Name: BONIFACIO, REDJ GUILLIAN F.	**Date Performed:** September 12, 2025
Course/Section: CPE31S4	**Date Submitted:** September 12, 2025
Instructor: Engr. VALENZUELA, ROBIN	**Semester and SY:** 2nd Semester SY 2025 - 2026

Activity 6: Targeting Specific Nodes and Managing Services

1. **Objectives:**

1.1 Individualize hosts

1.2 Apply tags in selecting plays to run

1.3 Managing Services from remote servers using playbooks

2. **Discussion**:

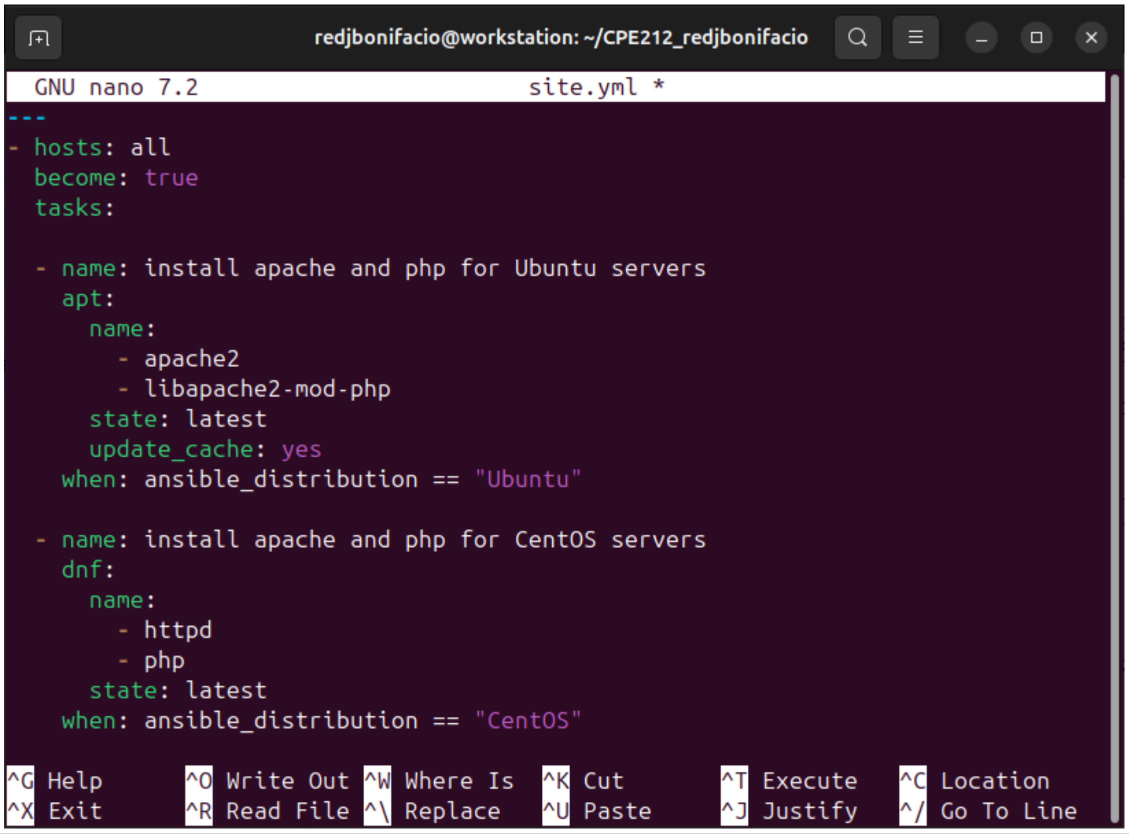
In this activity, we try to individualize hosts. For example, we don't want apache on all our servers, or maybe only one of our servers is a web server, or maybe we have different servers like database or file servers running different things on different categories of servers and that is what we are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

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

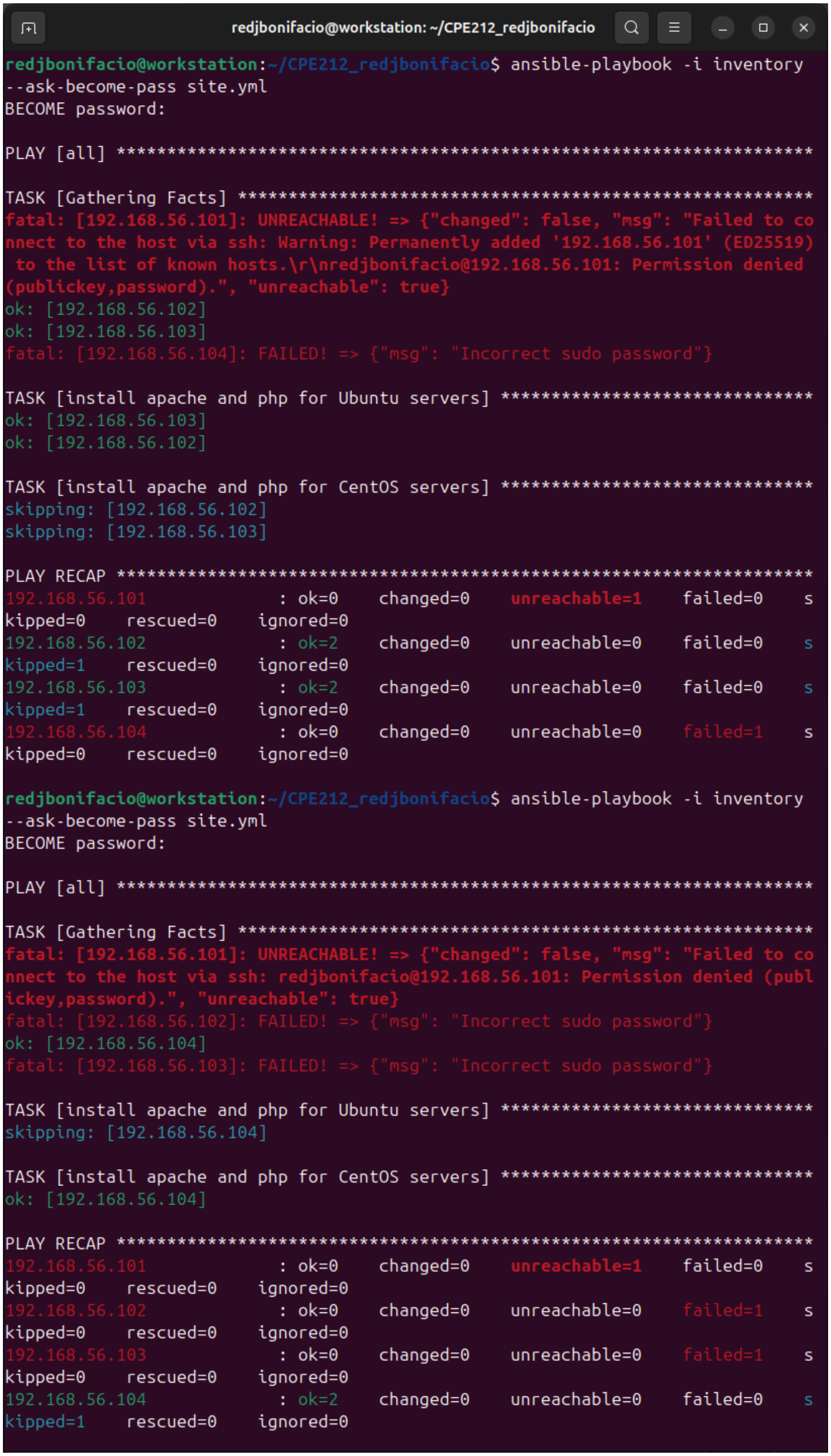
Requirement:
In this activity, you will need to create another Ubuntu VM and name it Server 3. Likewise, you need to activate the second adapter to a host-only adapter after the installations. Take note of the IP address of the Server 3. Make sure to use the command *ssh-copy-id* to copy the public key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

Task 1: Targeting Specific Nodes

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

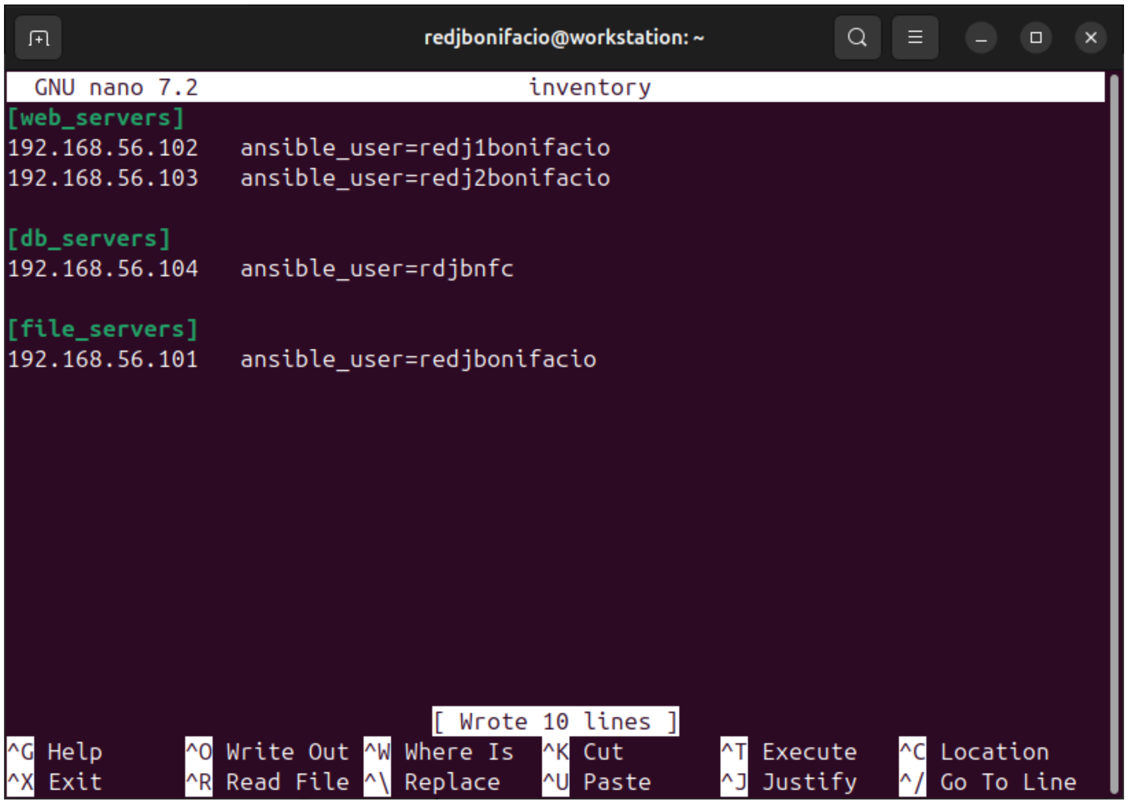
```
  GNU nano 7.2                         site.yml *
---
- hosts: all
  become: true
  tasks:

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

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

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```



redjbonifacio@workstation: ~/CPE212_redjbonifacio

```
redjbonifacio@workstation:~/CPE212_redjbonifacio$ ansible-playbook -i inventory
--ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] ***********************************************************
fatal: [192.168.56.101]: UNREACHABLE! => {"changed": false, "msg": "Failed to co
nnect to the host via ssh: Warning: Permanently added '192.168.56.101' (ED25519)
 to the list of known hosts.\r\nredjbonifacio@192.168.56.101: Permission denied
(publickey,password).", "unreachable": true}
ok: [192.168.56.102]
ok: [192.168.56.103]
fatal: [192.168.56.104]: FAILED! => {"msg": "Incorrect sudo password"}

TASK [install apache and php for Ubuntu servers] *********************************
ok: [192.168.56.103]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] *********************************
skipping: [192.168.56.102]
skipping: [192.168.56.103]

PLAY RECAP ***********************************************************************
192.168.56.101             : ok=0    changed=0    unreachable=1    failed=0    s
kipped=0    rescued=0    ignored=0
192.168.56.102             : ok=2    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.103             : ok=2    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.104             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0

redjbonifacio@workstation:~/CPE212_redjbonifacio$ ansible-playbook -i inventory
--ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] ***********************************************************
fatal: [192.168.56.101]: UNREACHABLE! => {"changed": false, "msg": "Failed to co
nnect to the host via ssh: redjbonifacio@192.168.56.101: Permission denied (publ
ickey,password).", "unreachable": true}
fatal: [192.168.56.102]: FAILED! => {"msg": "Incorrect sudo password"}
ok: [192.168.56.104]
fatal: [192.168.56.103]: FAILED! => {"msg": "Incorrect sudo password"}

TASK [install apache and php for Ubuntu servers] *********************************
skipping: [192.168.56.104]

TASK [install apache and php for CentOS servers] *********************************
ok: [192.168.56.104]

PLAY RECAP ***********************************************************************
192.168.56.101             : ok=0    changed=0    unreachable=1    failed=0    s
kipped=0    rescued=0    ignored=0
192.168.56.102             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
192.168.56.103             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
192.168.56.104             : ok=2    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
```

2. Edit the inventory file. Remove the variables we put in our last activity and group according to the image shown below:



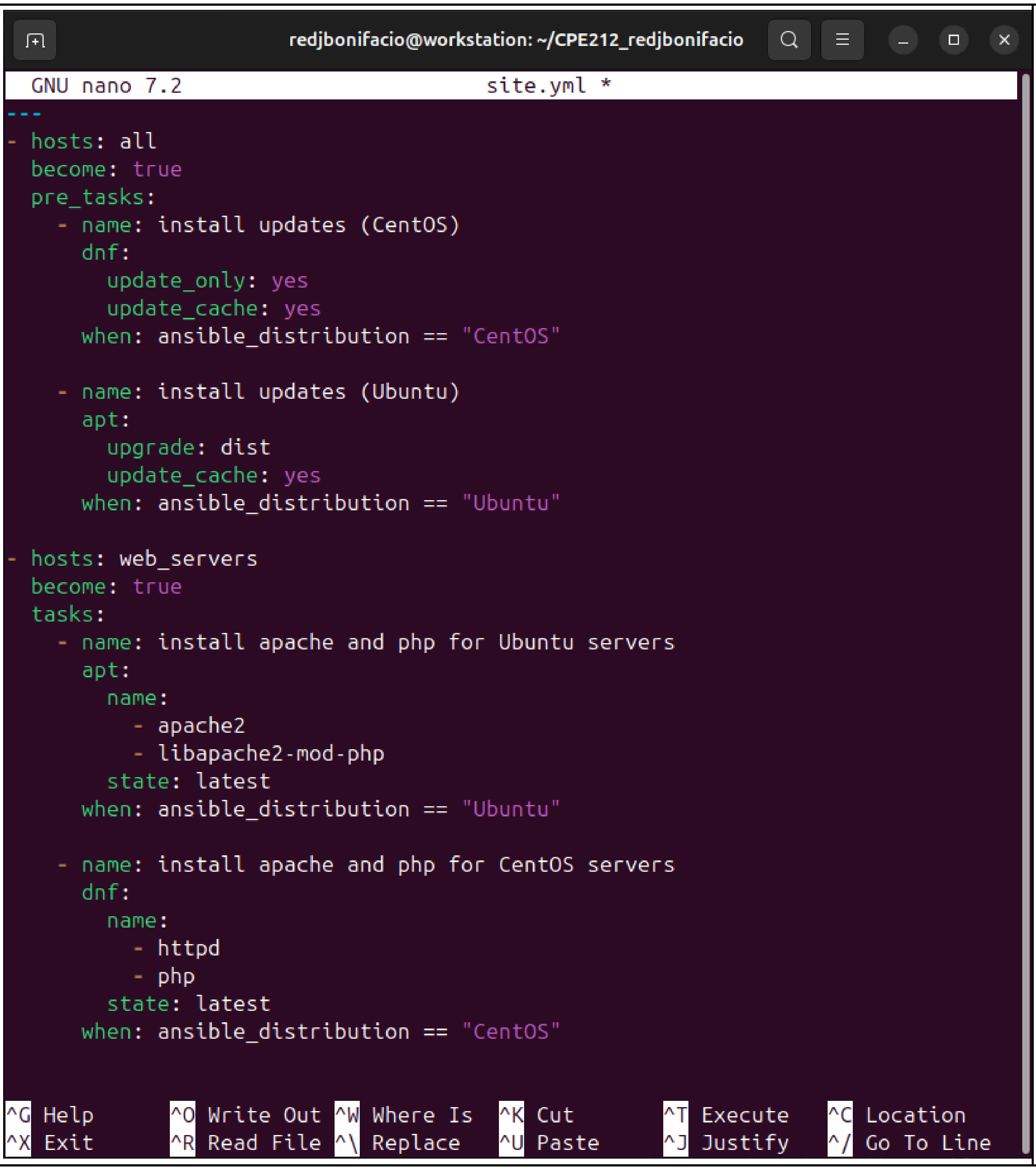
```
  GNU nano 7.2                           inventory
[web_servers]
192.168.56.102   ansible_user=redj1bonifacio
192.168.56.103   ansible_user=redj2bonifacio

[db_servers]
192.168.56.104   ansible_user=rdjbnfc

[file_servers]
192.168.56.101   ansible_user=redjbonifacio
```

```
                                [ Wrote 10 lines ]
^G Help        ^O Write Out ^W Where Is  ^K Cut        ^T Execute    ^C Location
^X Exit        ^R Read File ^\ Replace   ^U Paste      ^J Justify    ^/ Go To Line
```

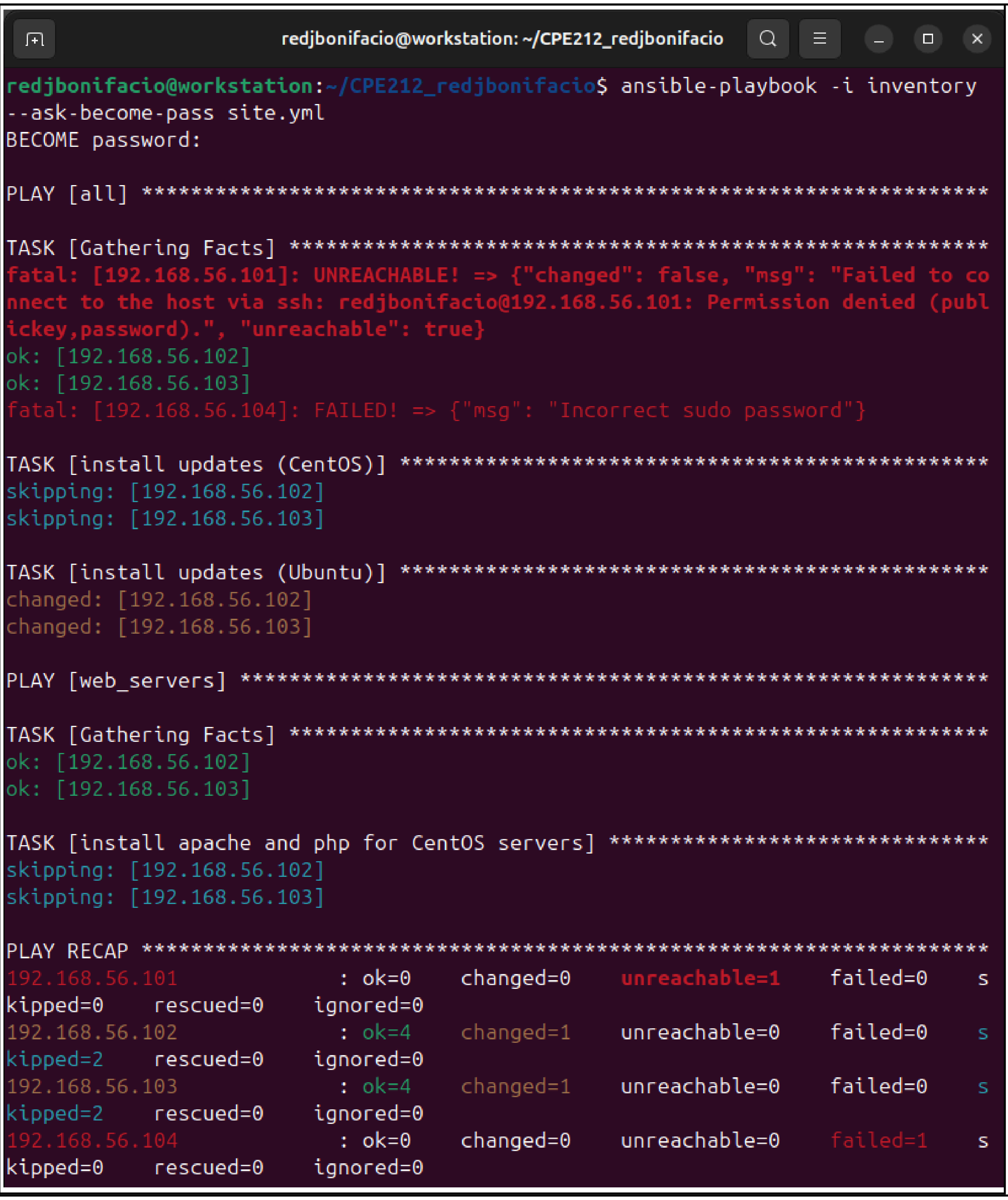
Make sure to save the file and exit.

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.

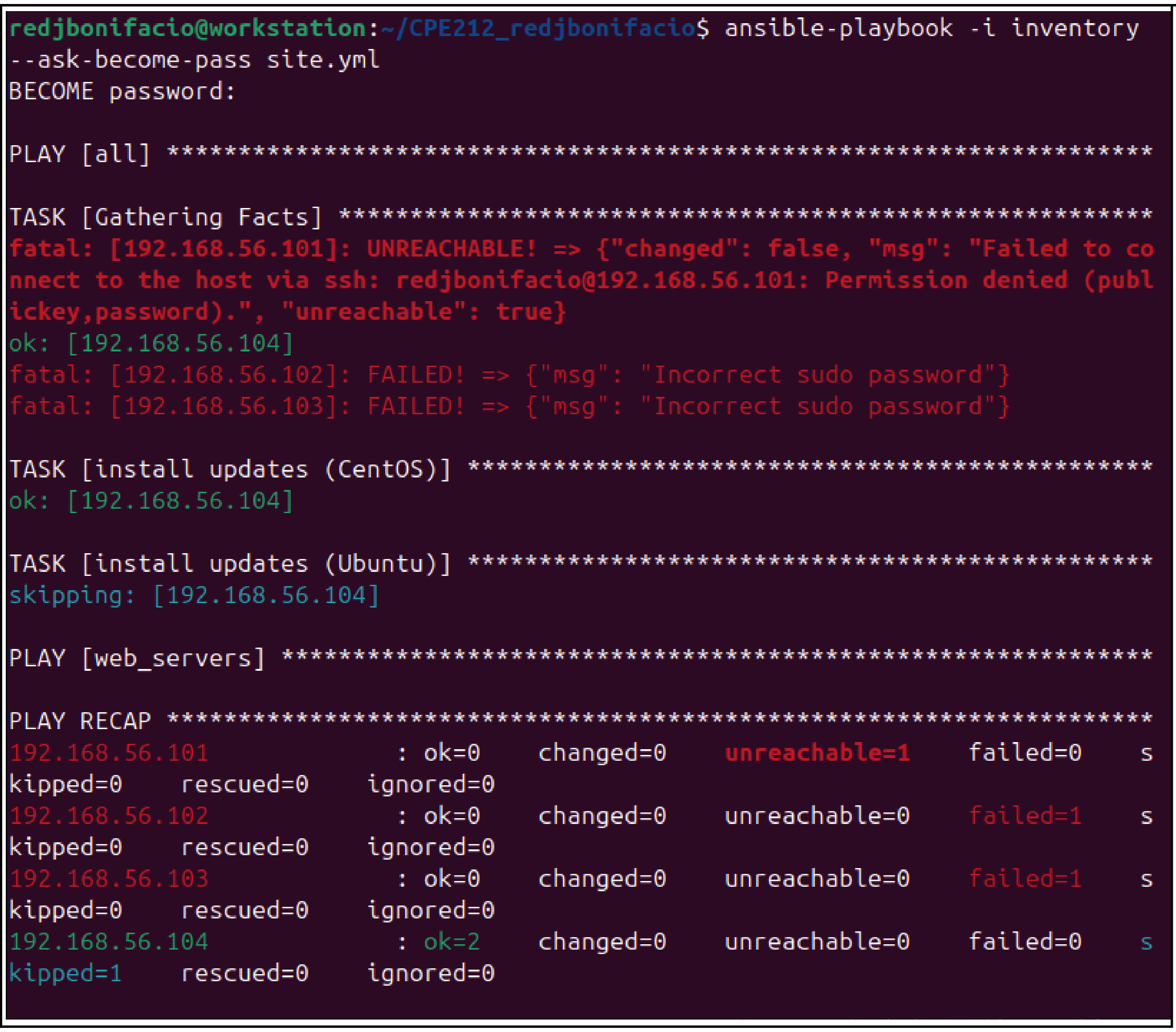
3. Edit the *site.yml* by following the image below:



```
  GNU nano 7.2                          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
      when: ansible_distribution == "Ubuntu"

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


^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

```
redjbonifacio@workstation: ~/CPE212_redjbonifacio
redjbonifacio@workstation:~/CPE212_redjbonifacio$ ansible-playbook -i inventory
--ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] ***********************************************************
fatal: [192.168.56.101]: UNREACHABLE! => {"changed": false, "msg": "Failed to co
nnect to the host via ssh: redjbonifacio@192.168.56.101: Permission denied (publ
ickey,password).", "unreachable": true}
ok: [192.168.56.102]
ok: [192.168.56.103]
fatal: [192.168.56.104]: FAILED! => {"msg": "Incorrect sudo password"}

TASK [install updates (CentOS)] **************************************************
skipping: [192.168.56.102]
skipping: [192.168.56.103]

TASK [install updates (Ubuntu)] **************************************************
changed: [192.168.56.102]
changed: [192.168.56.103]

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

TASK [Gathering Facts] ***********************************************************
ok: [192.168.56.102]
ok: [192.168.56.103]

TASK [install apache and php for CentOS servers] *********************************
skipping: [192.168.56.102]
skipping: [192.168.56.103]

PLAY RECAP ***********************************************************************
192.168.56.101             : ok=0    changed=0    unreachable=1    failed=0    s
kipped=0    rescued=0    ignored=0
192.168.56.102             : ok=4    changed=1    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.103             : ok=4    changed=1    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.104             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
```



```
redjbonifacio@workstation:~/CPE212_redjbonifacio$ ansible-playbook -i inventory
--ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
fatal: [192.168.56.101]: UNREACHABLE! => {"changed": false, "msg": "Failed to co
nnect to the host via ssh: redjbonifacio@192.168.56.101: Permission denied (publ
ickey,password).", "unreachable": true}
ok: [192.168.56.104]
fatal: [192.168.56.102]: FAILED! => {"msg": "Incorrect sudo password"}
fatal: [192.168.56.103]: FAILED! => {"msg": "Incorrect sudo password"}

TASK [install updates (CentOS)] ************************************************
ok: [192.168.56.104]

TASK [install updates (Ubuntu)] ************************************************
skipping: [192.168.56.104]

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

PLAY RECAP *********************************************************************
192.168.56.101             : ok=0    changed=0    unreachable=1    failed=0    s
kipped=0    rescued=0    ignored=0
192.168.56.102             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
192.168.56.103             : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
192.168.56.104             : ok=2    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
```

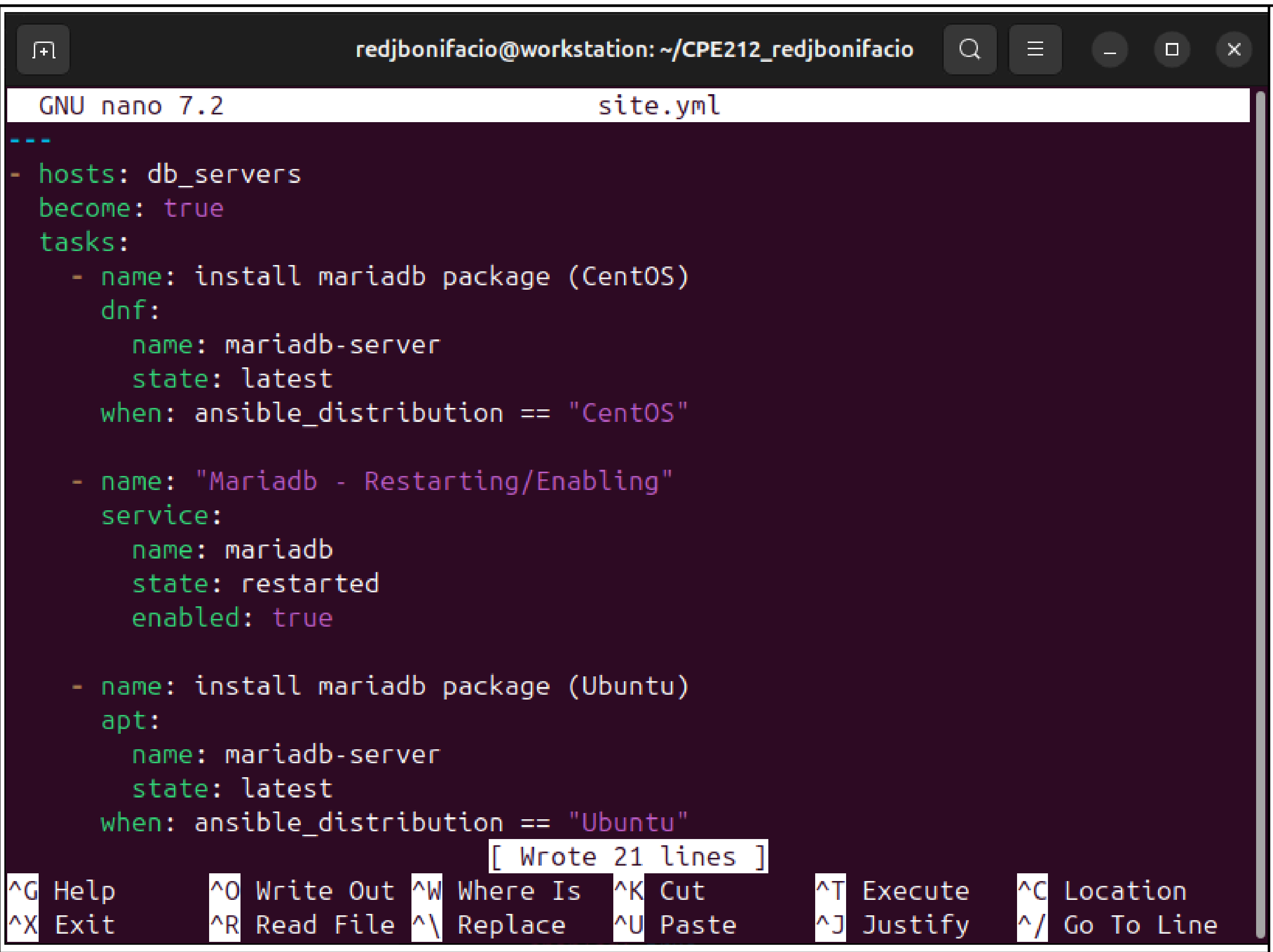
Make sure to save the file and exit.

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.

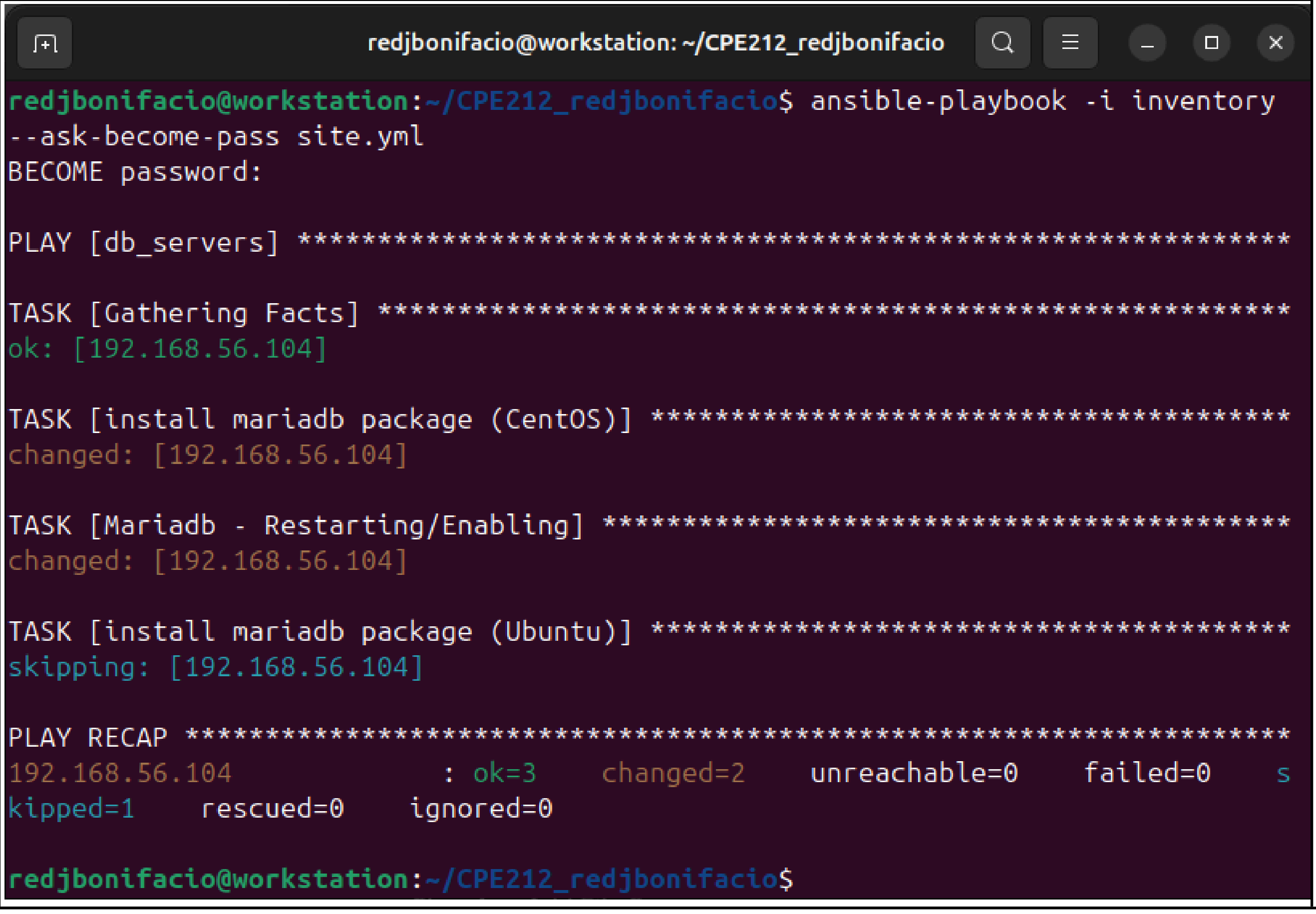
Run the *site.yml* file and describe the result.

- I had to run site.yml two times to make it successful because I have a different password for centOS. In the figure above, it is technically successful

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.



```
 GNU nano 7.2                         site.yml
---
- hosts: db_servers
  become: true
  tasks:
    - name: install mariadb package (CentOS)
      dnf:
        name: mariadb-server
        state: latest
      when: ansible_distribution == "CentOS"

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

    - name: install mariadb package (Ubuntu)
      apt:
        name: mariadb-server
        state: latest
      when: ansible_distribution == "Ubuntu"
                                   [ Wrote 21 lines ]
^G Help       ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit       ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```



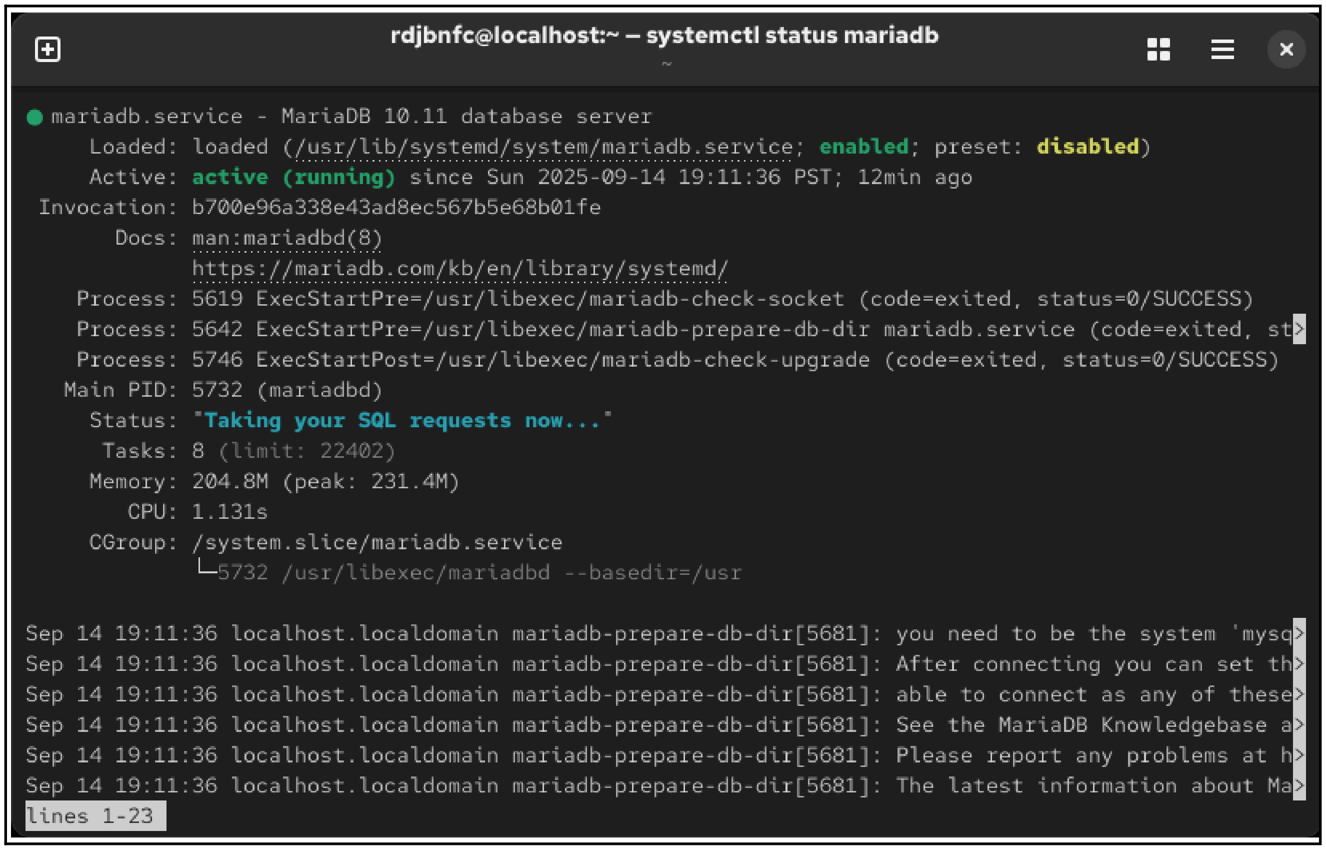
```
redjbonifacio@workstation:~/CPE212_redjbonifacio$ ansible-playbook -i inventory
--ask-become-pass site.yml
BECOME password:

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

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

TASK [install mariadb package (CentOS)] *****************************************
changed: [192.168.56.104]

TASK [Mariadb - Restarting/Enabling] ********************************************
changed: [192.168.56.104]

TASK [install mariadb package (Ubuntu)] *****************************************
skipping: [192.168.56.104]

PLAY RECAP **********************************************************************
192.168.56.104             : ok=3    changed=2    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0

redjbonifacio@workstation:~/CPE212_redjbonifacio$
```

Make sure to save the file and exit.

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

- The results showed that the status for installing mariadb package and restarting/enabling is ‘changed’, as well as installing mariadb package for ubuntu

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

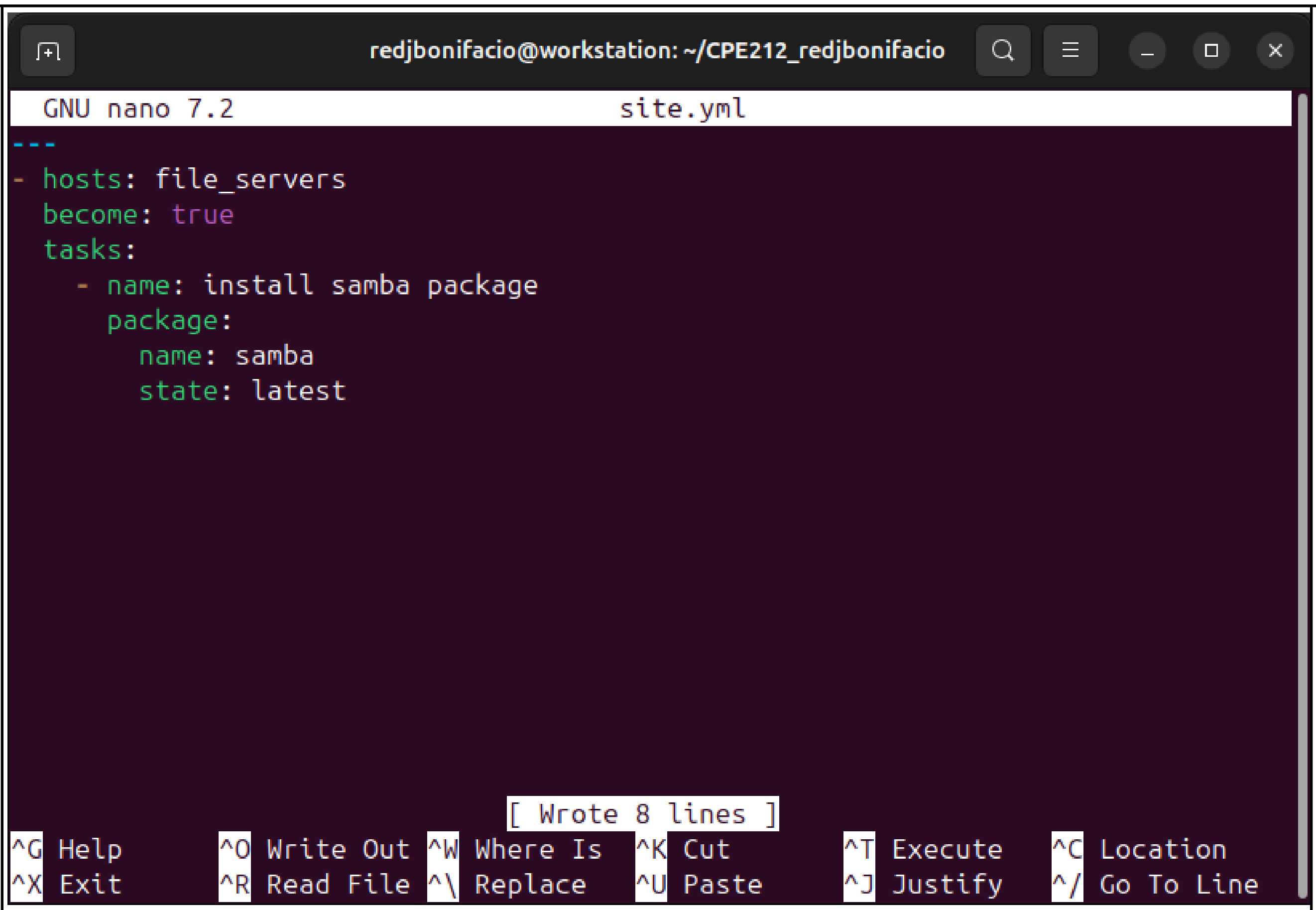


```
● mariadb.service - MariaDB 10.11 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: disabled)
     Active: active (running) since Sun 2025-09-14 19:11:36 PST; 12min ago
 Invocation: b700e96a338e43ad8ec567b5e68b01fe
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 5619 ExecStartPre=/usr/libexec/mariadb-check-socket (code=exited, status=0/SUCCESS)
    Process: 5642 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir mariadb.service (code=exited, st>
    Process: 5746 ExecStartPost=/usr/libexec/mariadb-check-upgrade (code=exited, status=0/SUCCESS)
   Main PID: 5732 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 8 (limit: 22402)
     Memory: 204.8M (peak: 231.4M)
        CPU: 1.131s
     CGroup: /system.slice/mariadb.service
             └─5732 /usr/libexec/mariadbd --basedir=/usr

Sep 14 19:11:36 localhost.localdomain mariadb-prepare-db-dir[5681]: you need to be the system 'mysq>
Sep 14 19:11:36 localhost.localdomain mariadb-prepare-db-dir[5681]: After connecting you can set th>
Sep 14 19:11:36 localhost.localdomain mariadb-prepare-db-dir[5681]: able to connect as any of these>
Sep 14 19:11:36 localhost.localdomain mariadb-prepare-db-dir[5681]: See the MariaDB Knowledgebase a>
Sep 14 19:11:36 localhost.localdomain mariadb-prepare-db-dir[5681]: Please report any problems at h>
Sep 14 19:11:36 localhost.localdomain mariadb-prepare-db-dir[5681]: The latest information about Ma>
lines 1-23
```

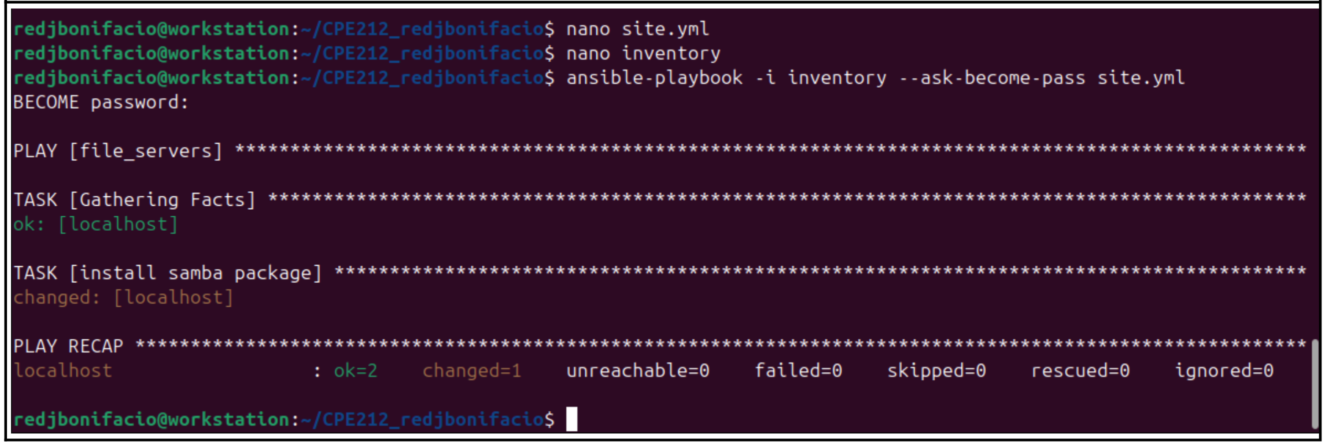
Describe the output.

- when ‘systemctl status mariadb’ ran, it showed that mariaDB is active/running

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.



```
redjbonifacio@workstation: ~/CPE212_redjbonifacio
  GNU nano 7.2                          site.yml
---
- hosts: file_servers
  become: true
  tasks:
    - name: install samba package
      package:
        name: samba
        state: latest

                              [ Wrote 8 lines ]
^G Help        ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit        ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```



```
redjbonifacio@workstation:~/CPE212_redjbonifacio$ nano site.yml
redjbonifacio@workstation:~/CPE212_redjbonifacio$ nano inventory
redjbonifacio@workstation:~/CPE212_redjbonifacio$ ansible-playbook -i inventory --ask-become-pass site.yml
BECOME password:

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

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

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

PLAY RECAP **********************************************************************************************
localhost                  : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

redjbonifacio@workstation:~/CPE212_redjbonifacio$ 
```

Make sure to save the file and exit.

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

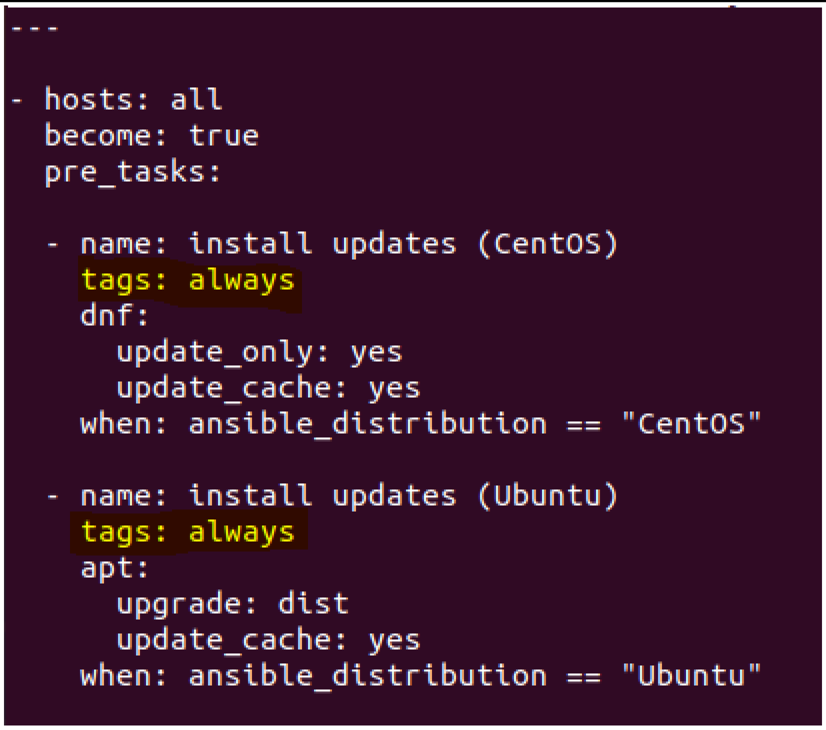
- It successfully installed the samba package on all servers

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.

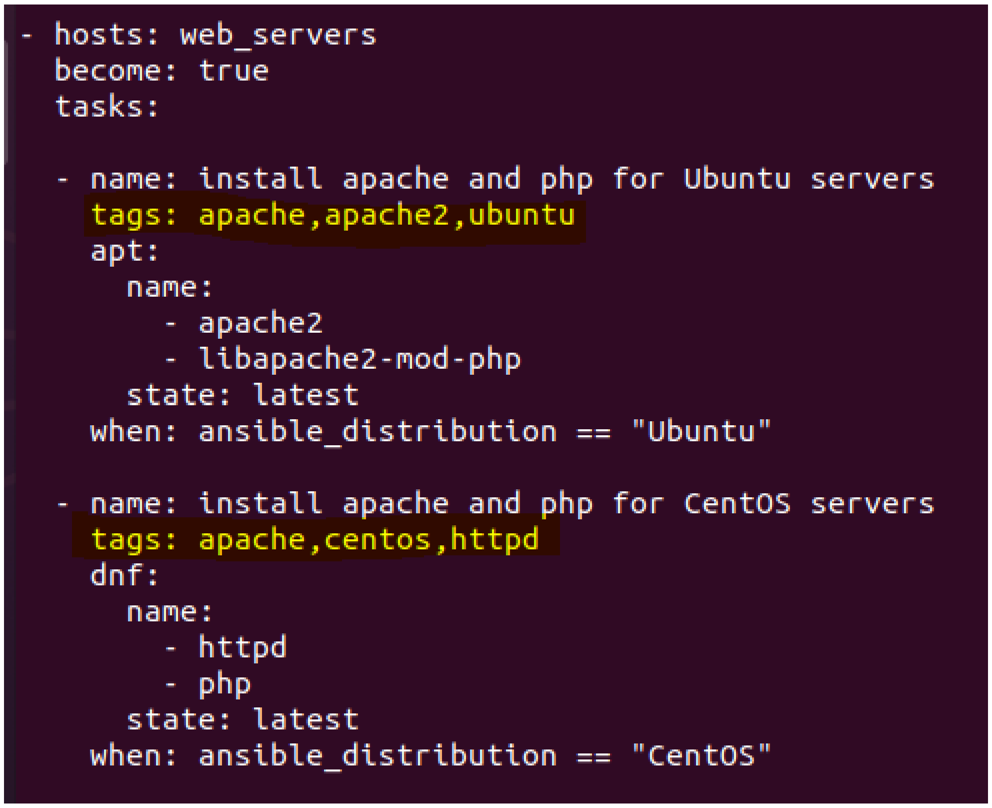
Task 2: Using Tags in running playbooks

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.

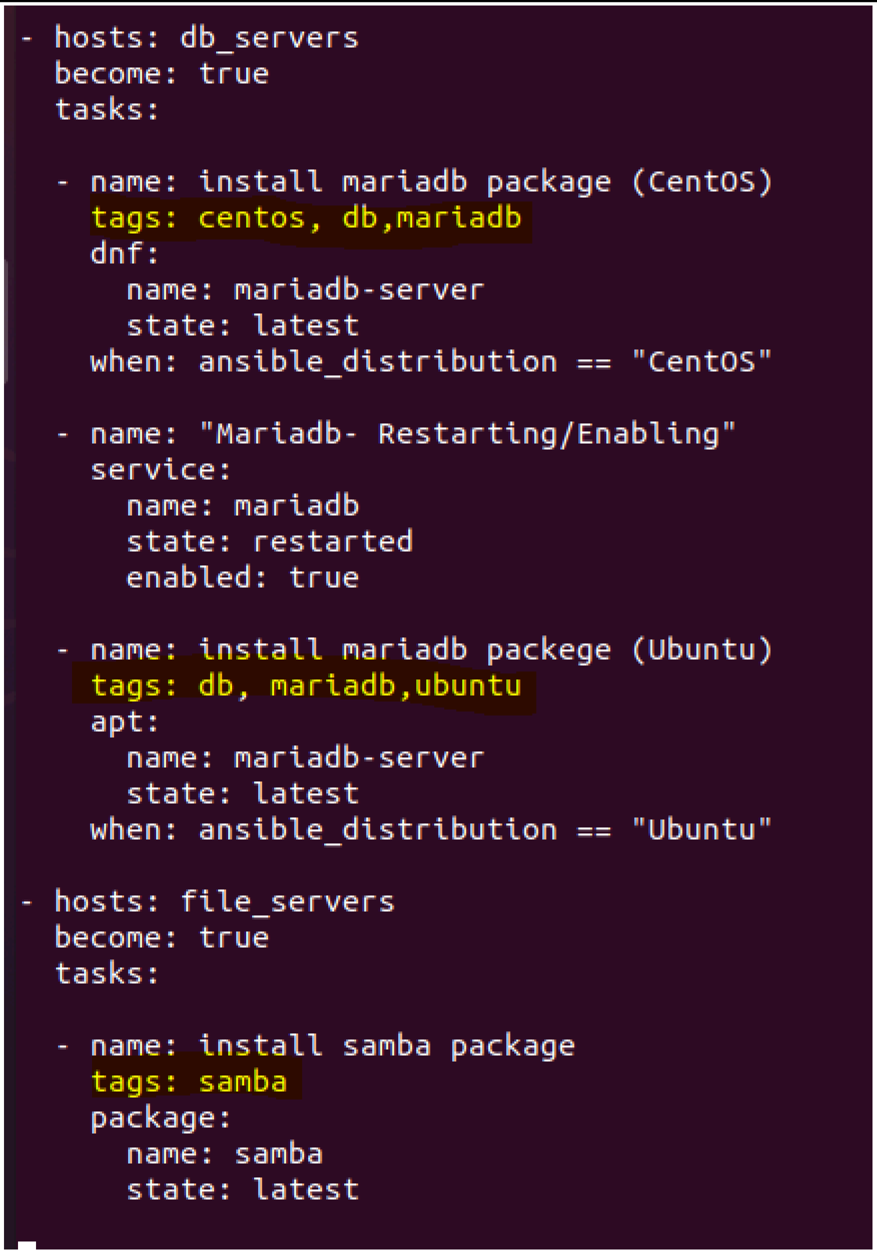
1. Edit the *site.yml* file. Add tags to the playbook. After the name, we can place the tags: *name_of_tag*. This is an arbitrary command, which means you can use any name for a tag.



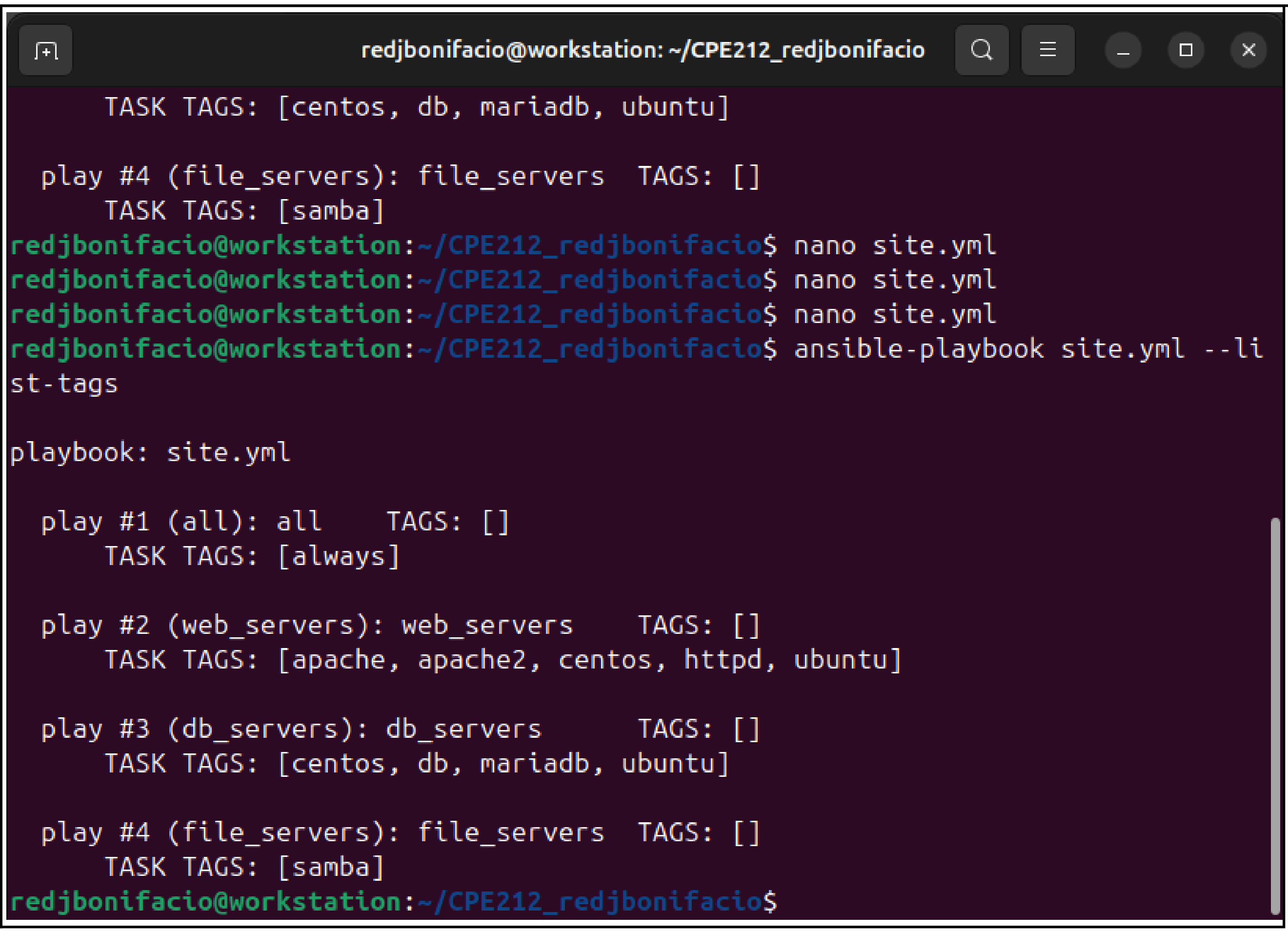
```
---

- 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
    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
```


```
      TASK TAGS: [centos, db, mariadb, ubuntu]

  play #4 (file_servers): file_servers  TAGS: []
      TASK TAGS: [samba]
redjbonifacio@workstation:~/CPE212_redjbonifacio$ nano site.yml
redjbonifacio@workstation:~/CPE212_redjbonifacio$ nano site.yml
redjbonifacio@workstation:~/CPE212_redjbonifacio$ nano site.yml
redjbonifacio@workstation:~/CPE212_redjbonifacio$ ansible-playbook site.yml --li
st-tags

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]
redjbonifacio@workstation:~/CPE212_redjbonifacio$
```

Make sure to save the file and exit.

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

- When 'site.yml' is ran, it showed all the tags for play #1, 2, 3, and 4. These tags specifically are 'always', 'apache', 'apache2', 'centos', 'httpd', 'ubuntu', 'db', 'mariadb', and 'samba'.

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

   2.1 ansible-playbook --list-tags site.yml
   - The playbook site.yml contains four plays targeting different server groups, each with specific task tags like apache, db, and samba.

   2.2 ansible-playbook --tags centos --ask-become-pass site.yml
   - The playbook ran successfully for the reachable hosts with correct sudo passwords, applying CentOS-related tasks where applicable and skipping others as needed.

   2.3 ansible-playbook --tags db --ask-become-pass site.yml
   - The playbook ran successfully on one reachable host with the correct sudo password, installing and configuring MariaDB as expected.

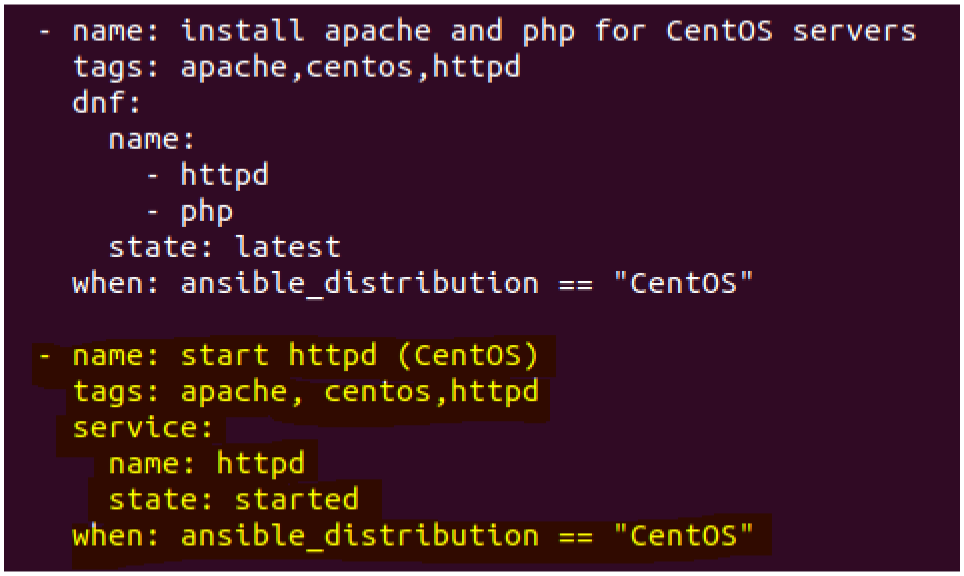
   2.4 ansible-playbook --tags apache --ask-become-pass site.yml
   - The playbook ran successfully on one host, applying Apache-related tasks for CentOS.

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

- The playbook successfully applied Apache and database-related tasks on one reachable host.

Task 3: Managing Services

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

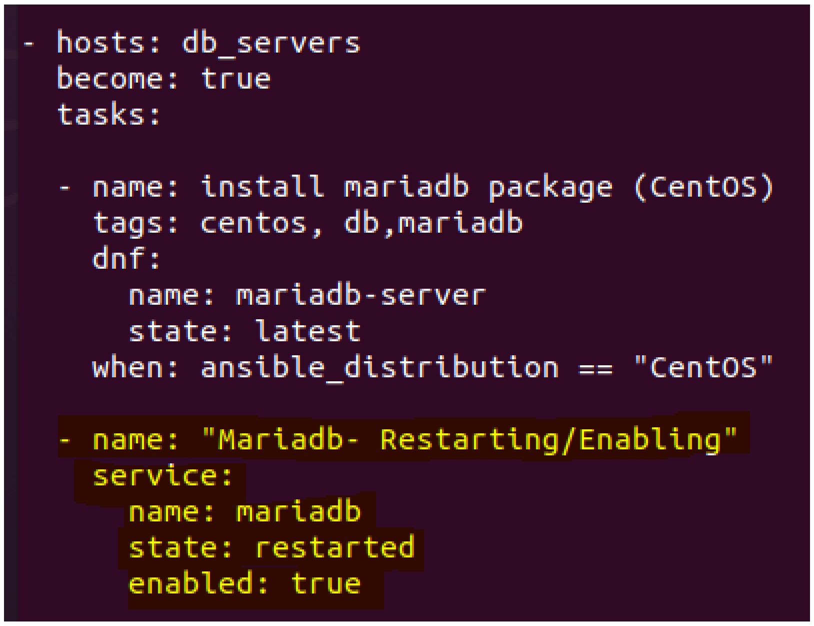


```
  - name: install apache and php for CentOS servers
    tags: apache,centos,httpd
    dnf:
      name:
        - httpd
        - php
      state: latest
    when: ansible_distribution == "CentOS"

  - 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.

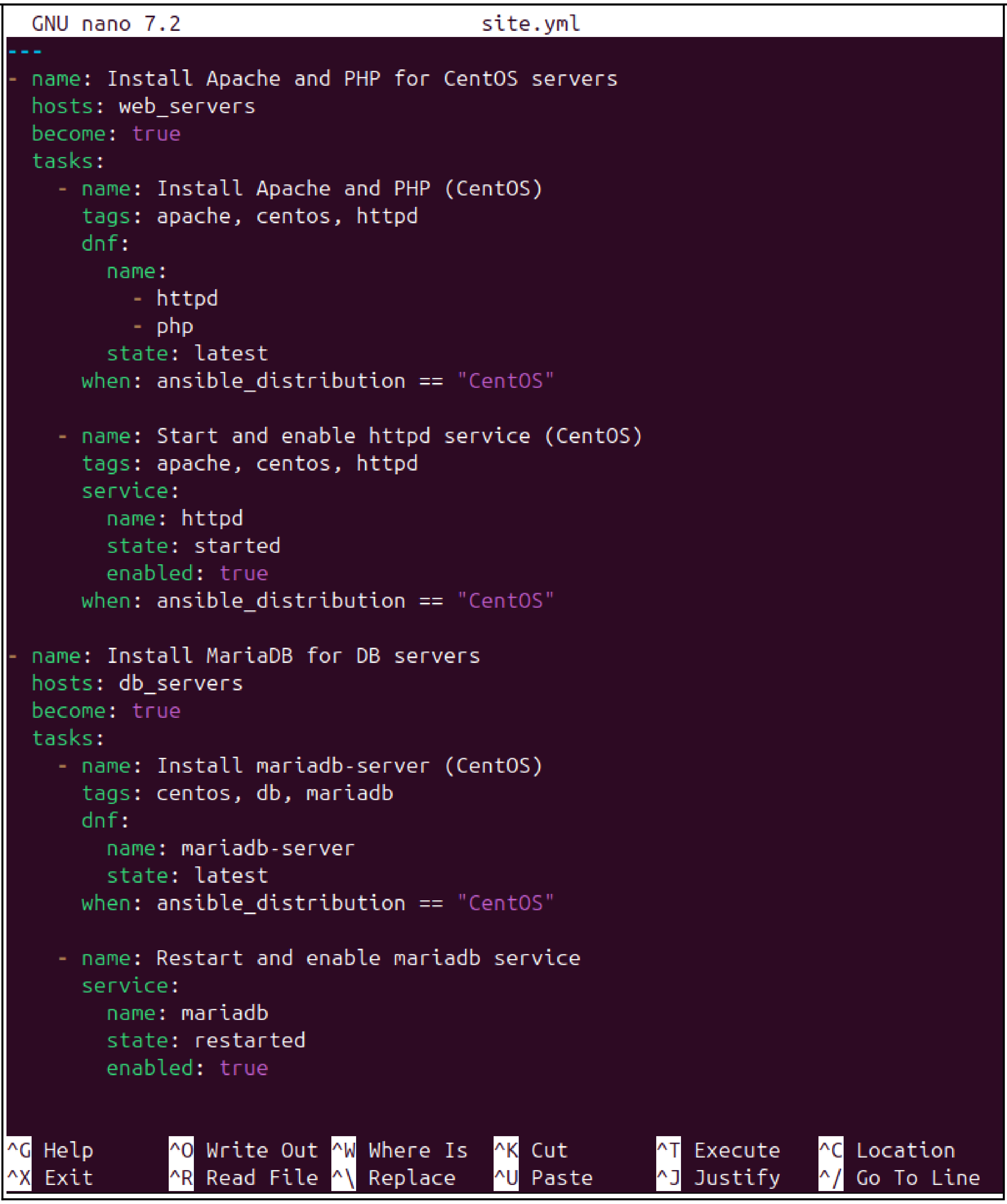
You would also notice from our previous activity that we already created a module that runs a service.



```
- 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
```

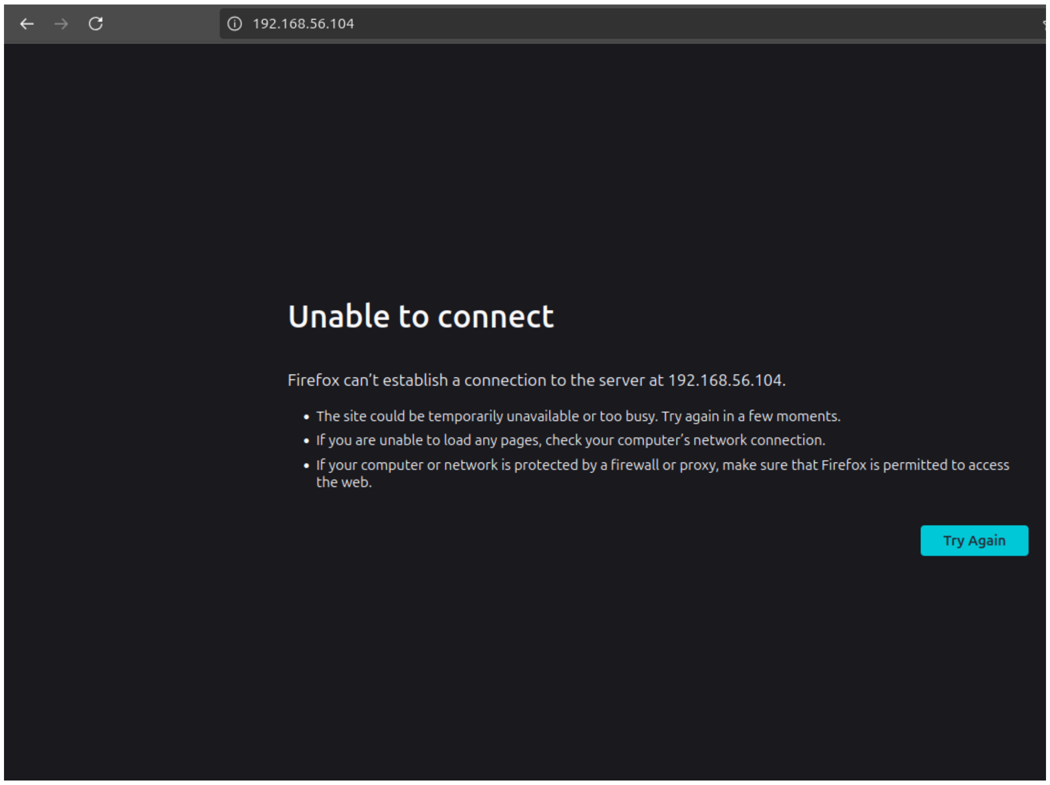
Figure 3.1.2

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



```
  GNU nano 7.2                          site.yml
---
- name: Install Apache and PHP for CentOS servers
  hosts: web_servers
  become: true
  tasks:
    - name: Install Apache and PHP (CentOS)
      tags: apache, centos, httpd
      dnf:
        name:
          - httpd
          - php
        state: latest
      when: ansible_distribution == "CentOS"

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

- name: Install MariaDB for DB servers
  hosts: db_servers
  become: true
  tasks:
    - name: Install mariadb-server (CentOS)
      tags: centos, db, mariadb
      dnf:
        name: mariadb-server
        state: latest
      when: ansible_distribution == "CentOS"

    - name: Restart and enable mariadb service
      service:
        name: mariadb
        state: restarted
        enabled: true


^G Help        ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit        ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

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.

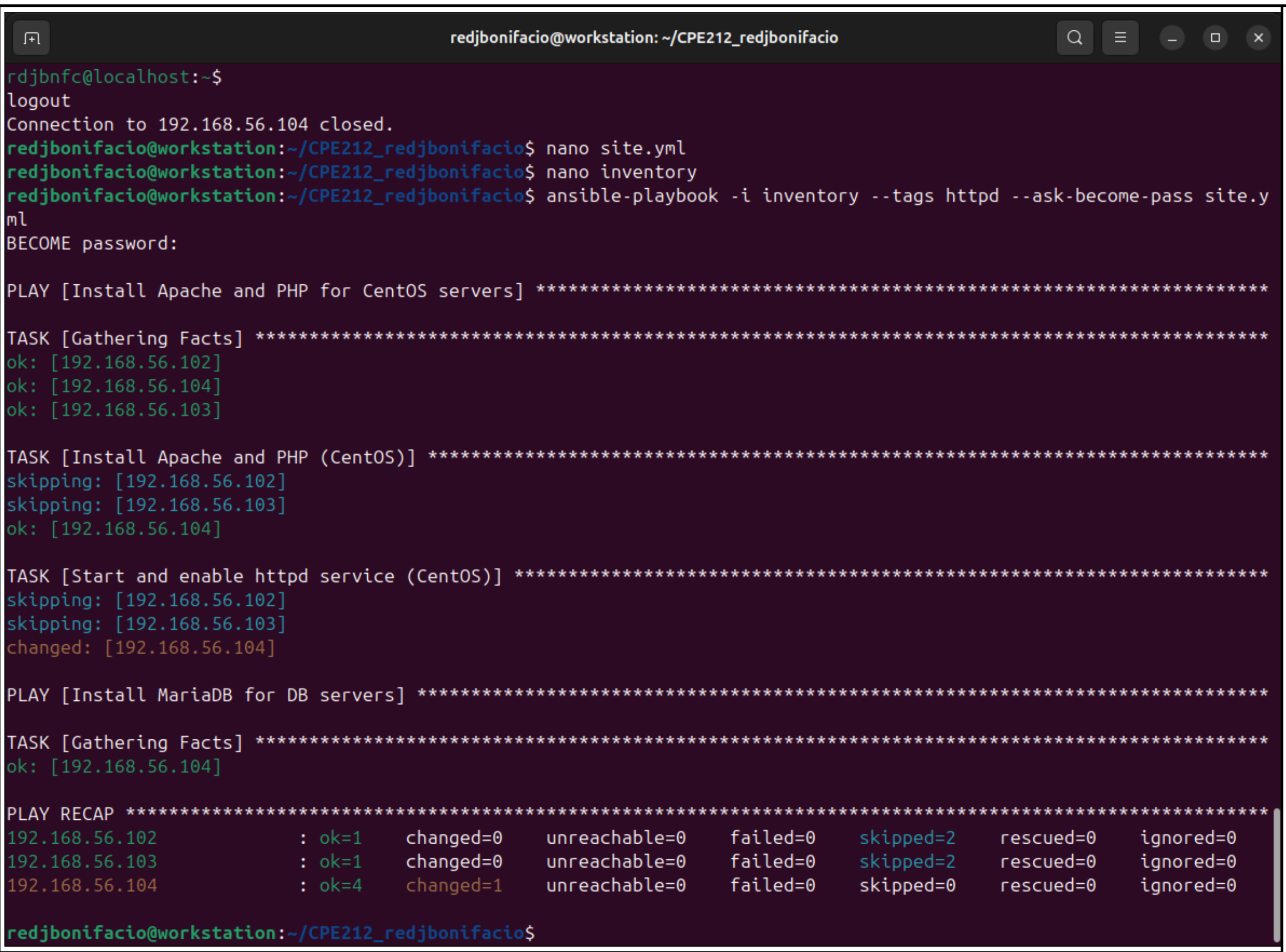

192.168.56.104

Unable to connect

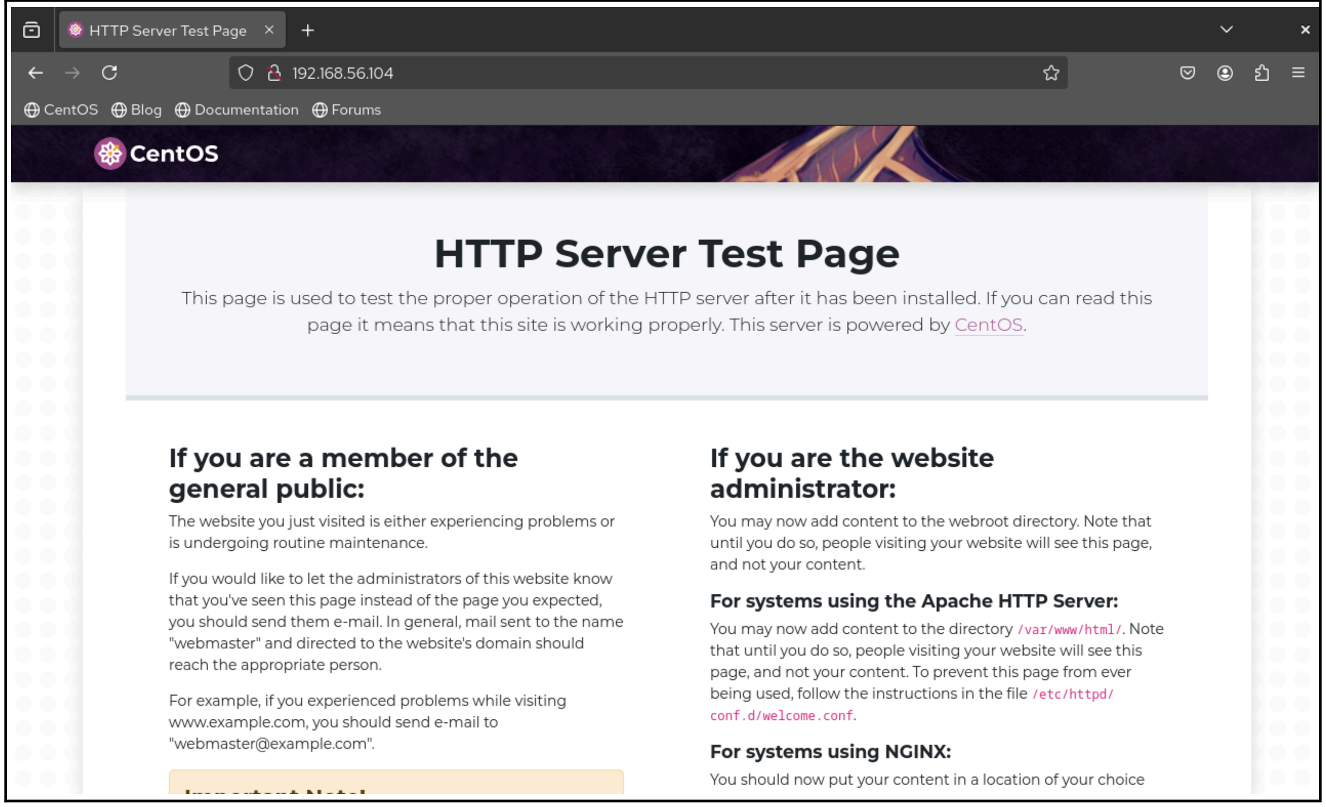
Firefox can't establish a connection to the server at 192.168.56.104.

- 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.
   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.



```
rdjbnfc@localhost:~$
logout
Connection to 192.168.56.104 closed.
redjbonifacio@workstation:~/CPE212_redjbonifacio$ nano site.yml
redjbonifacio@workstation:~/CPE212_redjbonifacio$ nano inventory
redjbonifacio@workstation:~/CPE212_redjbonifacio$ ansible-playbook -i inventory --tags httpd --ask-become-pass site.y
ml
BECOME password:

PLAY [Install Apache and PHP for CentOS servers] *****************************************************************

TASK [Gathering Facts] *******************************************************************************************
ok: [192.168.56.102]
ok: [192.168.56.104]
ok: [192.168.56.103]

TASK [Install Apache and PHP (CentOS)] ***************************************************************************
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [Start and enable httpd service (CentOS)] *******************************************************************
skipping: [192.168.56.102]
skipping: [192.168.56.103]
changed: [192.168.56.104]

PLAY [Install MariaDB for DB servers] ****************************************************************************

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

PLAY RECAP *******************************************************************************************************
192.168.56.102             : ok=1    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.103             : ok=1    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.104             : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

redjbonifacio@workstation:~/CPE212_redjbonifacio$
```



HTTP Server Test Page

This page is used to test the proper operation of the HTTP server after it has been installed. If you can read this page it means that this site is working properly. This server is powered by CentOS.

If you are a member of the general public:

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

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

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

If you are the website administrator:

You may now add content to the webroot directory. Note that until you do so, people visiting your website will see this page, and not your content.

For systems using the Apache HTTP Server:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

For systems using NGINX:

You should now put your content in a location of your choice

Reflections:

Answer the following:

1. What is the importance of putting our remote servers into groups?
    - tt makes it easier to organize and target servers based on their roles

2. What is the importance of tags in playbooks?
    - tags let us run only the tasks we want without running the whole playbook

3. Why do think some services need to be managed automatically in playbooks?
    - Because some services don’t start by default, and automation ensures they always run when needed