Name: Gabriel Soriano	**Date Performed: October 06, 2022**
Course/Section: CPE 232 - Managing Enterprise Servers	**Date Submitted: October 07, 2022**
Instructor: Engr. Taylar	**Semester and SY: 1st sem - SY 2022-2023**

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**:

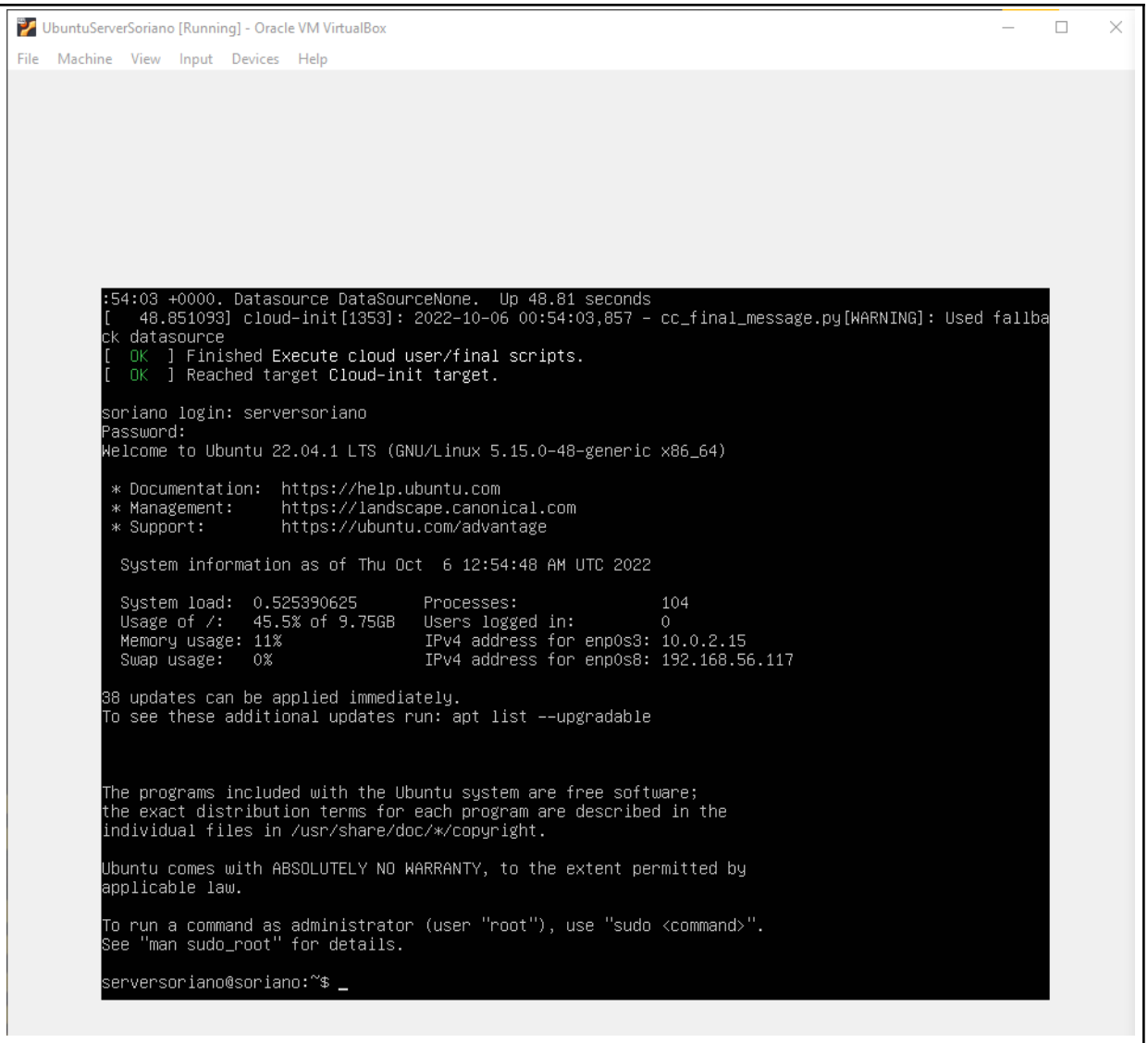
In this activity, we try to individualize hosts. For example, we don't want apache on all our servers, or maybe only one of our servers is a web server, or maybe we have different servers like database or file servers running different things on different categories of servers and that is what we are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

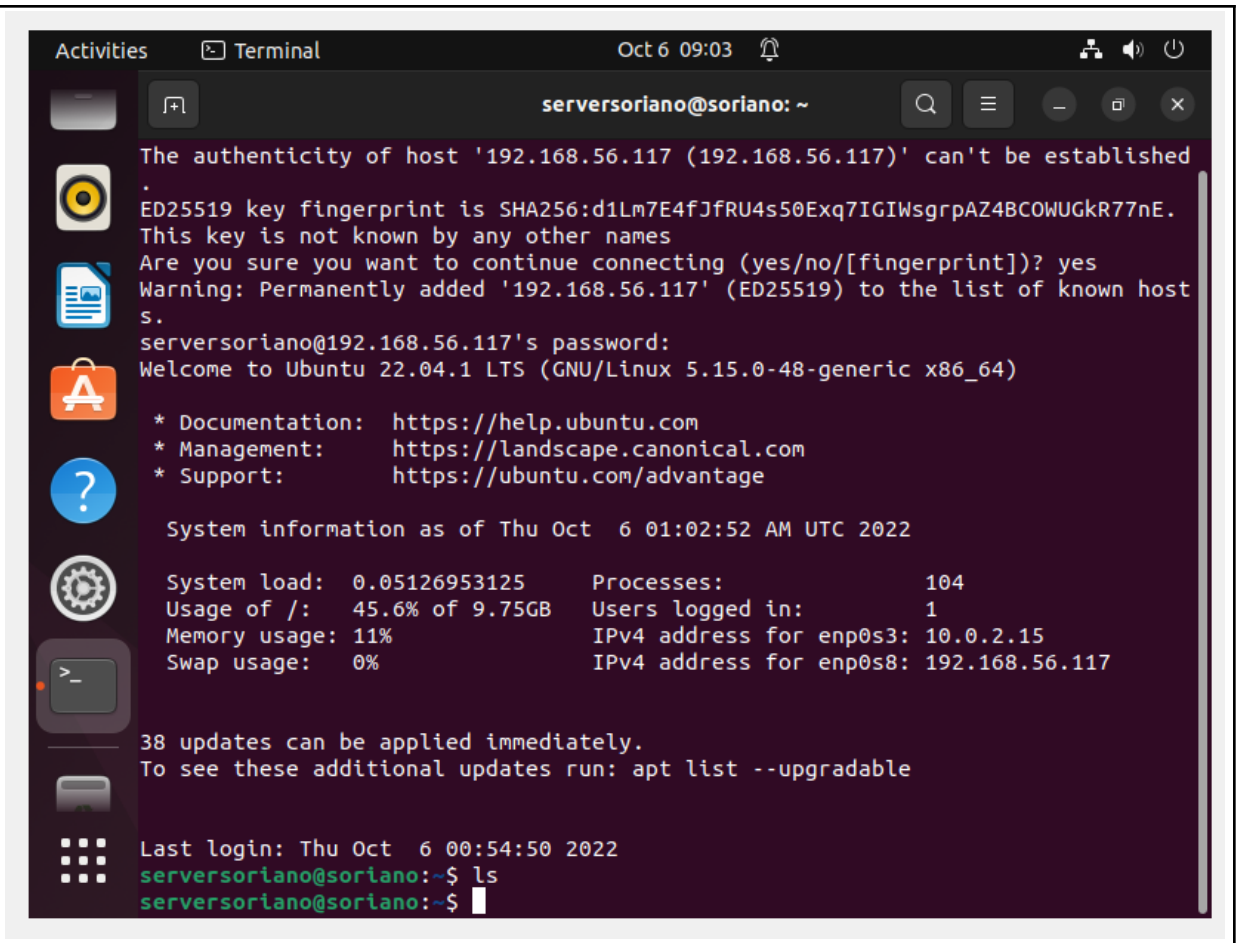
Requirement:

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.

Installation of UbuntuServer
SCREENSHOTS:



- **This shows that the Ubuntu server is successfully downloaded and installed in a new Virtual machine.**



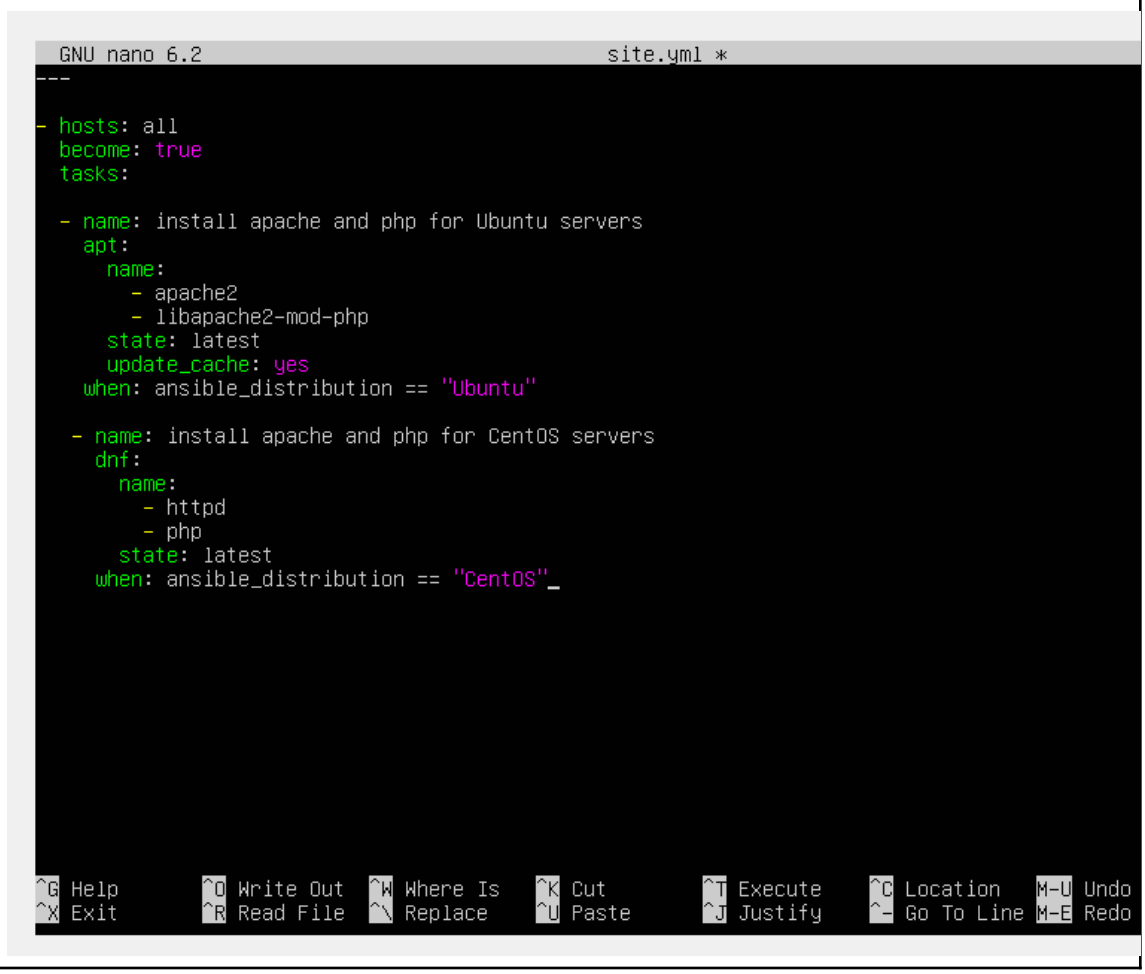
- **This shows that the ssh connection between the server 1 and the UbuntuServer is successful.**

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.

```yaml
---

- 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"
```

SCREENSHOT:



- **This shows the creation of a new .yml file name site.yml, this inside the Server3.**

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

```
[web_servers]
192.168.56.120
192.168.56.121

[db_servers]
192.168.56.122


[file_servers]
192.168.56.123
```

Make sure to save the file and exit.

SCREENSHOT:

```
  GNU nano 6.2                                  inventory *
192.168.56.105
192.168.56.106

[remote_servers]
192.168.56.105
192.168.56.106

[web_server]
192.168.56.105
[db_server]
192.168.56.105
[file server]
192.168.56.106_

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

- **This shows the editing of the file "inventory", then added the ip addresses of the remote server, web server, db server, and the file server.**

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

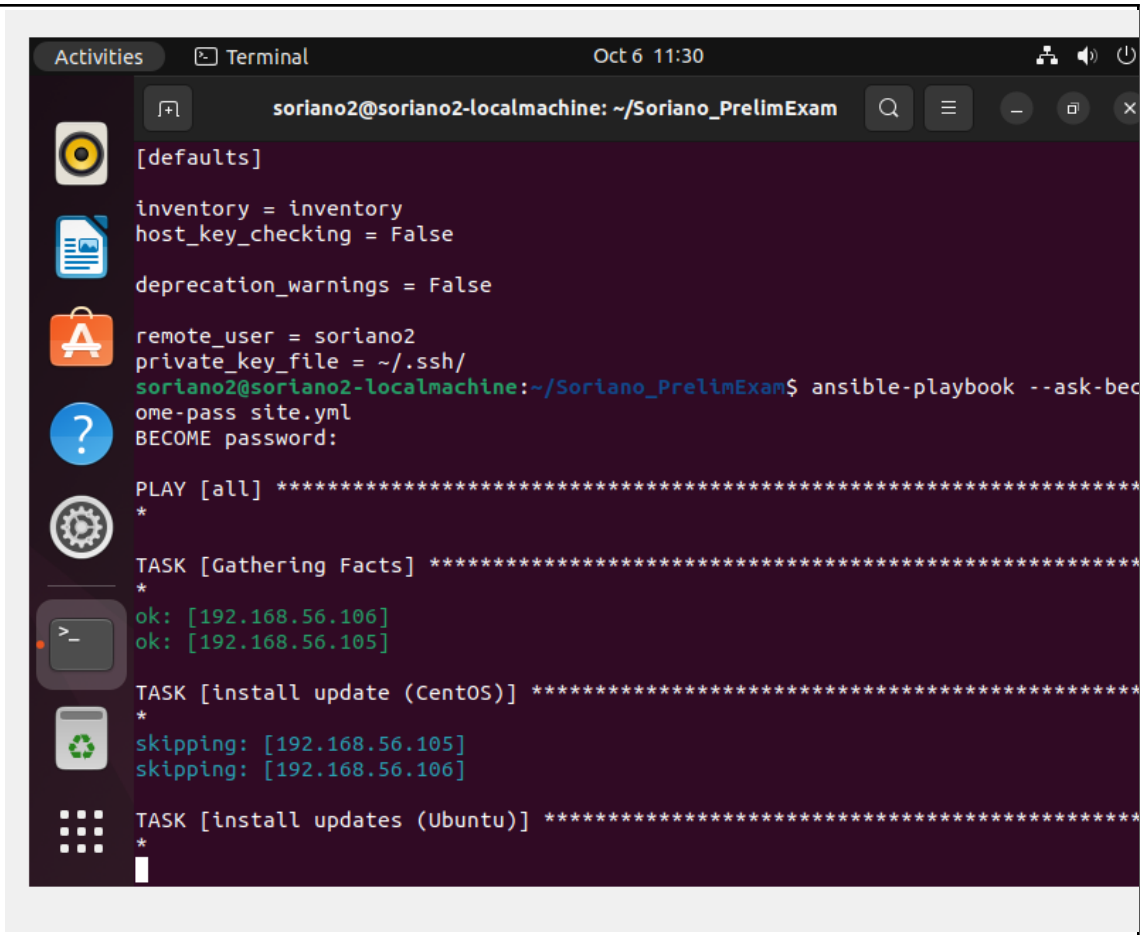
3. Edit the *site.yml* by following the image below:

```
---

- hosts: all
  become: true
  pre_tasks:

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

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


- hosts: web_servers
  become: true
  tasks:

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

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

Make sure to save the file and exit.

SCREENSHOT:

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

- hosts: all
  become: true
  pre_tasks:

  - name: instlal update (CentOS)
    dnf:
      update_only: yes
      update_cache: yes
    when: ansible_distribution == "CentOS"

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

- hosts: web_servers
  become: true
  tasks:

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

   - name: install apache and php for CentOS servers
     dnf:
                                      [ Wrote 38 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut       ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste     ^J Justify    ^- Go To Line M-E Redo
```



- **This shows the editing of the site.yml and adding the pre tasks commands. Also the process of running the task "install updates" for the Ubuntu distributions OS.**

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

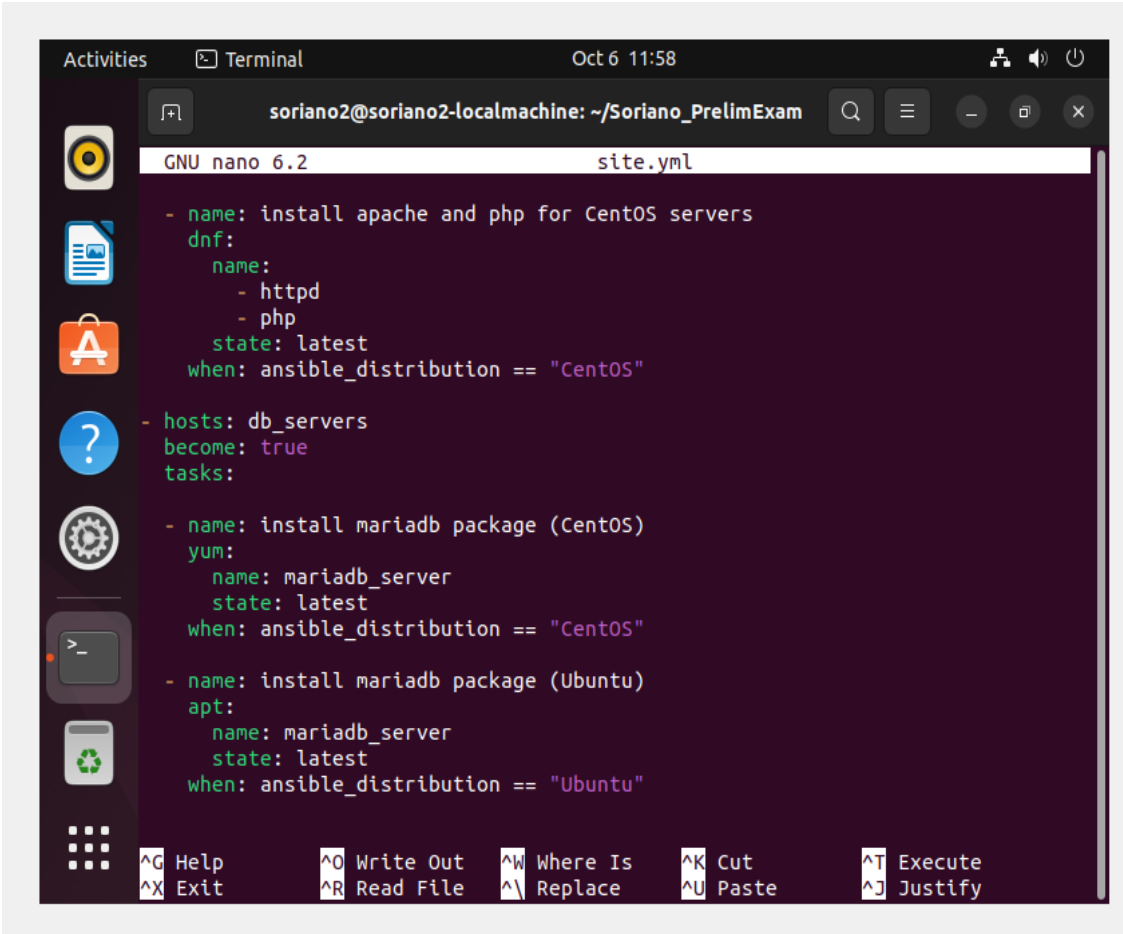
Run the *site.yml* file and describe the result.

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

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

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

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

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

Make sure to save the file and exit.

SCREENSHOT:



- **This shows the editing process of the site.yml and added the lines of "Mariadb". After the editing of the yml file, after running the playbook, it now did run the new commands included in the edited yml file.**

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

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

   Describe the output.

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.

SCREENSHOTS:

```
  GNU nano 5.4
    when: ansible_distribution == "CentOS"

- hosts: db_servers
  become: true
  tasks:

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

  - name: "Mariabd- Restarting/Enabling"
    service:
      name: maridb
      state: restarted
      enabled: true

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

- hosts: file_servers
  become: true
  tasks:

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

- **This shows the edited version of the yml file, and a new group is added which is the file_servers. And also the addition of the installation of the**

samba package. After the execution of the command to run the playbook, the additional commands was also running, which is the installation of samba, the process now took longer time to complete.

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"
```

```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.

SCREENSHOT:

```
GNU nano 5.4
  tasks:

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

  - name: "Mariabd- Restarting/Enabling"
    service:
      name: maridb
      state: restarted
      enabled: true

  - name: install mariadb package (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
```

- **This shows the editing of the yml file where there are tags being added to the commands and also in each of the groups. After the execution of the command to run the playbook, all jobs with tags that match those inserted at the command line are runned by Ansible, or they are skipped, depends on the matching.**

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

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

   SCREENSHOT:

```
qgsoriano1@cloudshell:~/Soriano_PrelimExam$ ansible-playbook --list-tags site.yml
/home/qgsoriano1/.local/lib/python2.7/site-packages/ansible/parsing/vault/__init__.py:44: CryptographyDeprecationWarning: Python 2 is no longer supported by the Pyth
on core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
  from cryptography.exceptions import InvalidSignature

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: []

  play #5 (file_servers): file_servers  TAGS: []
      TASK TAGS: [samba]
qgsoriano1@cloudshell:~/Soriano_PrelimExam$
```

- *This command shows all of the inserted tags in the playbook file or in the yml file, depending on what file is specified.*

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

SCREENSHOT:

```
qgsoriano1@cloudshell:~/Soriano_PrelimExam$ ansible-playbook --tags centos --ask-become-pass site.yml
/home/qgsoriano1/.local/lib/python2.7/site-packages/ansible/parsing/vault/__init__.py:44: CryptographyDeprecationWarning: Python 2 is
on core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
  from cryptography.exceptions import InvalidSignature
BECOME password:

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

TASK [Gathering Facts] ***********************************************************************************************************
fatal: [192.168.56.105]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.
med out", "unreachable": true}
fatal: [192.168.56.106]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.
med out", "unreachable": true}

PLAY RECAP ***********************************************************************************************************************
192.168.56.105             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.106             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0

qgsoriano1@cloudshell:~/Soriano_PrelimExam$
```

- *This shows that the playbook is being runned again, but this time I have encountered an error, as seen above in the screenshot. It also asked for the password of the user. But I think the command should have runned the playbook based only on the specified tags which is centos in the command.*

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

SCREENSHOT:

```
qgsoriano1@cloudshell:~/Soriano_PrelimExam$ ansible-playbook --tags db --ask-become-pass site.yml
/home/qgsoriano1/.local/lib/python2.7/site-packages/ansible/parsing/vault/__init__.py:44: CryptographyDeprecationWarning: Python 2 is no longer suppo
on core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
  from cryptography.exceptions import InvalidSignature
BECOME password:

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

TASK [Gathering Facts] ***********************************************************************************************************
fatal: [192.168.56.105]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.56.105 port 2
med out", "unreachable": true}
fatal: [192.168.56.106]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.56.106 port 2
med out", "unreachable": true}

PLAY RECAP ***********************************************************************************************************************
192.168.56.105             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.106             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0

qgsoriano1@cloudshell:~/Soriano_PrelimExam$
```

- *I wasn't able to fix the error because the workstation is lagging and hanging too much. It also asked for the password of the user. But I think the command*

should have runned the playbook based only on the specified tags which is db in the command.

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

SCREENSHOT:

```
qgsoriano1@cloudshell:~/Soriano_PrelimExam$ ansible-playbook --tags apache --ask-become-pass site.yml
/home/qgsoriano1/.local/lib/python2.7/site-packages/ansible/parsing/vault/__init__.py:44: CryptographyDeprecationWarning: Python 2 is no longer supported by the Pyth
on core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
  from cryptography.exceptions import InvalidSignature
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************************************************
fatal: [192.168.56.105]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.56.105 port 22: Connection ti
med out", "unreachable": true}
fatal: [192.168.56.106]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.56.106 port 22: Connection ti
med out", "unreachable": true}

PLAY RECAP *********************************************************************************************************************************************
192.168.56.105             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.106             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0

qgsoriano1@cloudshell:~/Soriano_PrelimExam$
```

- *I wasn't able to fix the error because the workstation is lagging and hanging too much. It also asked for the password of the user. But I think the command should have runned the playbook based only on the specified tags which is apache in the command.*

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

SCREENSHOT:

```
qgsoriano1@cloudshell:~/Soriano_PrelimExam$ ansible-playbook --tags "apache,db" --ask-become-pass site.yml
/home/qgsoriano1/.local/lib/python2.7/site-packages/ansible/parsing/vault/__init__.py:44: CryptographyDeprecationWarning: Python 2 is no longer supported by the Pyth
on core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
  from cryptography.exceptions import InvalidSignature
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************************************************
fatal: [192.168.56.105]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.56.105 port 22: Connection ti
med out", "unreachable": true}
fatal: [192.168.56.106]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.56.106 port 22: Connection ti
med out", "unreachable": true}

PLAY RECAP *********************************************************************************************************************************************
192.168.56.105             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.106             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
```

- *I wasn't able to fix the error because the workstation is lagging and hanging too much. It also asked for the password of the user. But I think the command should have runned the playbook based only on the specified tags which is apache and db in the command.*

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.

SCREENSHOT:

```
  GNU nano 5.4
      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
    when: ansible_distribution == "CentOS"

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

- **This shows the insertion of an additional line of command to automatically start httpd for a specified OS only, which is CentOS.**

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?
   - The importance of putting our remote servers into groups is to effectively manage the ip addresses accordingly. And to minimize the performing time of the tasks that are assigned to each of the servers. The significance of placing our distant servers into bunches is to really deal with the ip addresses in like manner. What's more, to limit the performing season of the undertakings that are appointed to every one of the servers.

2. What is the importance of tags in playbooks?
   - In essence, tags are basically associated with tasks, roles, and plays. The running and debugging of playbooks can be streamlined and time-saving by using tags. Additionally, it strengthens and organizes the playbook. Generally, labels are fundamentally connected with errands, jobs, and plays. The running and investigating of playbooks can be smoothed out and efficient by utilizing labels. Moreover, it fortifies and arranges the playbook.

3. Why do you think some services need to be managed automatically in playbooks?
   - This is to save the time of performing, assigning, and running tasks. Why? Because if I try to run them manually and one by one, the loading screen will be endless.