Name: Ian Carlo T. Bello	**Date Performed:** October 8, 2022
Course/Section: CPE31S24	**Date Submitted:** October 8, 2022
Instructor: Dr. Jonathan V. Taylar	**Semester and SY:** 1st sem – 3rd year

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

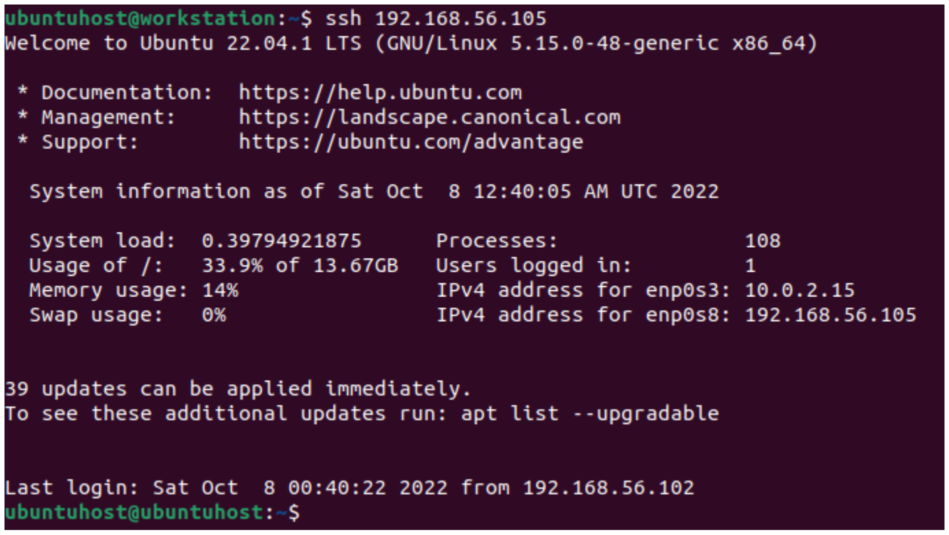
2. **Discussion**:

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:

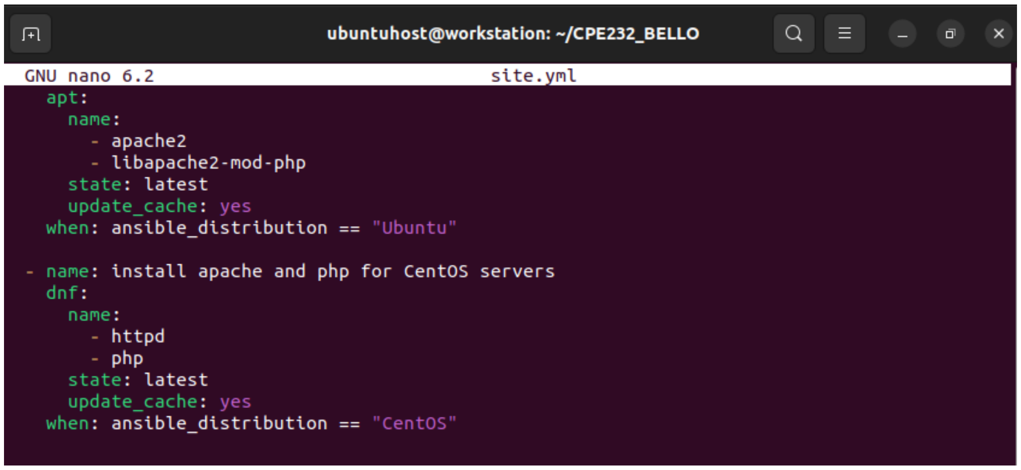
In this activity, you will need to create another Ubuntu VM and name it Server 3. Likewise, you need to activate the second adapter to a host-only adapter after the installations. Take note of the IP address of the Server 3. Make sure to use the command *ssh-copy-id* to copy the public key to Server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



```
ubuntuhost@workstation:~$ ssh 192.168.56.105
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-48-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Oct  8 12:40:05 AM UTC 2022

  System load:  0.39794921875      Processes:               108
  Usage of /:   33.9% of 13.67GB   Users logged in:         1
  Memory usage: 14%                IPv4 address for enp0s3: 10.0.2.15
  Swap usage:   0%                 IPv4 address for enp0s8: 192.168.56.105


39 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


Last login: Sat Oct  8 00:40:22 2022 from 192.168.56.102
ubuntuhost@ubuntuhost:~$
```

Task 1: Targeting Specific Nodes

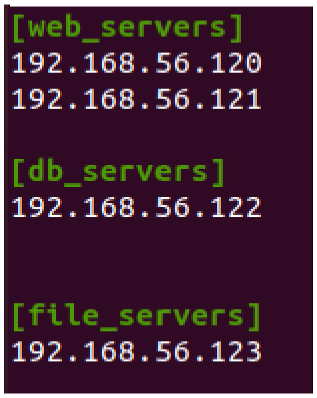
1. Create a new playbook and named it site.yml. Follow the commands as shown in the image below. Make sure to save the file and exit.



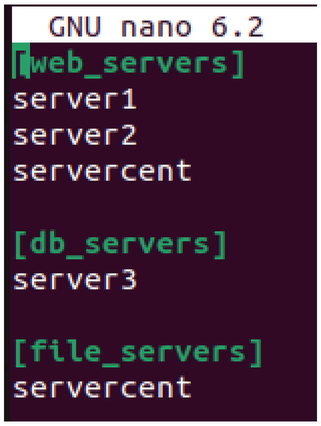
ubuntuhost@workstation: ~/CPE232_BELLO

```
GNU nano 6.2                              site.yml
  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
    update_cache: yes
  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
```

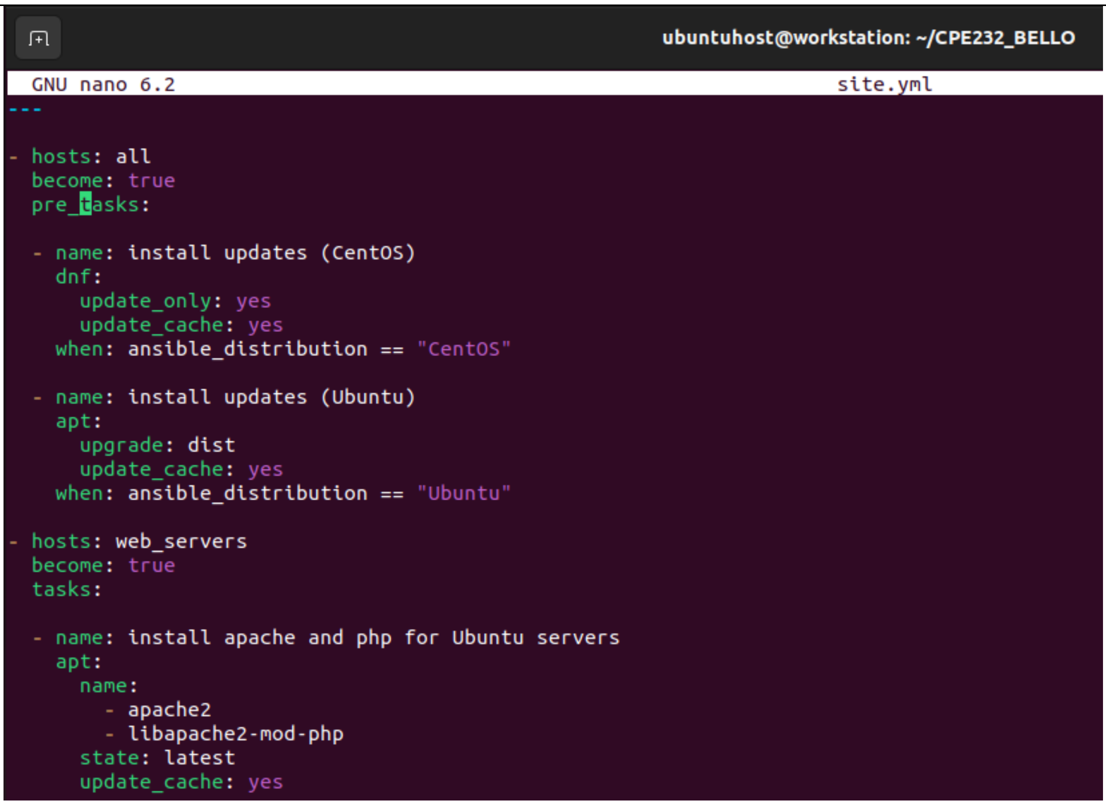


```
GNU nano 6.2
[web_servers]
server1
server2
servercent

[db_servers]
server3

[file_servers]
servercent
```

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:



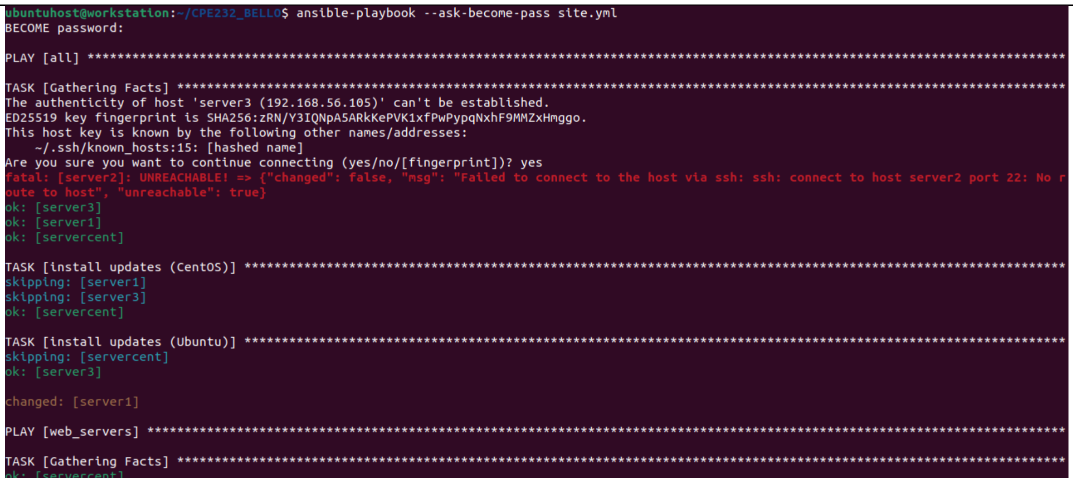
```
GNU nano 6.2                                                     site.yml
---

- hosts: all
  become: true
  pre_tasks:

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

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

- hosts: web_servers
  become: true
  tasks:

  - name: install apache and php for Ubuntu servers
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
      update_cache: yes
```

Make sure to save the file and exit.

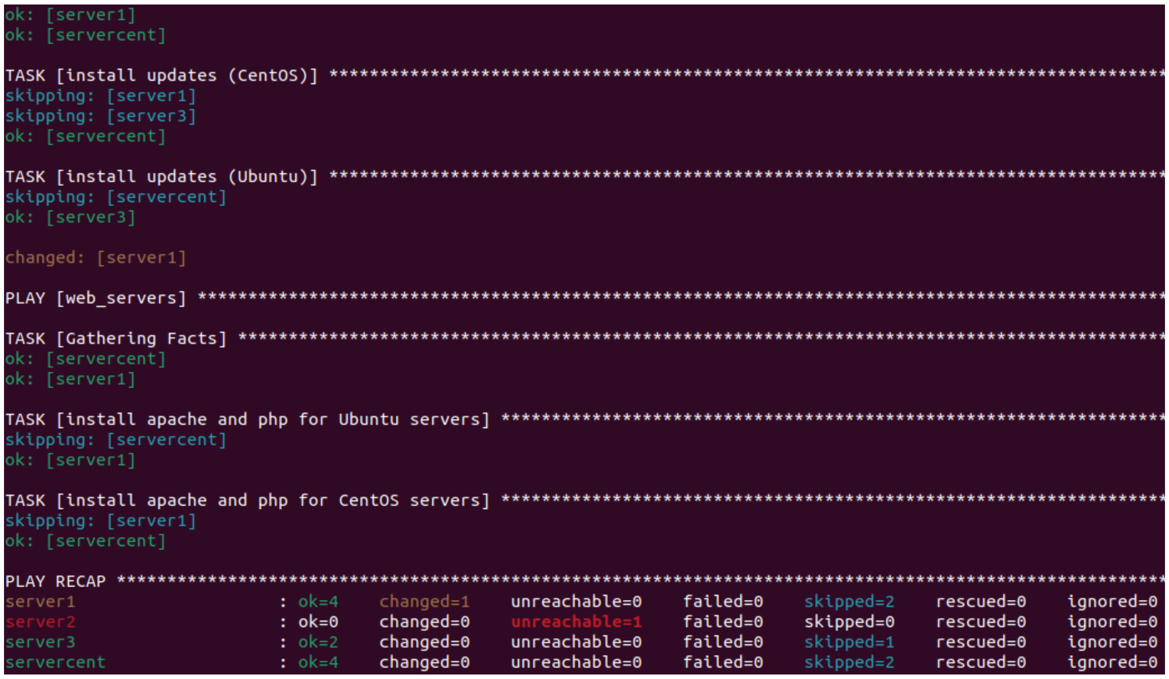
The *pre-tasks* command tells the ansible to run it before any other thing. In the *pre-tasks*, CentOS will install updates while Ubuntu will upgrade its distribution package. This will run before running the second play, which is targeted at *web_servers*. In the second play, apache and php will be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

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

First run does have an error because ansible doesn't know the newly added vm and asked for the ssh key after that all servers got updated and checked if apache and php are installed.

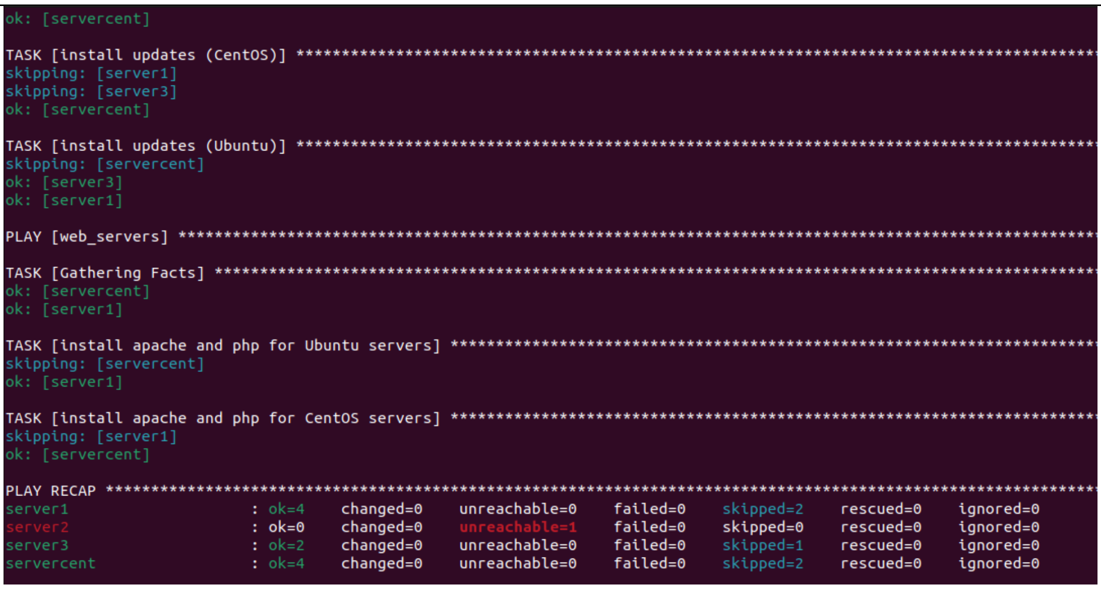


```
ubuntuhost@workstation:~/CPE232_BELLO$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] ****************************************************************************************
The authenticity of host 'server3 (192.168.56.105)' can't be established.
ED25519 key fingerprint is SHA256:zRN/Y3IQNpA5ARkKePVK1xfPwPypqNxhF9MMZxHmggo.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:15: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
fatal: [server2]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host server2 port 22: No r
oute to host", "unreachable": true}
ok: [server3]
ok: [server1]
ok: [servercent]

TASK [install updates (CentOS)] *******************************************************************************
skipping: [server1]
skipping: [server3]
ok: [servercent]

TASK [install updates (Ubuntu)] *******************************************************************************
skipping: [servercent]
ok: [server3]

changed: [server1]

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

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



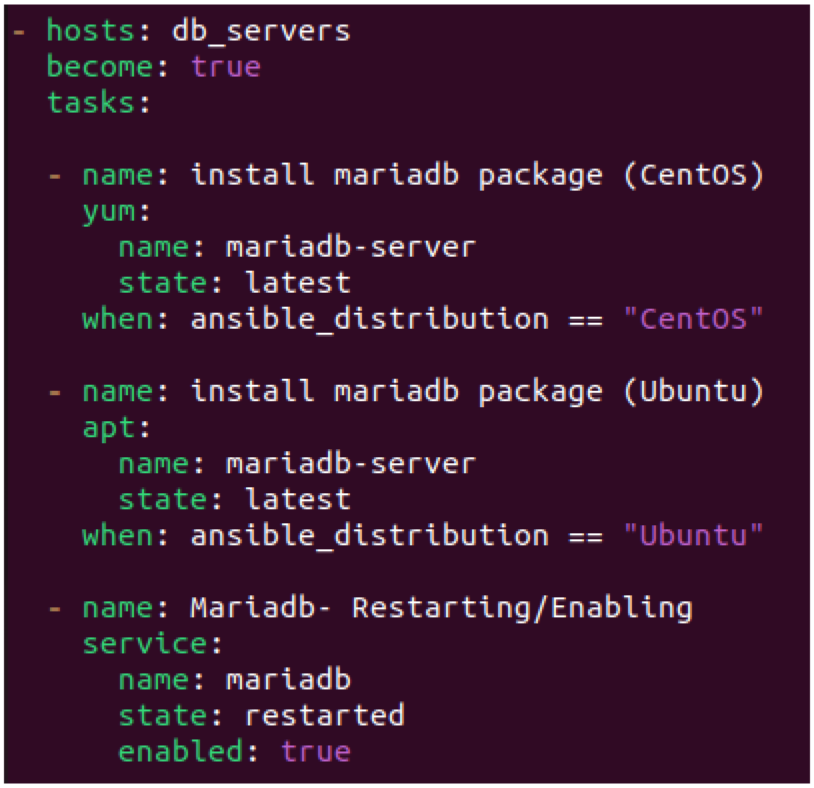
```
ok: [server1]
ok: [servercent]

TASK [install updates (CentOS)] ******************************************************************
skipping: [server1]
skipping: [server3]
ok: [servercent]

TASK [install updates (Ubuntu)] ******************************************************************
skipping: [servercent]
ok: [server3]

changed: [server1]

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

TASK [Gathering Facts] ***************************************************************************
ok: [servercent]
ok: [server1]

TASK [install apache and php for Ubuntu servers] *************************************************
skipping: [servercent]
ok: [server1]

TASK [install apache and php for CentOS servers] *************************************************
skipping: [server1]
ok: [servercent]

PLAY RECAP ***************************************************************************************
server1                    : ok=4    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
server2                    : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
server3                    : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
servercent                 : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

Second run went smoothly and just updates server1



```
ok: [servercent]

TASK [install updates (CentOS)] **************************************************************************************
skipping: [server1]
skipping: [server3]
ok: [servercent]

TASK [install updates (Ubuntu)] **************************************************************************************
skipping: [servercent]
ok: [server3]
ok: [server1]

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

TASK [Gathering Facts] ***********************************************************************************************
ok: [servercent]
ok: [server1]

TASK [install apache and php for Ubuntu servers] *********************************************************************
skipping: [servercent]
ok: [server1]

TASK [install apache and php for CentOS servers] *********************************************************************
skipping: [server1]
ok: [servercent]

PLAY RECAP ***********************************************************************************************************
server1                    : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
server2                    : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
server3                    : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
servercent                 : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

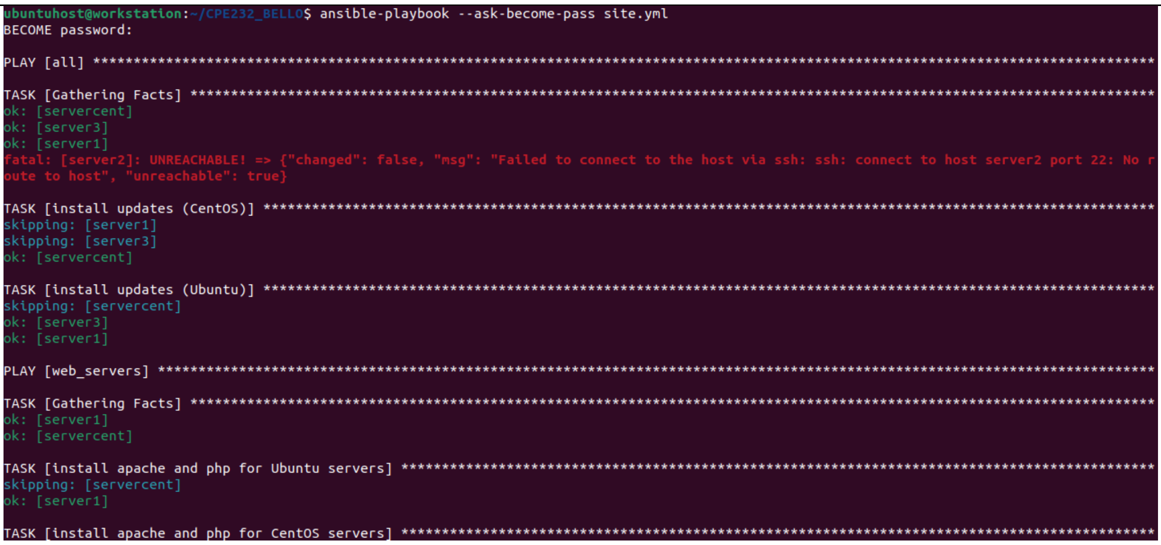
4. Let's try to edit again the *site.yml* file. This time, we are going to add plays targeting the other servers. This time we target the *db_servers* by adding it on the current *site.yml*. Below is an example: (Note add this at the end of the playbooks from task 1.3.



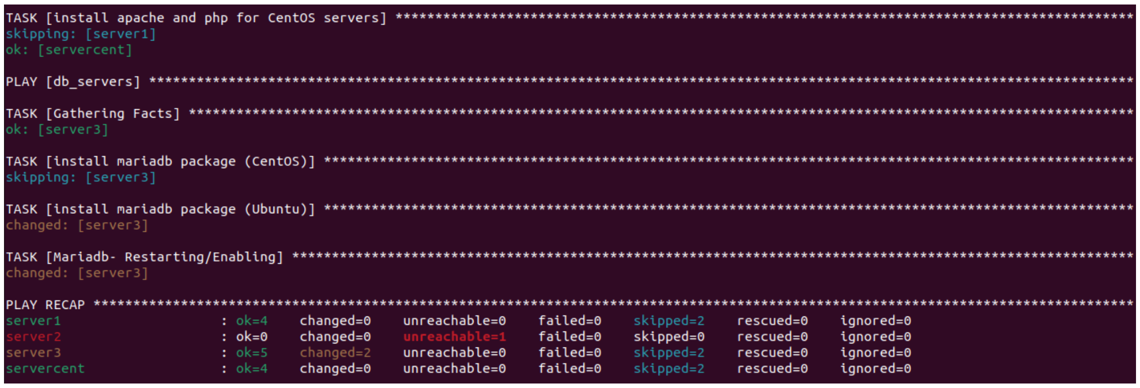
```
- hosts: db_servers
  become: true
  tasks:

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

  - name: 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
```

Make sure to save the file and exit.

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



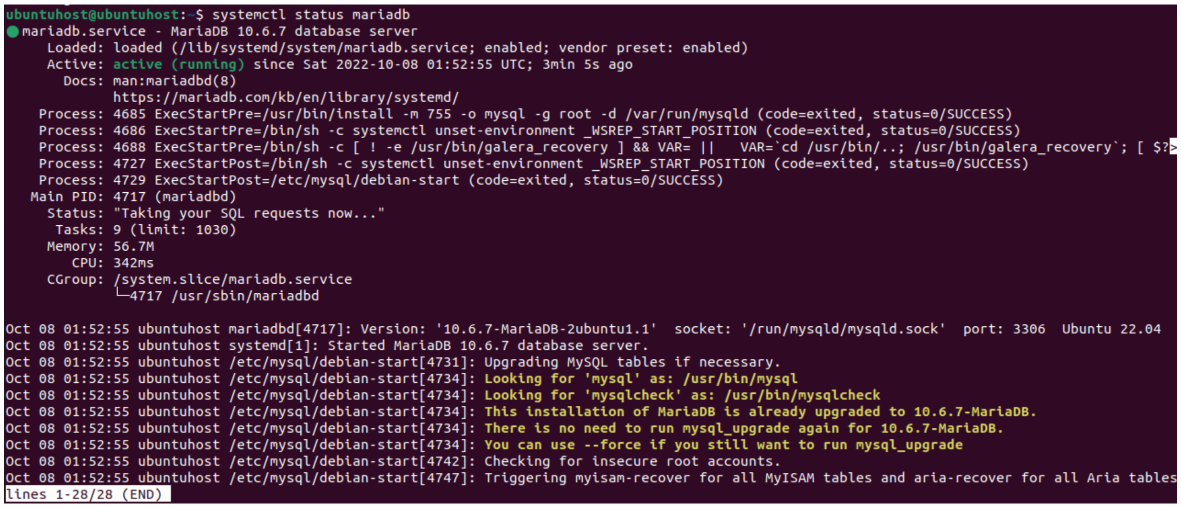
```
ubuntuhost@workstation:~/CPE232_BELLO$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************************
ok: [servercent]
ok: [server3]
ok: [server1]
fatal: [server2]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host server2 port 22: No r
oute to host", "unreachable": true}

TASK [install updates (CentOS)] ************************************************************************************************
skipping: [server1]
skipping: [server3]
ok: [servercent]

TASK [install updates (Ubuntu)] ************************************************************************************************
skipping: [servercent]
ok: [server3]
ok: [server1]

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

TASK [Gathering Facts] *********************************************************************************************************
ok: [server1]
ok: [servercent]

TASK [install apache and php for Ubuntu servers] *******************************************************************************
skipping: [servercent]
ok: [server1]

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



```
TASK [install apache and php for CentOS servers] *******************************************************************************
skipping: [server1]
ok: [servercent]

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

TASK [Gathering Facts] *********************************************************************************************************
ok: [server3]

TASK [install mariadb package (CentOS)] ****************************************************************************************
skipping: [server3]

TASK [install mariadb package (Ubuntu)] ****************************************************************************************
changed: [server3]

TASK [Mariadb- Restarting/Enabling] ********************************************************************************************
changed: [server3]

PLAY RECAP *********************************************************************************************************************
server1                    : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
server2                    : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
server3                    : ok=5    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
servercent                 : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

The mariadb package has been installed in the server3

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.

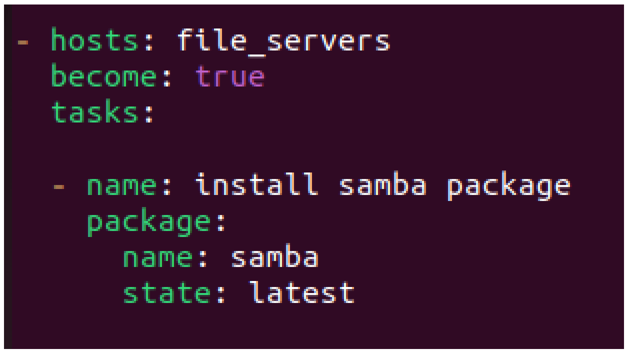


```
ubuntuhost@ubuntuhost:~$ systemctl status mariadb
● mariadb.service - MariaDB 10.6.7 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2022-10-08 01:52:55 UTC; 3min 5s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 4685 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysqld (code=exited, status=0/SUCCESS)
    Process: 4686 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
    Process: 4688 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= ||  VAR=`cd /usr/bin/..; /usr/bin/galera_recovery`; [ $?>
    Process: 4727 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
    Process: 4729 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
   Main PID: 4717 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 9 (limit: 1030)
     Memory: 56.7M
        CPU: 342ms
     CGroup: /system.slice/mariadb.service
             └─4717 /usr/sbin/mariadbd

Oct 08 01:52:55 ubuntuhost mariadbd[4717]: Version: '10.6.7-MariaDB-2ubuntu1.1'  socket: '/run/mysqld/mysqld.sock'  port: 3306  Ubuntu 22.04
Oct 08 01:52:55 ubuntuhost systemd[1]: Started MariaDB 10.6.7 database server.
Oct 08 01:52:55 ubuntuhost /etc/mysql/debian-start[4731]: Upgrading MySQL tables if necessary.
Oct 08 01:52:55 ubuntuhost /etc/mysql/debian-start[4734]: Looking for 'mysql' as: /usr/bin/mysql
Oct 08 01:52:55 ubuntuhost /etc/mysql/debian-start[4734]: Looking for 'mysqlcheck' as: /usr/bin/mysqlcheck
Oct 08 01:52:55 ubuntuhost /etc/mysql/debian-start[4734]: This installation of MariaDB is already upgraded to 10.6.7-MariaDB.
Oct 08 01:52:55 ubuntuhost /etc/mysql/debian-start[4734]: There is no need to run mysql_upgrade again for 10.6.7-MariaDB.
Oct 08 01:52:55 ubuntuhost /etc/mysql/debian-start[4734]: You can use --force if you still want to run mysql_upgrade
Oct 08 01:52:55 ubuntuhost /etc/mysql/debian-start[4742]: Checking for insecure root accounts.
Oct 08 01:52:55 ubuntuhost /etc/mysql/debian-start[4747]: Triggering myisam-recover for all MyISAM tables and aria-recover for all Aria tables
lines 1-28/28 (END)
```

Describe the output.

It shows the mariadb is installed and running in server3

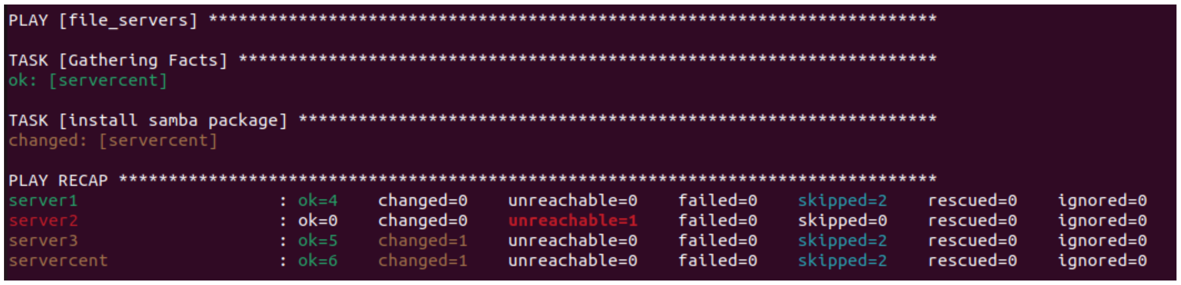
6. Edit the *site.yml* again. This time we will append the code to configure installation on the *file_servers* group. We can add the following on our file.



```
- 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.



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

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

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

PLAY RECAP ************************************************************************************
server1                    : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
server2                    : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
server3                    : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
servercent                 : ok=6    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

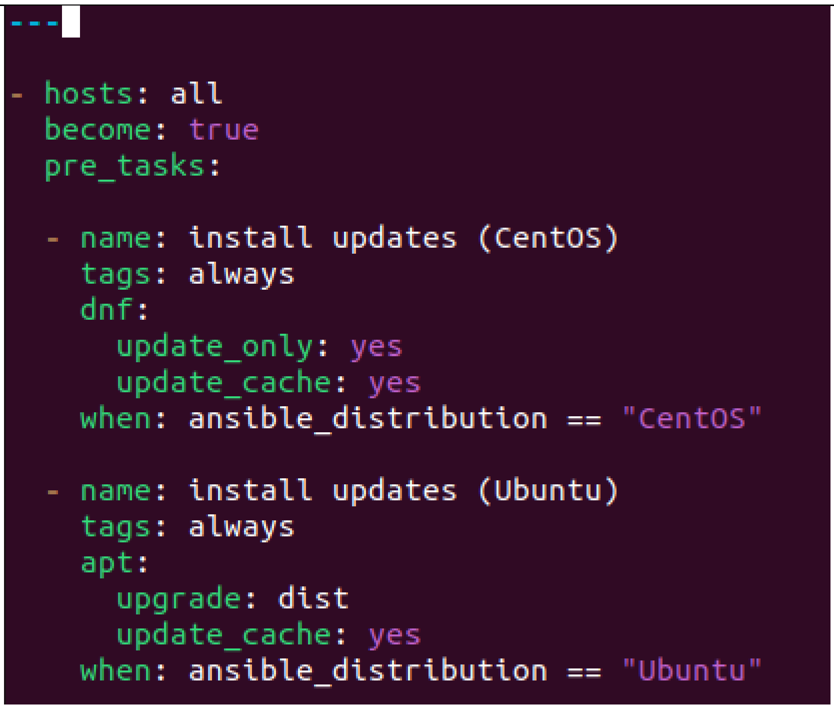
Samba was installed for servercent which is our file_servers

The testing of the *file_servers* is beyond the scope of this activity, and as well as our topics and objectives. However, in this activity we were able to show that we can target hosts or servers using grouping in ansible playbooks.

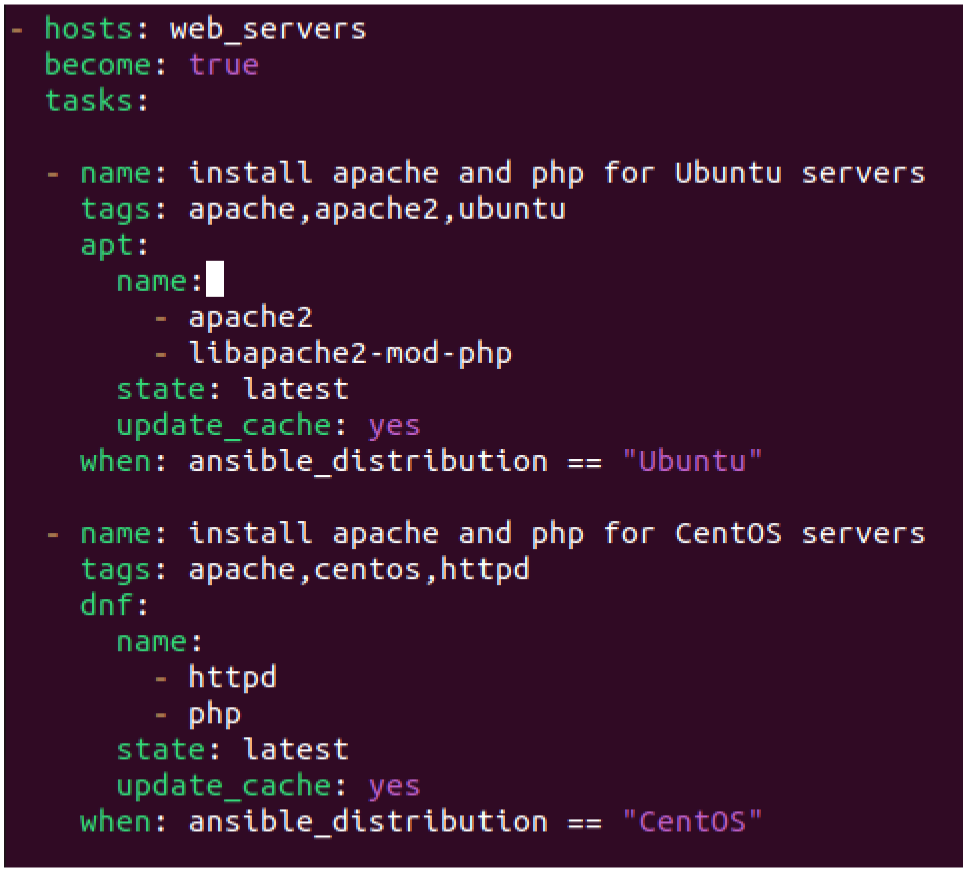
Task 2: Using Tags in running playbooks

In this task, our goal is to add metadata to our plays so that we can only run the plays that we want to run, and not all the plays in our playbook.

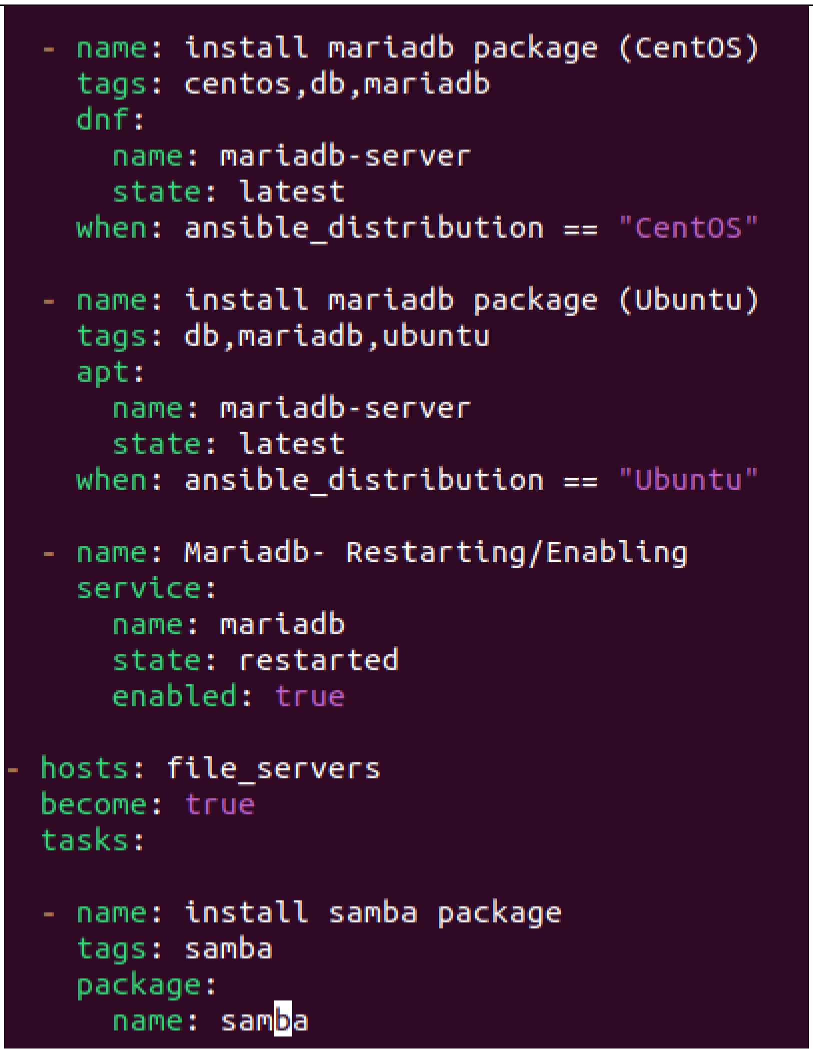
1. Edit the *site.yml* file. Add tags to the playbook. After the name, we can place the tags: *name_of_tag*. This is an arbitrary command, which means you can use any name for a tag.



```
---

- hosts: all
  become: true
  pre_tasks:

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

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



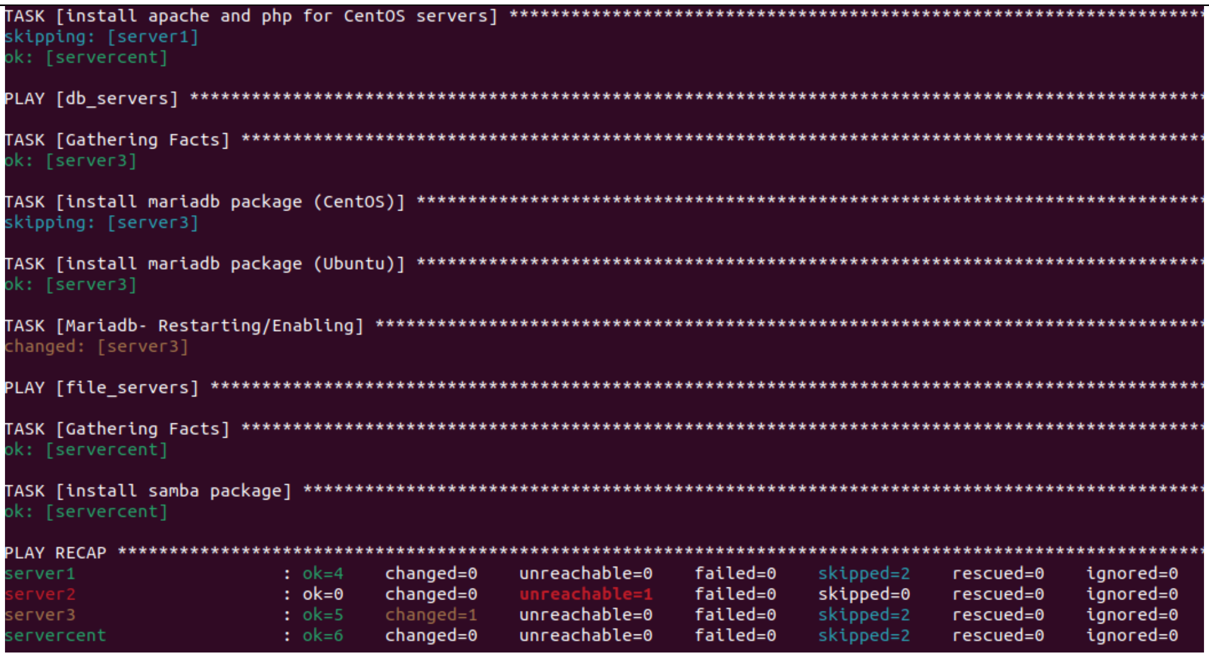
```
- hosts: web_servers
  become: true
  tasks:

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

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



```
  - name: install mariadb package (CentOS)
    tags: centos,db,mariadb
    dnf:
      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
```

Make sure to save the file and exit.

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

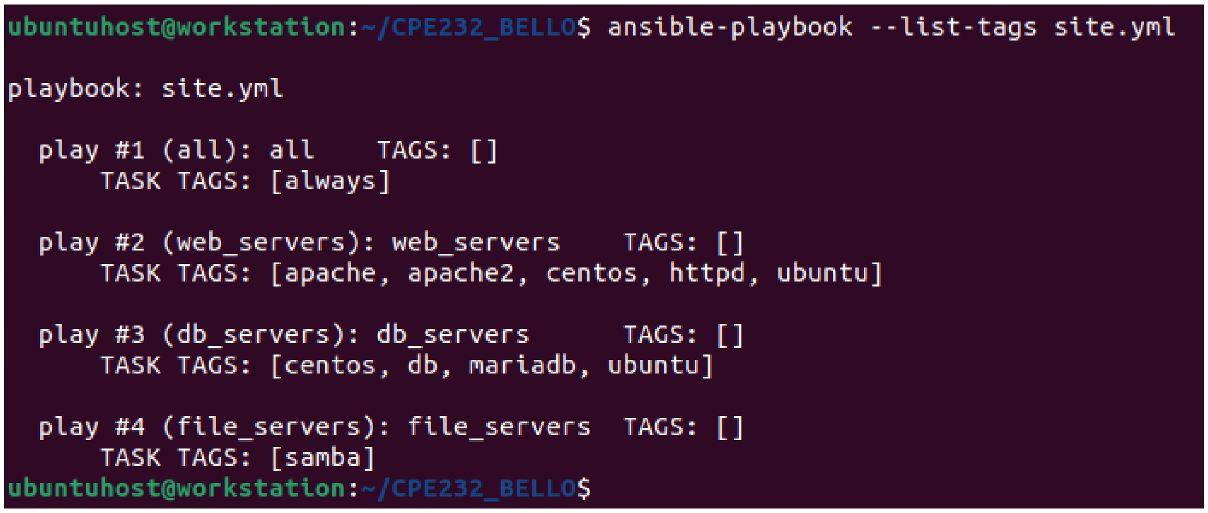


```
TASK [install apache and php for CentOS servers] ****************************************************************
skipping: [server1]
ok: [servercent]

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

TASK [Gathering Facts] ******************************************************************************************
ok: [server3]

TASK [install mariadb package (CentOS)] *************************************************************************
skipping: [server3]

TASK [install mariadb package (Ubuntu)] *************************************************************************
ok: [server3]

TASK [Mariadb- Restarting/Enabling] *****************************************************************************
changed: [server3]

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

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

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

PLAY RECAP ******************************************************************************************************
server1                    : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
server2                    : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
server3                    : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
servercent                 : ok=6    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

Same thing happened as last time we just added tags for tag issue command. We can also see that mariadb always restart/enable.

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

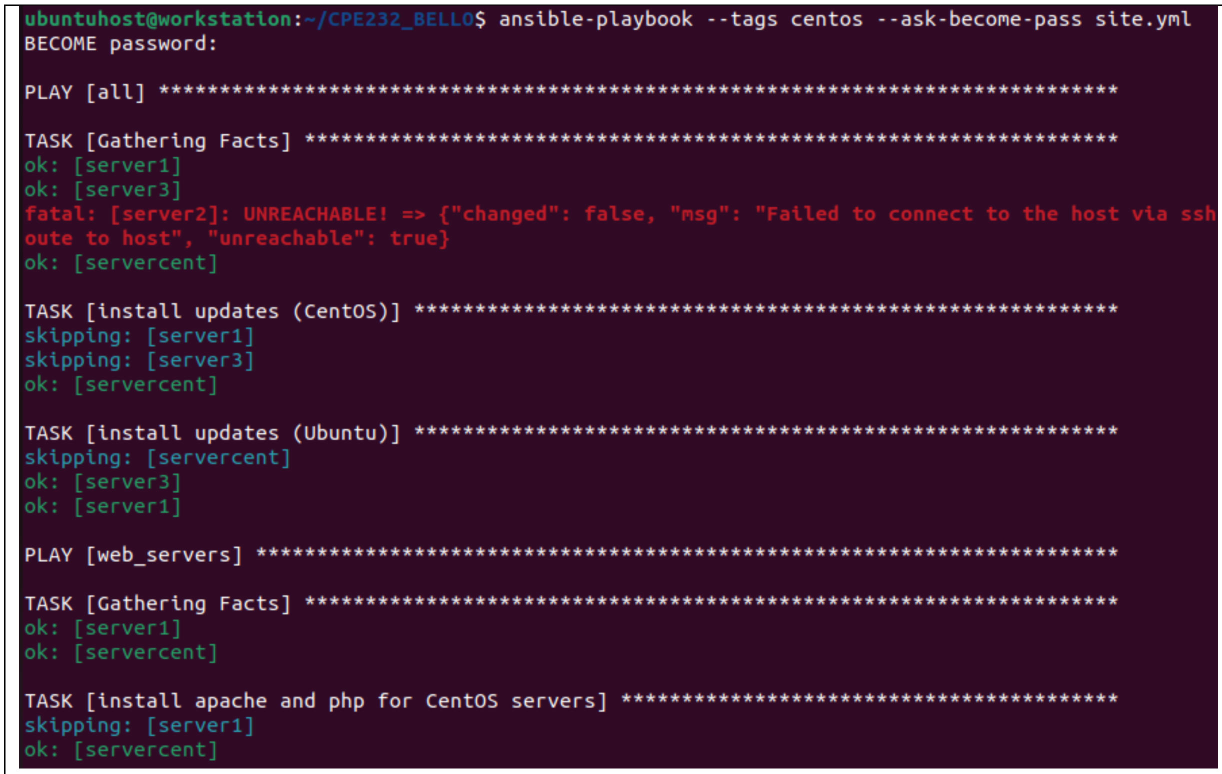
   2.1 *ansible-playbook --list-tags site.yml*



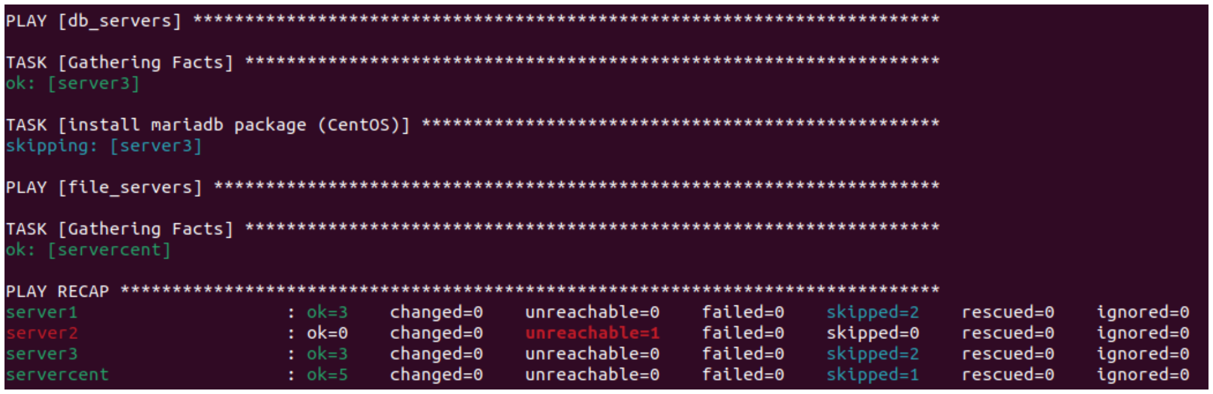
```
ubuntuhost@workstation:~/CPE232_BELLO$ ansible-playbook --list-tags site.yml

playbook: site.yml

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

  play #2 (web_servers): web_servers	TAGS: []
      TASK TAGS: [apache, apache2, centos, httpd, ubuntu]

  play #3 (db_servers): db_servers	TAGS: []
      TASK TAGS: [centos, db, mariadb, ubuntu]

  play #4 (file_servers): file_servers	TAGS: []
      TASK TAGS: [samba]
ubuntuhost@workstation:~/CPE232_BELLO$
```

This shows the list of tags in our site.yml.

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



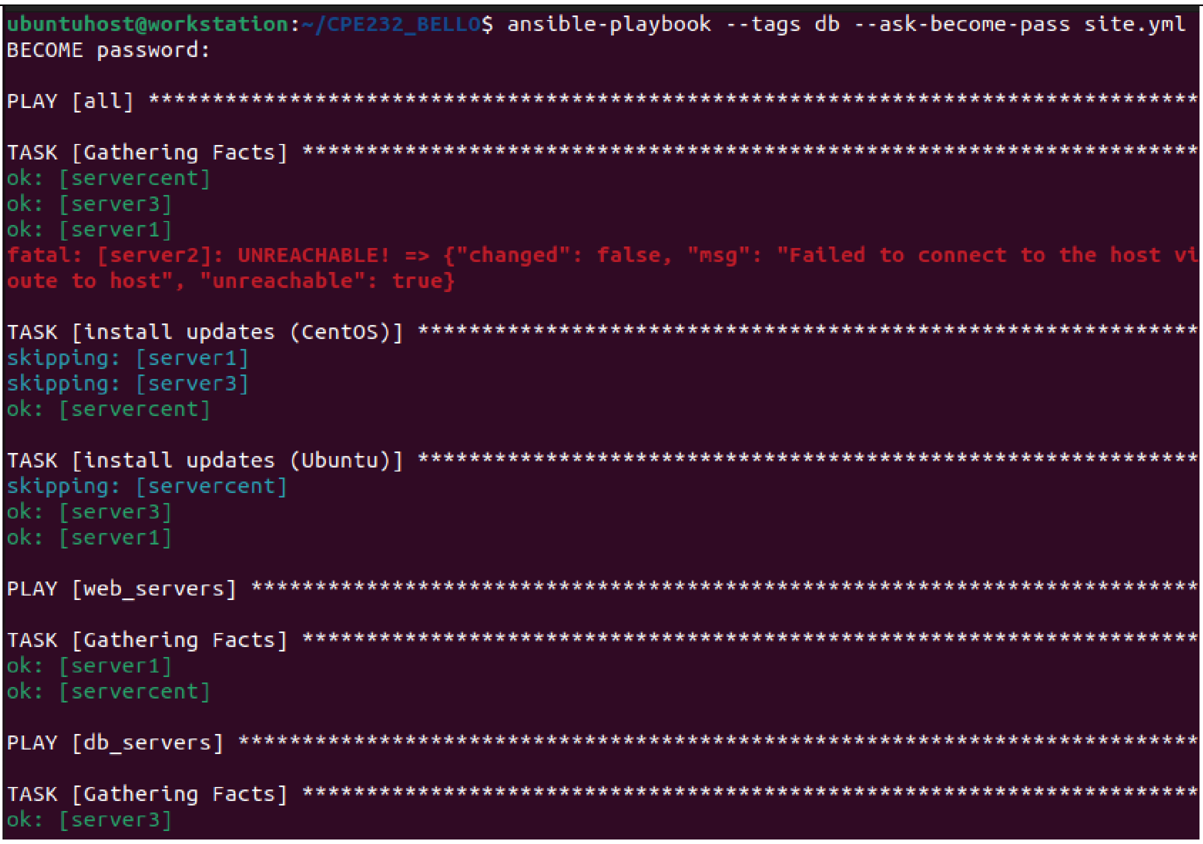
```
ubuntuhost@workstation:~/CPE232_BELLO$ ansible-playbook --tags centos --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] ***************************************************************
ok: [server1]
ok: [server3]
fatal: [server2]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh
oute to host", "unreachable": true}
ok: [servercent]

TASK [install updates (CentOS)] ******************************************************
skipping: [server1]
skipping: [server3]
ok: [servercent]

TASK [install updates (Ubuntu)] ******************************************************
skipping: [servercent]
ok: [server3]
ok: [server1]

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

TASK [Gathering Facts] ***************************************************************
ok: [server1]
ok: [servercent]

TASK [install apache and php for CentOS servers] *************************************
skipping: [server1]
ok: [servercent]
```



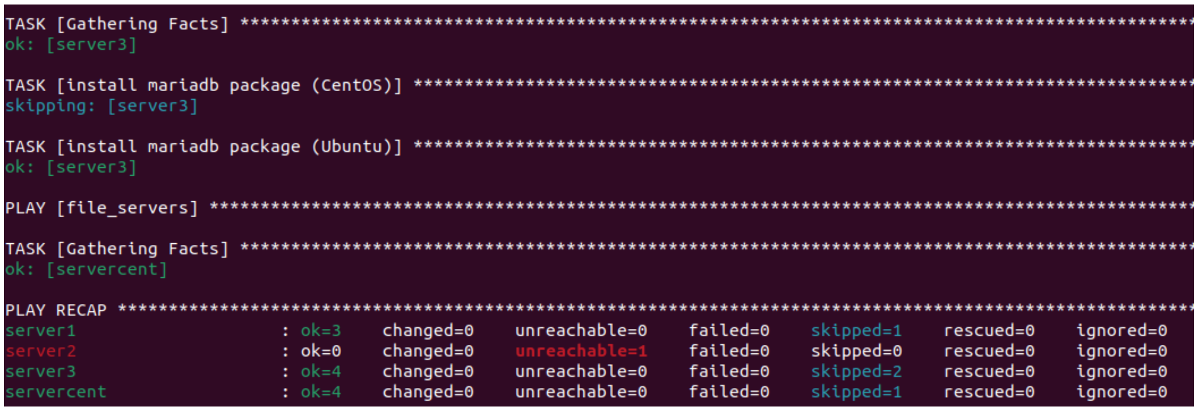
```
PLAY [db_servers] ********************************************************************

TASK [Gathering Facts] ***************************************************************
ok: [server3]

TASK [install mariadb package (CentOS)] **********************************************
skipping: [server3]

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

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

PLAY RECAP ***************************************************************************
server1                    : ok=3    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
server2                    : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
server3                    : ok=3    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
servercent                 : ok=5    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

This only runs tasks with the tag of centos.

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



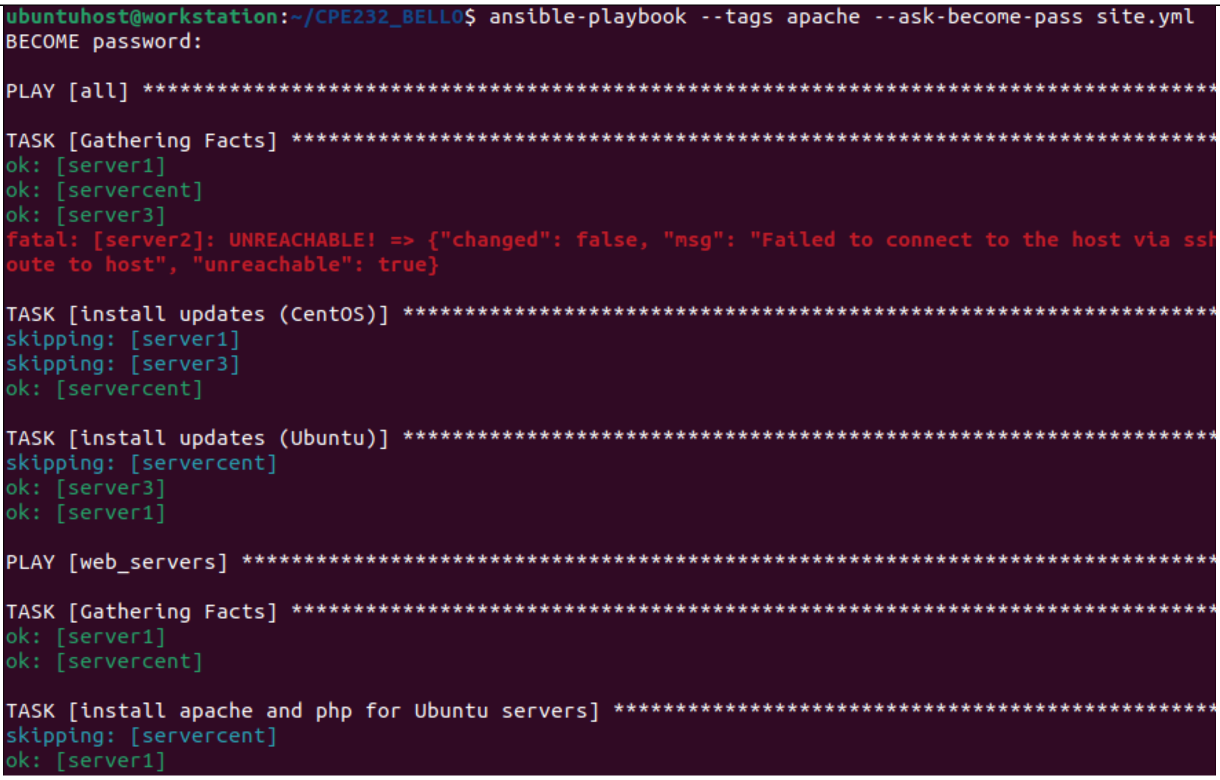
```
ubuntuhost@workstation:~/CPE232_BELLO$ ansible-playbook --tags db --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] ********************************************************************
ok: [servercent]
ok: [server3]
ok: [server1]
fatal: [server2]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host vi
oute to host", "unreachable": true}

TASK [install updates (CentOS)] ***********************************************************
skipping: [server1]
skipping: [server3]
ok: [servercent]

TASK [install updates (Ubuntu)] ***********************************************************
skipping: [servercent]
ok: [server3]
ok: [server1]

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

TASK [Gathering Facts] ********************************************************************
ok: [server1]
ok: [servercent]

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

TASK [Gathering Facts] ********************************************************************
ok: [server3]
```



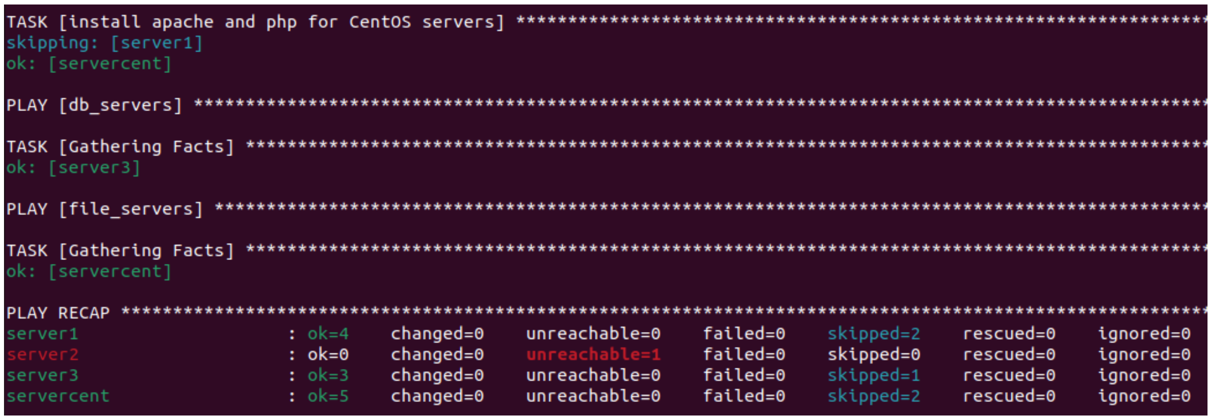
```
TASK [Gathering Facts] ***********************************************************************************
ok: [server3]

TASK [install mariadb package (CentOS)] ******************************************************************
skipping: [server3]

TASK [install mariadb package (Ubuntu)] ******************************************************************
ok: [server3]

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

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

PLAY RECAP ***********************************************************************************************
server1                    : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
server2                    : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
server3                    : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
servercent                 : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

This only runs tasks with the tag of db.

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



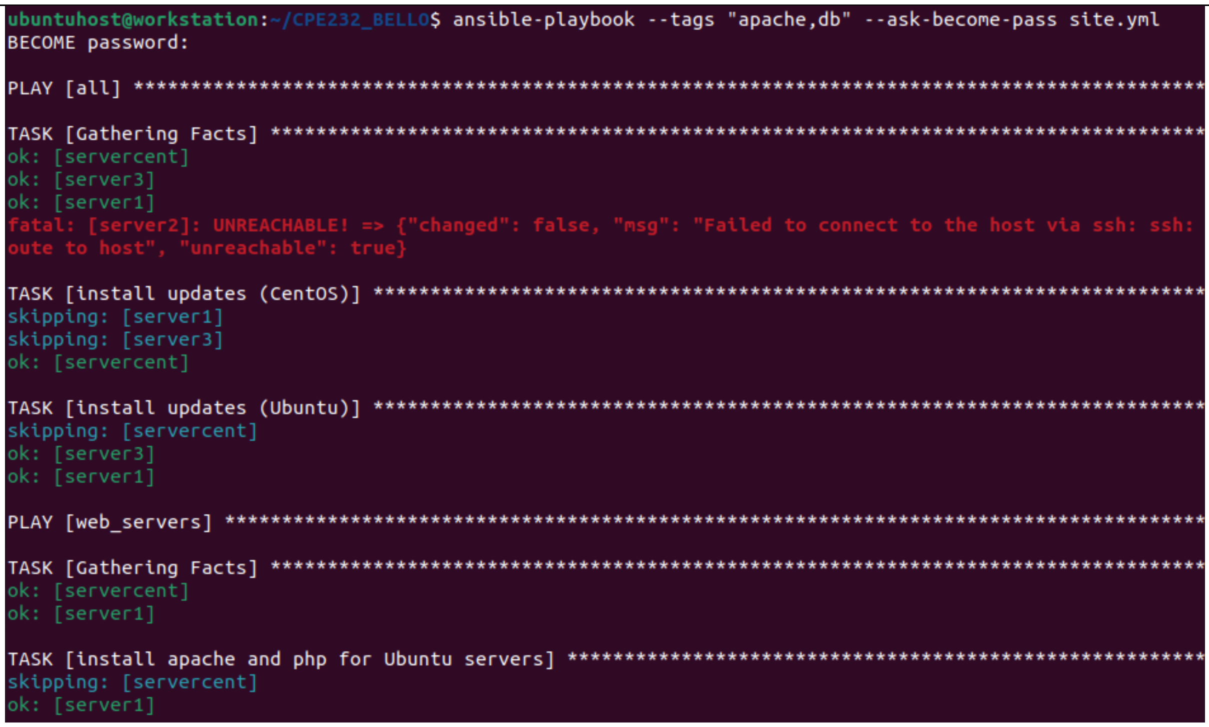
```
ubuntuhost@workstation:~/CPE232_BELLO$ ansible-playbook --tags apache --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] ********************************************************************
ok: [server1]
ok: [servercent]
ok: [server3]
fatal: [server2]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh
oute to host", "unreachable": true}

TASK [install updates (CentOS)] ***********************************************************
skipping: [server1]
skipping: [server3]
ok: [servercent]

TASK [install updates (Ubuntu)] ***********************************************************
skipping: [servercent]
ok: [server3]
ok: [server1]

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

TASK [Gathering Facts] ********************************************************************
ok: [server1]
ok: [servercent]

TASK [install apache and php for Ubuntu servers] ******************************************
skipping: [servercent]
ok: [server1]
```



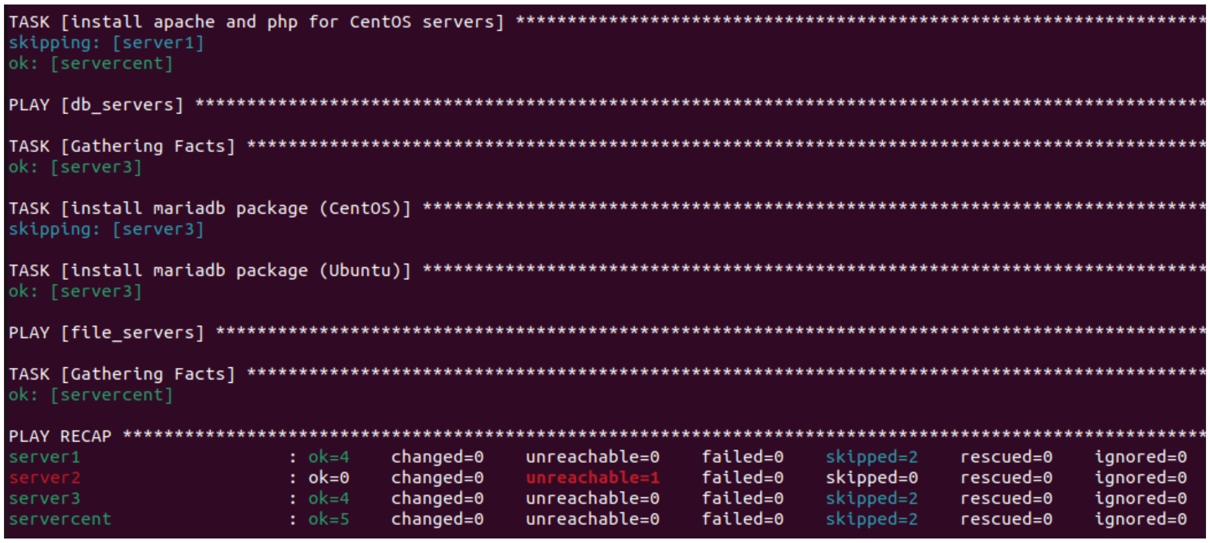
```
TASK [install apache and php for CentOS servers] ******************************************
skipping: [server1]
ok: [servercent]

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

TASK [Gathering Facts] ********************************************************************
ok: [server3]

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

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

PLAY RECAP ********************************************************************************
server1                    : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
server2                    : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
server3                    : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
servercent                 : ok=5    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

This only runs tasks with the tag of apache.

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



```
ubuntuhost@workstation:~/CPE232_BELLO$ ansible-playbook --tags "apache,db" --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************
ok: [servercent]
ok: [server3]
ok: [server1]
fatal: [server2]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh:
oute to host", "unreachable": true}

TASK [install updates (CentOS)] ************************************************************************
skipping: [server1]
skipping: [server3]
ok: [servercent]

TASK [install updates (Ubuntu)] ************************************************************************
skipping: [servercent]
ok: [server3]
ok: [server1]

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

TASK [Gathering Facts] *********************************************************************************
ok: [servercent]
ok: [server1]

TASK [install apache and php for Ubuntu servers] *******************************************************
skipping: [servercent]
ok: [server1]
```

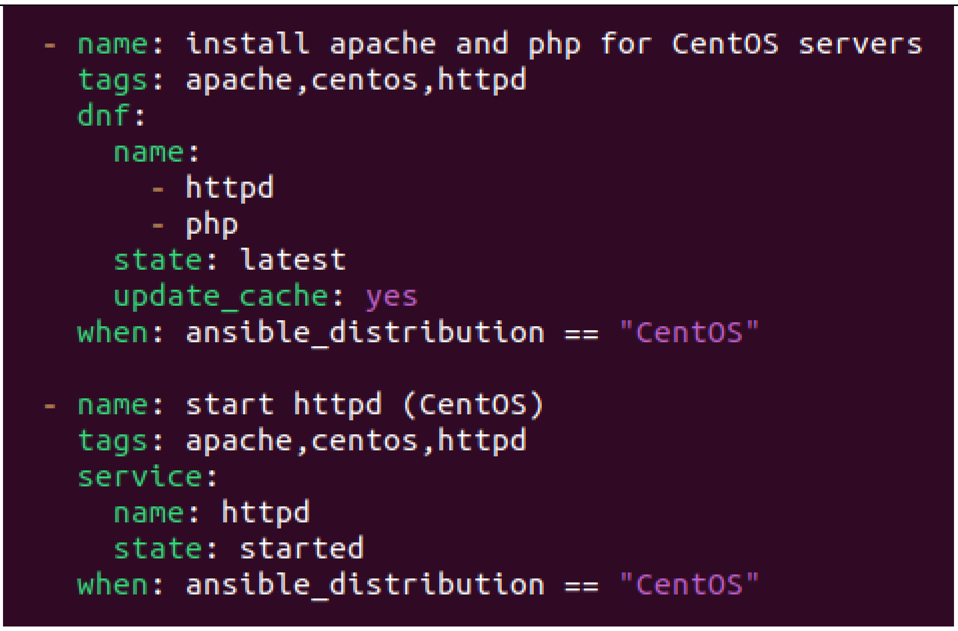


```
TASK [install apache and php for CentOS servers] *********************************************************
skipping: [server1]
ok: [servercent]

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

TASK [Gathering Facts] ***********************************************************************************
ok: [server3]

TASK [install mariadb package (CentOS)] ******************************************************************
skipping: [server3]

TASK [install mariadb package (Ubuntu)] ******************************************************************
ok: [server3]

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

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

PLAY RECAP ***********************************************************************************************
server1                    : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
server2                    : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
server3                    : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
servercent                 : ok=5    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

This only runs tasks with the tag of apache and db.

Task 3: Managing Services

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

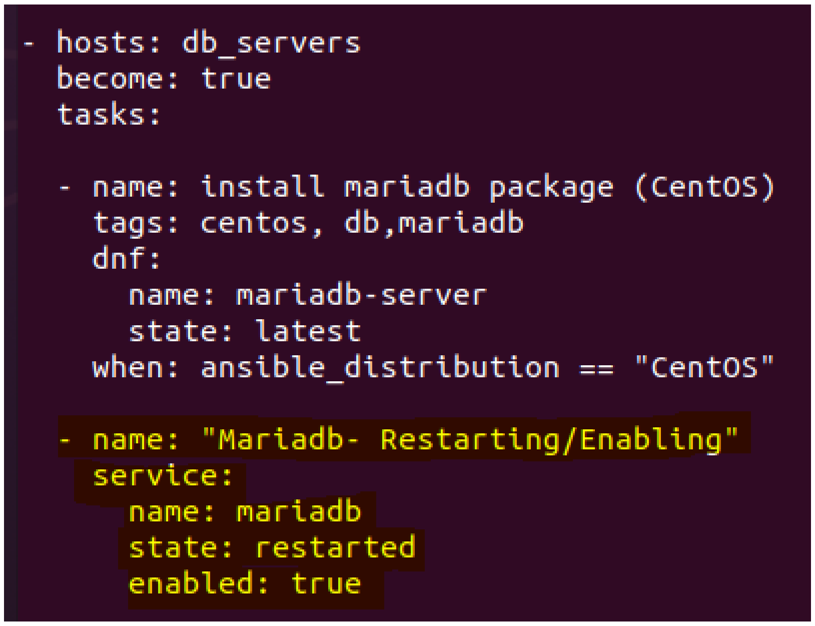


```
- name: install apache and php for CentOS servers
  tags: apache,centos,httpd
  dnf:
    name:
      - httpd
      - php
    state: latest
    update_cache: yes
  when: ansible_distribution == "CentOS"

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

Figure 3.1.1

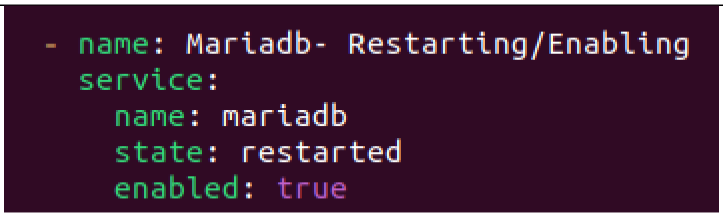
Make sure to save the file and exit.

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
```

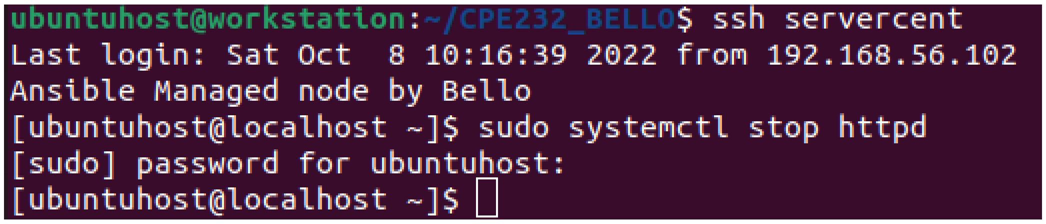
Figure 3.1.2



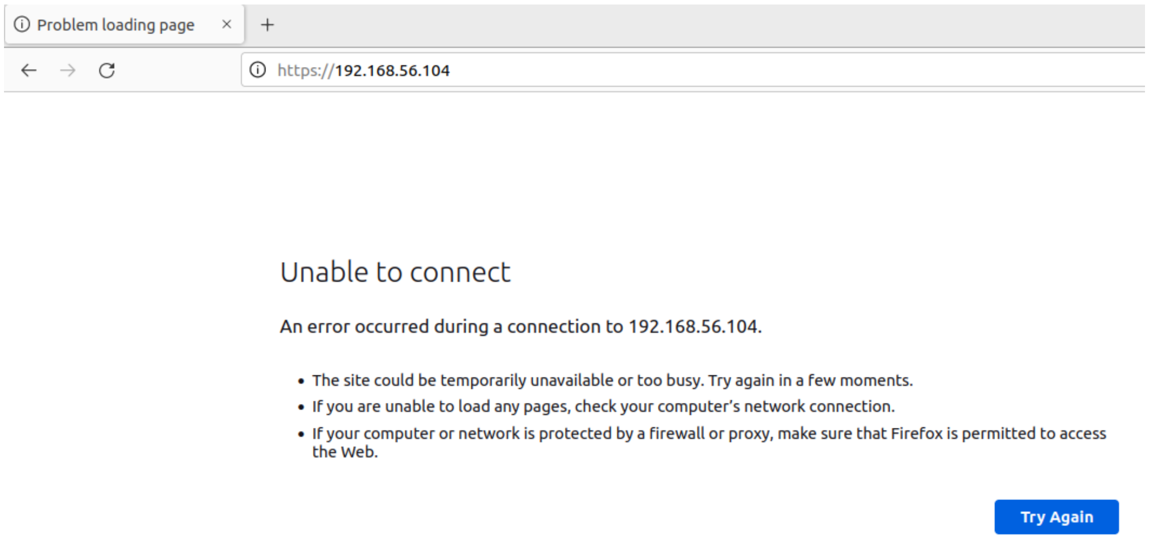
```
- name: Mariadb- Restarting/Enabling
  service:
    name: mariadb
    state: restarted
    enabled: true
```

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.



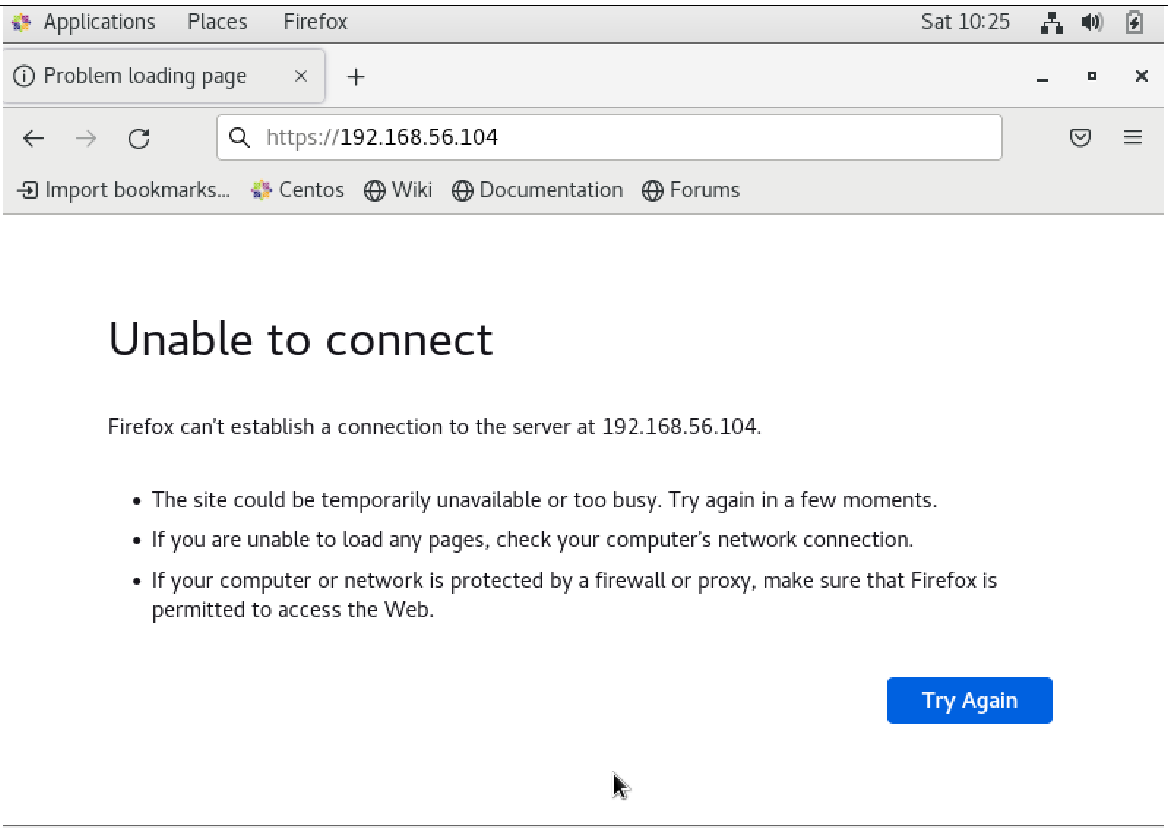
```
ubuntuhost@workstation:~/CPE232_BELLO$ ssh servercent
Last login: Sat Oct  8 10:16:39 2022 from 192.168.56.102
Ansible Managed node by Bello
[ubuntuhost@localhost ~]$ sudo systemctl stop httpd
[sudo] password for ubuntuhost:
[ubuntuhost@localhost ~]$ 
```



Unable to connect

An error occurred during a connection to 192.168.56.104.

- 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.



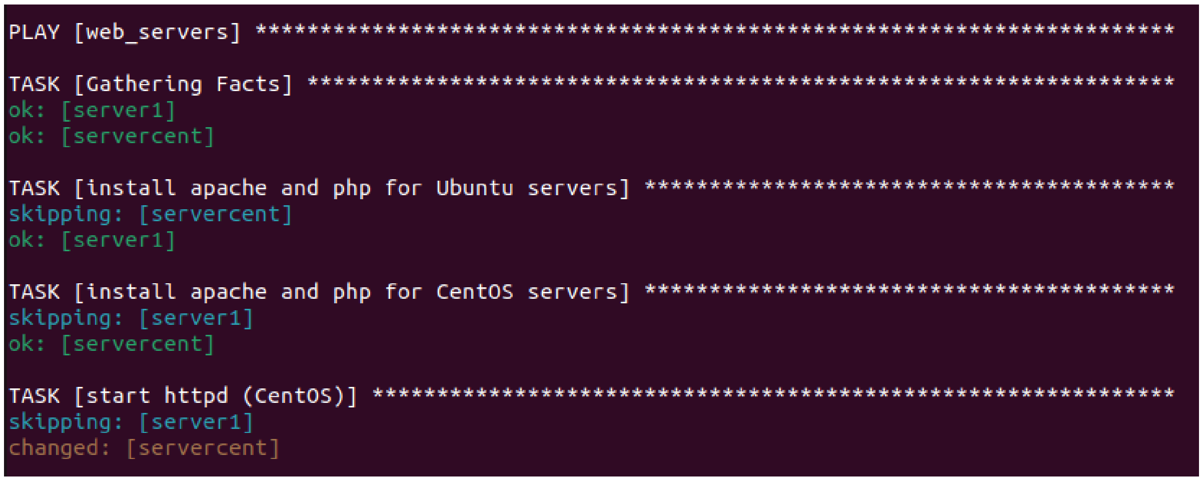
Unable to connect

Firefox can't establish a connection to the server at 192.168.56.104.

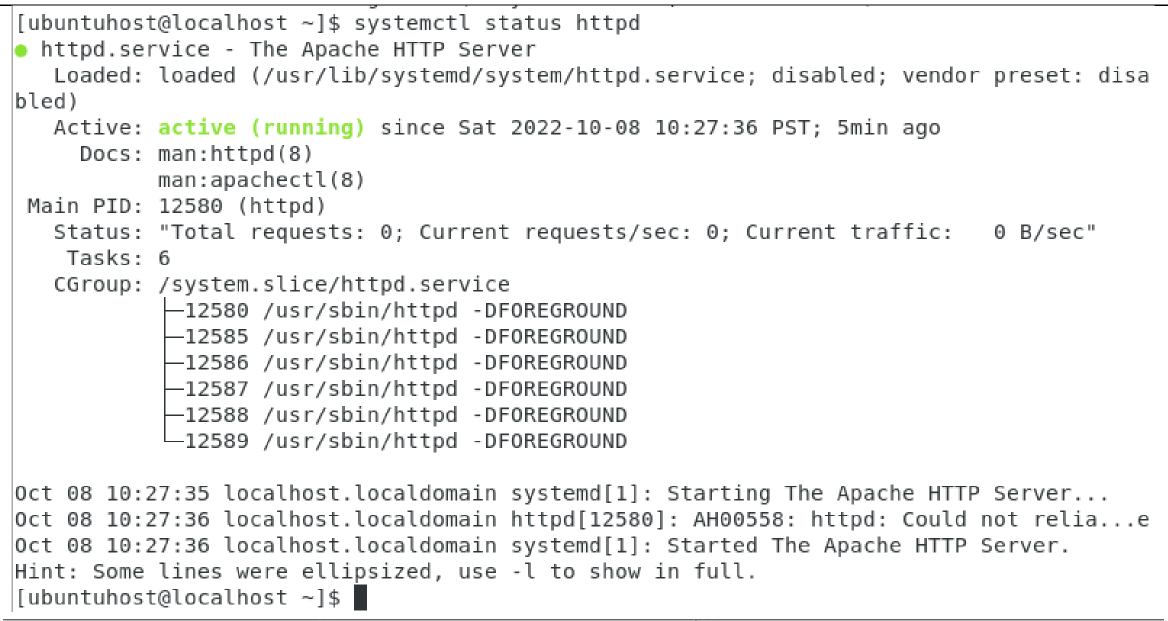
- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

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.

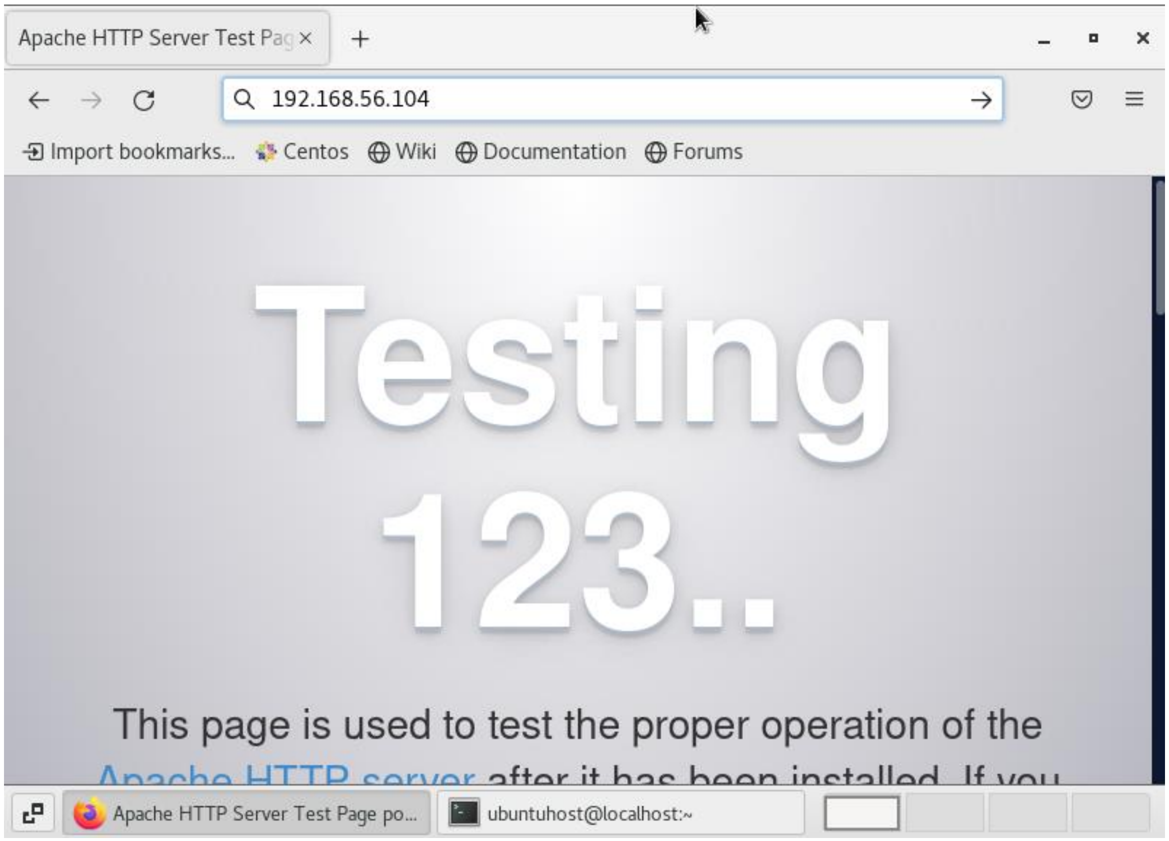
Here we can see that httpd or apache hase been started and up running we can also see that we can access it in the browser.



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

TASK [Gathering Facts] ********************************************************************
ok: [server1]
ok: [servercent]

TASK [install apache and php for Ubuntu servers] ******************************************
skipping: [servercent]
ok: [server1]

TASK [install apache and php for CentOS servers] ******************************************
skipping: [server1]
ok: [servercent]

TASK [start httpd (CentOS)] ***************************************************************
skipping: [server1]
changed: [servercent]
```



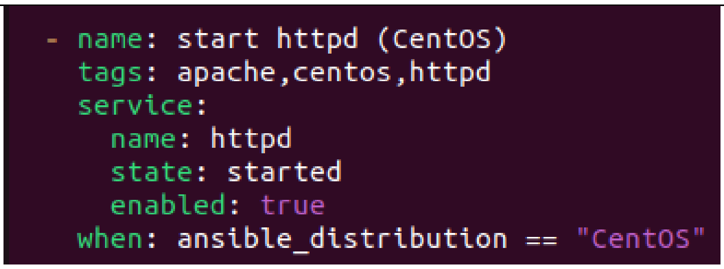
```
[ubuntuhost@localhost ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disa
bled)
   Active: active (running) since Sat 2022-10-08 10:27:36 PST; 5min ago
     Docs: man:httpd(8)
           man:apachectl(8)
 Main PID: 12580 (httpd)
   Status: "Total requests: 0; Current requests/sec: 0; Current traffic:   0 B/sec"
    Tasks: 6
   CGroup: /system.slice/httpd.service
           ├─12580 /usr/sbin/httpd -DFOREGROUND
           ├─12585 /usr/sbin/httpd -DFOREGROUND
           ├─12586 /usr/sbin/httpd -DFOREGROUND
           ├─12587 /usr/sbin/httpd -DFOREGROUND
           ├─12588 /usr/sbin/httpd -DFOREGROUND
           └─12589 /usr/sbin/httpd -DFOREGROUND

Oct 08 10:27:35 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
Oct 08 10:27:36 localhost.localdomain httpd[12580]: AH00558: httpd: Could not relia...e
Oct 08 10:27:36 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
[ubuntuhost@localhost ~]$ 
```



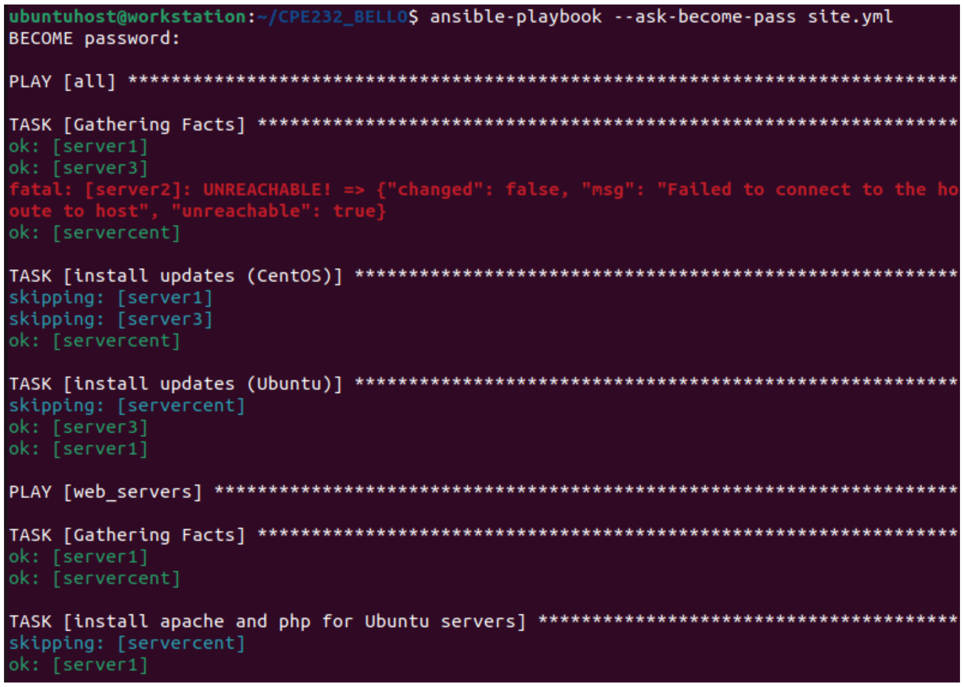
Testing 123..

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you

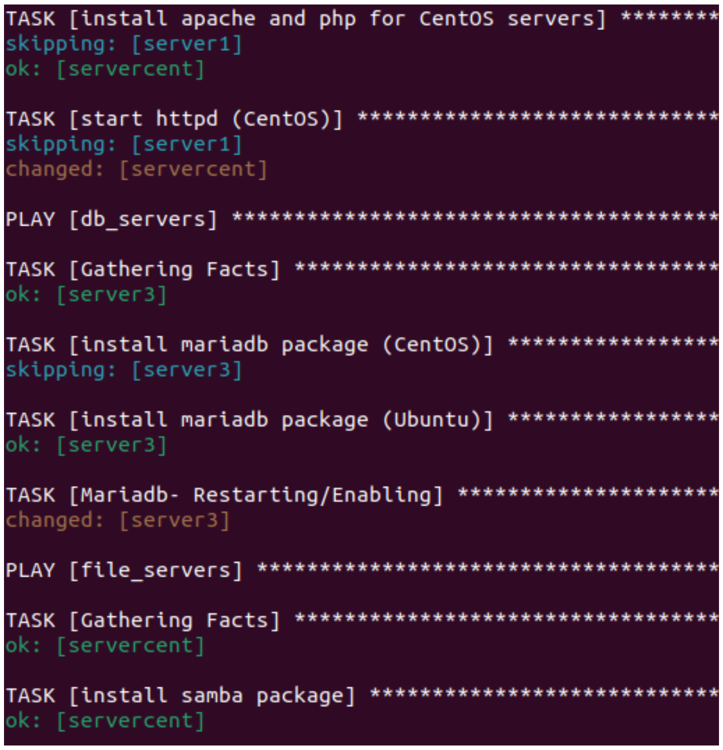
To automatically enable the service every time we run the playbook, use the command *enabled: true* similar to Figure 7.1.2 and save the playbook.



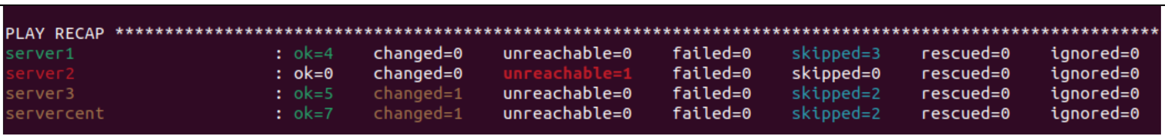
```
  - name: start httpd (CentOS)
    tags: apache,centos,httpd
    service:
      name: httpd
      state: started
      enabled: true
    when: ansible_distribution == "CentOS"
```



```
ubuntuhost@workstation:~/CPE232_BELLO$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *************************************************************
ok: [server1]
ok: [server3]
fatal: [server2]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the ho
oute to host", "unreachable": true}
ok: [servercent]

TASK [install updates (CentOS)] ****************************************************
skipping: [server1]
skipping: [server3]
ok: [servercent]

TASK [install updates (Ubuntu)] ****************************************************
skipping: [servercent]
ok: [server3]
ok: [server1]

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

TASK [Gathering Facts] *************************************************************
ok: [server1]
ok: [servercent]

TASK [install apache and php for Ubuntu servers] ***********************************
skipping: [servercent]
ok: [server1]
```



```
TASK [install apache and php for CentOS servers] ********
skipping: [server1]
ok: [servercent]

TASK [start httpd (CentOS)] *****************************
skipping: [server1]
changed: [servercent]

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

TASK [Gathering Facts] **********************************
ok: [server3]

TASK [install mariadb package (CentOS)] *****************
skipping: [server3]

TASK [install mariadb package (Ubuntu)] *****************
ok: [server3]

TASK [Mariadb- Restarting/Enabling] *********************
changed: [server3]

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

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

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



```
PLAY RECAP *********************************************************************************************************
server1                    : ok=4    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
server2                    : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
server3                    : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
servercent                 : ok=7    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

Reflections:

Answer the following:

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

They can specify which task is for them and isolate them from other groups, so tasks or commands won't be running, and we can also specify which remote server to run the task on, so we're not going to run a whole task code. This simplifies the playbook and how it will be used in the larger network.

2. What is the importance of tags in playbooks?

Tags are used to specify tasks that will only run for a specific command. For example, the apache tag will only run tasks with apache and won't do anything more. They enable us to target specific tasks and tell what to run and what not to run.

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

Larger networks and many devices will be a huge pain if we're doing it one by one, so we use playbooks to run it in one go. We can also debug and look for logs with this command, so we know who needs updates, installation, and fixes.