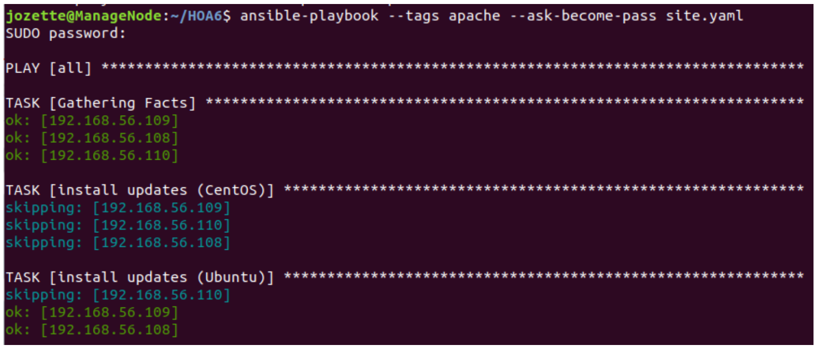


```
jozette@ManageNode:~/HOA6$ ansible-playbook --tags db --ask-become-pass site.yaml
SUDO password:

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

TASK [Gathering Facts] *************************************************
ok: [192.168.56.109]
ok: [192.168.56.108]
ok: [192.168.56.110]

TASK [install updates (CentOS)] ****************************************
skipping: [192.168.56.109]
skipping: [192.168.56.110]
skipping: [192.168.56.108]

TASK [install updates (Ubuntu)] ****************************************
skipping: [192.168.56.110]
ok: [192.168.56.109]
ok: [192.168.56.108]
```

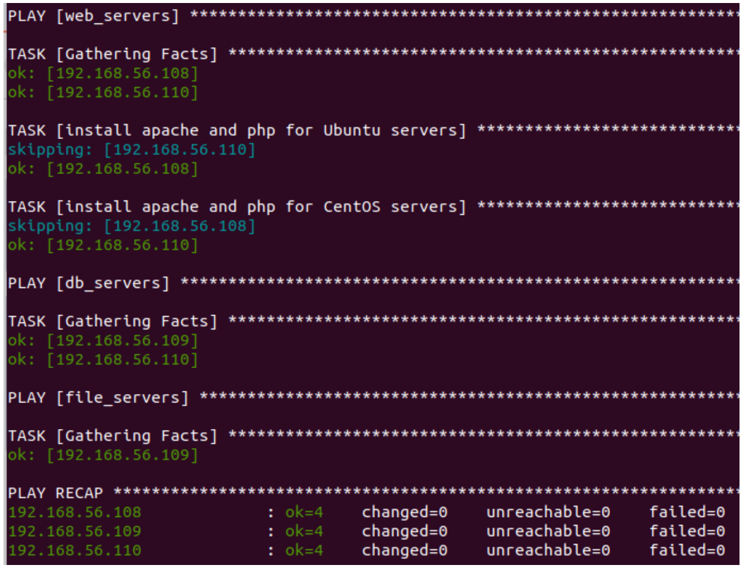


```
PLAY [web_servers] *****************************************************

TASK [Gathering Facts] *************************************************
ok: [192.168.56.108]
ok: [192.168.56.110]

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

TASK [Gathering Facts] *************************************************
ok: [192.168.56.109]
ok: [192.168.56.110]

TASK [install mariadb package (CentOS)] ********************************
skipping: [192.168.56.109]
ok: [192.168.56.110]

TASK [install mariadb package (Ubuntu)] ********************************
skipping: [192.168.56.110]
ok: [192.168.56.109]

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

TASK [Gathering Facts] *************************************************
ok: [192.168.56.109]

PLAY RECAP *************************************************************
192.168.56.108             : ok=3    changed=0    unreachable=0    failed=0
192.168.56.109             : ok=5    changed=0    unreachable=0    failed=0
192.168.56.110             : ok=4    changed=0    unreachable=0    failed=0
```

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

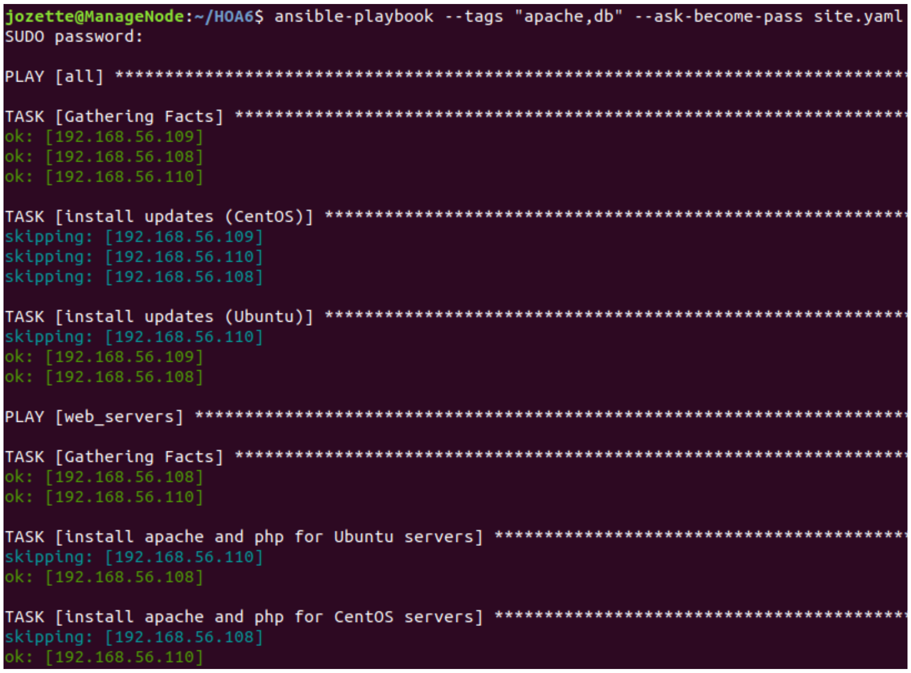


```
jozette@ManageNode:~/HOA6$ ansible-playbook --tags apache --ask-become-pass site.yaml
SUDO password:

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

TASK [Gathering Facts] *************************************************************
ok: [192.168.56.109]
ok: [192.168.56.108]
ok: [192.168.56.110]

TASK [install updates (CentOS)] ****************************************************
skipping: [192.168.56.109]
skipping: [192.168.56.110]
skipping: [192.168.56.108]

TASK [install updates (Ubuntu)] ****************************************************
skipping: [192.168.56.110]
ok: [192.168.56.109]
ok: [192.168.56.108]
```

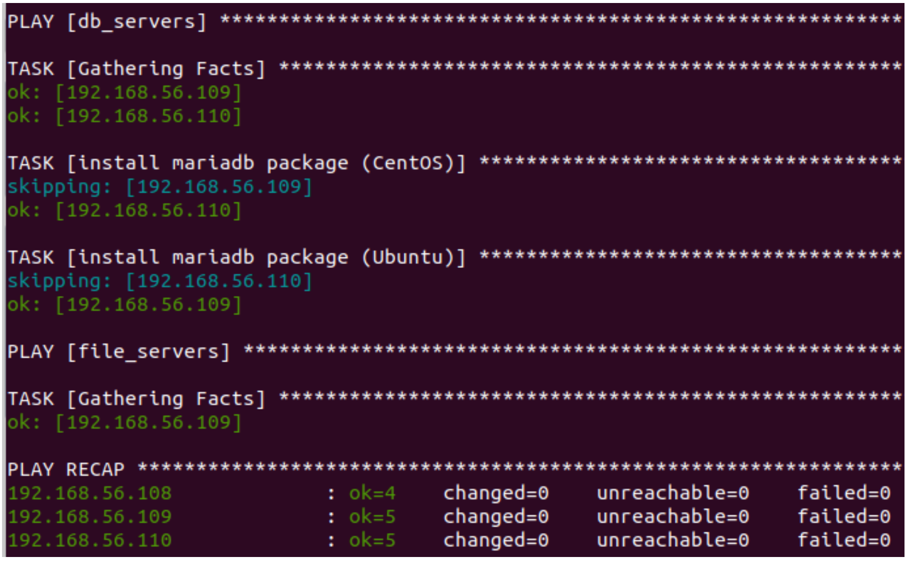


```
PLAY [web_servers] *********************************************************

TASK [Gathering Facts] *****************************************************
ok: [192.168.56.108]
ok: [192.168.56.110]

TASK [install apache and php for Ubuntu servers] ***************************
skipping: [192.168.56.110]
ok: [192.168.56.108]

TASK [install apache and php for CentOS servers] ***************************
skipping: [192.168.56.108]
ok: [192.168.56.110]

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

TASK [Gathering Facts] *****************************************************
ok: [192.168.56.109]
ok: [192.168.56.110]

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

TASK [Gathering Facts] *****************************************************
ok: [192.168.56.109]

PLAY RECAP *****************************************************************
192.168.56.108             : ok=4    changed=0    unreachable=0    failed=0
192.168.56.109             : ok=4    changed=0    unreachable=0    failed=0
192.168.56.110             : ok=4    changed=0    unreachable=0    failed=0
```

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



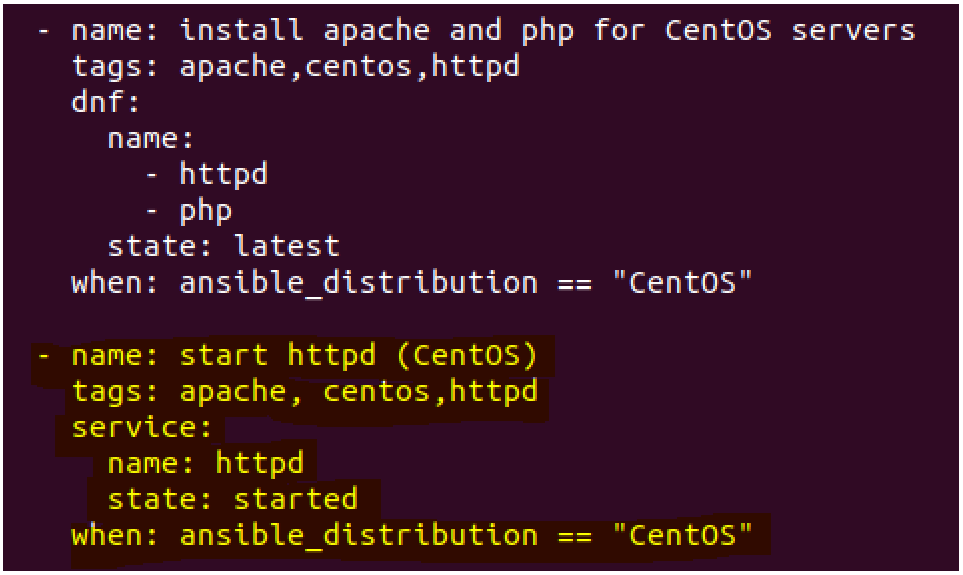
```
jozette@ManageNode:~/HOA6$ ansible-playbook --tags "apache,db" --ask-become-pass site.yaml
SUDO password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.109]
ok: [192.168.56.108]
ok: [192.168.56.110]

TASK [install updates (CentOS)] ************************************************
skipping: [192.168.56.109]
skipping: [192.168.56.110]
skipping: [192.168.56.108]

TASK [install updates (Ubuntu)] ************************************************
skipping: [192.168.56.110]
ok: [192.168.56.109]
ok: [192.168.56.108]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.108]
ok: [192.168.56.110]

TASK [install apache and php for Ubuntu servers] *******************************
skipping: [192.168.56.110]
ok: [192.168.56.108]

TASK [install apache and php for CentOS servers] *******************************
skipping: [192.168.56.108]
ok: [192.168.56.110]
```



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

TASK [Gathering Facts] *****************************************************
ok: [192.168.56.109]
ok: [192.168.56.110]

TASK [install mariadb package (CentOS)] ************************************
skipping: [192.168.56.109]
ok: [192.168.56.110]

TASK [install mariadb package (Ubuntu)] ************************************
skipping: [192.168.56.110]
ok: [192.168.56.109]

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

TASK [Gathering Facts] *****************************************************
ok: [192.168.56.109]

PLAY RECAP *****************************************************************
192.168.56.108             : ok=4    changed=0    unreachable=0    failed=0
192.168.56.109             : ok=5    changed=0    unreachable=0    failed=0
192.168.56.110             : ok=5    changed=0    unreachable=0    failed=0
```

## Task 3: Managing Services

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

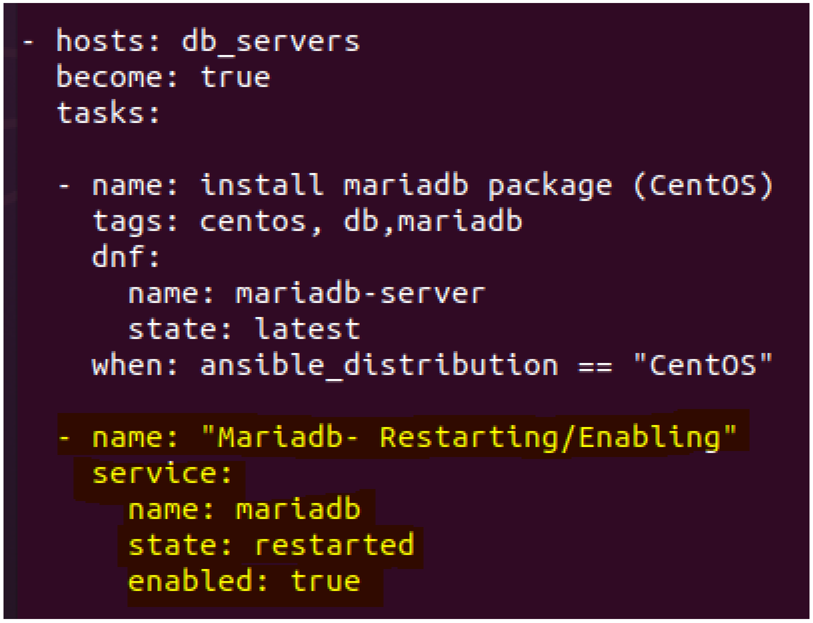


```
- name: install apache and php for CentOS servers
  tags: apache,centos,httpd
  dnf:
    name:
      - httpd
      - php
    state: latest
  when: ansible_distribution == "CentOS"

- 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.

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

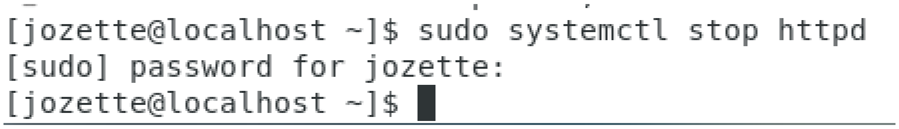


```
- hosts: db_servers
  become: true
  tasks:

  - 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
```

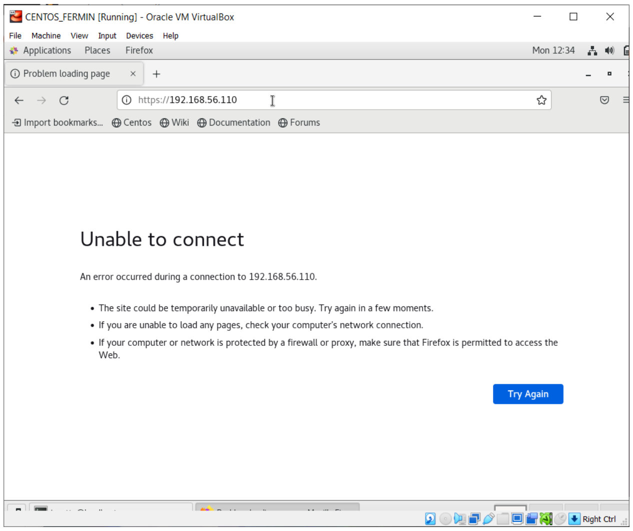
Figure 3.1.2

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.



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



CENTOS_FERMIN [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Firefox
Mon 12:34
Problem loading page
https://192.168.56.110
Import bookmarks... Centos Wiki Documentation Forums
Try Again
Right Ctrl

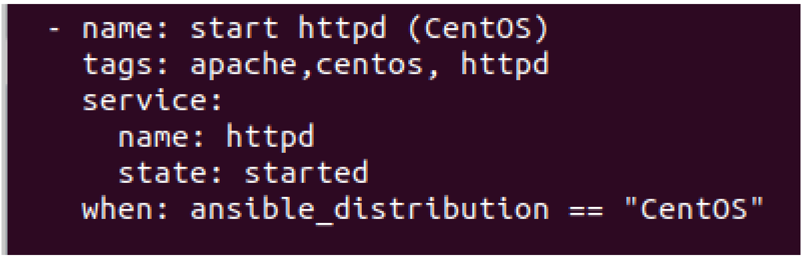
Unable to connect

An error occurred during a connection to 192.168.56.110.

- 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.

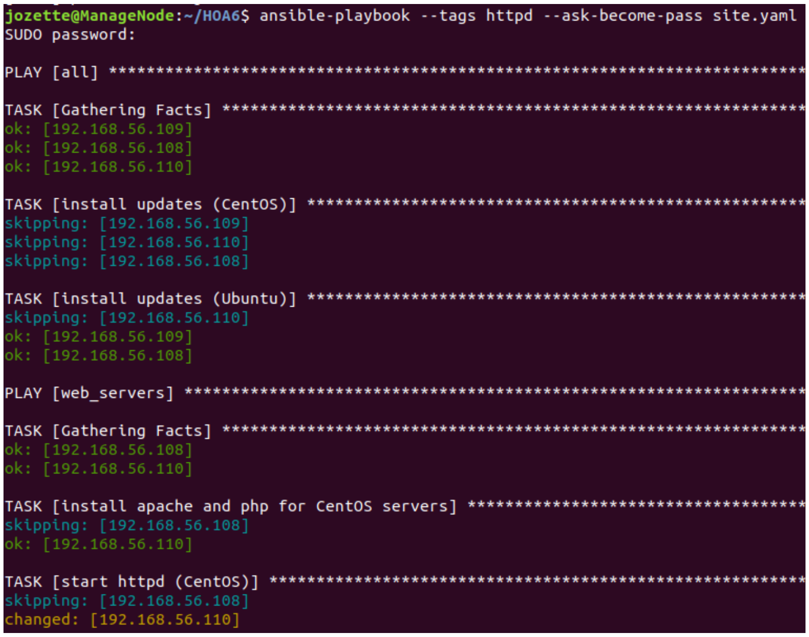
3. Go to the local machine and this time, run the *site.yml* file. Then after running the file, go again to the CentOS server and enter its IP address on the browser. Describe the result.

Input

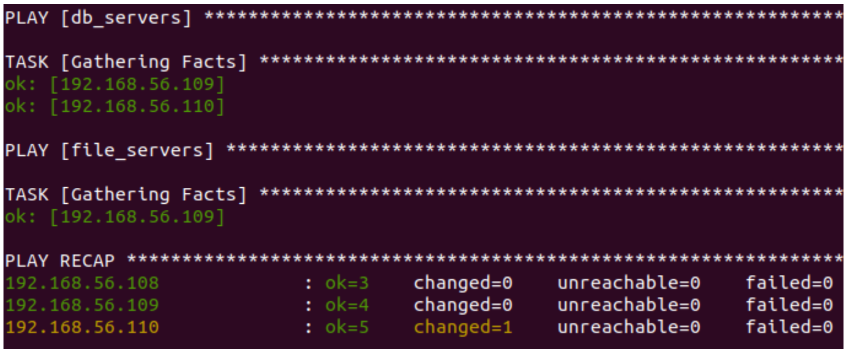


```
  - name: start httpd (CentOS)
    tags: apache,centos, httpd
    service:
      name: httpd
      state: started
    when: ansible_distribution == "CentOS"
```

Process

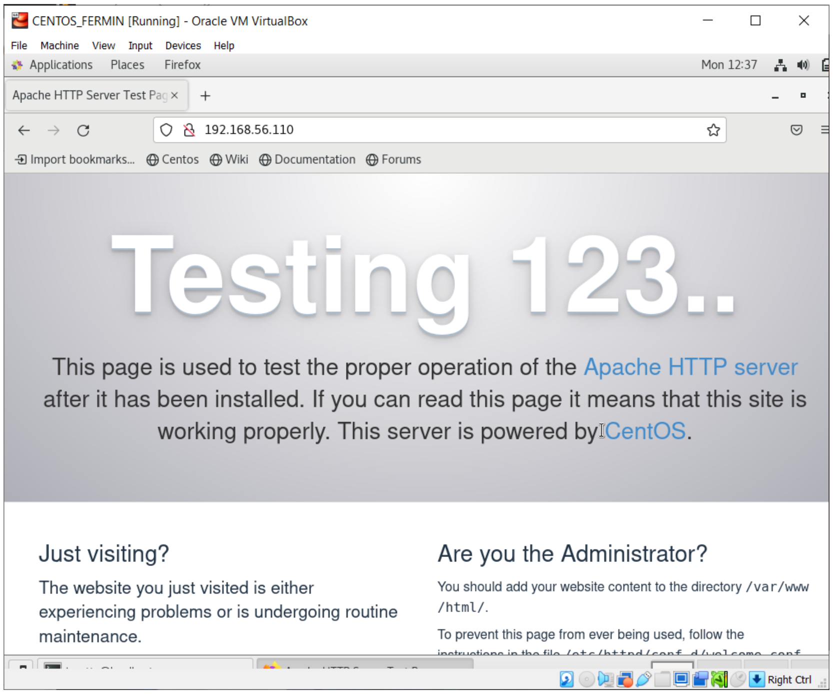


```
jozette@ManageNode:~/HOA6$ ansible-playbook --tags httpd --ask-become-pass site.yaml
SUDO password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.109]
ok: [192.168.56.108]
ok: [192.168.56.110]

TASK [install updates (CentOS)] ************************************************
skipping: [192.168.56.109]
skipping: [192.168.56.110]
skipping: [192.168.56.108]

TASK [install updates (Ubuntu)] ************************************************
skipping: [192.168.56.110]
ok: [192.168.56.109]
ok: [192.168.56.108]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.108]
ok: [192.168.56.110]

TASK [install apache and php for CentOS servers] *******************************
skipping: [192.168.56.108]
ok: [192.168.56.110]

TASK [start httpd (CentOS)] ****************************************************
skipping: [192.168.56.108]
changed: [192.168.56.110]
```



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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.109]
ok: [192.168.56.110]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.109]

PLAY RECAP *********************************************************************
192.168.56.108             : ok=3    changed=0    unreachable=0    failed=0
192.168.56.109             : ok=4    changed=0    unreachable=0    failed=0
192.168.56.110             : ok=5    changed=1    unreachable=0    failed=0
```

Output


CENTOS_FERMIN [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Firefox
Mon 12:37
Apache HTTP Server Test Pag
192.168.56.110
Import bookmarks... Centos Wiki Documentation Forums
Right Ctrl

Testing 123..

This page is used to test the proper operation of the Apache 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.

Just visiting?

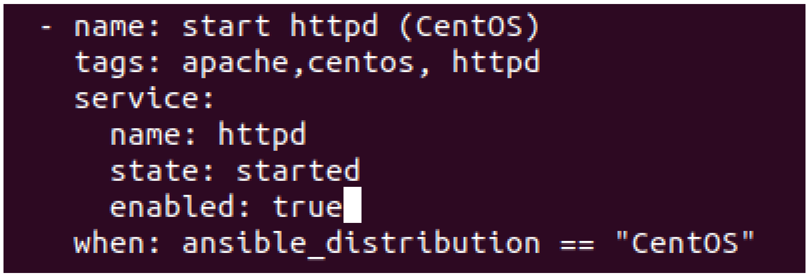
The website you just visited is either experiencing problems or is undergoing routine maintenance.

Are you the Administrator?

You should add your website content to the directory /var/www/html/.

To prevent this page from ever being used, follow the instructions in the file /etc/httpd/conf.d/welcome.conf

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"
```

**Reflections:**

Answer the following:

1. What is the importance of putting our remote servers into groups?
   **The importance of putting the remote servers into a groups is it organized the servers and helps in improving the efficiency in managing, configurations, and granular access control of your topology.**

2. What is the importance of tags in playbooks?
   **The importance of playing tags in playbook is it allows for selective execution of tasks, fine-grained control, and efficient use of resources, making autonation in the servers more precise and manageable.**

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

   **Some services need to be managed automatically in playbooks to ensure consistency, scalability, timely responses to changes, and error handling, making infrastructure more reliable and adaptable to dynamic environments.**

**Conclusions:**

**In this hands-on activity, I learned that Automation in service management maintains consistency and scalability in my infrastructure, ensuring reliable service operations and timely responses to changes and issues, including error handling for service availability during failures. I also learned that by targeting specific nodes in managing servers are helps in improving the efficiency. I find that grouping servers helps me streamline tasks and access control, while tags in my playbooks enable me to execute tasks more precisely, ultimately saving me time and resources. In short, these practices allow me to efficiently manage my infrastructure, making it more robust and adaptable to evolving IT demands.**