Name: Perez, Dean Lenard D.	**Date Performed:** 10/04/2024
Course/Section: CPE 212-CPE31S21	**Date Submitted:** 10/04/2024
Instructor:	**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**:

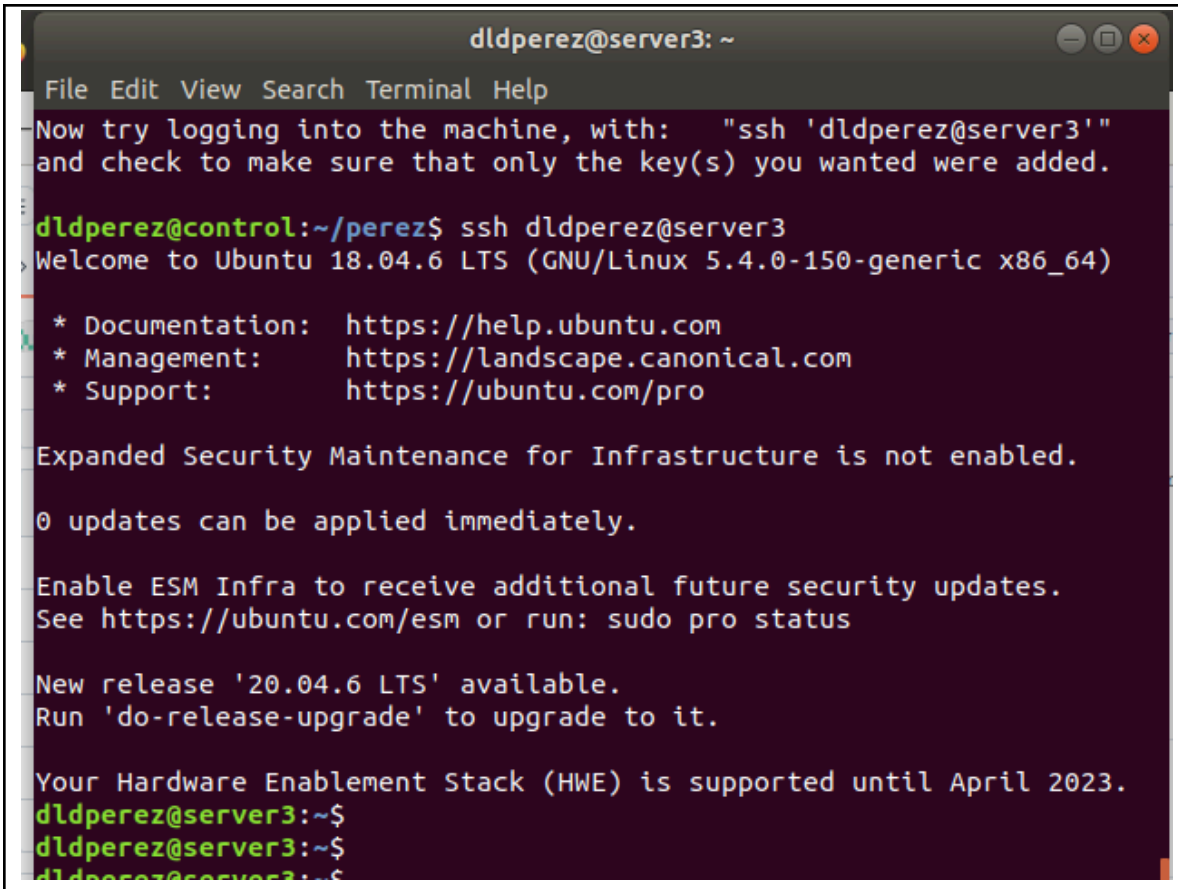
In this activity, we try to individualize hosts. For example, we don't want apache on all our servers, or maybe only one of our servers is a web server, or maybe we have different servers like database or file servers running different things on different categories of servers and that is what we are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

Connection to Server3

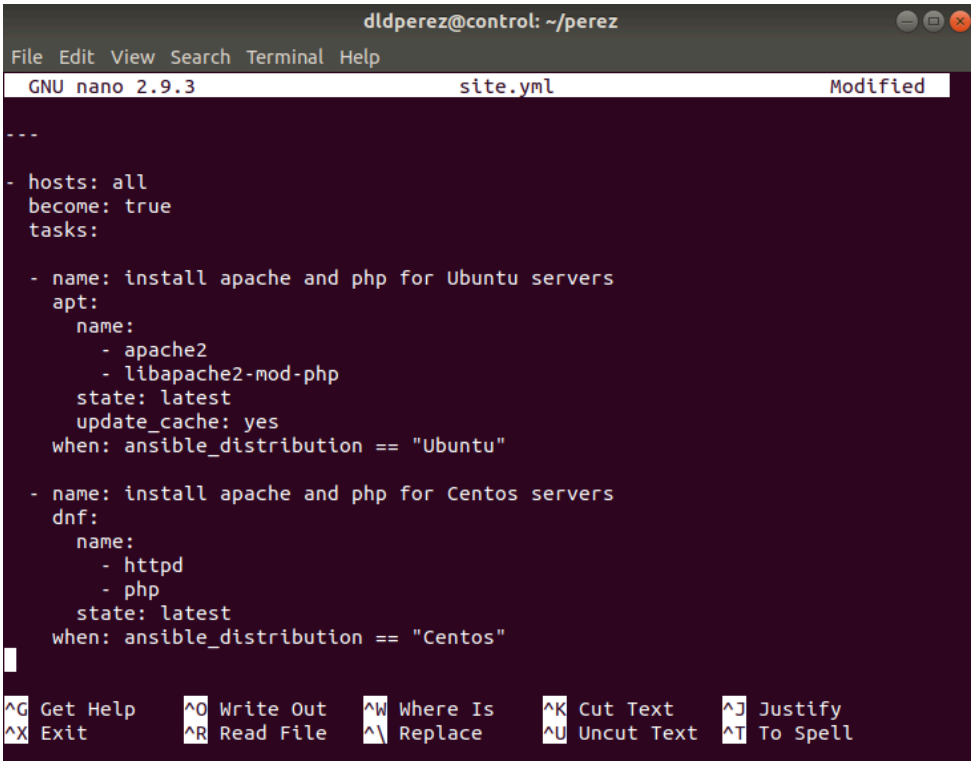


We can ssh from control to server3.

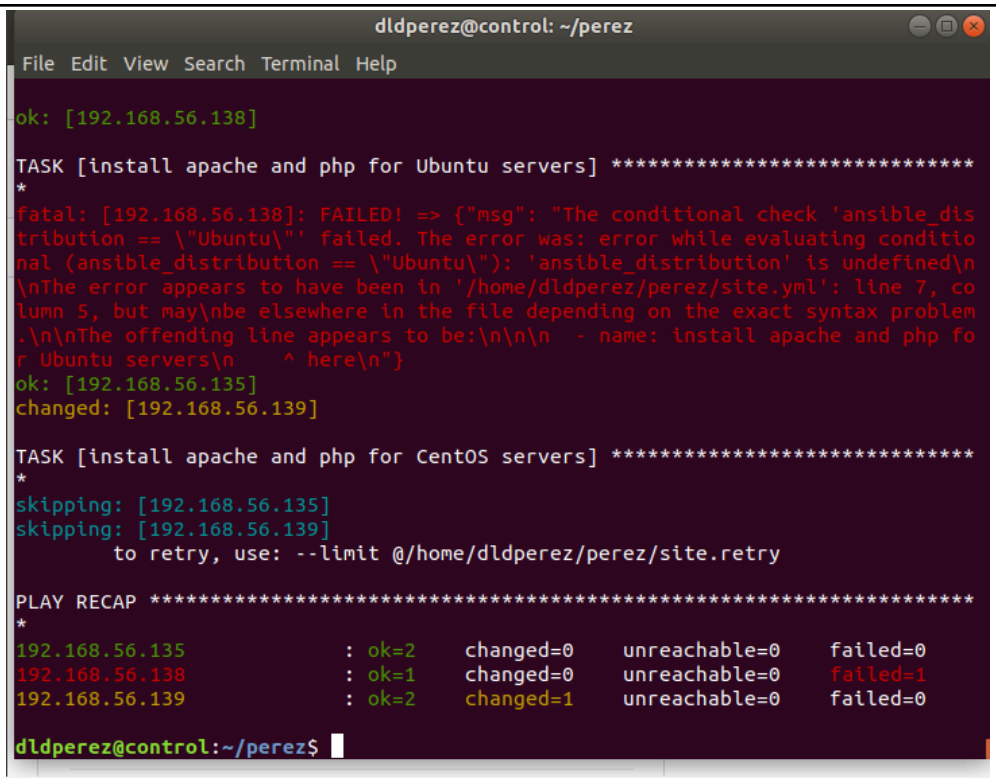
Task 1: Targeting Specific Nodes

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"
```



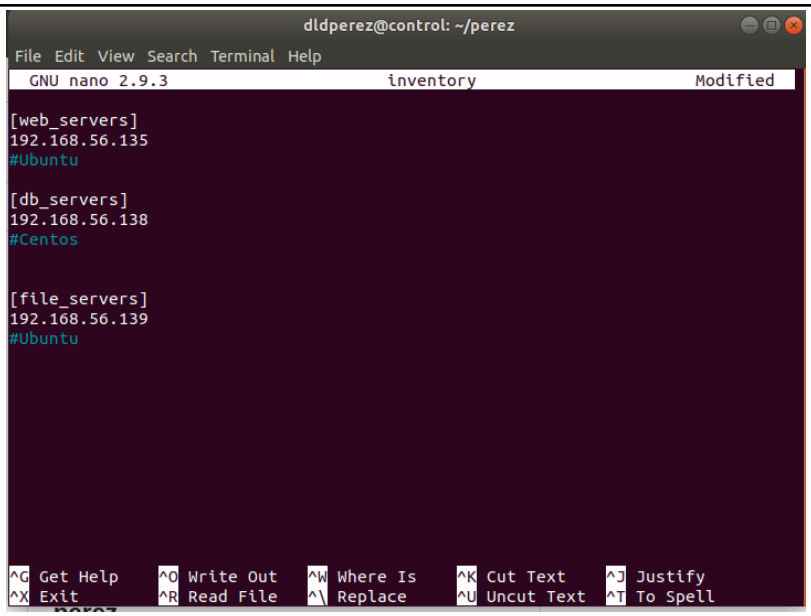
Created site.yml and configured.



The first task was successful on the Ubuntu servers. The second task skipps the ubuntu servers.

-

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



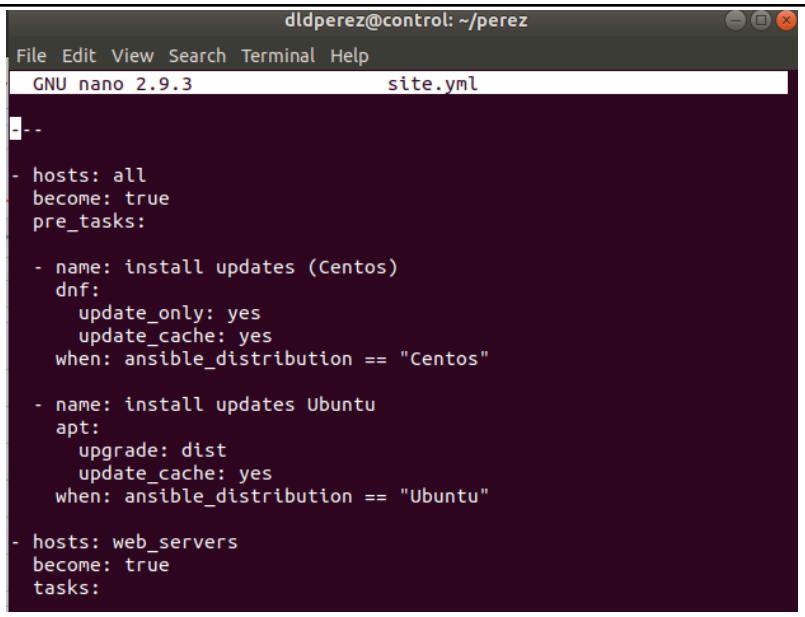
Edited inventory which we removed the variables and separated the ip addresses.

Make sure to save the file and exit.

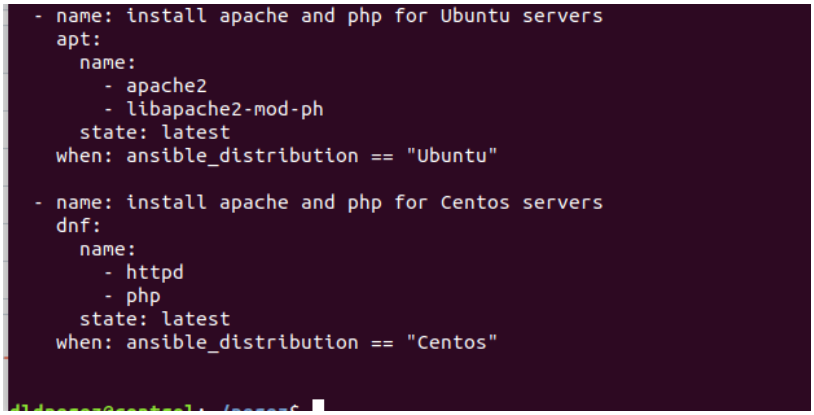
Right now, we have created groups in our inventory file and put each server in its own group. In other cases, you can have a server be a member of multiple groups, for example you have a test server that is also a web server.

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"
```



First-half of the edited version of the site.yml.



Second-half of the edited version of the site.yml.

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 Ubansibuntu 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.

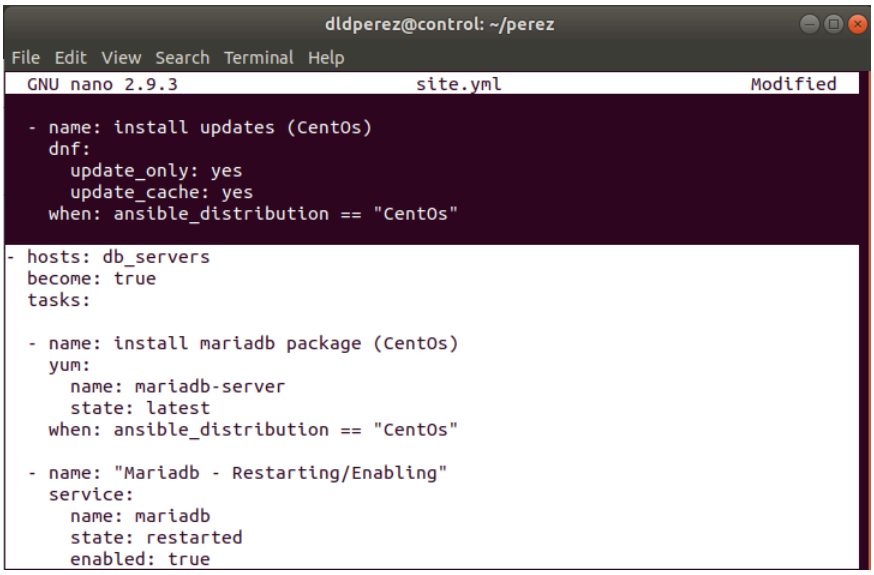
Run the *site.yml* file and describe the result.

```
dldperez@control:~/perez$ nano site.yml
dldperez@control:~/perez$ ansible-playbook --ask-become-pass site.yml
SUDO password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.135]
ok: [192.168.56.139]
 [WARNING]: Module invocation had junk after the JSON data:
AttributeError("module 'platform' has no attribute 'dist'")

ok: [192.168.56.138]

TASK [install updates (Centos)] ************************************************
*
skipping: [192.168.56.135]
skipping: [192.168.56.139]
fatal: [192.168.56.138]: FAILED! => {"msg": "The conditional check 'ansible_dis
tribution == \"Centos\"' failed. The error was: error while evaluating conditio
nal (ansible_distribution == \"Centos\"): 'ansible_distribution' is undefined\n
\nThe error appears to have been in '/home/dldperez/perez/site.yml': line 7, co
lumn 5, but may\nbe elsewhere in the file depending on the exact syntax problem
.\n\nThe offending line appears to be:\n\n\n  - name: install updates (Centos)\
n    ^ here\n"}

TASK [install updates Ubuntu] **************************************************
*
ok: [192.168.56.139]
changed: [192.168.56.135]

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.135]

TASK [install apache and php for Ubuntu servers] *******************************
*
fatal: [192.168.56.135]: FAILED! => {"changed": false, "msg": "No package match
ing 'libapache2-mod-ph' is available"}
        to retry, use: --limit @/home/dldperez/perez/site.retry

PLAY RECAP *********************************************************************
*
192.168.56.135             : ok=3    changed=1    unreachable=0    failed=1
192.168.56.138             : ok=1    changed=0    unreachable=0    failed=1
192.168.56.139             : ok=2    changed=0    unreachable=0    failed=0

dldperez@control:~/perez$
```

The result shows that it failed on the two servers. There is also an error on the first task which 2 are skipped but the last failed because of the conditional statement.

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"
```

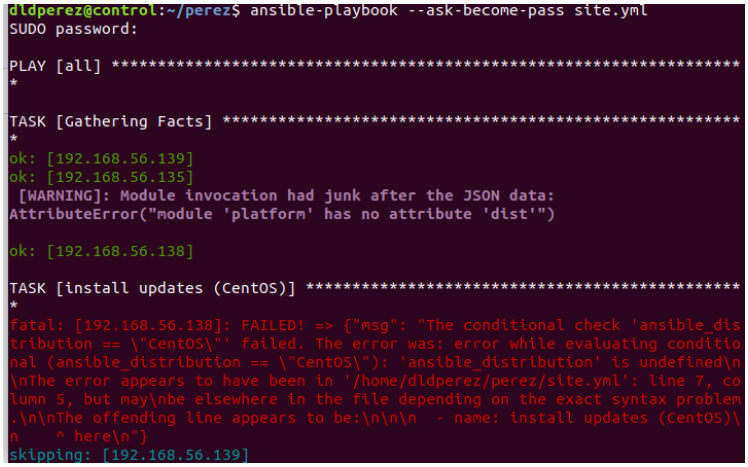


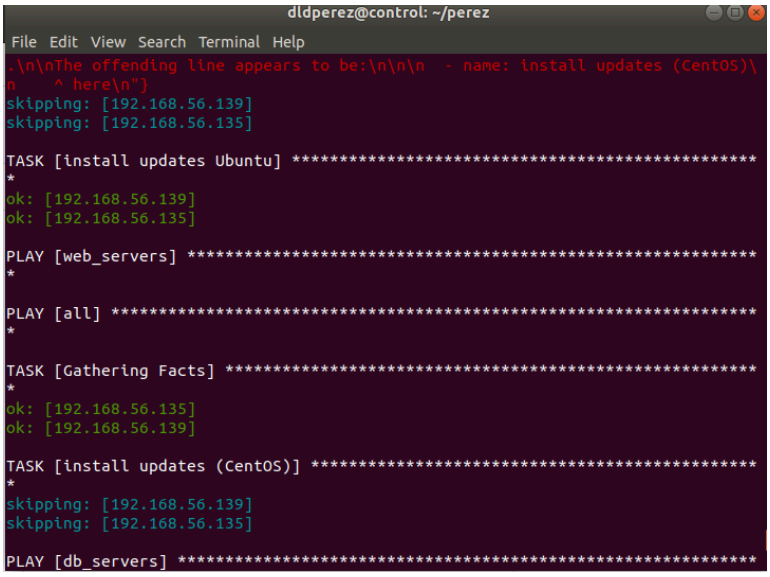
The shaded part is the part that I add to the site.yml.

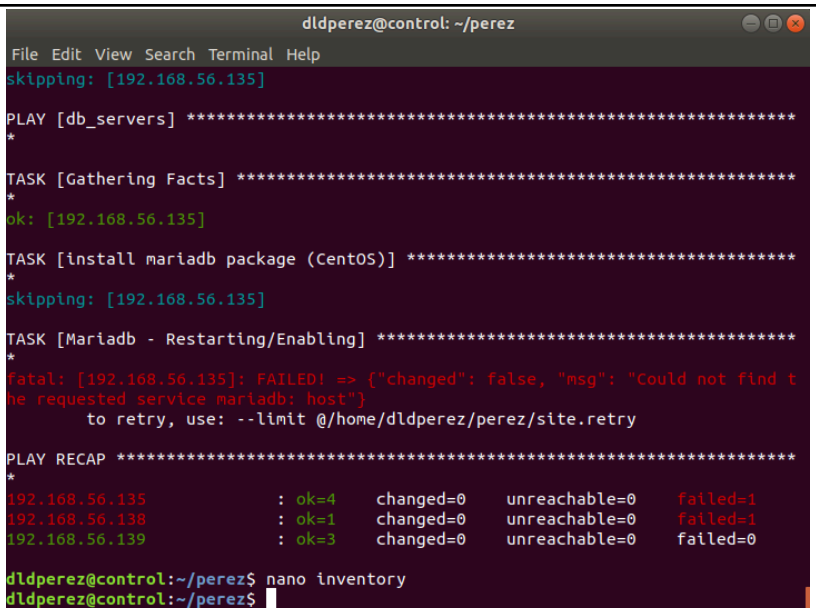
Make sure to save the file and exit.

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

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





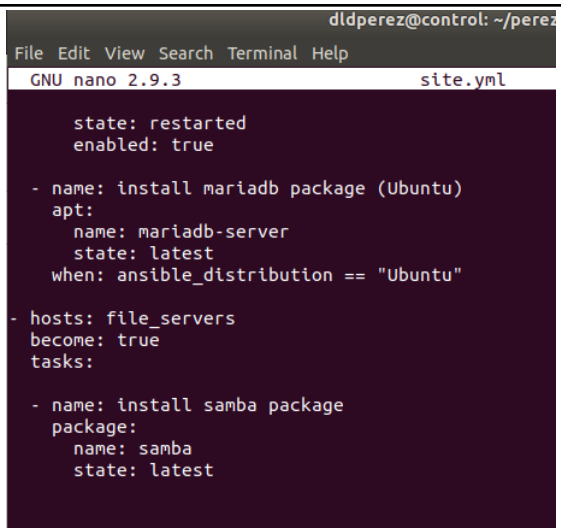


Two failed.

Describe the output.

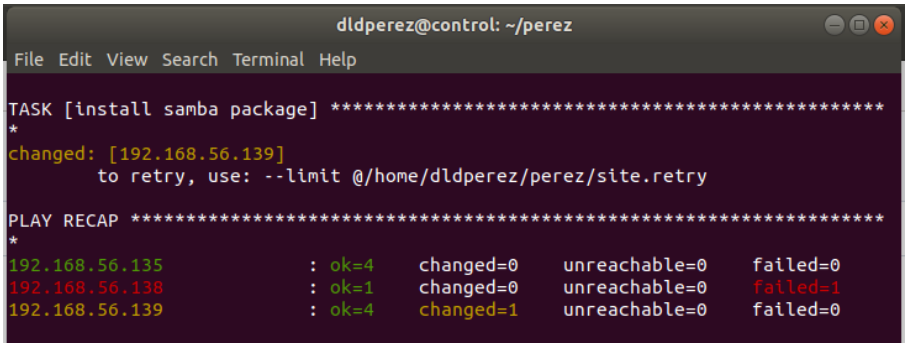
6. Edit the *site.yml* again. This time we will append the code to configure installation on the *file_servers* group. We can add the following on our file.

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

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



Make sure to save the file and exit.

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



It is successful, evidently by the word changed.

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

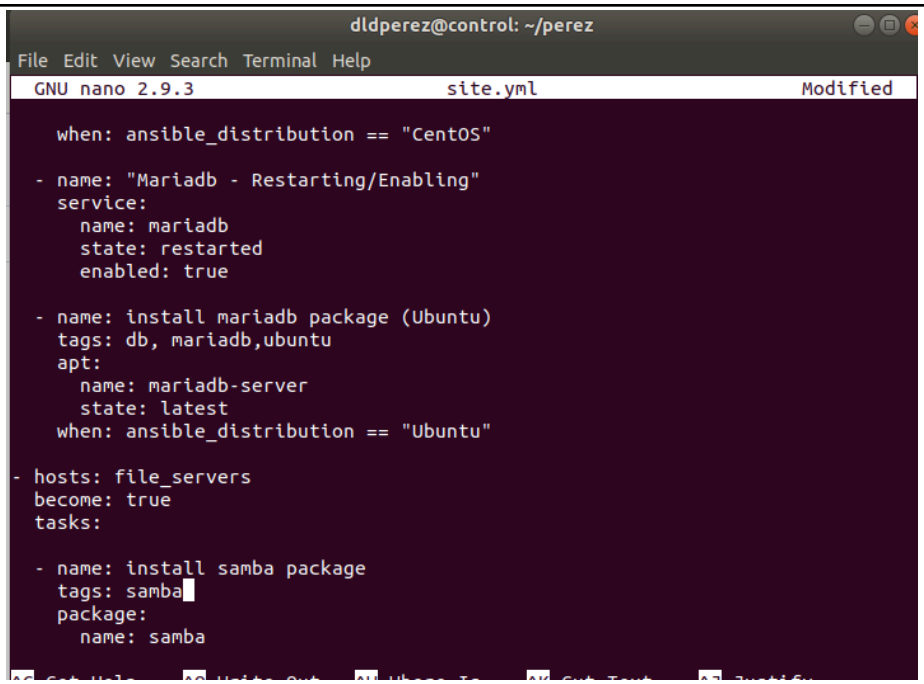
In this task, our goal is to add metadata to our plays so that we can only run the plays that we want to run, and not all the plays in our playbook.

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.

```yaml
---

- hosts: all
  become: true
  pre_tasks:

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

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

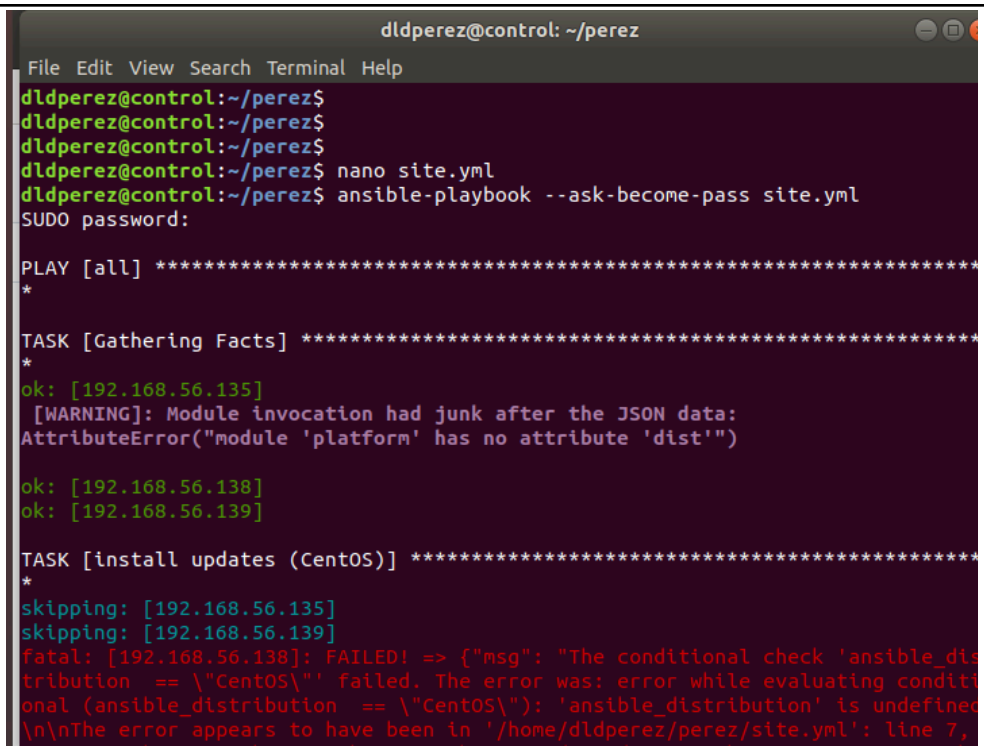
```yaml
- hosts: web_servers
  become: true
  tasks:

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

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

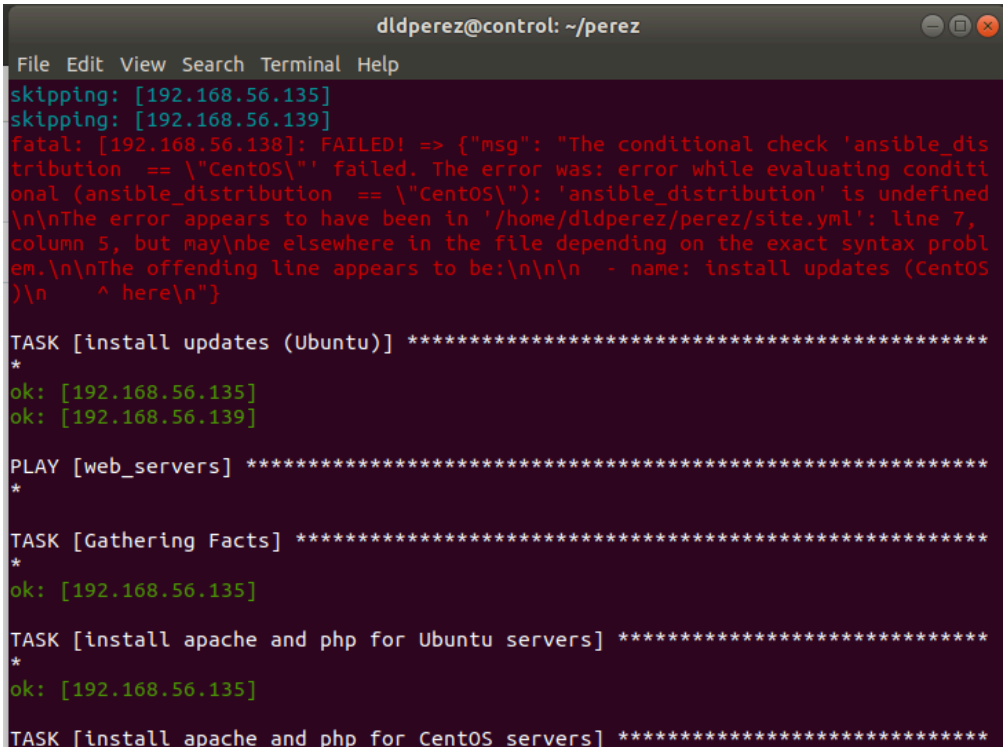
```
- hosts: db_servers
  become: true
  tasks:

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

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

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

- hosts: file_servers
  become: true
  tasks:

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

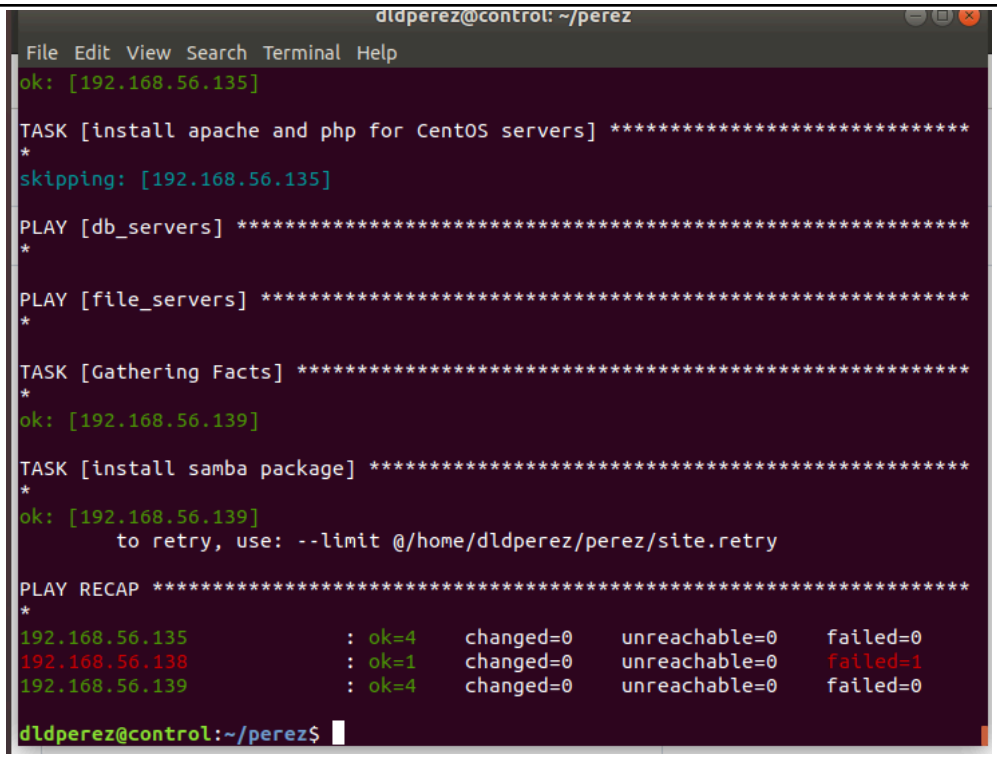


Make sure to save the file and exit.

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


dldperez@control: ~/perez
File Edit View Search Terminal Help
dldperez@control:~/perez$
dldperez@control:~/perez$
dldperez@control:~/perez$
dldperez@control:~/perez$ nano site.yml
dldperez@control:~/perez$ ansible-playbook --ask-become-pass site.yml
SUDO password:
PLAY [all] ***
*
TASK [Gathering Facts] ***
*
ok: [192.168.56.135]
[WARNING]: Module invocation had junk after the JSON data:
AttributeError("module 'platform' has no attribute 'dist'")
ok: [192.168.56.138]
ok: [192.168.56.139]
TASK [install updates (CentOS)] **
*
skipping: [192.168.56.135]
skipping: [192.168.56.139]
fatal: [192.168.56.138]: FAILED! => {"msg": "The conditional check 'ansible_dis
tribution == \"CentOS\"' failed. The error was: error while evaluating conditi
onal (ansible_distribution == \"CentOS\"): 'ansible_distribution' is undefined
\n\nThe error appears to have been in '/home/dldperez/perez/site.yml': line 7,

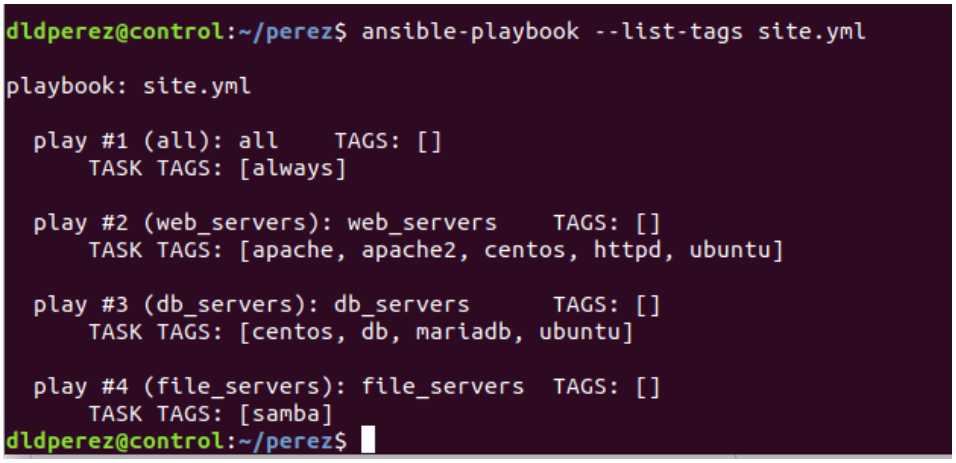

dldperez@control: ~/perez
File Edit View Search Terminal Help
skipping: [192.168.56.135]
skipping: [192.168.56.139]
fatal: [192.168.56.138]: FAILED! => {"msg": "The conditional check 'ansible_dis
tribution == \"CentOS\"' failed. The error was: error while evaluating conditi
onal (ansible_distribution == \"CentOS\"): 'ansible_distribution' is undefined
\n\nThe error appears to have been in '/home/dldperez/perez/site.yml': line 7,
column 5, but may\nbe elsewhere in the file depending on the exact syntax probl
em.\n\nThe offending line appears to be:\n\n\n - name: install updates (CentOS
)\n ^ here\n"}
TASK [install updates (Ubuntu)] **
*
ok: [192.168.56.135]
ok: [192.168.56.139]
PLAY [web_servers] ***
*
TASK [Gathering Facts] ***
*
ok: [192.168.56.135]
TASK [install apache and php for Ubuntu servers] *********************************
*
ok: [192.168.56.135]
TASK [install apache and php for CentOS servers] *********************************



Nothing much happened. It is still the precious result.

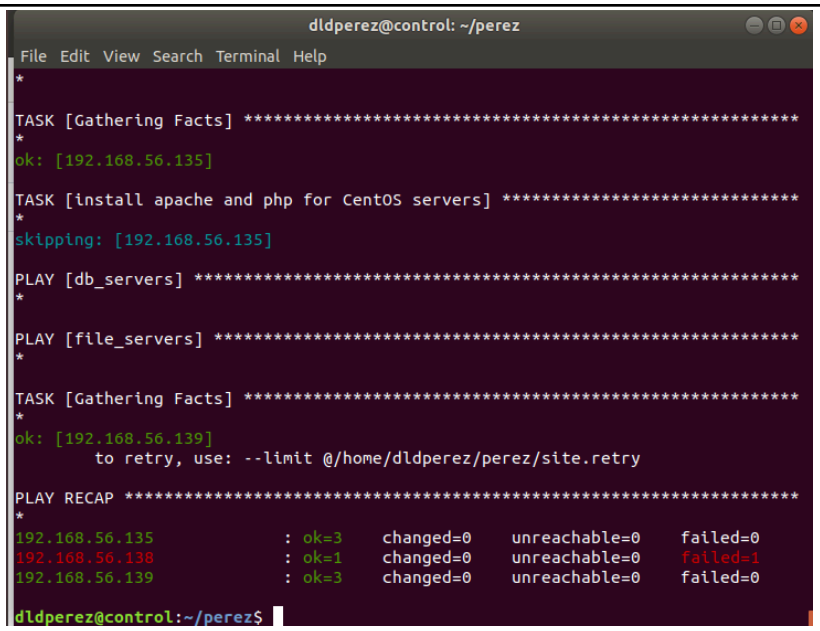
2. On the local machine, try to issue the following commands and describe each result:

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



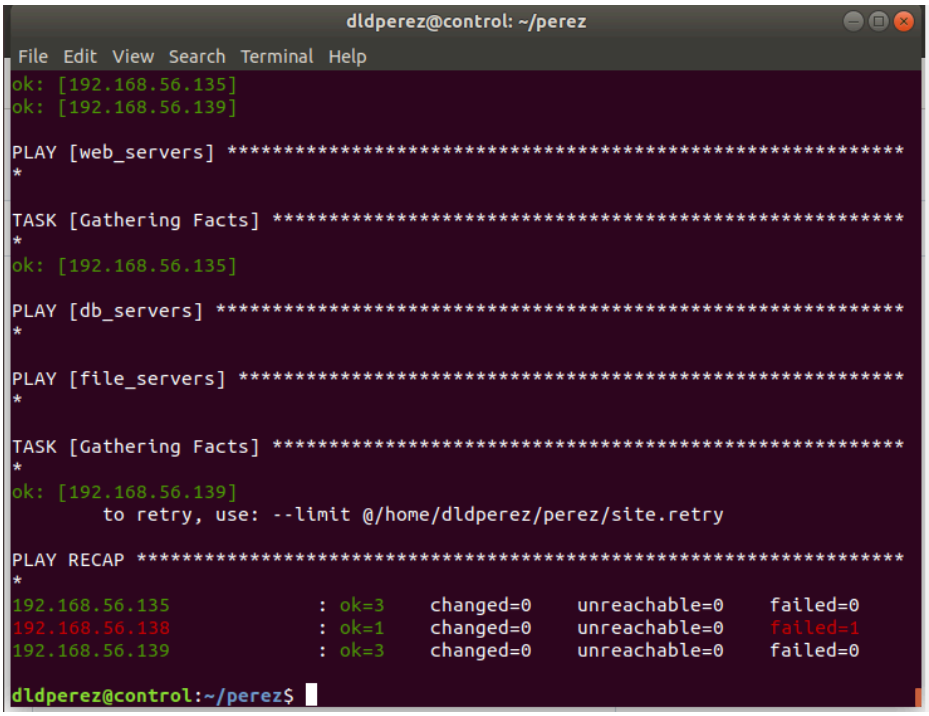
It listed the tags we used and where we used it. It also included the ip addresses or servers on which they are used.

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



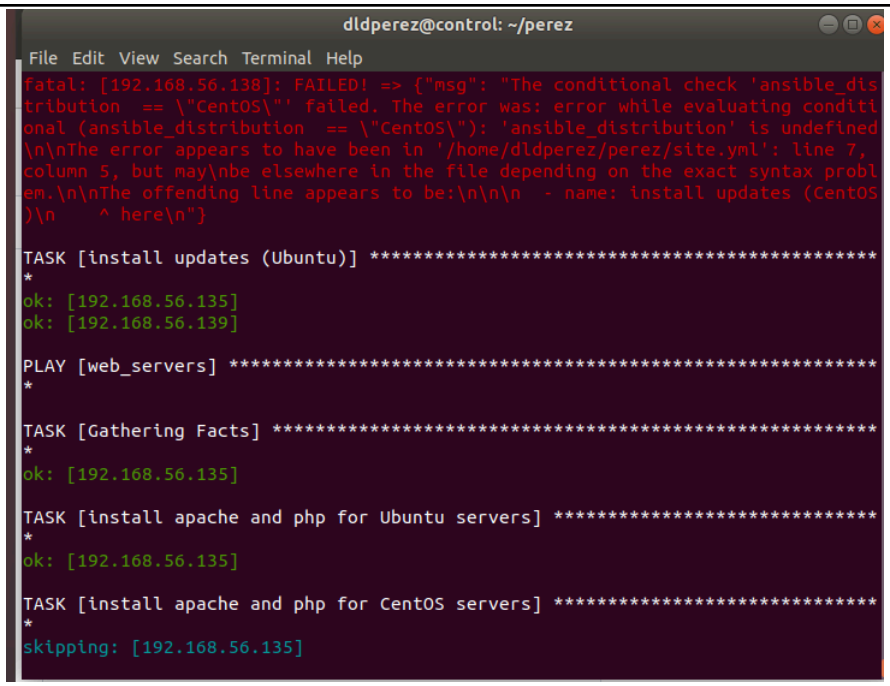
Based on my observation and results, the command executed the tasks that have the tag centos located in the site.yml.

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



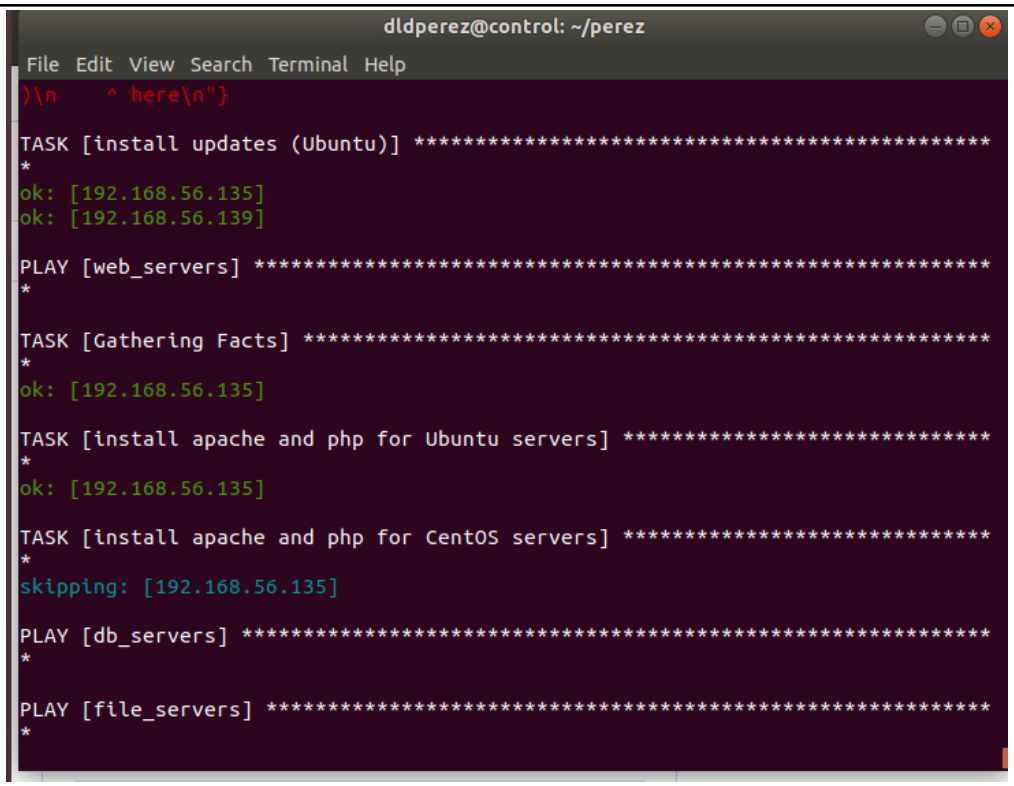
Based on my observation and results, the command executed the tasks that have the tag db located in the site.yml.

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



Based on my observation and results, the command executed the tasks that have the tag apache located in the site.yml.

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



Based on my observation and results, the command executed the tasks that have the tag "apache,db" located in the 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"
```

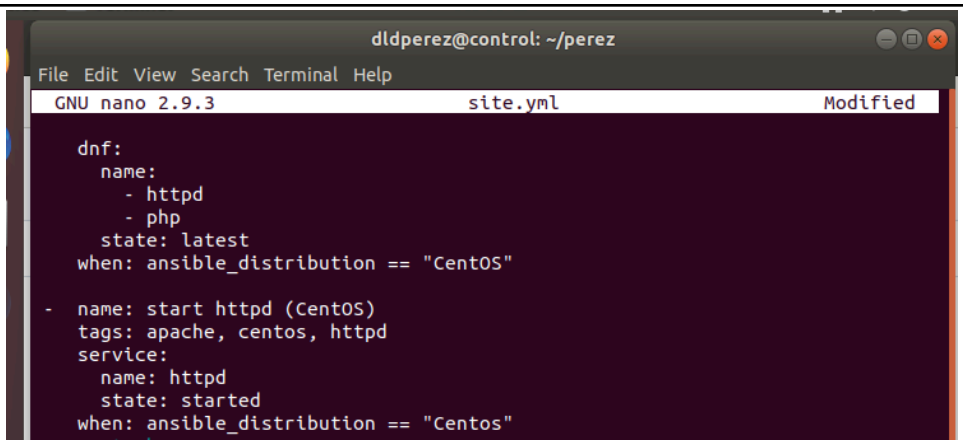
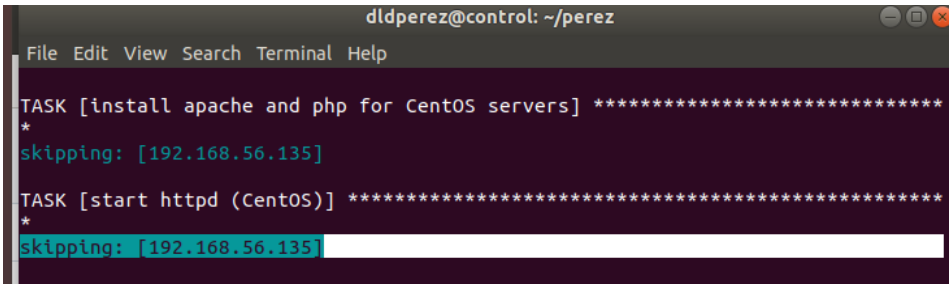


Figure 3.1.1

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.
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?

   It is important so that we will not be confused about the ip addresses and hostname that we used. It looks neat and it will be easy for the user to read the information that he needs.
2. What is the importance of tags in playbooks?

   Tags are important to provide an easy way to strings because some string may be long. Lessening the work is good.
3. Why do think some services need to be managed automatically in playbooks?

   Some services need to be managed automatically because a lot of services updates periodically, setting it automatically lessens the time.