

Name: Seruelas, Ronn Kristoper H.	Date Performed: 02-10-2023
Course/Section: CPE31S4	Date Submitted: 02-11-2023
Instructor: Dr. Jonathan V. Taylar	Semester and SY: 2nd Sem. SY23-24

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 0: Preparation

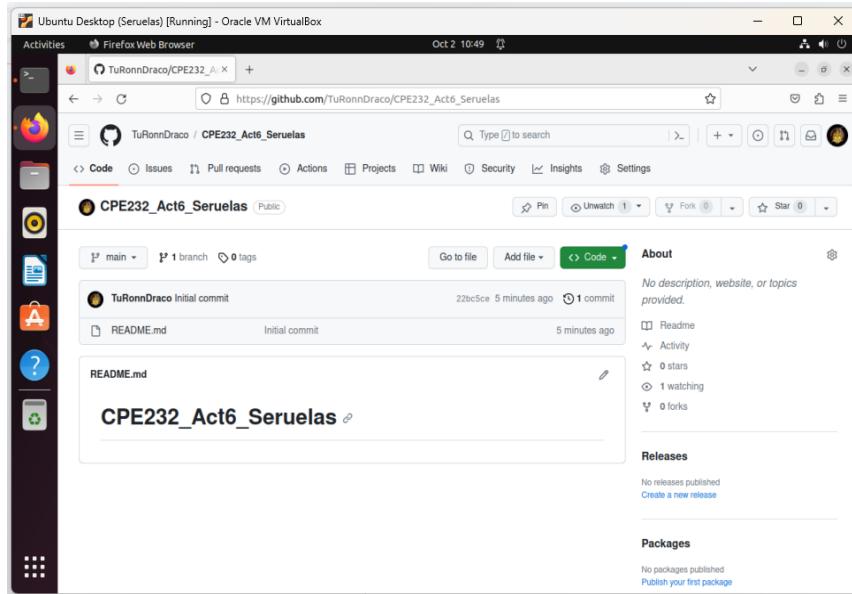


Figure 0.1 - Creation of a new repository dedicated for the activity.

```
seruelas@workstation: ~/CPE232_Act6_Seruelas$ git clone git@github.com:TuRonnDraco/CPE232_Act6_Seruelas.git
Cloning into 'CPE232_Act6_Seruelas'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
seruelas@workstation: $ ls
Ansible54          CPE232_TESTREPOSITORY  Music           snap
CPE232_Act5_Seruelas  Desktop          Pictures        Templates
CPE232_Act6_Seruelas  Documents        Public         Videos
CPE232_Seruelas1     Downloads        Seruelas_PrelimExam
seruelas@workstation: $ cd CPE232_Act6_Seruelas
seruelas@workstation: ~/CPE232_Act6_Seruelas$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Figure 0.2 - Cloning of new repository onto the workstation.

```
GNU nano 6.2
[defaults]

inventory = inventory
host_key_checking = False

deprecation_warning = False

remote_user = seruelas
private_key_file = ~/.ssh/
```

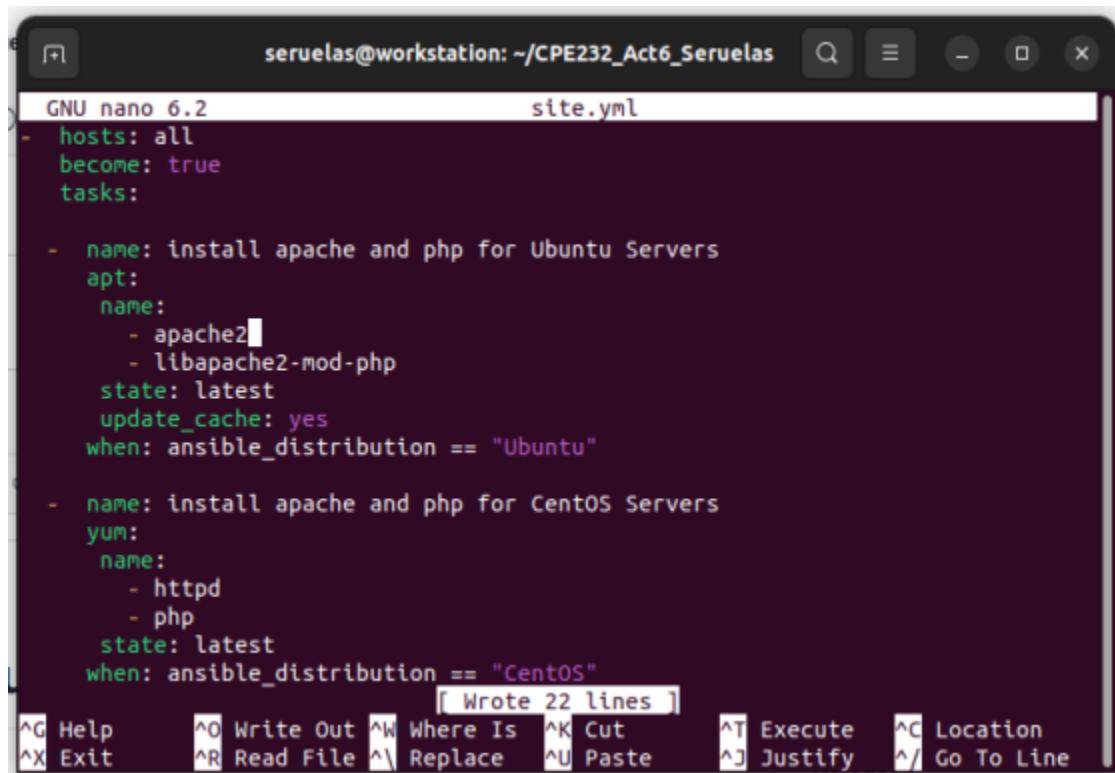
Figure 0.3 - Configuration of the ansible.cfg for the new repository.

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"
```

Figure 1.1.1 - Commands for the site.yml



The screenshot shows a terminal window titled "seruelas@workstation: ~/CPE232_Act6_Seruelas". The window contains the YAML code for the site.yml playbook. The code defines a playbook with a single host group ("hosts: all") and become privilege level ("become: true"). It contains two tasks: one for Ubuntu servers using the "apt" module to install Apache and PHP, and another for CentOS servers using the "dnf" module to install Apache and PHP. Both tasks set the "state" to "latest" and "update_cache" to "yes". A condition "when: ansible_distribution == 'Ubuntu'" is applied to the first task, and "when: ansible_distribution == 'CentOS'" is applied to the second task. The bottom status bar of the terminal window indicates "[Wrote 22 lines]".

```
GNU nano 6.2                                     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  
yum:  
  name:  
    - httpd  
    - php  
  state: latest  
when: ansible_distribution == "CentOS"
```

Figure 1.1.2 - Creation of the site.yml for the new repository with the corresponding commands.

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

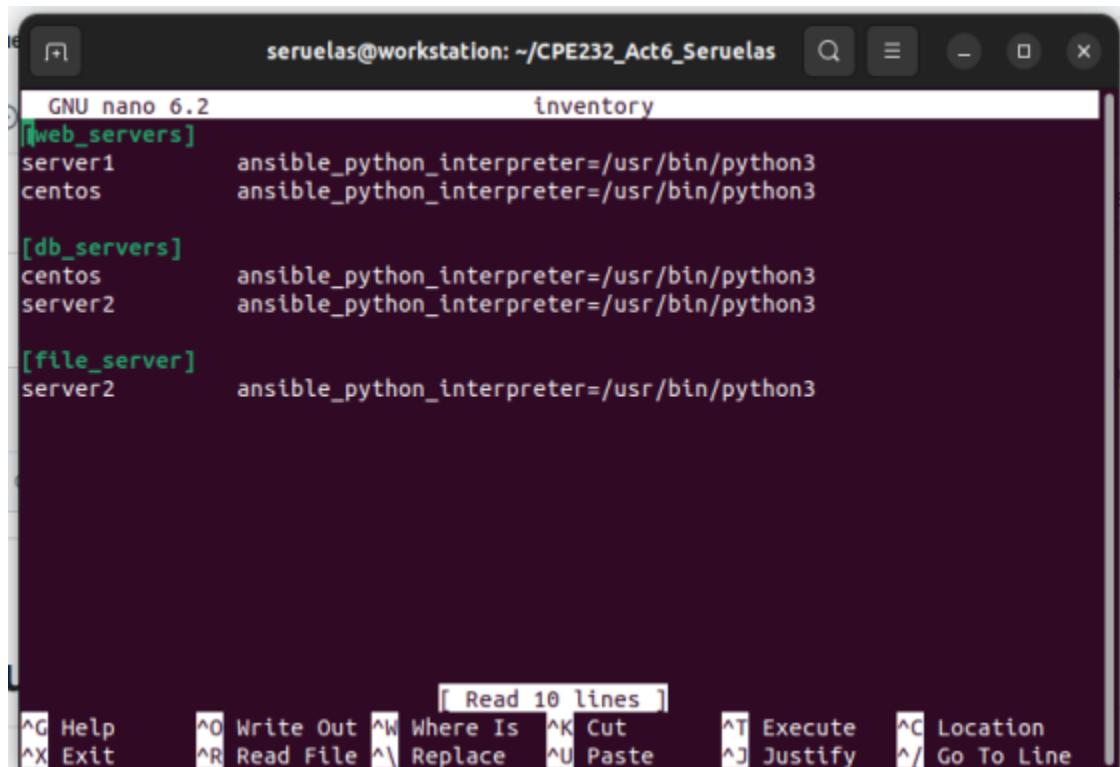
```
[web_servers]
192.168.56.120
192.168.56.121

[db_servers]
192.168.56.122

[file_servers]
192.168.56.123
```

Figure 1.2.1 - Inventory file for the activity.

Make sure to save the file and exit.



The screenshot shows a terminal window titled "seruelas@workstation: ~/CPE232_Act6_Seruelas". The user is using the "nano" text editor to create an inventory file. The file contains the following content:

```
GNU nano 6.2                               inventory
[web_servers]
server1      ansible_python_interpreter=/usr/bin/python3
centos       ansible_python_interpreter=/usr/bin/python3

[db_servers]
centos       ansible_python_interpreter=/usr/bin/python3
server2      ansible_python_interpreter=/usr/bin/python3

[file_server]
server2      ansible_python_interpreter=/usr/bin/python3
```

The terminal window includes a menu bar at the top and a toolbar at the bottom with various keyboard shortcuts for editing.

Figure 1.2.2 - Creation of the inventory file with the corresponding servers (By using the hosts instead of the standard IP Address)

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"

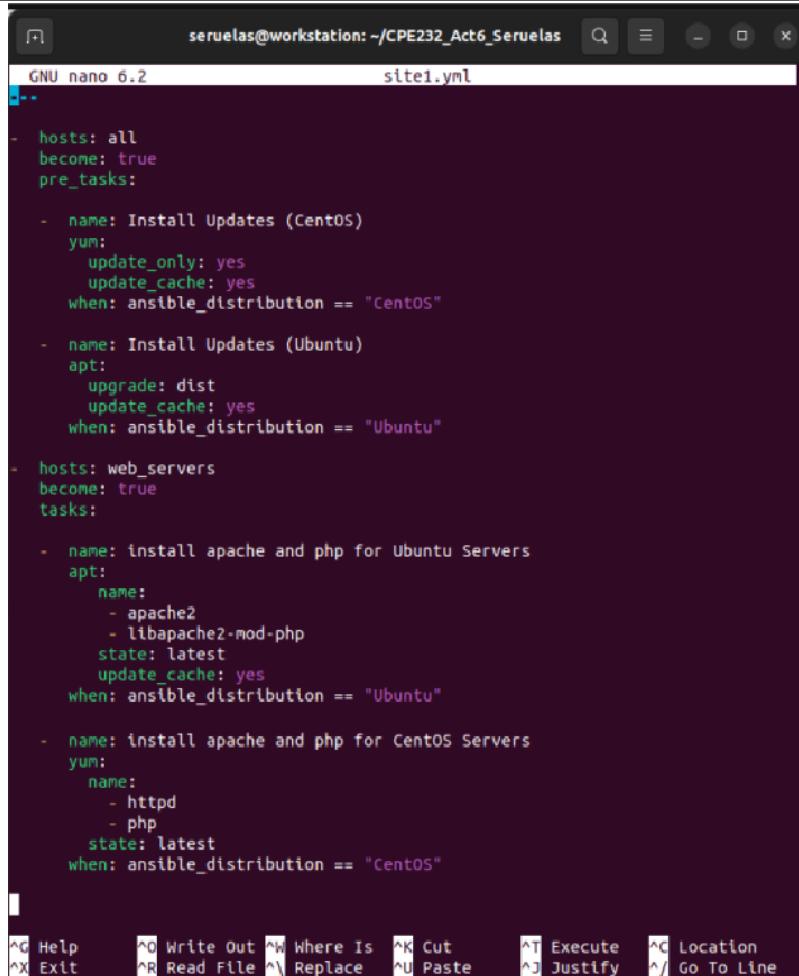
    - name: install updates (Ubuntu)
      apt:
        upgrade: dist
        update_cache: yes
      when: ansible_distribution == "Ubuntu"

- 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"
```

Figure 1.3.1 - Additional Commands for site.yml

Make sure to save the file and exit.



The screenshot shows a terminal window titled "seruelas@workstation: ~/CPE232_Act6_Seruelas". The file being edited is "site1.yml". The code in the file is as follows:

```
GNU nano 6.2             site1.yml
- hosts: all
  become: true
  pre_tasks:
    - name: Install Updates (CentOS)
      yum:
        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
      when: ansible_distribution == "Ubuntu"
    - name: install apache and php for CentOS Servers
      yum:
        name:
          - httpd
          - php
        state: latest
      when: ansible_distribution == "CentOS"
```

The terminal window includes standard nano editor navigation keys at the bottom: Help, Write Out, Where Is, Cut, Execute, Location, Exit, Read File, Replace, Paste, Justify, and Go To Line.

Figure 1.3.2 - Modification of the site.yml with new commands. (Created a new version of site.yml for new modifications)

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.

```

seruelas@workstation: ~/CPE232_Act6_Seruelas
BECOME password:
PLAY [all] ****
TASK [Gathering Facts] ****
ok: [server2]
ok: [server1]
ok: [centos]

TASK [Install Updates (CentOS)] ****
skipping: [server1]
skipping: [server2]
ok: [centos]

TASK [Install Updates (Ubuntu)] ****
skipping: [centos]
changed: [server1]
changed: [server2]

PLAY [web_servers] ****
TASK [Gathering Facts] ****
ok: [server1]
ok: [centos]

TASK [install apache and php for Ubuntu Servers] ****
(skipping: [centos]
ok: [server1]

TASK [install apache and php for CentOS Servers] ****
skipping: [server1]
ok: [centos]

PLAY RECAP ****
centos           : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2        : rescued=0   ignored=0
server1          : ok=4    changed=1    unreachable=0    failed=0    s
kipped=2        : rescued=0   ignored=0
server2          : ok=2    changed=1    unreachable=0    failed=0    s
kipped=1        : rescued=0   ignored=0

```

Figure 1.3.3 - By executing the modified site.yml, it was able to perform the additional commands for the playbook, while first executing the pre_tasks before executing the main tasks of the playbook. In the web_server hosts, it was able to only execute the commands for the remote hosts that are a part of the web_server group, hence only executing its tasks for the remote hosts, server1 and centos.

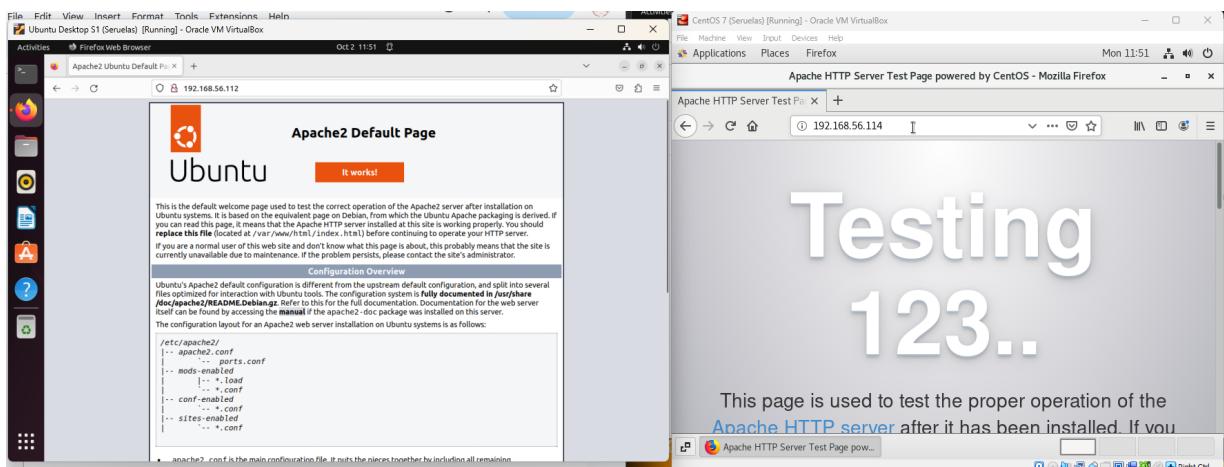


Figure 1.3.4 - Verification of the installation of apache/httpd in server1 (Ubuntu) and centos (CentOs 7)

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:

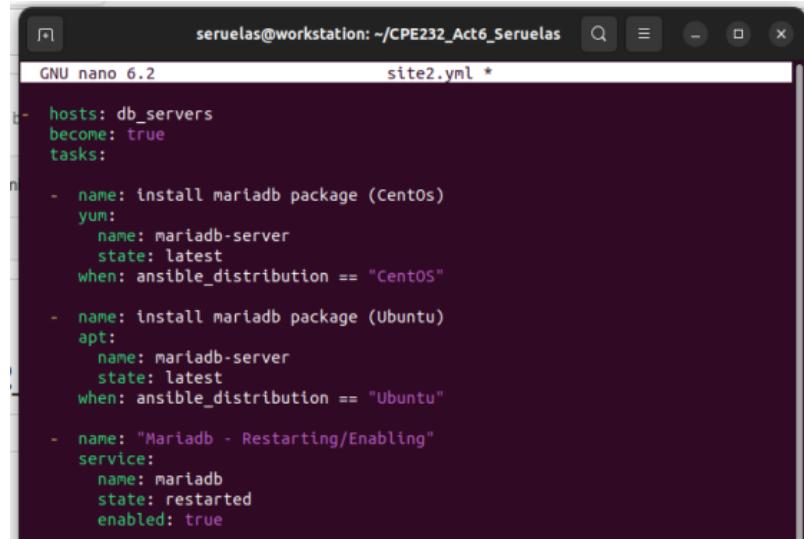
    - name: install mariadb package (CentOS)
      yum:
        name: mariadb-server
        state: latest
      when: ansible_distribution == "CentOS"

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

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

Figure 1.4.1 - Additional commands including the group hosts, db_servers.

Make sure to save the file and exit.



```

seruelas@workstation: ~/CPE232_Act6_Seruelas
GNU nano 6.2          site2.yml *

- 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
  
```

Figure 1.4.2 - Modification of the site.yml by including the db_servers with a new installation of packages. (Creation of site2.yml as another version of site.yml).

Changed the arrangement of tasks as well so that both ansible distributions will have mariadb installed before their services are enabled and restarted.

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

```
PLAY [db_servers] *****
TASK [Gathering Facts] *****
ok: [server2]
ok: [centos]

TASK [Install mariadb package (CentOS)] *****
skipping: [server2]
ok: [centos]

TASK [Install mariadb package (Ubuntu)] *****
skipping: [centos]
changed: [server2]

TASK [Mariadb - Restarting/Enabling] *****
changed: [server2]
changed: [centos]

PLAY RECAP *****
centos              : ok=7    changed=1    unreachable=0    failed=0    s
kipped=3  rescued=0  ignored=0
server1             : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2  rescued=0  ignored=0
server2             : ok=5    changed=2    unreachable=0    failed=0    s
kipped=2  rescued=0  ignored=0
```

Figure 1.4.3 - Executing the modified site.yml allowed us to install mariadb into the db_server remote hosts, allowing us to also enable and restart their services.

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.

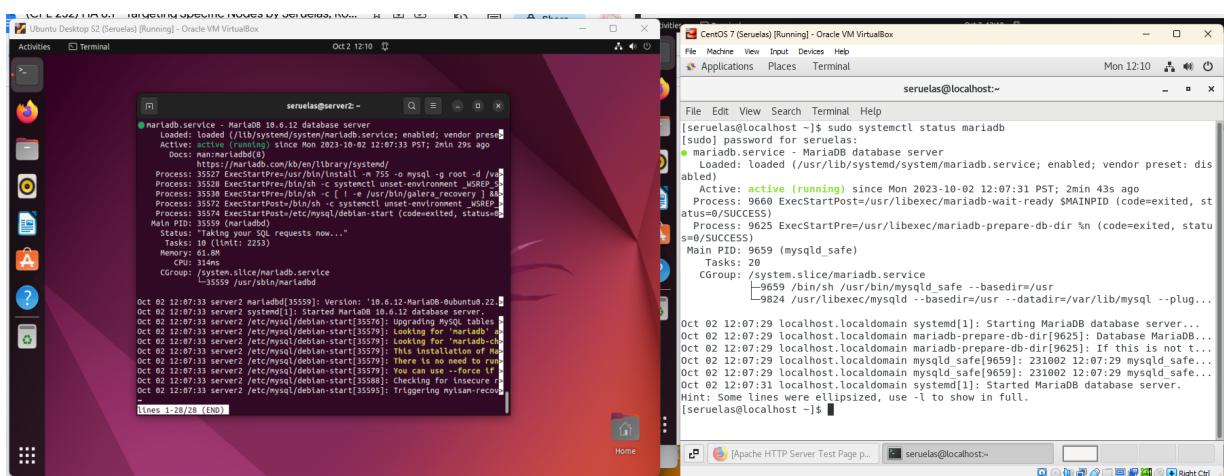


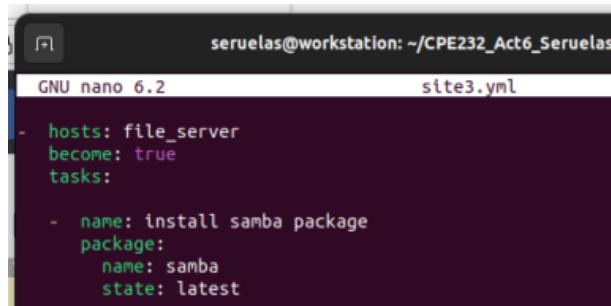
Figure 1.5.1 - Verification of the status of mariadb in the db_server remote hosts. We were able to verify that the service is running and installed in both remote hosts.

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
```

Figure 1.6.1 - Additional commands for site.yml, adding commands for file_server remote hosts.

Make sure to save the file and exit.



```
seruelas@workstation: ~/CPE232_Act6_Seruelas
$ nano site3.yml
GNU nano 6.2                               site3.yml

- hosts: file_server
  become: true
  tasks:
    - name: install samba package
      package:
        name: samba
        state: latest
```

Figure 1.6.2 - Modification of the site.yml with the additional commands for the file_server remote hosts. (Creation of site3.yml for another version of site.yml)

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



```
PLAY [file_server] ****
TASK [Gathering Facts] ****
ok: [server2]

TASK [install samba package] ****
changed: [server2]

PLAY RECAP ****
centos          : ok=7    changed=1    unreachable=0    failed=0    s
kipped=3       rescued=0   ignored=0
server1         : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2       rescued=0   ignored=0
server2         : ok=7    changed=2    unreachable=0    failed=0    s
kipped=2       rescued=0   ignored=0
```

Figure 1.6.3 - Execution of the modified site.yml allowed us to install the samba package into the file_server remote host.

```
seruelas@server2:~$ sudo systemctl status smbd
● smbd.service - Samba SMB Daemon
   Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor preset: ▶
   Active: active (running) since Mon 2023-10-02 12:21:17 PST; 2min 7s ago
     Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
   Process: 38032 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile ▶
 Main PID: 38041 (smbd)
   Status: "smbd: ready to serve connections..."
      Tasks: 4 (limit: 2253)
     Memory: 15.5M
        CPU: 127ms
       CGroup: /system.slice/smbd.service
               ├─38041 /usr/sbin/smbd --foreground --no-process-group
               ├─38043 /usr/sbin/smbd --foreground --no-process-group
               ├─38044 /usr/sbin/smbd --foreground --no-process-group
               └─38045 /usr/lib/x86_64-linux-gnu/samba/bgqd --ready-signal->

Oct 02 12:21:17 server2 systemd[1]: Starting Samba SMB Daemon...
Oct 02 12:21:17 server2 update-apparmor-samba-profile[38035]: grep: /etc/apparm...
Oct 02 12:21:17 server2 update-apparmor-samba-profile[38038]: diff: /etc/apparm...
Oct 02 12:21:17 server2 systemd[1]: Started Samba SMB Daemon.
Lines 1-22/22 (END)
```

Figure 1.6.4 - Verification of the installation of the service samba in the server2 or the file_server remote host.

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"
```

Figure 2.1.1 - Inclusion of tags in the pre_tasks.

```
- 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"
```

Figure 2.1.2 - Inclusion of tags in the tasks of web_servers.

```
- 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
```

Figure 2.1.3 - Inclusion of tags in tasks of db_servers and file_servers

Make sure to save the file and exit.

```
GNU nano 6.2             site4.yml
...
- hosts: all
  become: true
  pre_tasks:
    - name: Install Updates (CentOS)
      tags: always
      yum:
        update_only: yes
        update_cache: yes
      when: ansible_distribution == "CentOS"
    - name: Install Updates (Ubuntu)
      tags: always
      apt:
        upgrade: dist
        update_cache: yes
      when: ansible_distribution == "Ubuntu"
- hosts: web_servers
  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
      yum:
        name:
          - httpd
          - php
        state: latest
      when: ansible_distribution == "CentOS"
- hosts: db_servers
  become: true
  tasks:
    - name: install mariadb package (CentOS)
      tags: centos,db,mariadb
      yum:
        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_server
  become: true
  tasks:
    - name: install samba package
      tags: samba
      package:
        name: samba
        state: latest
```

Figure 2.1.4 - Modification of site.yml of adding tags to each tasks of remote_hosts
(Creation of site4.yml for another version)

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

```
seruelas@workstation:~/CPE232_Act6_Seruelas$ ansible-playbook --ask-become-pass
site4.yml
BECOME password:

PLAY [all] ****
TASK [Gathering Facts] ****
ok: [server2]
ok: [server1]
ok: [centos]

TASK [Install Updates (CentOS)] ****
skipping: [server1]
skipping: [server2]
ok: [centos]

TASK [Install Updates (Ubuntu)] ****
skipping: [centos]
ok: [server2]
ok: [server1]

PLAY [web_servers] ****
TASK [Gathering Facts] ****
ok: [server1]
ok: [centos]

TASK [install apache and php for Ubuntu Servers] ****
skipping: [centos]
ok: [server1]

TASK [install apache and php for CentOS Servers] ****
skipping: [server1]
ok: [centos]

PLAY [db_servers] ****
TASK [Gathering Facts] ****
ok: [server2]
ok: [centos]

TASK [install mariadb package (CentOs)] ****
skipping: [server2]
ok: [centos]

TASK [install mariadb package (Ubuntu)] ****
skipping: [centos]
ok: [server2]

TASK [Mariadb - Restarting/Enabling] ****
changed: [centos]
changed: [server2]

PLAY [file_server] ****
TASK [Gathering Facts] ****
ok: [server2]

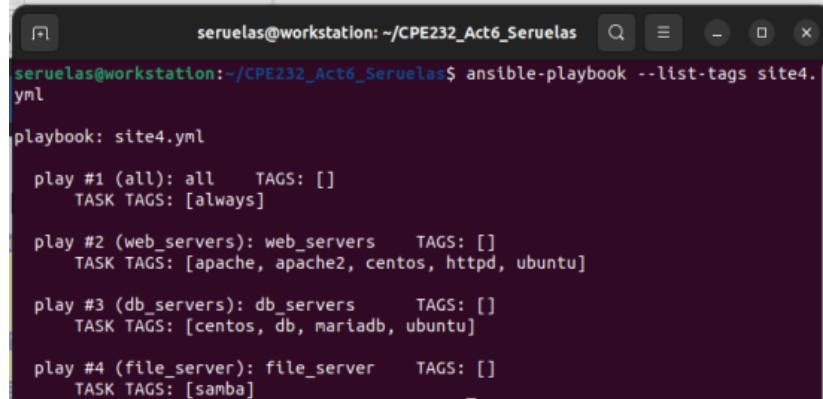
TASK [install samba package] ****
ok: [server2]

PLAY RECAP ****
centos      : ok=7    changed=1    unreachable=0    failed=0    s
kipped=3   rescued=0  ignored=0
server1     : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2   rescued=0  ignored=0
server2     : ok=7    changed=1    unreachable=0    failed=0    s
kipped=2   rescued=0  ignored=0
```

Figure 2.1.5 - Execution of the modified site.yml, having no changes after its modification.

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

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



```
seruelas@workstation:~/CPE232_Act6_Seruelas$ ansible-playbook --list-tags site4.yml
playbook: site4.yml

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

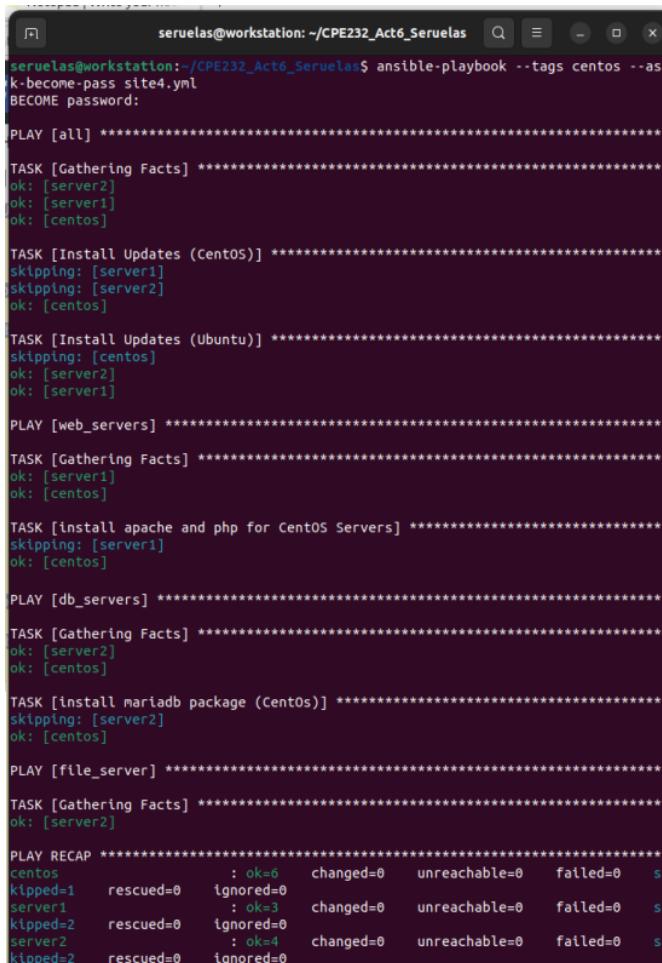
play #2 (web_servers): web_servers    TAGS: []
  TASK TAGS: [apache, apache2, centos, httpd, ubuntu]

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

play #4 (file_server): file_server    TAGS: []
  TASK TAGS: [samba]
```

Figure 2.2.1 - Executing the command listed the tags that are in the playbook.

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



```
seruelas@workstation:~/CPE232_Act6_Seruelas$ ansible-playbook --tags centos --ask-become-pass site4.yml
BECOME password:

PLAY [all] ****
TASK [Gathering Facts] ****
ok: [server2]
ok: [server1]
ok: [centos]

TASK [Install Updates (CentOS)] ****
skipping: [server1]
skipping: [server2]
ok: [centos]

TASK [Install Updates (Ubuntu)] ****
skipping: [centos]
ok: [server2]
ok: [server1]

PLAY [web_servers] ****
TASK [Gathering Facts] ****
ok: [server1]
ok: [centos]

TASK [install apache and php for CentOS Servers] ****
skipping: [server1]
ok: [centos]

PLAY [db_servers] ****
TASK [Gathering Facts] ****
ok: [server2]
ok: [centos]

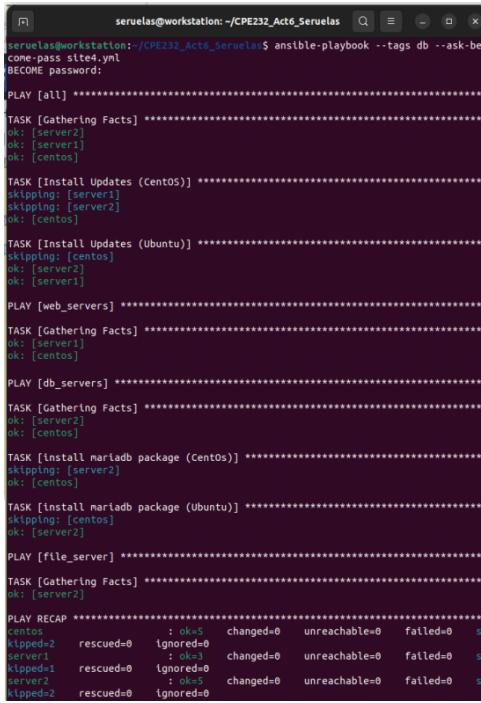
TASK [install mariadb package (CentOS)] ****
skipping: [server2]
ok: [centos]

PLAY [file_server] ****
TASK [Gathering Facts] ****
ok: [server2]

PLAY RECAP ****
centos             : ok=6    changed=0    unreachable=0    failed=0    s
kipped=1   rescued=0  ignored=0
server1            : ok=3    changed=0    unreachable=0    failed=0    s
kipped=2   rescued=0  ignored=0
server2            : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2   rescued=0  ignored=0
```

Figure 2.2.2 - Executing tasks only at specified tasks with the tag, centos.

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



```
seruelas@workstation:~/CPE232_Act6_Seruelas$ ansible-playbook --tags db --ask-become-pass site4.yml
BECOME password:

PLAY [all] *****
TASK [Gathering Facts] *****
ok: [server2]
ok: [server1]
ok: [centos]

TASK [Install Updates (CentOS)] *****
skipping: [server1]
skipping: [server2]
ok: [centos]

TASK [Install Updates (Ubuntu)] *****
skipping: [centos]
ok: [server2]
ok: [server1]

PLAY [web_servers] *****
TASK [Gathering Facts] *****
ok: [server1]
ok: [centos]

PLAY [db_servers] *****
TASK [Gathering Facts] *****
ok: [server2]
ok: [centos]

TASK [Install mariadb package (CentOs)] *****
skipping: [server2]
ok: [centos]

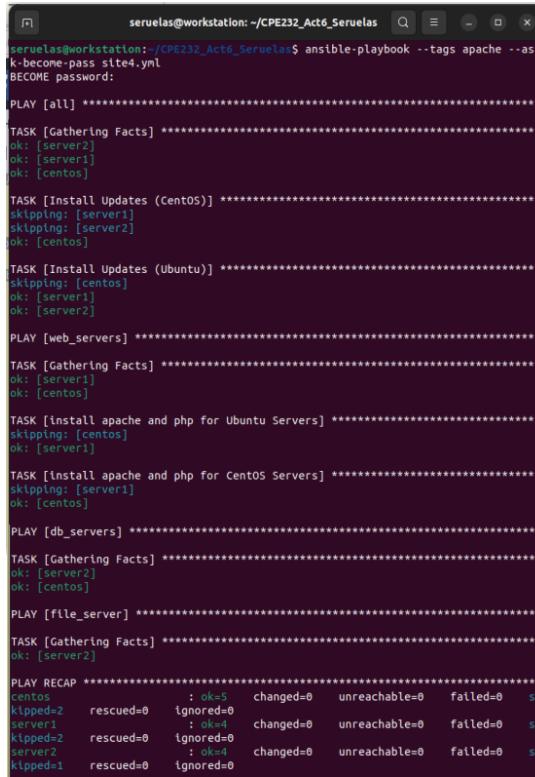
TASK [Install mariadb package (Ubuntu)] *****
skipping: [centos]
ok: [server2]

PLAY [file_server] *****
TASK [Gathering Facts] *****
ok: [server2]

PLAY RECAP *****
centos      : ok=5    changed=0    unreachable=0    failed=0    s
k1ped=2   rescued=0  ignored=0
server1     : ok=3    changed=0    unreachable=0    failed=0    s
k1ped=1   rescued=0  ignored=0
server2     : ok=5    changed=0    unreachable=0    failed=0    s
k1ped=2   rescued=0  ignored=0
```

Figure 2.2.3 - Executing tasks only at specified tasks with the tag, db

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



```
seruelas@workstation:~/CPE232_Act6_Seruelas$ ansible-playbook --tags apache --ask-become-pass site4.yml
BECOME password:

PLAY [all] *****
TASK [Gathering Facts] *****
ok: [server2]
ok: [server1]
ok: [centos]

TASK [Install Updates (CentOS)] *****
skipping: [server1]
skipping: [server2]
ok: [centos]

TASK [Install Updates (Ubuntu)] *****
skipping: [centos]
ok: [server1]
ok: [server2]

PLAY [web_servers] *****
TASK [Gathering Facts] *****
ok: [server1]
ok: [centos]

TASK [Install apache and php for Ubuntu Servers] *****
skipping: [centos]
ok: [server1]

TASK [Install apache and php for CentOS Servers] *****
skipping: [server1]
ok: [centos]

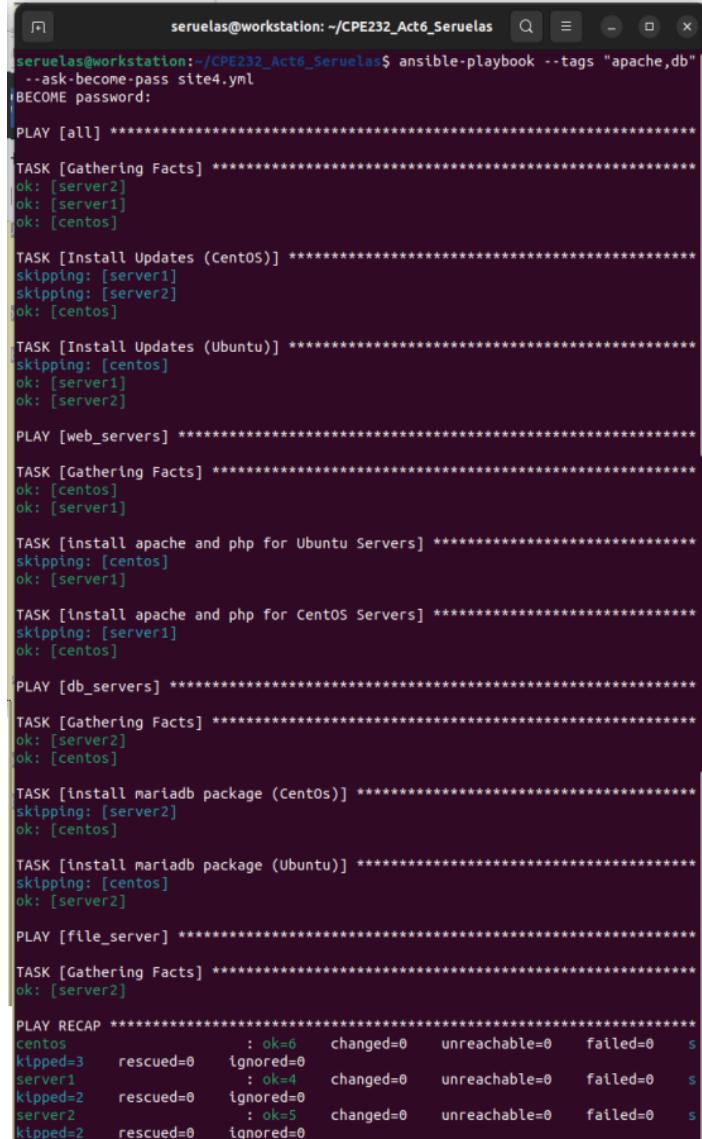
PLAY [db_servers] *****
TASK [Gathering Facts] *****
ok: [server2]
ok: [centos]

PLAY [file_server] *****
TASK [Gathering Facts] *****
ok: [server2]

PLAY RECAP *****
centos      : ok=5    changed=0    unreachable=0    failed=0    s
k1ped=2   rescued=0  ignored=0
server1     : ok=4    changed=0    unreachable=0    failed=0    s
k1ped=2   rescued=0  ignored=0
server2     : ok=4    changed=0    unreachable=0    failed=0    s
k1ped=1   rescued=0  ignored=0
```

Figure 2.2.4 - Executing tasks only at specified tasks with the tag, apache

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



```
seruelas@workstation:~/CPE232_Act6_Seruelas$ ansible-playbook --tags "apache,db" --ask-become-pass site4.yml
BECOME password:

PLAY [all] ****
TASK [Gathering Facts] ****
ok: [server2]
ok: [server1]
ok: [centos]

TASK [Install Updates (CentOS)] ****
skipping: [server1]
skipping: [server2]
ok: [centos]

TASK [Install Updates (Ubuntu)] ****
skipping: [centos]
ok: [server1]
ok: [server2]

PLAY [web_servers] ****
TASK [Gathering Facts] ****
ok: [centos]
ok: [server1]

TASK [install apache and php for Ubuntu Servers] ****
skipping: [centos]
ok: [server1]

TASK [install apache and php for CentOS Servers] ****
skipping: [server1]
ok: [centos]

PLAY [db_servers] ****
TASK [Gathering Facts] ****
ok: [server2]
ok: [centos]

TASK [install mariadb package (CentOs)] ****
skipping: [server2]
ok: [centos]

TASK [install mariadb package (Ubuntu)] ****
skipping: [centos]
ok: [server2]

PLAY [file_server] ****
TASK [Gathering Facts] ****
ok: [server2]

PLAY RECAP ****
centos      : ok=6    changed=0    unreachable=0    failed=0    s
kipped=3   rescued=0   ignored=0
server1     : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2   rescued=0   ignored=0
server2     : ok=5    changed=0    unreachable=0    failed=0    s
kipped=2   rescued=0   ignored=0
```

Figure 2.2.5 - Executing tasks only at specified tasks with the tags, apache and db.

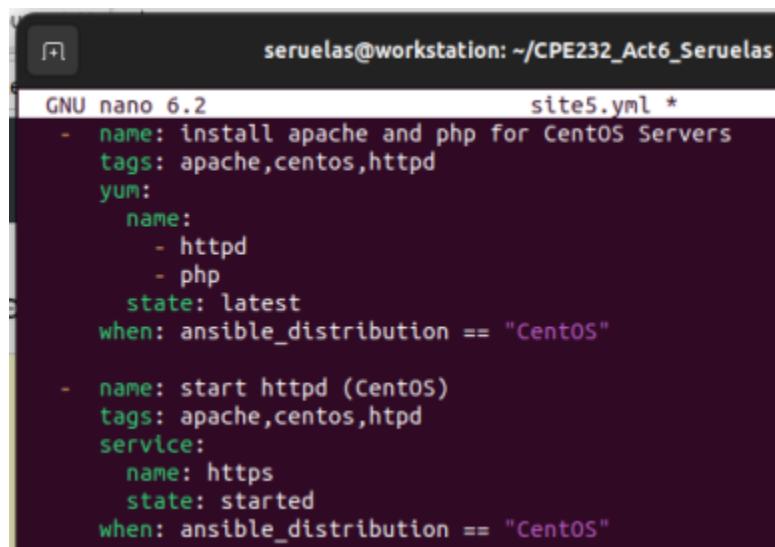
Task 3: Managing Services

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
  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 - Additional task for CentOS server, that will start the httpd server.



The screenshot shows a terminal window titled "seruelas@workstation: ~/CPE232_Act6_Seruelas". The command "GNU nano 6.2" is displayed at the top. The file "site5.yml" is being edited, containing the following YAML code:

```
site5.yml *
- name: install apache and php for CentOS Servers
  tags: apache,centos,httpd
  yum:
    name:
      - httpd
      - php
    state: latest
  when: ansible_distribution == "CentOS"

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

Figure 3.1.2 - Modification of the site.yml with a new task that will start the httpd server on the centos.

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 - A task that will enable and restart the service of mariadb, located in the db_servers tasks.

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.

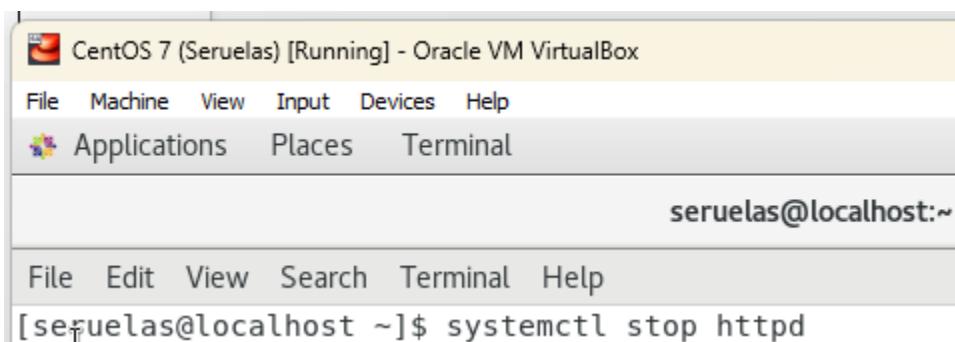


Figure 3.2.1 - Execution of a command that will stop the httpd service.

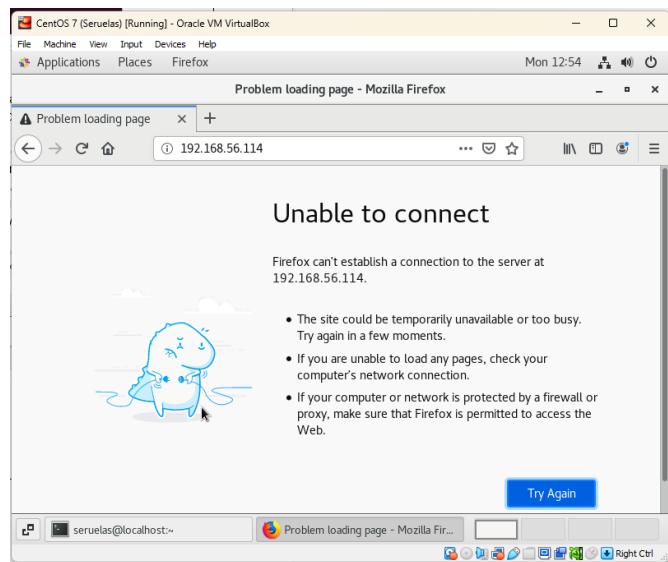


Figure 3.2.2 - Verification of the stopped service via browser.

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.

A screenshot of a terminal window on a workstation. The title bar says "seruelas@workstation: ~/CPE232_Act6_Seruelas". The terminal window displays the content of a file named "site5.yml":

```
GNU nano 6.2          site5.yml *
- name: start httpd (CentOS)
  tags: apache,centos,httpd
  service:
    name: httpd
    state: started
    enabled: true
  when: ansible_distribution == "CentOS"
```

Figure 3.3.1 - Adding the additional *enabled: true* command, allowing to enable the service every playbook run.

```
seruelas@workstation:~/CPE232_Act6_Seruelas$ ansible-playbook --tags "apache,centos,httpd" --ask-become-pass site5.yml
BECOME password:

PLAY [all] ****
TASK [Gathering Facts] ****
ok: [server2]
ok: [server1]
ok: [centos]

TASK [Install Updates (CentOS)] ****
skipping: [server1]
skipping: [server2]
ok: [centos]

TASK [Install Updates (Ubuntu)] ****
skipping: [centos]
ok: [server1]
ok: [server2]

PLAY [web_servers] ****
TASK [Gathering Facts] ****
ok: [server1]
ok: [centos]

TASK [Install apache and php for Ubuntu Servers] ****
skipping: [centos]
ok: [server1]

TASK [install apache and php for CentOS Servers] ****
skipping: [server1]
ok: [centos]

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

PLAY [db_servers] ****
TASK [Gathering Facts] ****
ok: [server2]
ok: [centos]

TASK [install mariadb package (CentOs)] ****
skipping: [server2]
ok: [centos]

PLAY [file_server] ****
TASK [Gathering Facts] ****
ok: [server2]

PLAY RECAP ****
centos      : ok=7    changed=1    unreachable=0    failed=0    s
kippeD=2   rescued=0   ignored=0
server1     : ok=4    changed=0    unreachable=0    failed=0    s
kippeD=3   rescued=0   ignored=0
server2     : ok=4    changed=0    unreachable=0    failed=0    s
kippeD=2   rescued=0   ignored=0
```

Figure 3.3.2 - Execution of the playbook, executing those tasks with the tags “apache,centos,httpd”, allowing to specify the httpd task and skipping the others.

```

CentOS 7 (Seruelas) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal Mon 13:09
seruelas@localhost:~ seruelas@localhost:~ - x
File Edit View Search Terminal Help
[seruelas@localhost ~]$ sudo systemctl status httpd
[sudo] password for seruelas:
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2023-10-02 13:08:05 PST; 1min 15s ago
     Docs: man:httpd(8)
           man:apachectl(8)
   Main PID: 20726 (httpd)
      Status: "Total requests: 0; Current requests/sec: 0; Current traffic: 0 B/sec"
     Tasks: 6
    CGroup: /system.slice/httpd.service
            └─20726 /usr/sbin/httpd -DFOREGROUND
              ├─20732 /usr/sbin/httpd -DFOREGROUND
              ├─20733 /usr/sbin/httpd -DFOREGROUND
              ├─20734 /usr/sbin/httpd -DFOREGROUND
              ├─20735 /usr/sbin/httpd -DFOREGROUND
              └─20736 /usr/sbin/httpd -DFOREGROUND

Oct 02 13:07:41 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
Oct 02 13:08:05 localhost.localdomain httpd[20726]: AH00558: httpd: Could not reli...
Oct 02 13:08:05 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
[seruelas@localhost ~]$ 

```

Figure 3.3.3 - Verification of the httpd service running after playbook run.

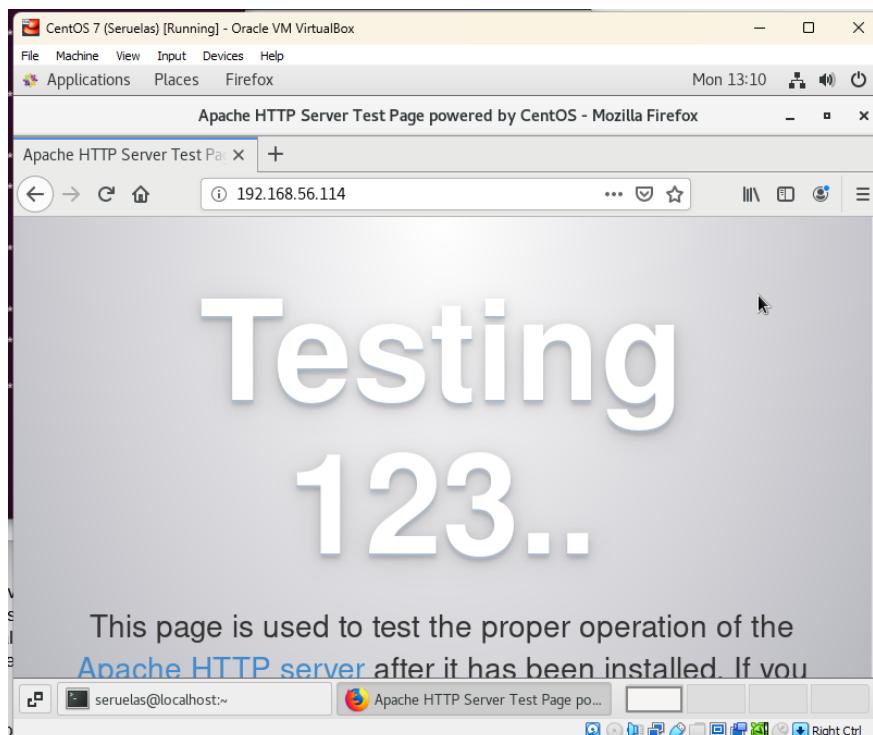
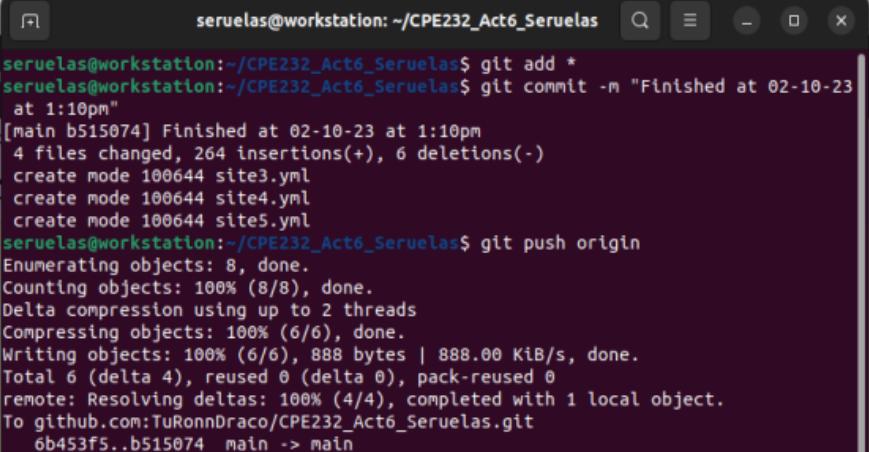


Figure 3.3.4 - Verification of the httpd service running after playbook run via browser.

4. Save all changes into github.



```
seruelas@workstation:~/CPE232_Act6_Seruelas$ git add *
seruelas@workstation:~/CPE232_Act6_Seruelas$ git commit -m "Finished at 02-10-23 at 1:10pm"
[main b515074] Finished at 02-10-23 at 1:10pm
  4 files changed, 264 insertions(+), 6 deletions(-)
  create mode 100644 site3.yml
  create mode 100644 site4.yml
  create mode 100644 site5.yml
seruelas@workstation:~/CPE232_Act6_Seruelas$ git push origin
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 2 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 888 bytes | 888.00 KiB/s, done.
Total 6 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:TuRonnDraco/CPE232_Act6_Seruelas.git
  6b453f5..b515074  main -> main
```

Figure 4.1.1 - Pushing and committing all changes into github repository.

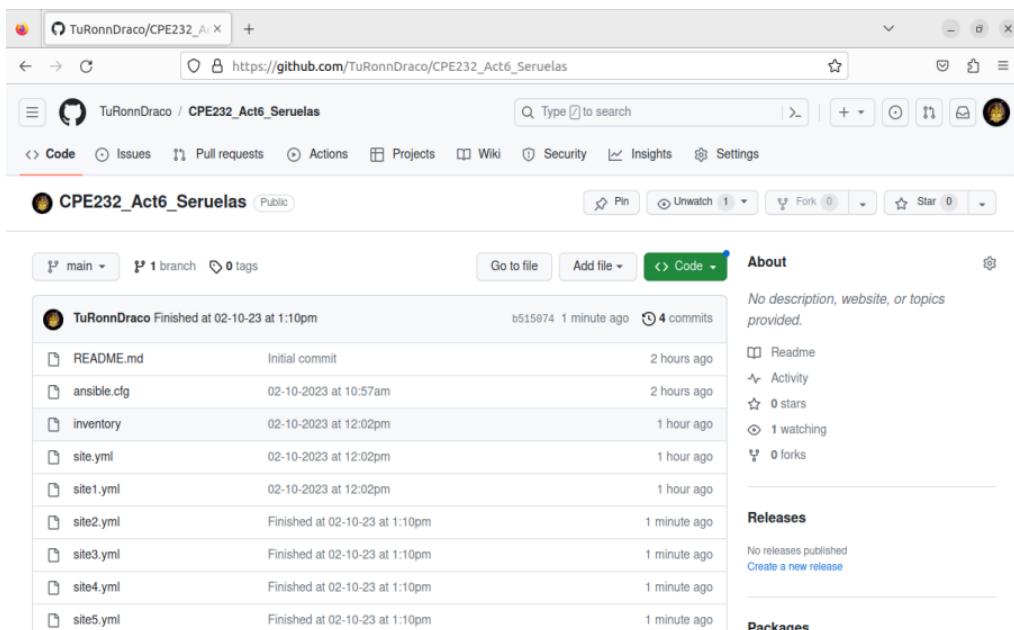


Figure 4.2.1 - Verification of all the committed changes in github.

https://github.com/TuRonnDraco/CPE232_Act6_Seruelas

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 that it allows us to efficiently manage our remote servers by allowing us to execute commands over multiple remote servers by specifying their respective group.
2. What is the importance of tags in playbooks?
 - The importance of using tags in our playbooks is that it gives the users the ability to run or execute a playbook and only execute those commands in which a tag is specified. As an example, if we were to specify a task with the tag, **remotedb**, it would allow us to execute the playbook while only executing the tasks with the specific tag.
3. Why do think some services need to be managed automatically in playbooks?
 - Some services are managed automatically by playbooks as it allows the user to execute and to become more efficient when it comes to finishing a certain task. By allowing an automation of management, it allows the servers to be ran automatically with the simplicity of executing a playbook, but also it allows the user to execute more commands and to become more efficient in completing their tasks.

Conclusion:

In conclusion, we were able to educate ourselves in using groups and tags for in a playbook for managing enterprise servers. We were able to learn on how we can implement groups in a repository, and how we can use them to specify the tasks that they are to execute. We were able to learn on how to use tags, in which it allows us to execute tasks in a playbook by specifying the tasks with certain tags, allowing us to only execute those tasks. To conclude this activity, we are able to implement the usage of groups and tasks in creating a more efficient and effective playbook for installing packages and execution of different commands.

Assessment:

Hands-On Rubric (2)						
Criteria	Ratings					Pts
Completeness This criterion specifies the analysis of the student of the task given.	5 pts Excellent Components of all tasks are present in the documentation and execution.	4 pts Good Components of most of the tasks are present in the documentation and execution.	3 pts Ok Components of half of the tasks are present in documentation and execution.	2 pts Poor Components some tasks are present in documentation and execution	1 pts Bad Components of all tasks lacks data in documentation and execution.	5 pts
Design This criterion measures the components and engineering of the Hands-on activity.	5 pts Excellent Design is robust and acceptable in the industry	4 pts Good Design is acceptable in the industry but can be improved.	3 pts Ok Design is a satisfactory level in the industry.	2 pts Poor Design is poorly architected and engineered needs improvement.	1 pts Bad Design is badly architected and engineered needs revisiting and rework.	5 pts
Documentation This criterion measures the context and completeness of artifacts of the activity.	5 pts Excellent The context of documentation is precise and understandable to readers.	4 pts Good The context of documentation is acceptable for readers.	3 pts Ok The documentation is satisfactory, has the main components needed, and grammar is acceptable.	2 pts Poor The documentation needs grammar checks but the content is complete.	1 pts Bad Documentation needs revisions from grammar to contexts.	5 pts
Total Points: 15						

"I affirm that I have not received or given any unauthorized help on this activity and that all work is my own."