Name: Jessie Robert Lazo	**Date Performed: 10/02/2024**
Course/Section:CPE 212-CPE31S2	**Date Submitted:10/02/2024**
Instructor: Engr. Robin Valenzuela	**Semester and SY:**

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.

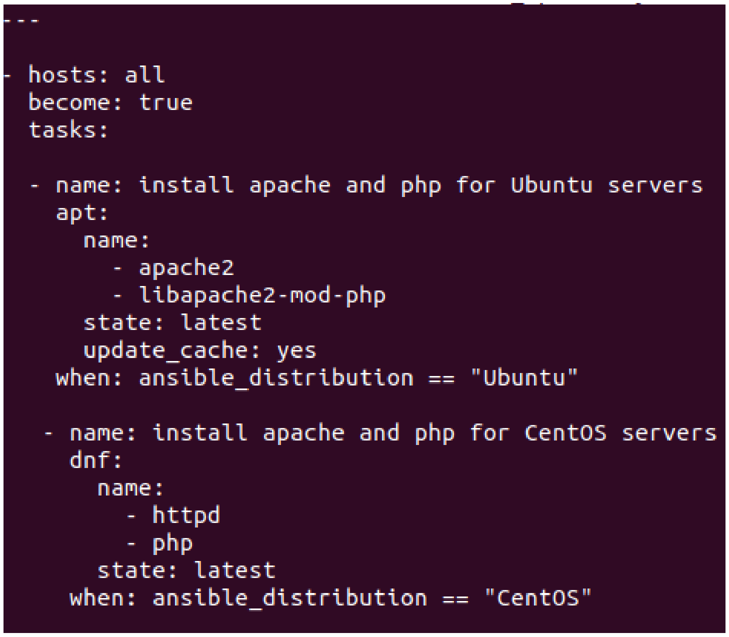
We also try to manage services that do not automatically run using the automations in playbook. For example, when we install web servers or httpd for CentOS, we notice that the service did not start automatically.

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.

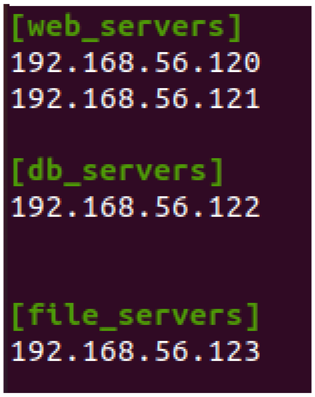
Task 1: Targeting Specific Nodes

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.

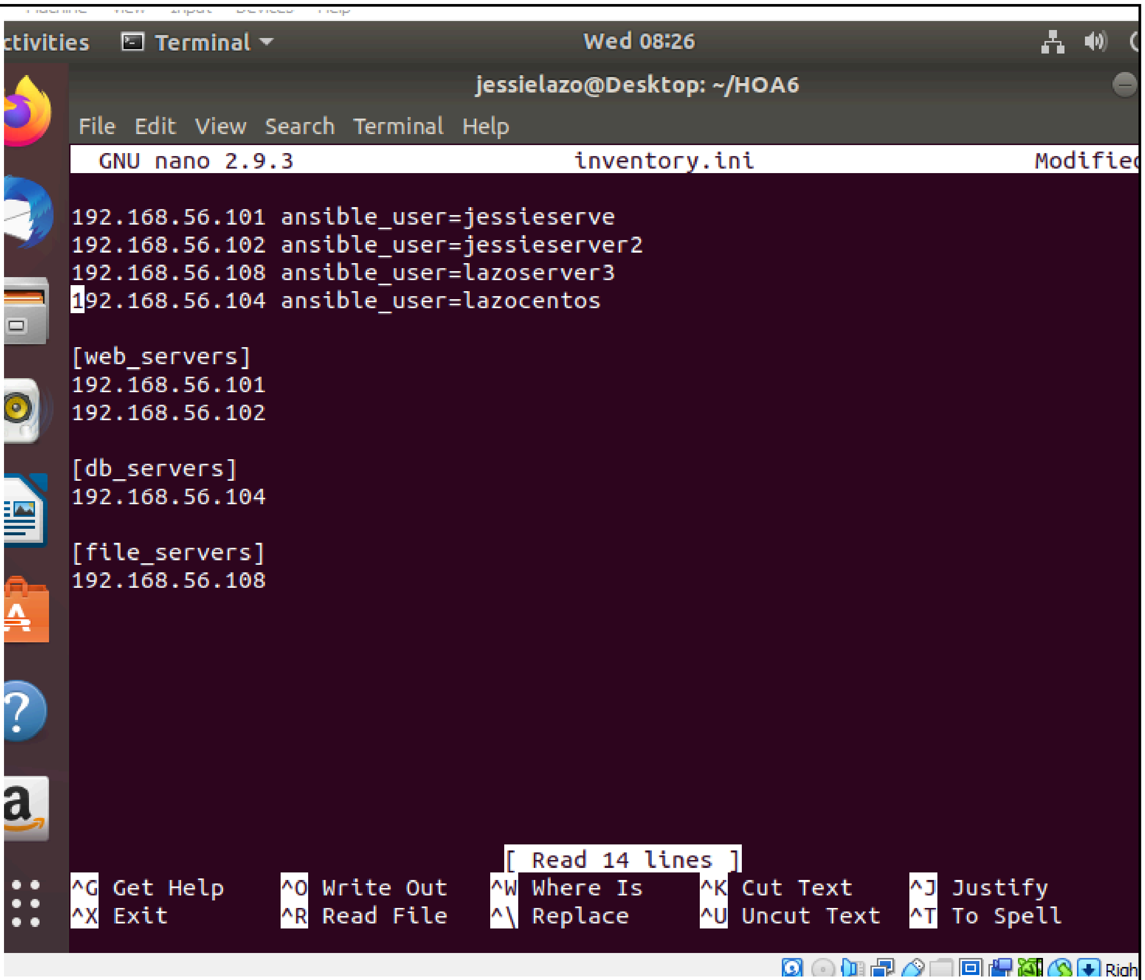


```
---

- 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
```

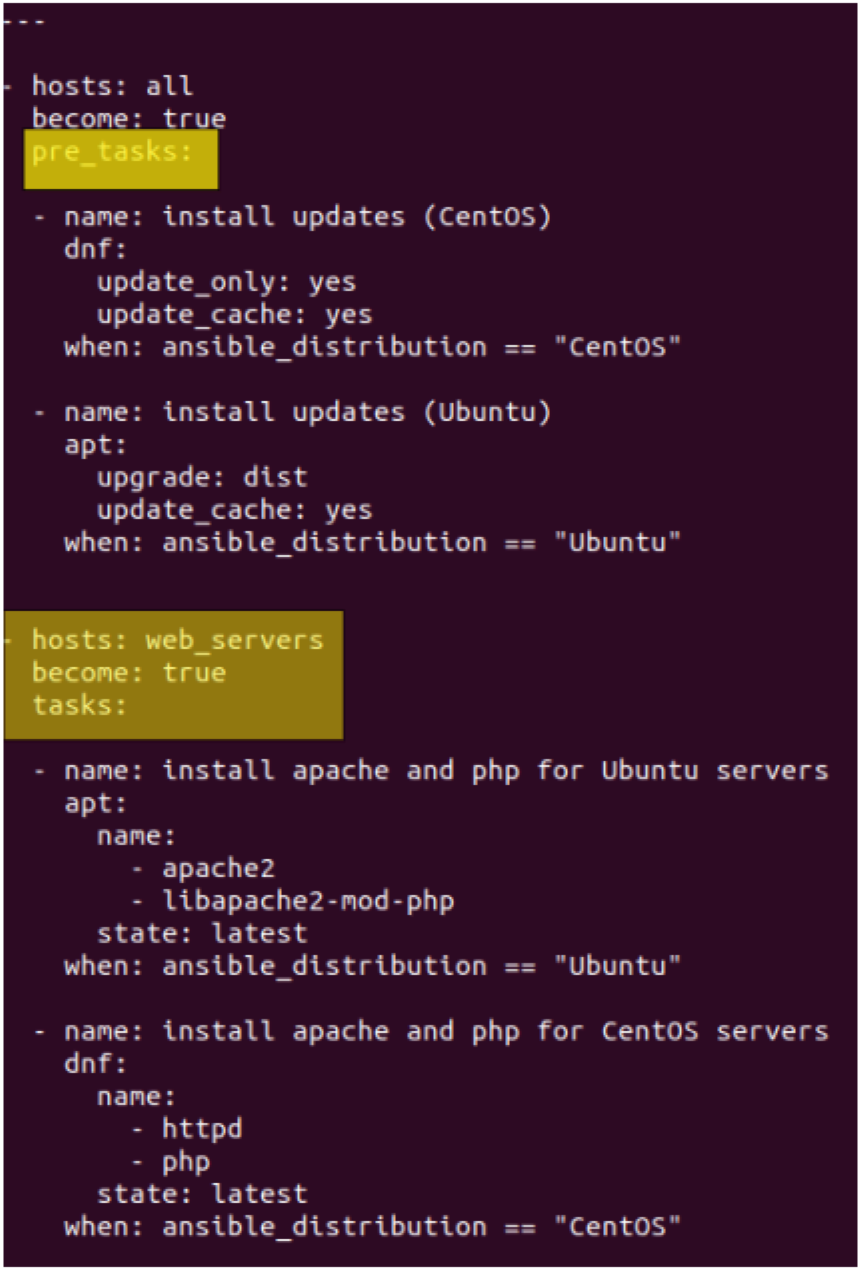

ctivities
Terminal
Wed 08:26
jessielazo@Desktop: ~/HOA6
File Edit View Search Terminal Help

```
  GNU nano 2.9.3                      inventory.ini                     Modifie

192.168.56.101 ansible_user=jessieserve
192.168.56.102 ansible_user=jessieserver2
192.168.56.108 ansible_user=lazoserver3
192.168.56.104 ansible_user=lazocentos

[web_servers]
192.168.56.101
192.168.56.102

[db_servers]
192.168.56.104

[file_servers]
192.168.56.108



                                    [ Read 14 lines ]
^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify
^X Exit         ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell
```

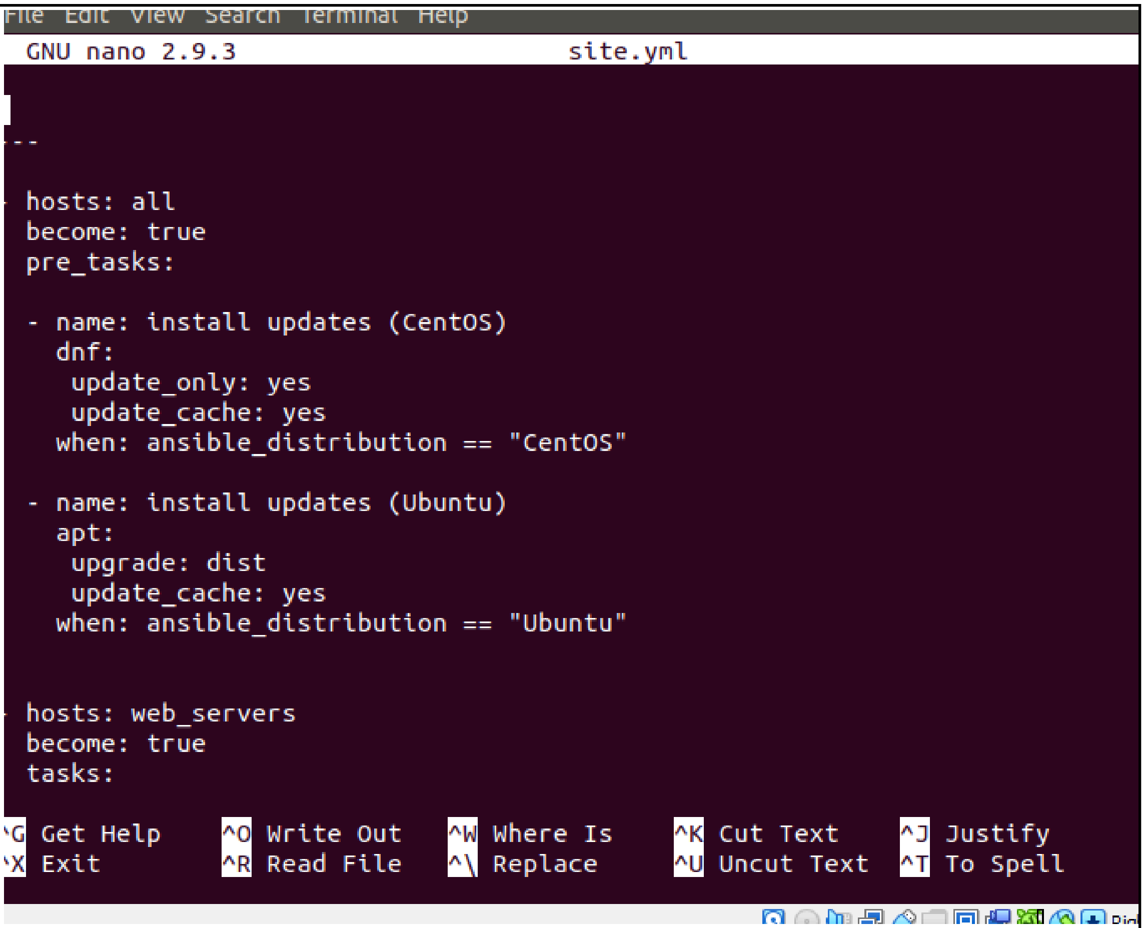
Make sure to save the file and exit.

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

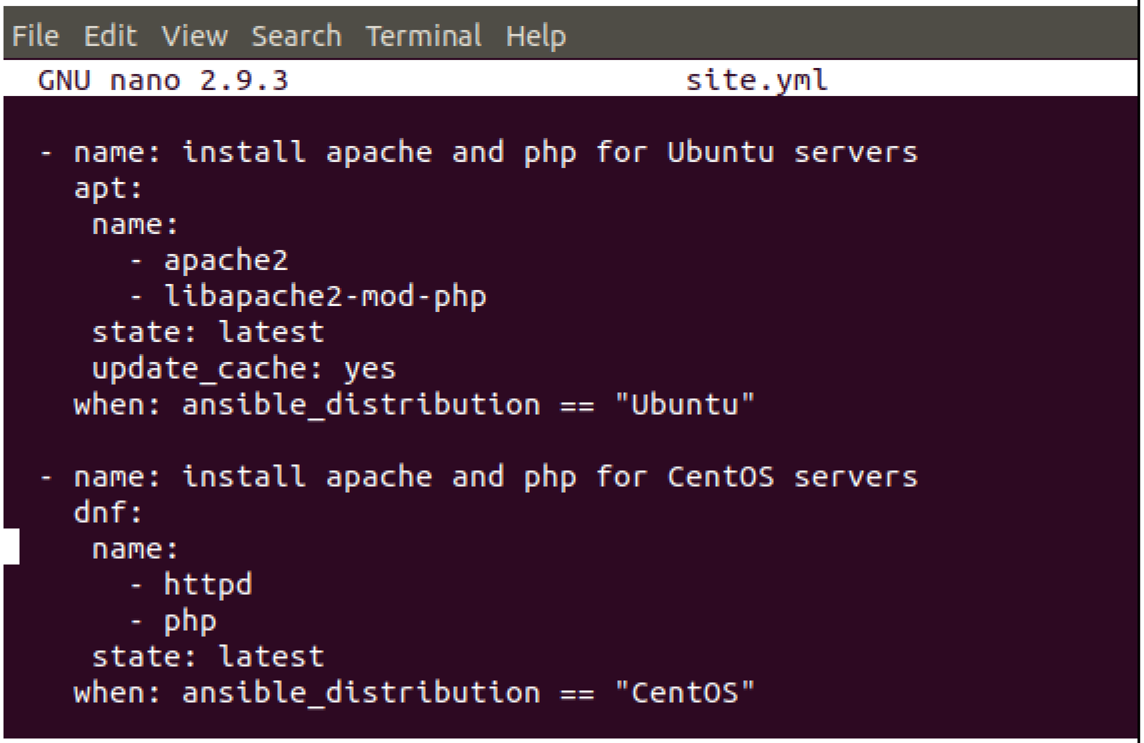
3. Edit the *site.yml* by following the image below:



```
---

- hosts: all
  become: true
  pre_tasks:

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

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


- hosts: web_servers
  become: true
  tasks:

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

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



```
  GNU nano 2.9.3                    site.yml

---

- hosts: all
  become: true
  pre_tasks:

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

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


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

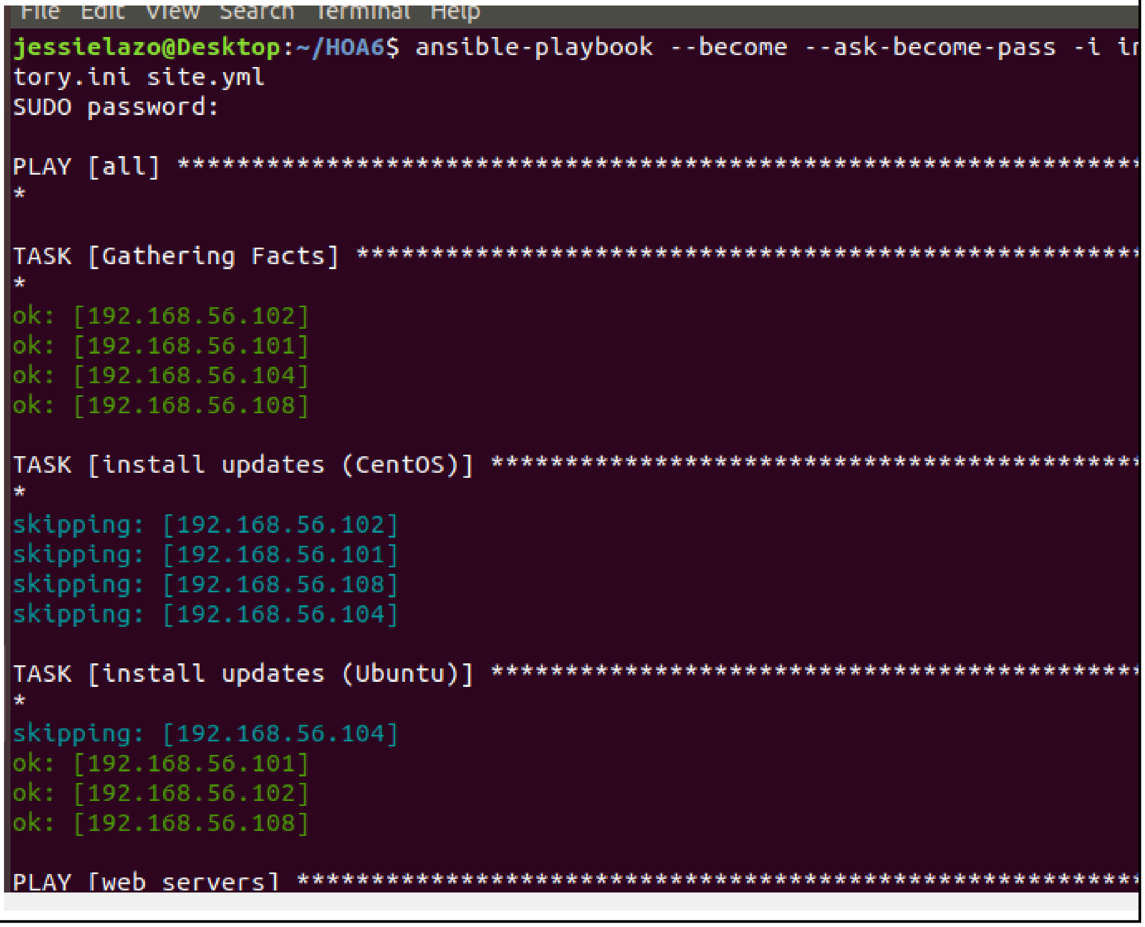


```
  GNU nano 2.9.3                    site.yml

  - 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"
```

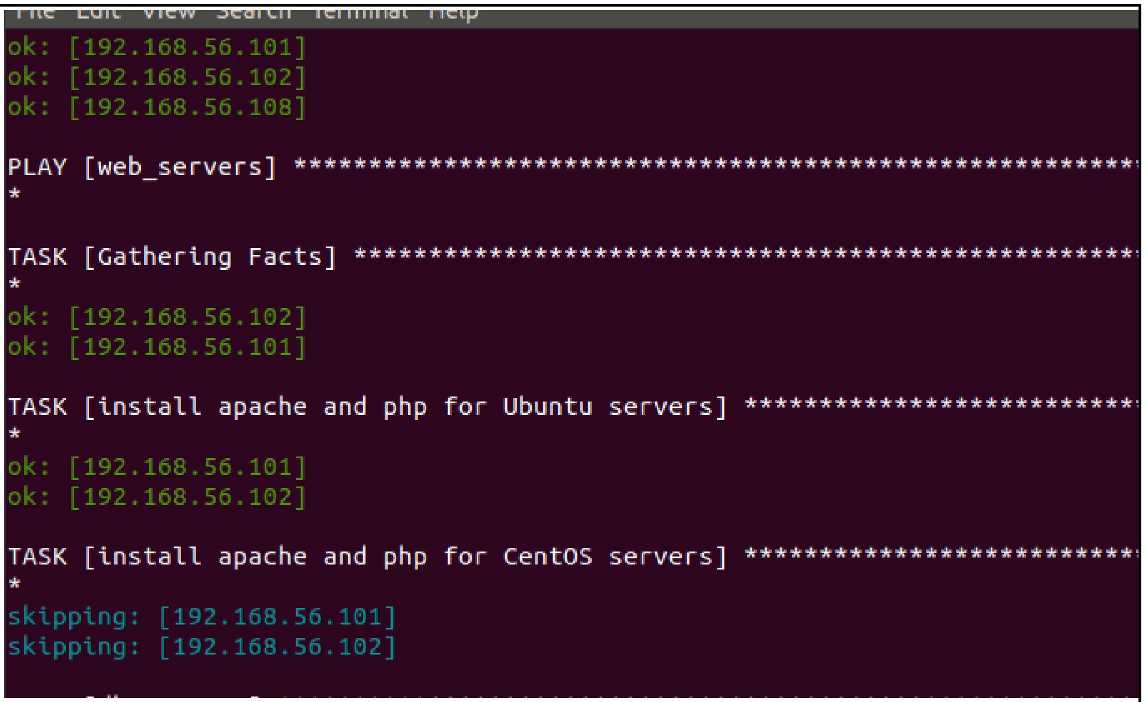
Make sure to save the file and exit.

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.



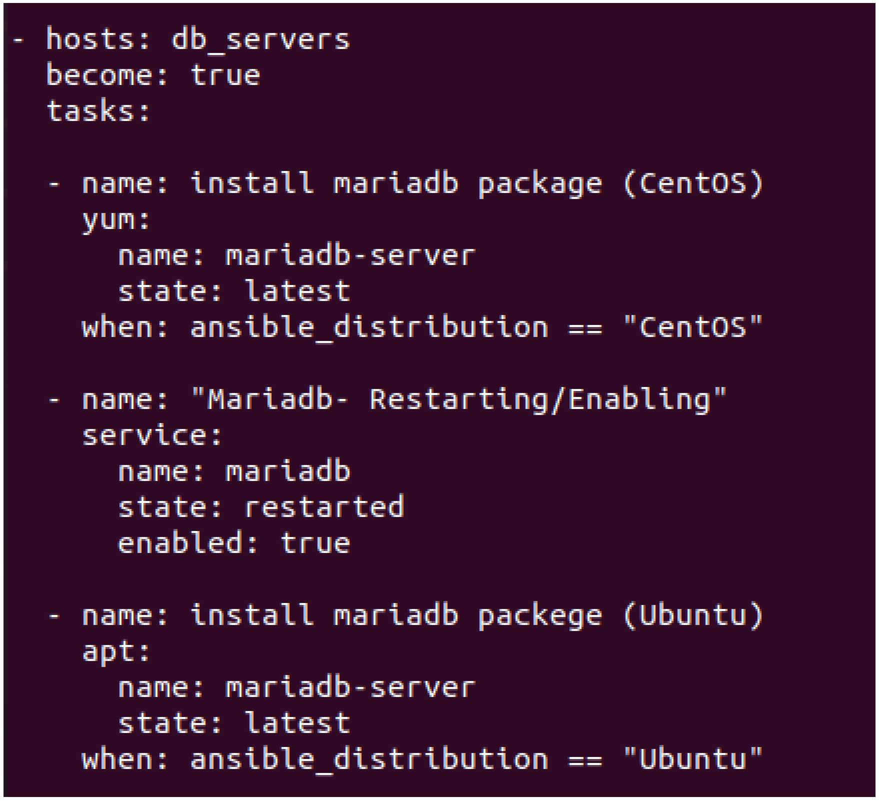
```
jessielazo@Desktop:~/HOA6$ ansible-playbook --become --ask-become-pass -i in
tory.ini site.yml
SUDO password:

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

TASK [Gathering Facts] *****************************************************
*
ok: [192.168.56.102]
ok: [192.168.56.101]
ok: [192.168.56.104]
ok: [192.168.56.108]

TASK [install updates (CentOS)] ********************************************
*
skipping: [192.168.56.102]
skipping: [192.168.56.101]
skipping: [192.168.56.108]
skipping: [192.168.56.104]

TASK [install updates (Ubuntu)] ********************************************
*
skipping: [192.168.56.104]
ok: [192.168.56.101]
ok: [192.168.56.102]
ok: [192.168.56.108]

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



```
ok: [192.168.56.101]
ok: [192.168.56.102]
ok: [192.168.56.108]

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

TASK [Gathering Facts] ****************************************************
*
ok: [192.168.56.102]
ok: [192.168.56.101]

TASK [install apache and php for Ubuntu servers] **************************
*
ok: [192.168.56.101]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] **************************
*
skipping: [192.168.56.101]
skipping: [192.168.56.102]
```

it runs without error and skips ubuntu servers for centos tasks.

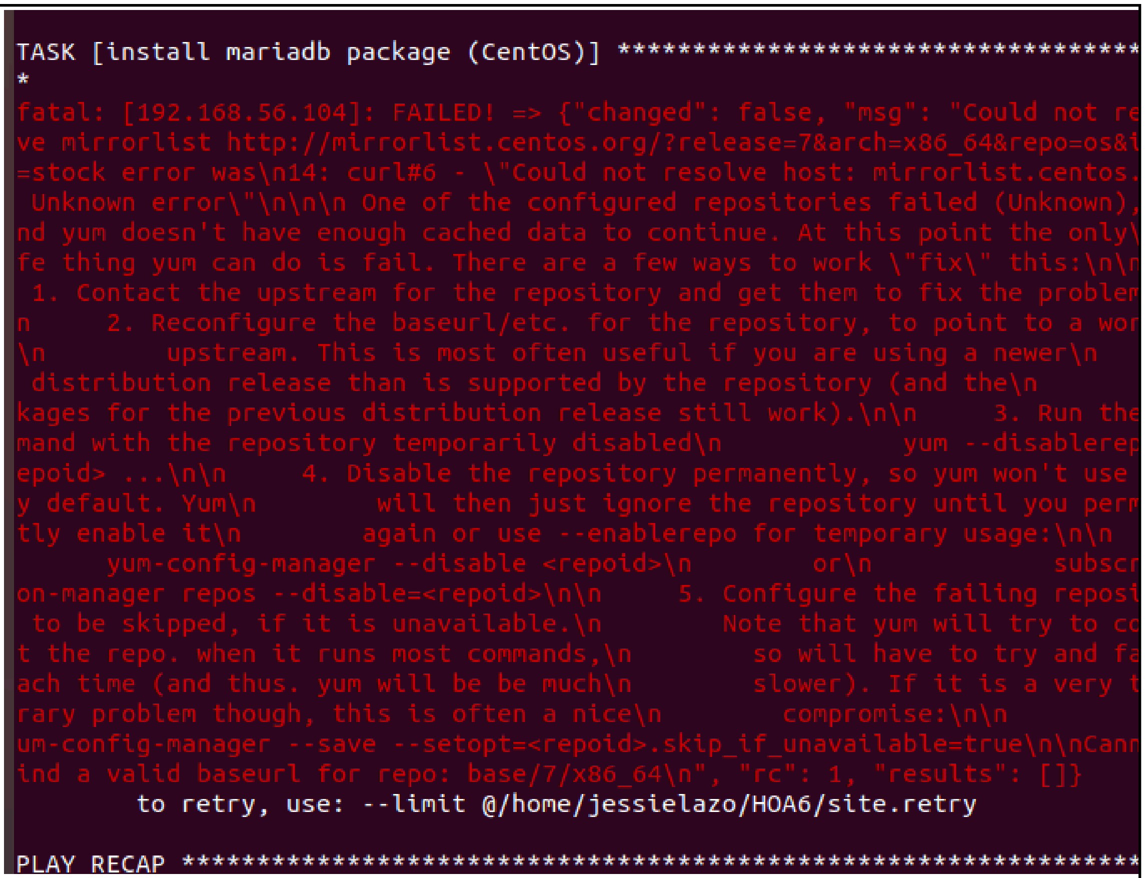
4. Let’s try to edit again the *site.yml* file. This time, we are going to add plays targeting the other servers. This time we target the *db_servers* by adding it on the current *site.yml*. Below is an example: (Note add this at the end of the playbooks from task 1.3.



```
- 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"
```

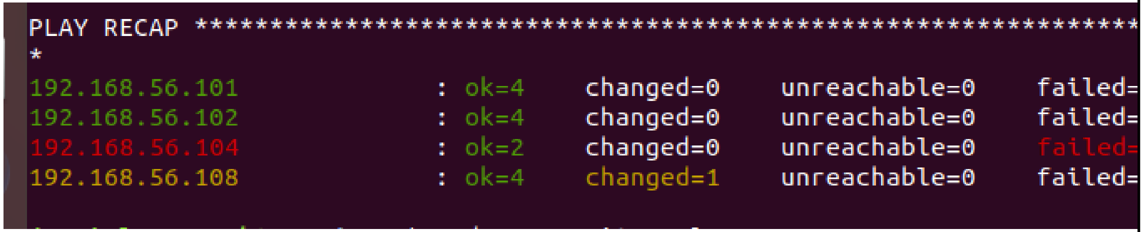
Make sure to save the file and exit.

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



```
TASK [install mariadb package (CentOS)] ************************************
*
fatal: [192.168.56.104]: FAILED! => {"changed": false, "msg": "Could not re
ve mirrorlist http://mirrorlist.centos.org/?release=7&arch=x86_64&repo=os&i
=stock error was\n14: curl#6 - \"Could not resolve host: mirrorlist.centos.
 Unknown error\"\n\n\n One of the configured repositories failed (Unknown),
nd yum doesn't have enough cached data to continue. At this point the only\
fe thing yum can do is fail. There are a few ways to work \"fix\" this:\n\n
 1. Contact the upstream for the repository and get them to fix the problem
n     2. Reconfigure the baseurl/etc. for the repository, to point to a wor
\n        upstream. This is most often useful if you are using a newer\n
 distribution release than is supported by the repository (and the\n
kages for the previous distribution release still work).\n\n     3. Run the
mand with the repository temporarily disabled\n            yum --disablerep
epoid> ...\n\n     4. Disable the repository permanently, so yum won't use
y default. Yum\n        will then just ignore the repository until you perm
tly enable it\n        again or use --enablerepo for temporary usage:\n\n
      yum-config-manager --disable <repoid>\n        or\n            subscr
on-manager repos --disable=<repoid>\n\n     5. Configure the failing reposi
 to be skipped, if it is unavailable.\n        Note that yum will try to co
t the repo. when it runs most commands,\n        so will have to try and fa
ach time (and thus. yum will be be much\n        slower). If it is a very t
rary problem though, this is often a nice\n        compromise:\n\n
um-config-manager --save --setopt=<repoid>.skip_if_unavailable=true\n\nCann
ind a valid baseurl for repo: base/7/x86_64\n", "rc": 1, "results": []}
        to retry, use: --limit @/home/jessielazo/HOA6/site.retry

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

I got an error in my centos thus, it cannot install mariadb.

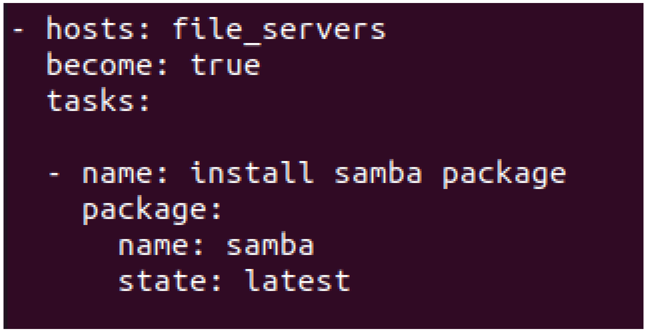


```
PLAY RECAP *****************************************************************
*
192.168.56.101             : ok=4    changed=0    unreachable=0    failed=
192.168.56.102             : ok=4    changed=0    unreachable=0    failed=
192.168.56.104             : ok=2    changed=0    unreachable=0    failed=
192.168.56.108             : ok=4    changed=1    unreachable=0    failed=
```

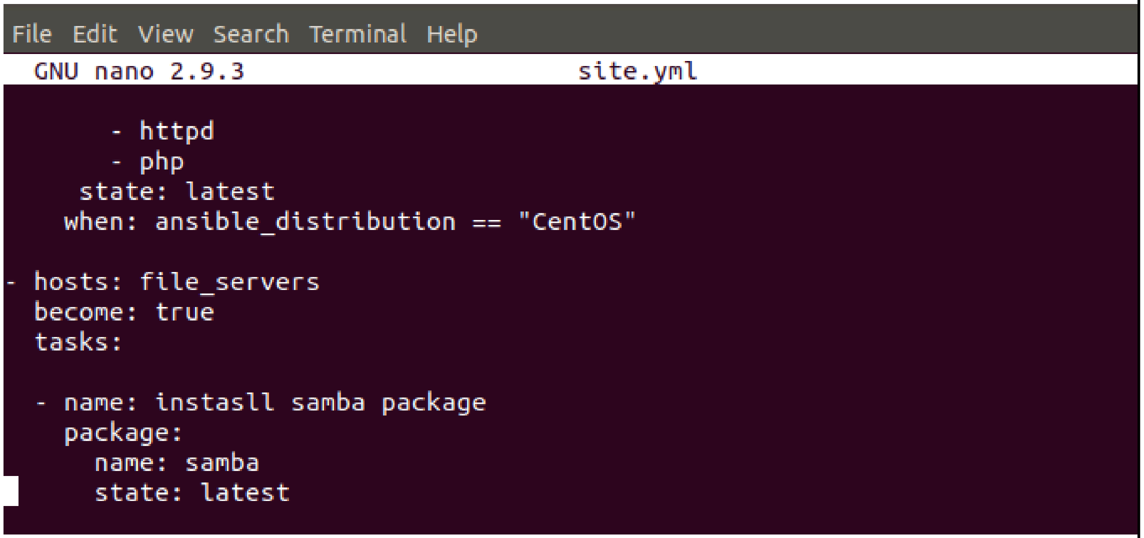
5. Go to the remote server (Ubuntu) terminal that belongs to the db_servers group and check the status for mariadb installation using the command: *systemctl status mariadb.* Do this on the CentOS server also.

   Describe the output.

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.



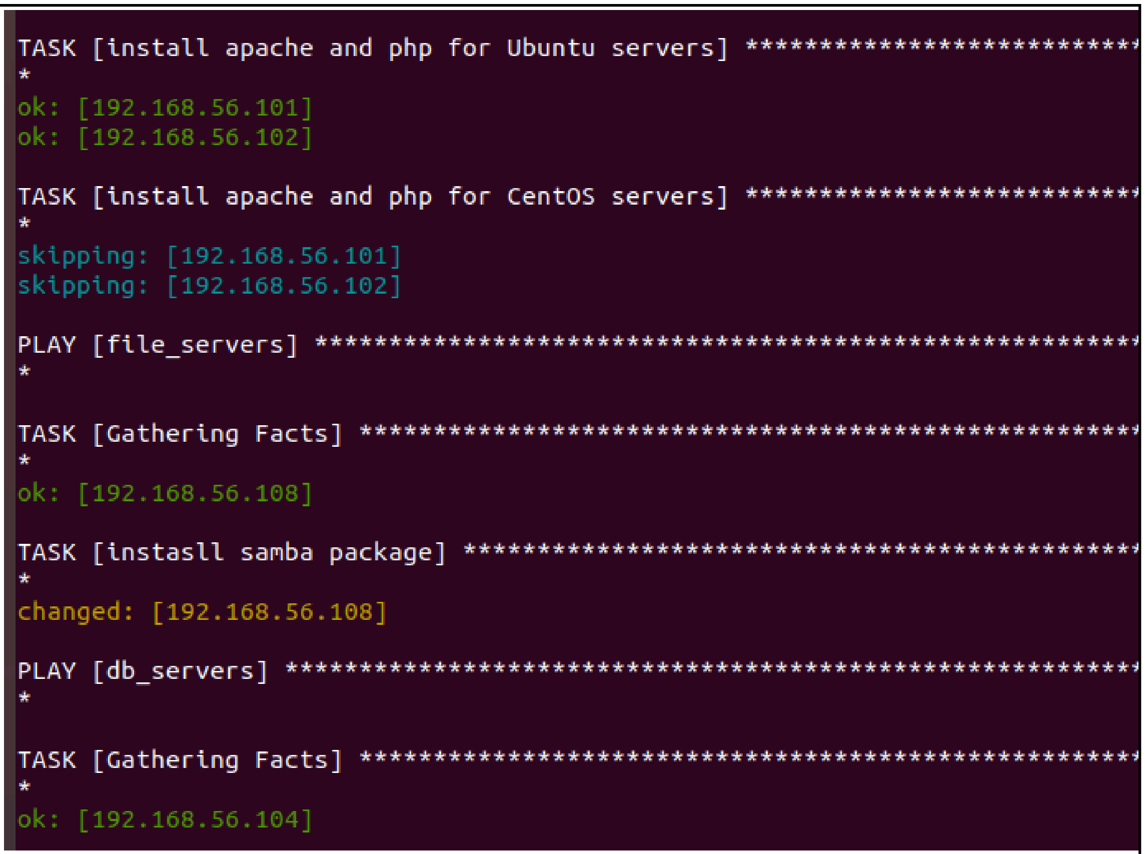
```
- hosts: file_servers
  become: true
  tasks:

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



```
      - httpd
      - php
    state: latest
  when: ansible_distribution == "CentOS"

- hosts: file_servers
  become: true
  tasks:

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

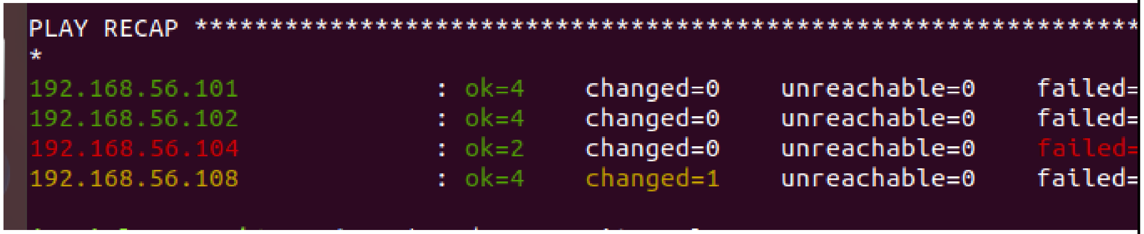
Make sure to save the file and exit.

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



```
TASK [install apache and php for Ubuntu servers] ***************************
*
ok: [192.168.56.101]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] ***************************
*
skipping: [192.168.56.101]
skipping: [192.168.56.102]

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

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

TASK [instasll samba package] **********************************************
*
changed: [192.168.56.108]

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

TASK [Gathering Facts] *****************************************************
*
ok: [192.168.56.104]
```

tasks for file_servers successfully processed. but maria db still error.



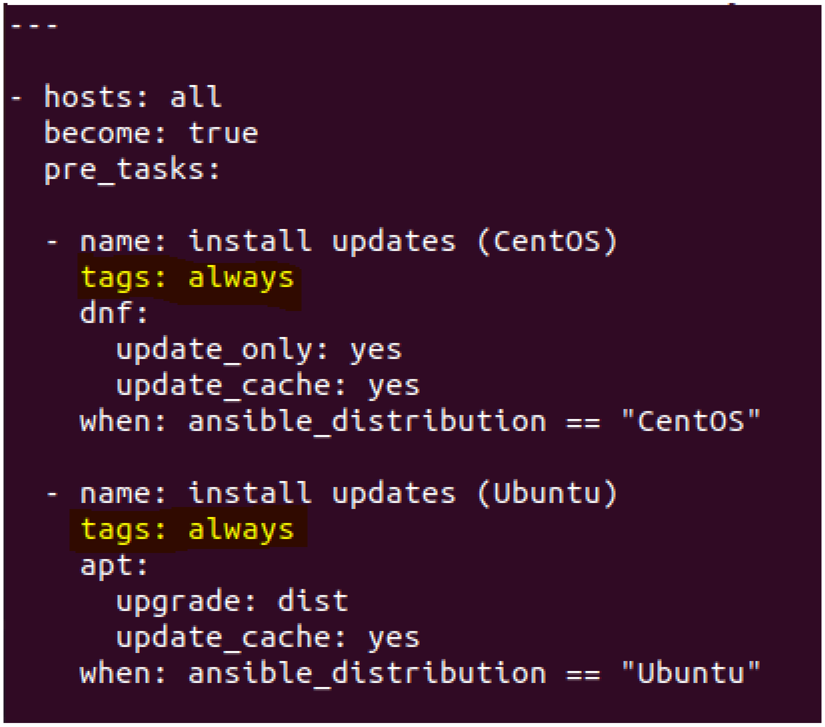
```
PLAY RECAP *****************************************************************
*
192.168.56.101             : ok=4    changed=0    unreachable=0    failed=
192.168.56.102             : ok=4    changed=0    unreachable=0    failed=
192.168.56.104             : ok=2    changed=0    unreachable=0    failed=
192.168.56.108             : ok=4    changed=1    unreachable=0    failed=
```

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.

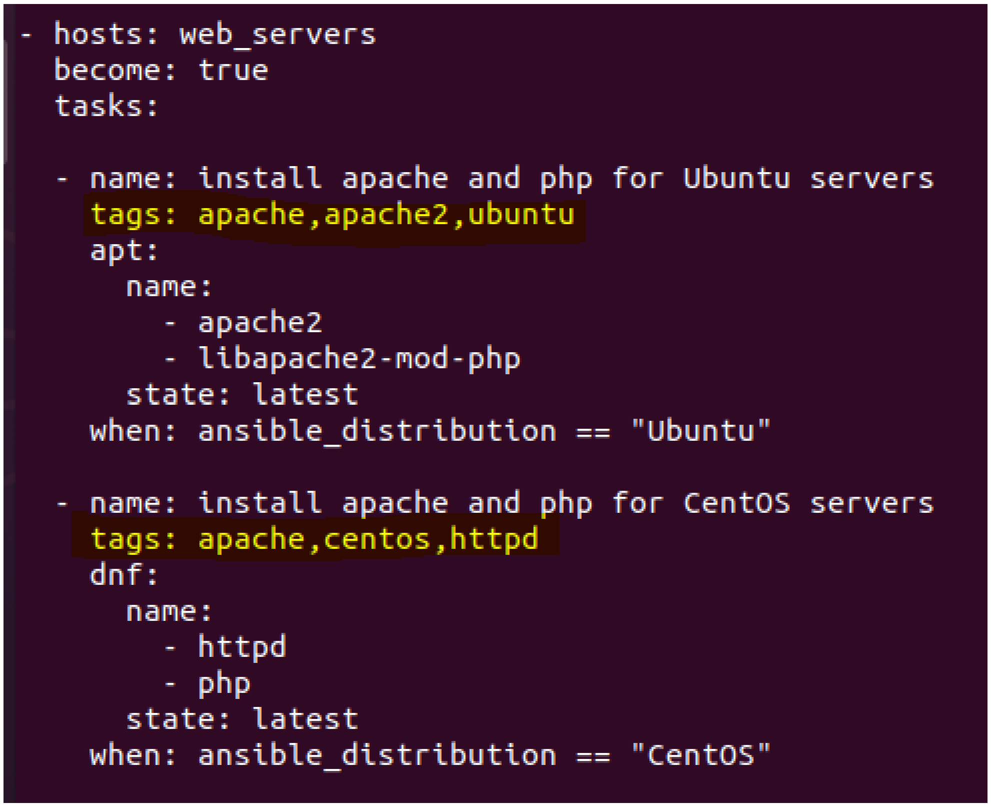
Task 2: Using Tags in running playbooks

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.

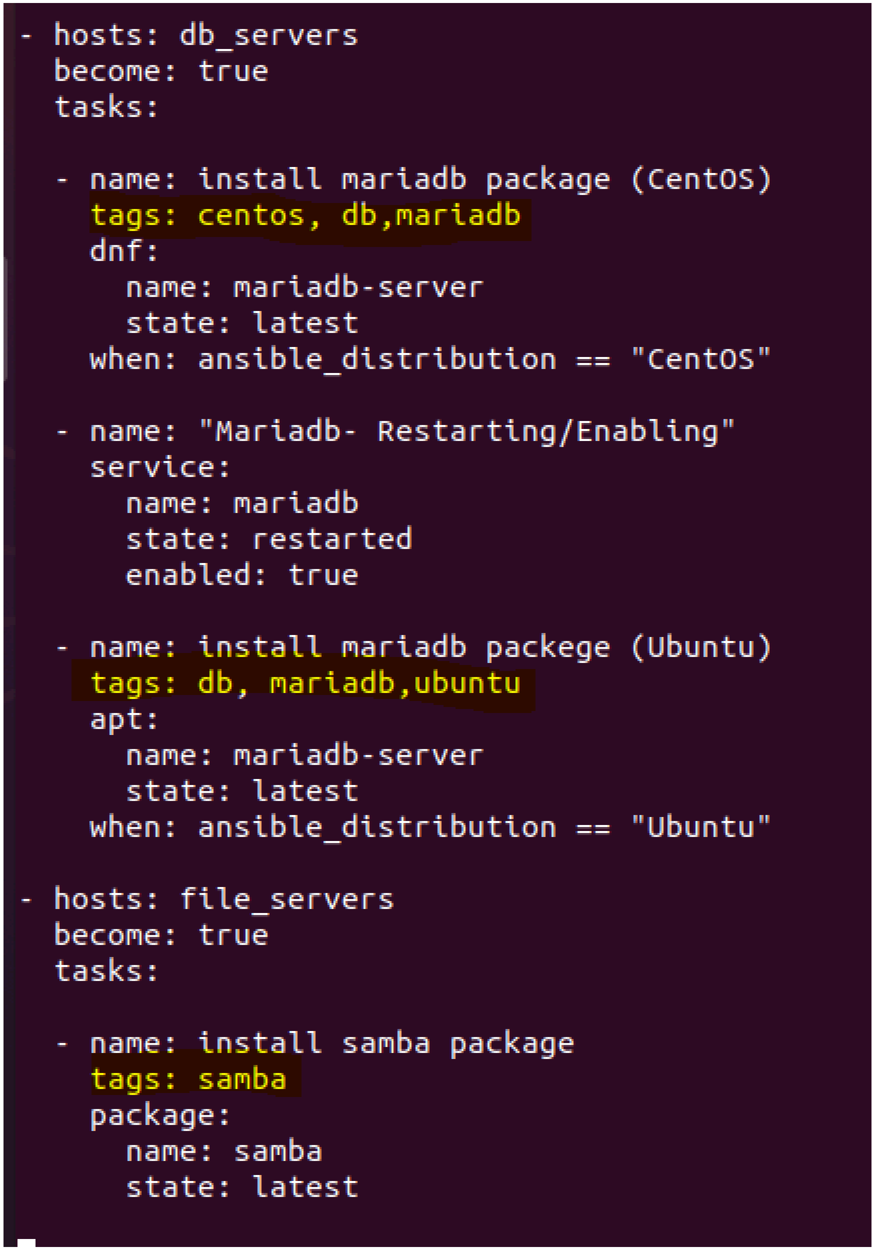
1. Edit the *site.yml* file. Add tags to the playbook. After the name, we can place the tags: *name_of_tag*. This is an arbitrary command, which means you can use any name for a tag.



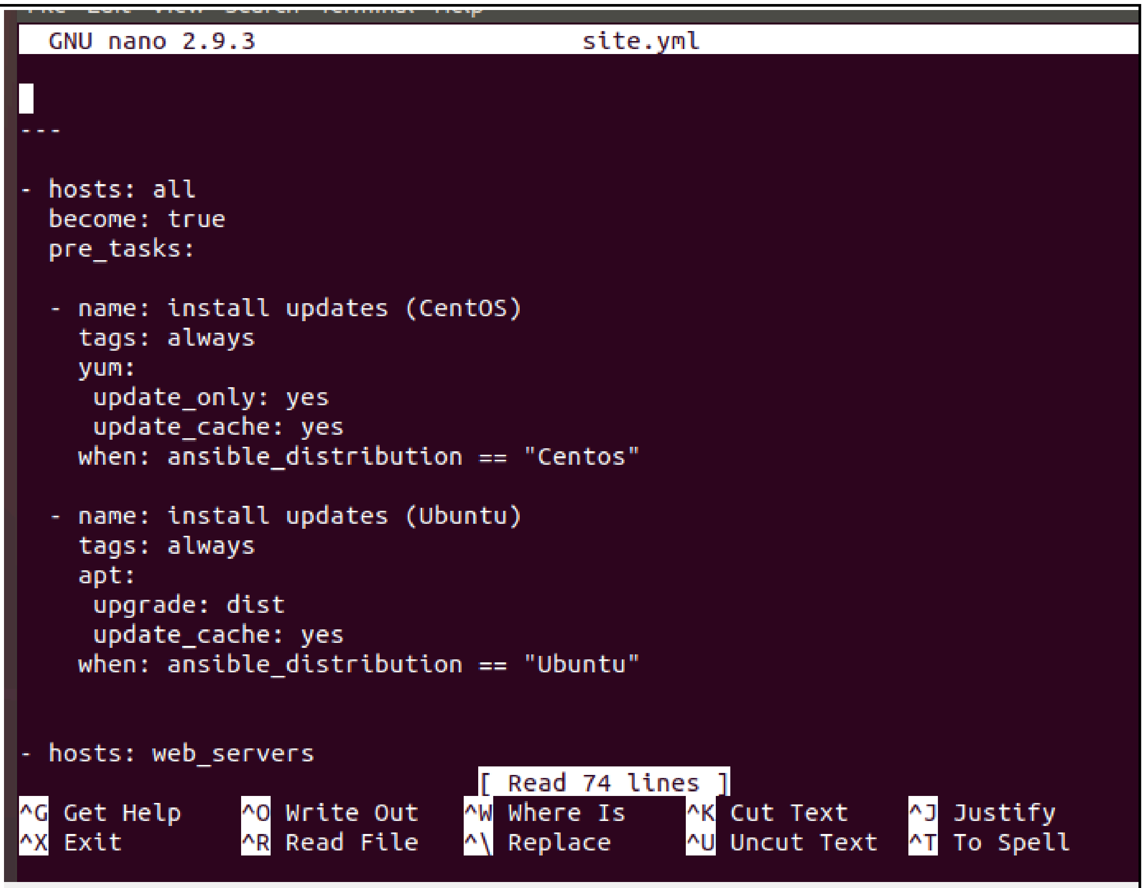
```
---

- 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"
```



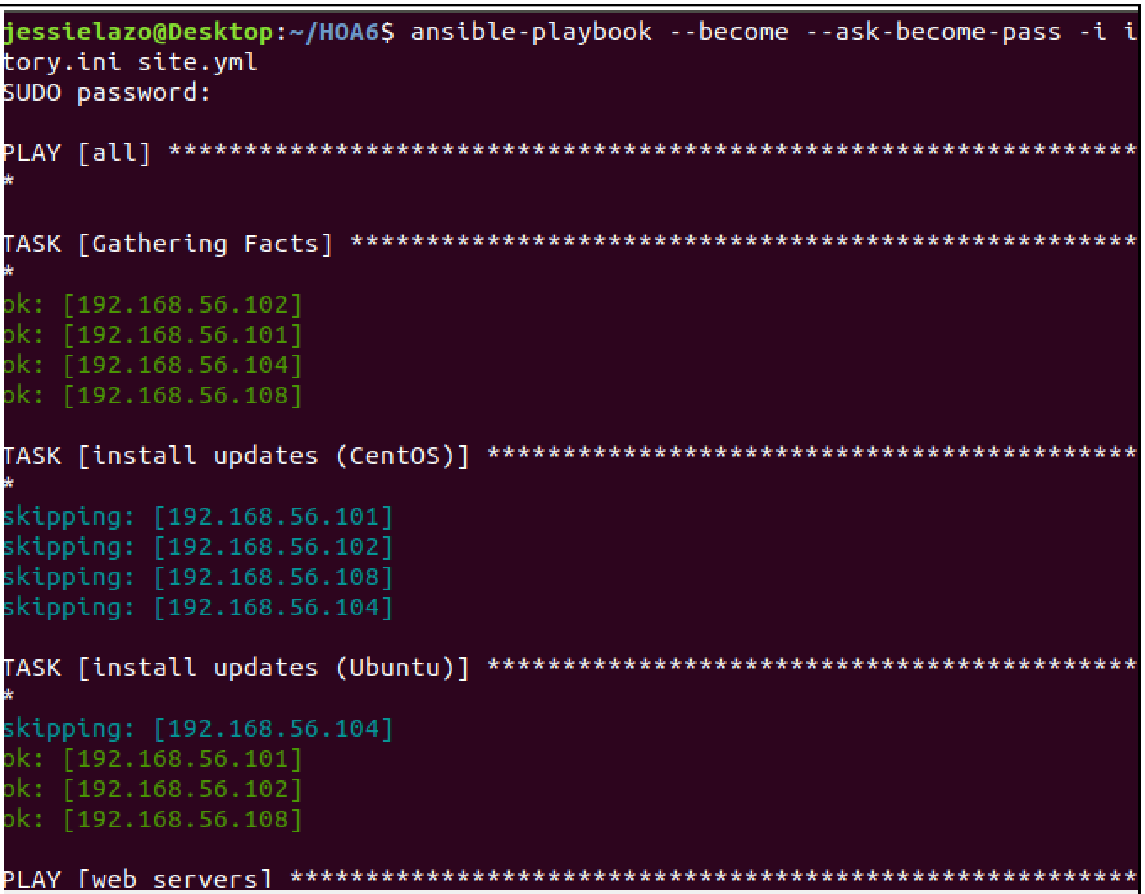
```
- hosts: db_servers
  become: true
  tasks:

  - 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
```



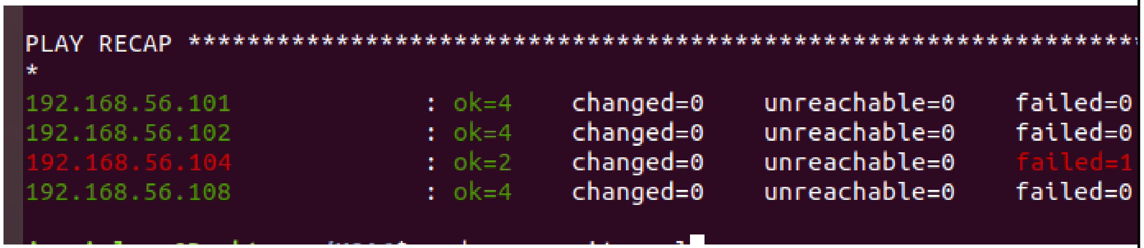
```
  GNU nano 2.9.3                                   site.yml


---

- hosts: all
  become: true
  pre_tasks:

  - name: install updates (CentOS)
    tags: always
    yum:
     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
                               [ Read 74 lines ]
^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify
^X Exit         ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell
```

Make sure to save the file and exit.

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



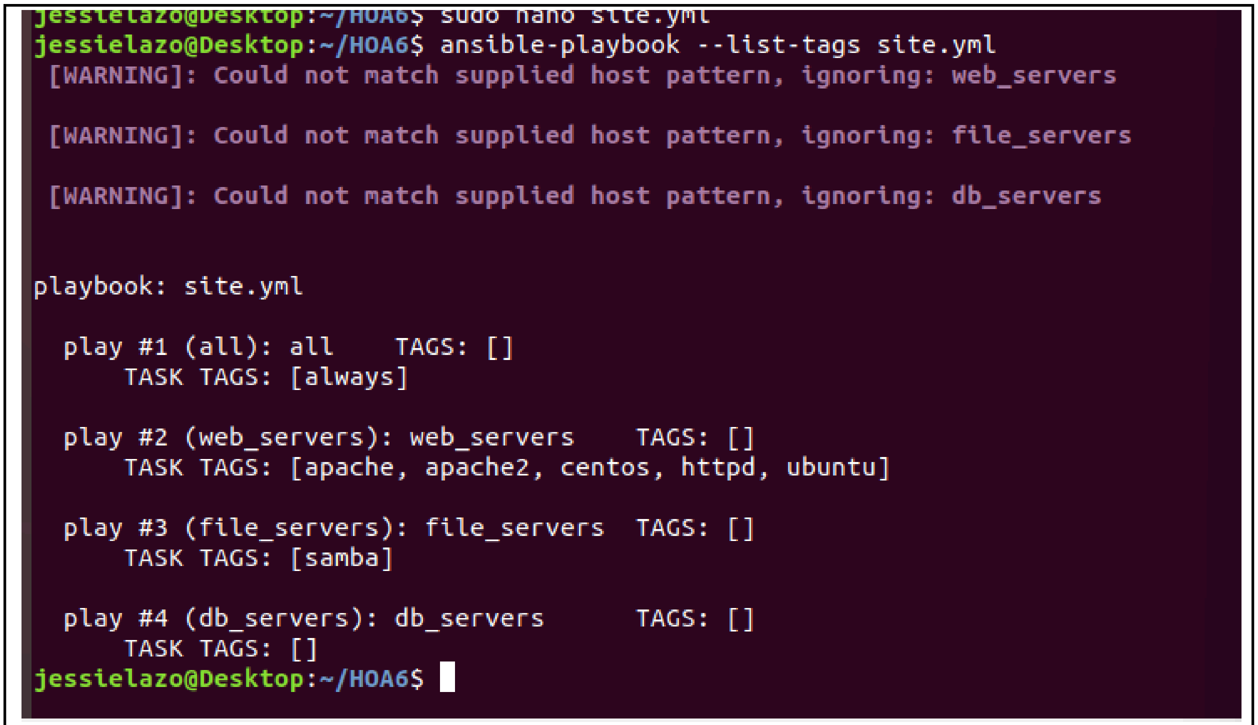
```
jessielazo@Desktop:~/HOA6$ ansible-playbook --become --ask-become-pass -i i
tory.ini site.yml
SUDO password:

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

TASK [Gathering Facts] *****************************************************
*
ok: [192.168.56.102]
ok: [192.168.56.101]
ok: [192.168.56.104]
ok: [192.168.56.108]

TASK [install updates (CentOS)] ********************************************
*
skipping: [192.168.56.101]
skipping: [192.168.56.102]
skipping: [192.168.56.108]
skipping: [192.168.56.104]

TASK [install updates (Ubuntu)] ********************************************
*
skipping: [192.168.56.104]
ok: [192.168.56.101]
ok: [192.168.56.102]
ok: [192.168.56.108]

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



```
PLAY RECAP *****************************************************************
*
192.168.56.101             : ok=4    changed=0    unreachable=0    failed=0
192.168.56.102             : ok=4    changed=0    unreachable=0    failed=0
192.168.56.104             : ok=2    changed=0    unreachable=0    failed=1
192.168.56.108             : ok=4    changed=0    unreachable=0    failed=0
```

it runs successfully, but mariadb still has an error.

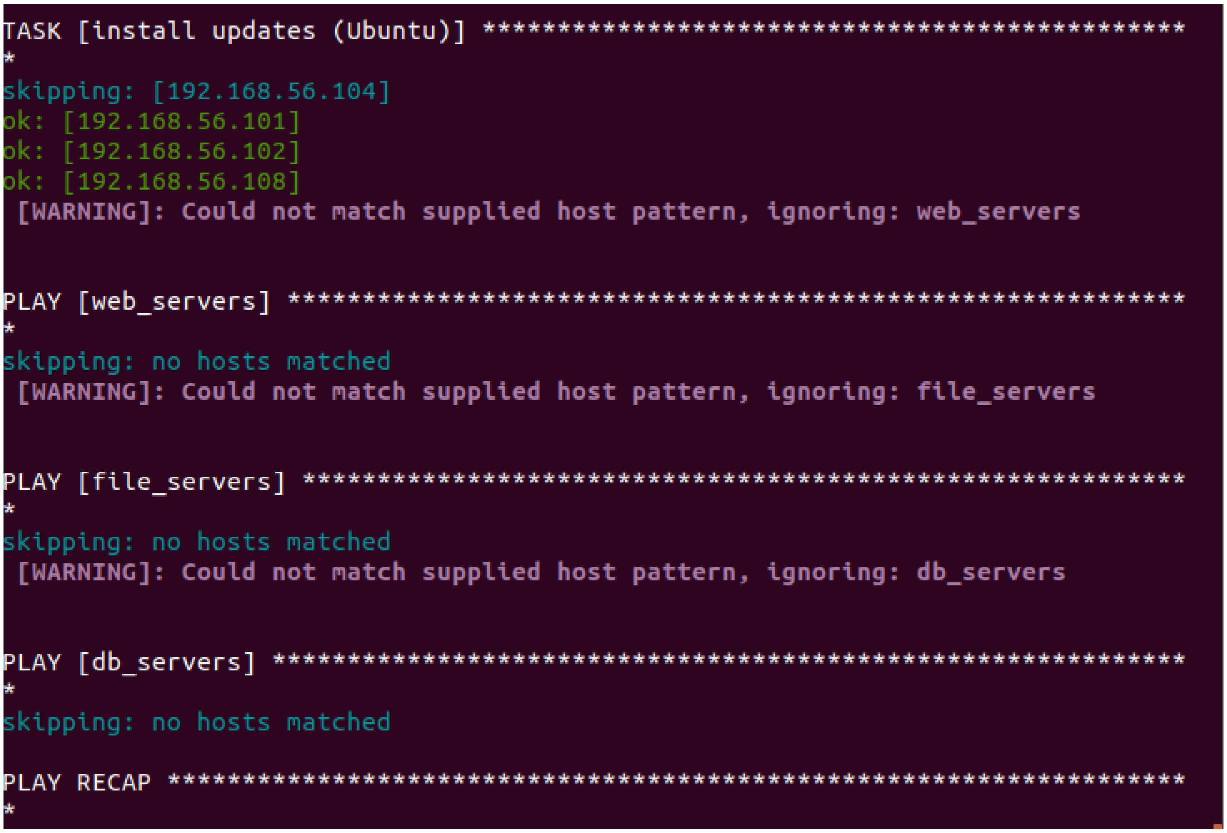
2. On the local machine, try to issue the following commands and describe each result:
   2.1 *ansible-playbook --list-tags site.yml*



```
jessielazo@Desktop:~/HOA6$ sudo nano site.yml
jessielazo@Desktop:~/HOA6$ ansible-playbook --list-tags site.yml
 [WARNING]: Could not match supplied host pattern, ignoring: web_servers

 [WARNING]: Could not match supplied host pattern, ignoring: file_servers

 [WARNING]: Could not match supplied host pattern, ignoring: db_servers


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 (file_servers): file_servers	TAGS: []
      TASK TAGS: [samba]

  play #4 (db_servers): db_servers	TAGS: []
      TASK TAGS: []
jessielazo@Desktop:~/HOA6$ 
```

It listed all of the available tags from the site.yml

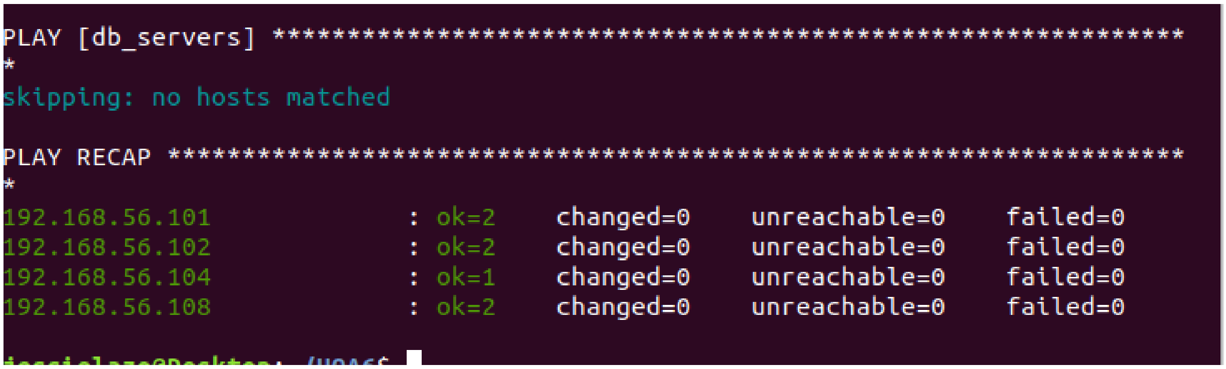
2.2 *ansible-playbook --tags centos --ask-become-pass site.yml*



```
TASK [install updates (Ubuntu)] ***********************************************
*
skipping: [192.168.56.104]
ok: [192.168.56.101]
ok: [192.168.56.102]
ok: [192.168.56.108]
 [WARNING]: Could not match supplied host pattern, ignoring: web_servers


PLAY [web_servers] ************************************************************
*
skipping: no hosts matched
 [WARNING]: Could not match supplied host pattern, ignoring: file_servers


PLAY [file_servers] ***********************************************************
*
skipping: no hosts matched
 [WARNING]: Could not match supplied host pattern, ignoring: db_servers


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

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

It skipped every process which does not include the ‘centos’ in its tags.

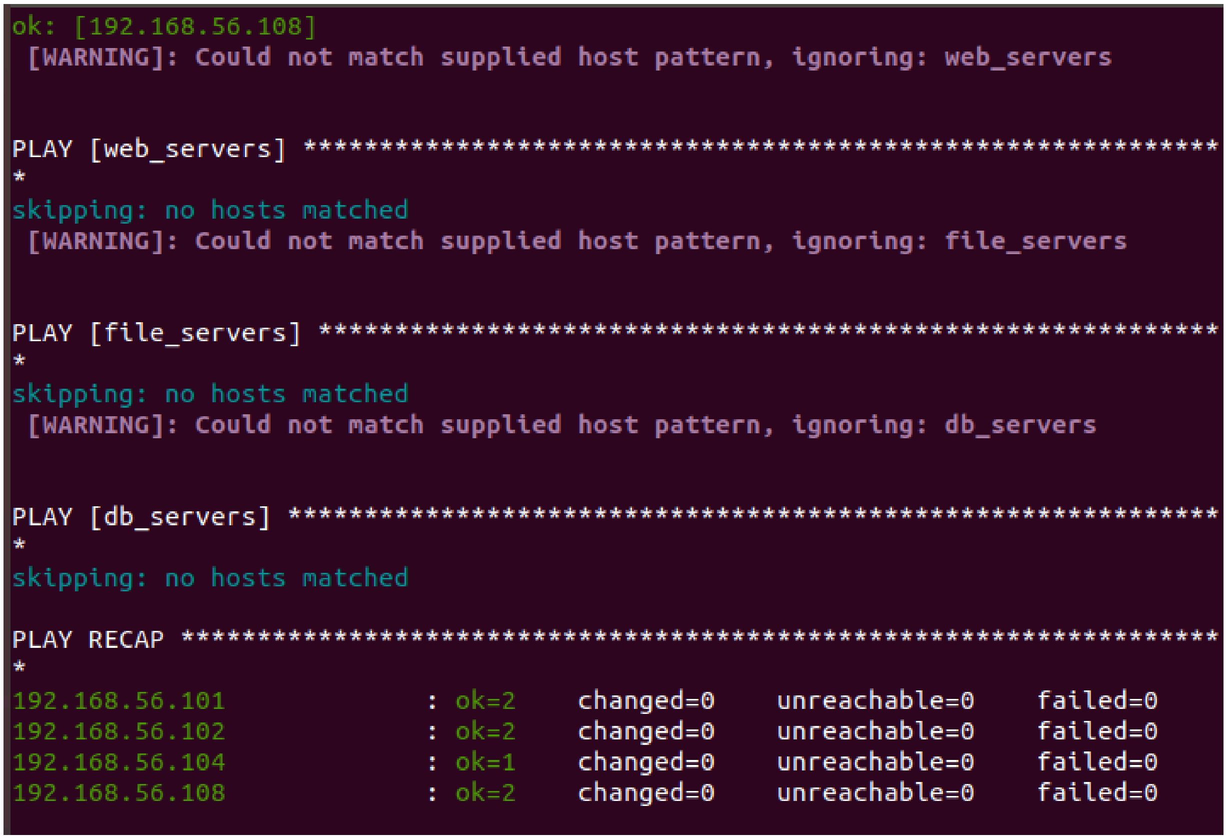
2.3 *ansible-playbook --tags db --ask-become-pass site.yml*



```
PLAY [db_servers] *************************************************************
*
skipping: no hosts matched

PLAY RECAP ********************************************************************
*
192.168.56.101             : ok=2    changed=0    unreachable=0    failed=0
192.168.56.102             : ok=2    changed=0    unreachable=0    failed=0
192.168.56.104             : ok=1    changed=0    unreachable=0    failed=0
192.168.56.108             : ok=2    changed=0    unreachable=0    failed=0

jessiclaze@Desktop:~/HOA6$
```

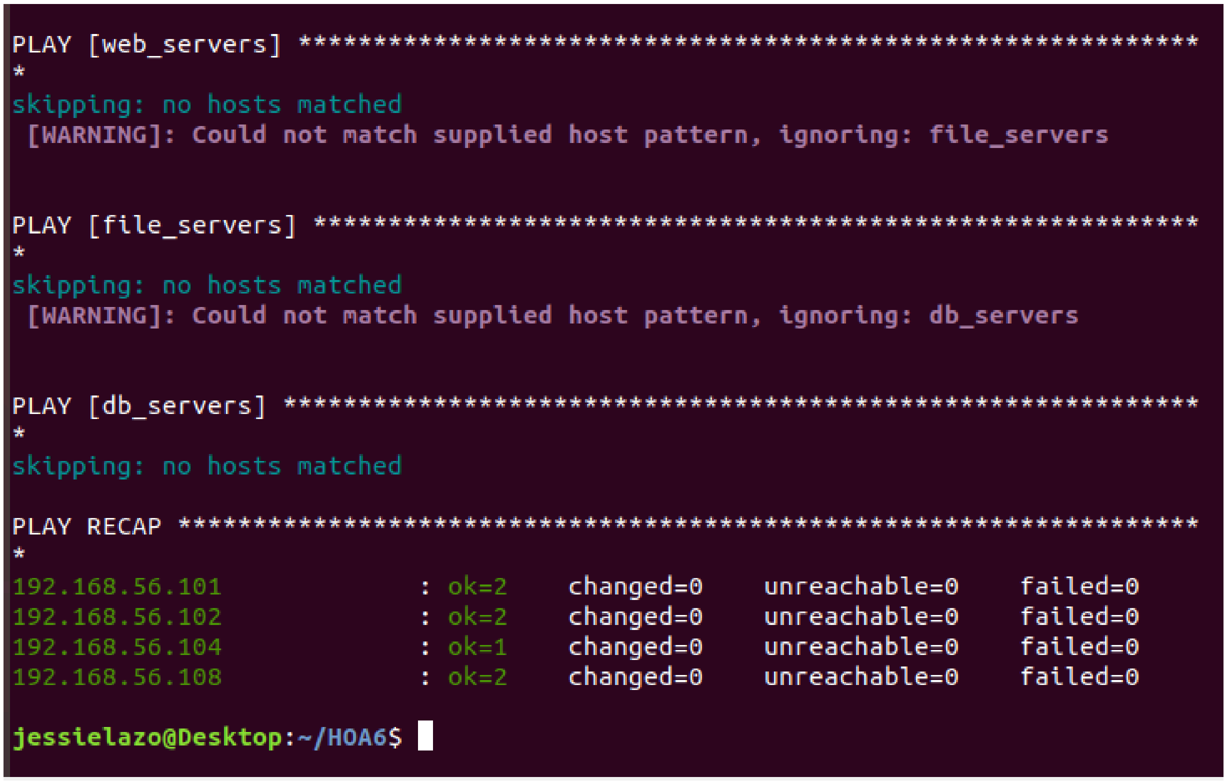
It skipped every process which does not include ‘db’ in its tags

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



```
ok: [192.168.56.108]
 [WARNING]: Could not match supplied host pattern, ignoring: web_servers


PLAY [web_servers] ************************************************************
*
skipping: no hosts matched
 [WARNING]: Could not match supplied host pattern, ignoring: file_servers


PLAY [file_servers] ***********************************************************
*
skipping: no hosts matched
 [WARNING]: Could not match supplied host pattern, ignoring: db_servers


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

PLAY RECAP ********************************************************************
*
192.168.56.101             : ok=2    changed=0    unreachable=0    failed=0
192.168.56.102             : ok=2    changed=0    unreachable=0    failed=0
192.168.56.104             : ok=1    changed=0    unreachable=0    failed=0
192.168.56.108             : ok=2    changed=0    unreachable=0    failed=0
```

It skipped every process which does not include the ‘apache’ in its tags.



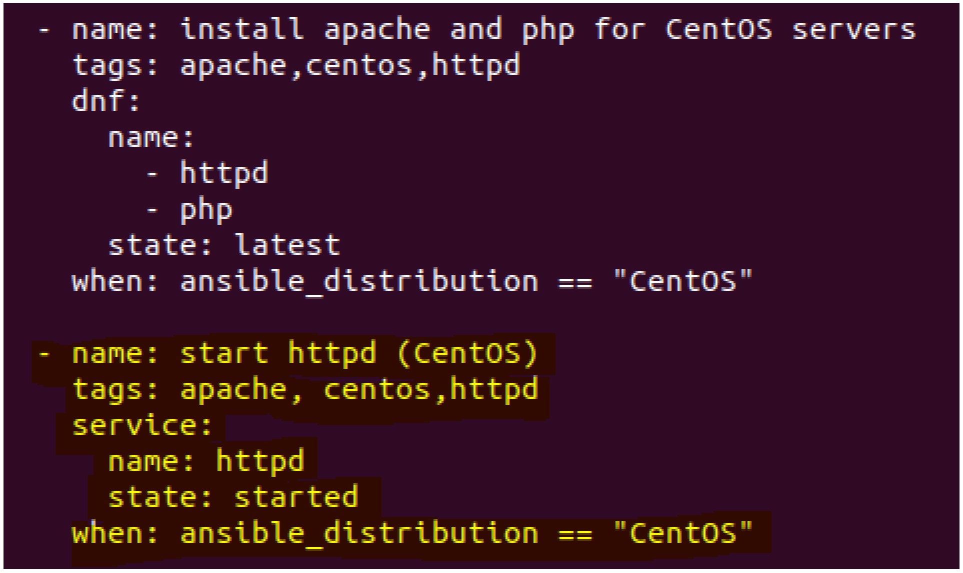
```
PLAY [web_servers] ************************************************************
*
skipping: no hosts matched
 [WARNING]: Could not match supplied host pattern, ignoring: file_servers


PLAY [file_servers] ***********************************************************
*
skipping: no hosts matched
 [WARNING]: Could not match supplied host pattern, ignoring: db_servers


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

PLAY RECAP ********************************************************************
*
192.168.56.101             : ok=2    changed=0    unreachable=0    failed=0
192.168.56.102             : ok=2    changed=0    unreachable=0    failed=0
192.168.56.104             : ok=1    changed=0    unreachable=0    failed=0
192.168.56.108             : ok=2    changed=0    unreachable=0    failed=0

jessielazo@Desktop:~/HOA6$ 
```

It skipped every process which does not include the ‘apache’, and ‘db’ in its tags

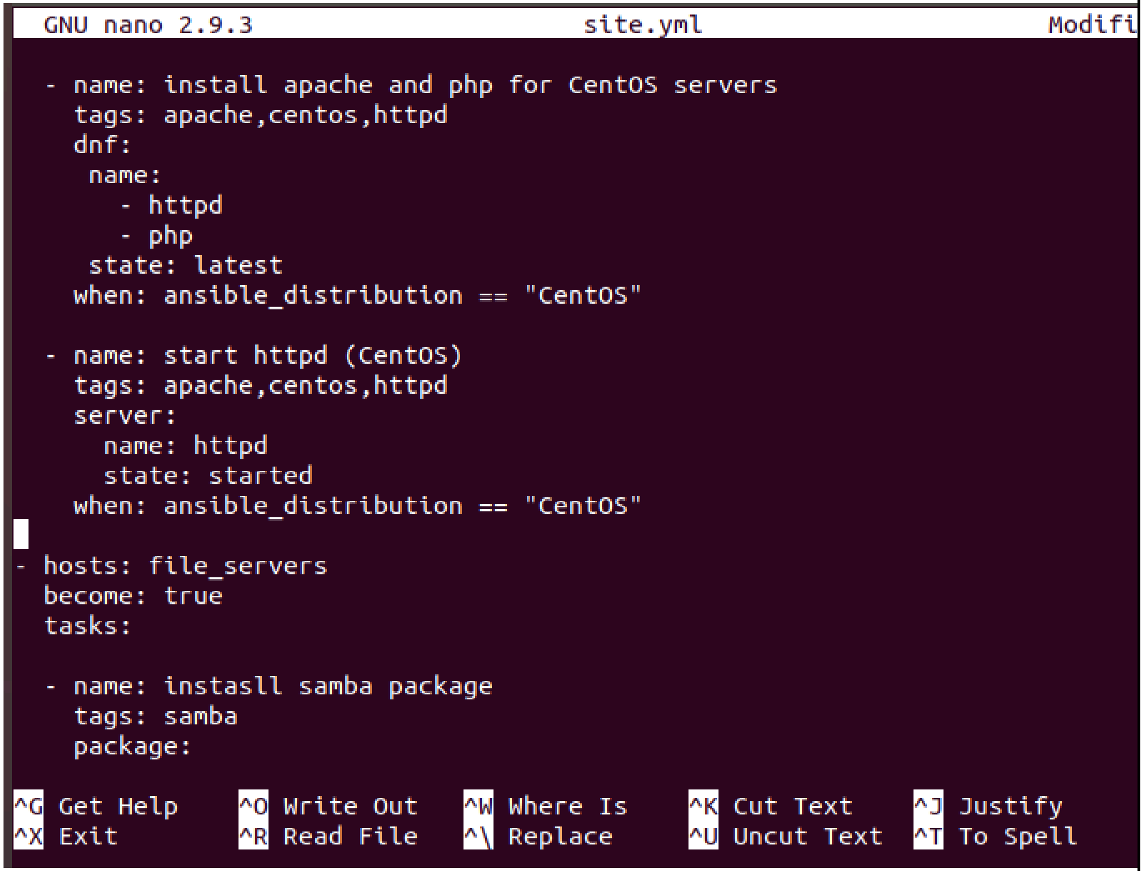
2.5 *ansible-playbook --tags “apache,db” --ask-become-pass site.yml*

Task 3: Managing Services

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



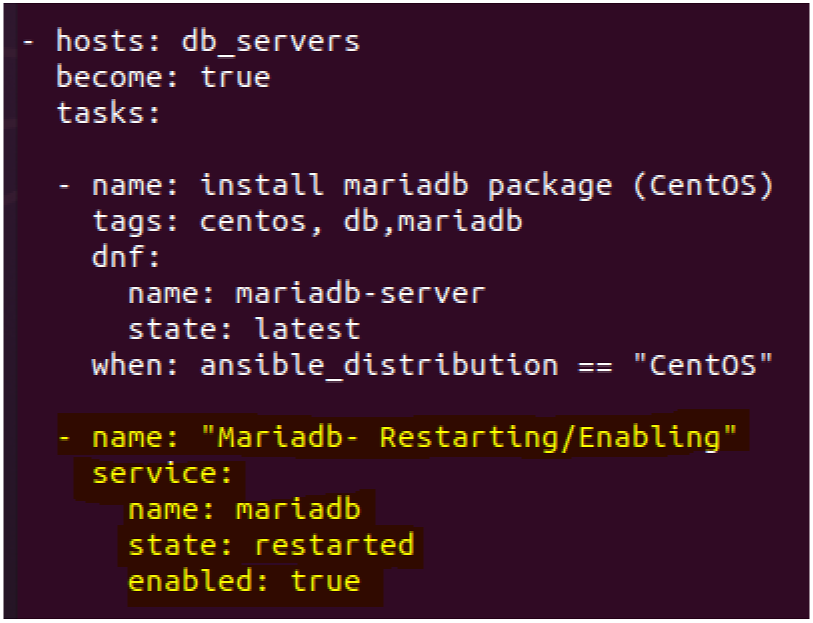
```
- name: install apache and php for CentOS servers
  tags: apache,centos,httpd
  dnf:
    name:
      - httpd
      - php
    state: latest
  when: ansible_distribution == "CentOS"

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



```
  GNU nano 2.9.3                    site.yml                    Modifi

  - 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
    server:
      name: httpd
      state: started
    when: ansible_distribution == "CentOS"

- hosts: file_servers
  become: true
  tasks:

  - name: instasll samba package
    tags: samba
    package:
```

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.

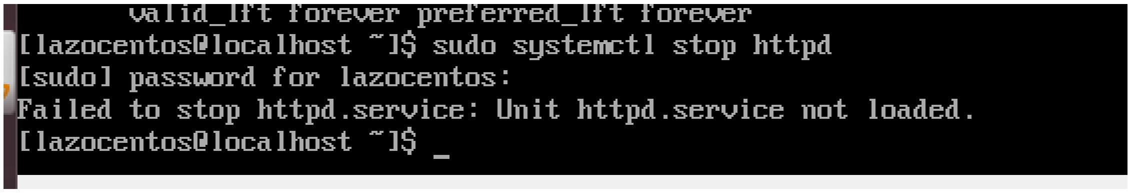


```
- hosts: db_servers
  become: true
  tasks:

  - 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.



```
       valid_lft forever preferred_lft forever
[lazocentos@localhost ~]$ sudo systemctl stop httpd
[sudo] password for lazocentos:
Failed to stop httpd.service: Unit httpd.service not loaded.
[lazocentos@localhost ~]$ _
```

3. Go to the local machine and this time, run the *site.yml* file. Then after running the file, go again to the CentOS server and enter its IP address on the browser. Describe the result.
   To automatically enable the service every time we run the playbook, use the command *enabled: true* similar to Figure 7.1.2 and save the playbook.

Reflections:

Answer the following:

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

Based on what I observed while performing the activity is that putting

remote servers into groups makes the management more organized, easier to debug for errors, and easier to specify code blocks to run inside a playbook file.

2. What is the importance of tags in playbooks?

Implementing tags on the playbook enables the administrator to specify the specific operations which are written inside the playbook.

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

Because some services do not automatically start after the installation, hence creating a service management automation in the playbook decreases the possibility of receiving an error after running a playbook