

<b>Name: Jozshua Amiel Alonzo</b>	<b>Date Performed: October 6, 2022</b>
<b>Course/Section: CPE31S23</b>	<b>Date Submitted: October 7, 2022</b>
<b>Instructor: Dr. Jonathan Taylar</b>	<b>Semester and SY: 2022-2023</b>
<b>Activity 6: Targeting Specific Nodes and Managing Services</b>	
<b>1. Objectives:</b> 1.1 Individualize hosts 1.2 Apply tags in selecting plays to run 1.3 Managing Services from remote servers using playbooks	
<b>2. Discussion:</b>  <b>Github Link:</b> <a href="https://github.com/jozshua/HOA6_Alonzo.git">https://github.com/jozshua/HOA6_Alonzo.git</a>  <p>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.</p> <p>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.</p> <p><b>Requirement:</b>  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 <i>ssh-copy-id</i> to copy the public key to Server 3. Verify if you can successfully SSH to Server 3.</p>	
<b>Task 1: Targeting Specific Nodes</b>	
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 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
    dnf:
      name:
        - httpd
        - php
      state: latest
      when: ansible_distribution == "CentOS"

```

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

```

GNU nano 6.2
[web_servers]
192.168.56.112
192.168.56.116

[db_servers]
192.168.56.112

[file_servers]
192.168.56.112

```

Make sure to save the file and exit.

```

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.116]
ok: [192.168.56.112]

TASK [install apache and php for Ubuntu servers] *****
*
skipping: [192.168.56.116]
changed: [192.168.56.112]

TASK [install apache and php for CentOS servers] *****
*
skipping: [192.168.56.112]
changed: [192.168.56.116]

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

```

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.

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.

```

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.112]
ok: [192.168.56.116]

TASK [install updates (CentOS)] *****
*
skipping: [192.168.56.112]
ok: [192.168.56.116]

TASK [install updates (Ubuntu)] *****
*
skipping: [192.168.56.116]
changed: [192.168.56.112]

PLAY [web_servers] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.116]
ok: [192.168.56.112]

TASK [install apache and php for Ubuntu servers] *****
*
skipping: [192.168.56.116]
ok: [192.168.56.112]

TASK [install apache and php for CentOS servers] *****
*
skipping: [192.168.56.112]
ok: [192.168.56.116]

PLAY RECAP *****
*
192.168.56.112      : ok=4    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.116      : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0

```

It successfully executed the 4 tasks and it has 1 changed state. However, there are 2 skipped states because it is a task for other servers.

- 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"

```

Make sure to save the file and exit.

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

```

jozshua@jozshua-VirtualBox:~/cpe_ansible_Act6$ ansible-playbook --ask-become-pass site.yml
BECOME password:

PLAY [db_servers] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.112]

TASK [install mariadb package (Ubuntu)] *****
*
changed: [192.168.56.112]

TASK [Mariadb- Restarting/Enabling] *****
*
changed: [192.168.56.112]

PLAY RECAP *****
*
192.168.56.112      : ok=3    changed=2    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0

```

It successfully executed the 3 tasks and it has 2 changed state. I used Ubuntu server for this db\_server.

- 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.

```
jozshua@Server1-VirtualBox:~$ systemctl status mariadb
● mariadb.service - MariaDB 10.6.7 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-10-06 10:59:01 PST; 4min 13s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 51852 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/lib/mysql (code=exited, status=0/SUCCESS)
   Process: 51853 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_MODE (code=exited, status=0/SUCCESS)
   Process: 51855 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && mv /usr/bin/galera_recovery_recovery.sh /tmp/galera_recovery_recovery.sh; systemctl start mariadb (code=exited, status=0/SUCCESS)
   Process: 51895 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_MODE (code=exited, status=0/SUCCESS)
   Process: 51897 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
  Main PID: 51884 (mariabdb)
    Status: "Taking your SQL requests now..."
     Tasks: 8 (limit: 1080)
    Memory: 59.3M
       CPU: 490ms
    CGroup: /system.slice/mariadb.service
            └─51884 /usr/sbin/mariabdb

Oct 06 10:59:01 Server1-VirtualBox mariabdb[51884]: Version: '10.6.7-MariaDB-2'
Oct 06 10:59:01 Server1-VirtualBox systemd[1]: Started MariaDB 10.6.7 database server.
Oct 06 10:59:01 Server1-VirtualBox /etc/mysql/debian-start[51899]: Upgrading MySQL to use new 'mysql_native_password' authentication mechanism from 'mysql_old_password' to 'mysql_native_password'.
Oct 06 10:59:01 Server1-VirtualBox /etc/mysql/debian-start[51902]: Looking for a valid user password in /etc/mysql/debian_init.
Oct 06 10:59:01 Server1-VirtualBox /etc/mysql/debian-start[51902]: Looking for a valid user password in /etc/mysql/debian_init.
Oct 06 10:59:01 Server1-VirtualBox /etc/mysql/debian-start[51902]: This installation is for secure server only (no anonymous users, no salts etc.)
Oct 06 10:59:01 Server1-VirtualBox /etc/mysql/debian-start[51902]: There is no password for the anonymous user by default.
Oct 06 10:59:01 Server1-VirtualBox /etc/mysql/debian-start[51902]: You can use the 'mysql_native_password' authentication mechanism for MySQL 5.7+ servers.
Oct 06 10:59:01 Server1-VirtualBox /etc/mysql/debian-start[51914]: Checking for updates to the 'mysql_native_password' authentication mechanism from 'mysql_old_password' to 'mysql_native_password'.
Oct 06 10:59:01 Server1-VirtualBox /etc/mysql/debian-start[51918]: Triggering a restart by running 'systemctl restart mariadb.service'.
```

```
[jozshua@localhost ~]$ systemctl status mariadb
Unit mariadb.service could not be found.
```

Describe the output. It is active now and currently running for Ubuntu and for the CentOS could not found because the db server has Ubuntu ip address only.

- 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.

```
jozshua@jozshua-VirtualBox:~/cpe_ansible_Act6$ ansible-playbook --ask-become-pass site.yml
BECOME password:

PLAY [file_servers] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.112]

TASK [install samba package] *****
*
changed: [192.168.56.112]

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

It has 2 successfully executed tasks and 1 changed state. I also used only 1 server here because it is the file\_server.

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"
```

```
GNU nano 6.2                                site.yml *
--
- hosts: db_servers
  become: true
  tasks:
    - name: install mariadb package (Ubuntu)
      tags: db,mariadb,ubuntu
      apt:
        name: mariadb
        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.

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

TASK [Gathering Facts] *****
*
ok: [192.168.56.108]

TASK [install updates (Ubuntu)] *****
*
changed: [192.168.56.108]

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

```

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.101]

TASK [install updates (CentOS)] *****
*
ok: [192.168.56.101]

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

```

It has a total of 4 successfully executed tasks and 1 changed state with the combination of the Ubuntu and CentOS servers.

```

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.101]

TASK [install apache and php for CentOS servers] *****
*
ok: [192.168.56.101]

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

```

```

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.108]

TASK [install apache and php for Ubuntu servers] *****
*
ok: [192.168.56.108]

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

```

It has a total of 4 successfully executed tasks for the web\_servers with the combination of Ubuntu and CentOS servers.

```

PLAY [db_servers] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.108]

TASK [install mariadb package (Ubuntu)] *****
*
changed: [192.168.56.108]

PLAY [file_servers] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.108]

TASK [install samba package] *****
*
changed: [192.168.56.108]

PLAY RECAP *****
*
192.168.56.108      : ok=4    changed=2    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0

```

I used only Ubuntu server for the db\_server group. It has 4 successfully executed tasks and 2 changed states on it.

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

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

```

jozshua@workstation-VirtualBox:~/CPE232_Jozshua/cpe_ansible_Alonzo$ ansible-playbook --list-tags site.yml

playbook: site.yml

play #1 (db_servers): db_servers    TAGS: []
TASK TAGS: [db, mariadb, ubuntu]

play #2 (file_servers): file_servers TAGS: []
TASK TAGS: [samba]

```

It shows the 2 groups with their personal tags in each for the tasks on each server.

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

```

PLAY [db_servers] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.108]

PLAY [file_servers] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.108]

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

```

It shows the 2 successfully executed tasks from the db\_servers group with the assigned Ubuntu server on it.

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

```

PLAY [db_servers] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.108]

TASK [install mariadb package (Ubuntu)] *****
*
ok: [192.168.56.108]

PLAY [file_servers] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.108]

PLAY RECAP *****
192.168.56.108      : ok=3    changed=0    unreachable=0    failed=0
skipped=0         rescued=0    ignored=0

```

It displays the 3 successfully executed tasks from the db\_servers and file\_servers group wherein the Ubuntu server is assigned on it.

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

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

TASK [Gathering Facts] *****
*
ok: [192.168.56.108]

PLAY [file_servers] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.108]

PLAY RECAP *****
192.168.56.108      : ok=2    changed=0    unreachable=0    failed=0
skipped=0         rescued=0    ignored=0
```

It displays the 2 successfully executed tasks. One task per each group from the db\_server and file\_server groups.

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

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

TASK [Gathering Facts] *****
*
ok: [192.168.56.108]

TASK [install mariadb package (Ubuntu)] *****
*
ok: [192.168.56.108]

PLAY [file_servers] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.108]

PLAY RECAP *****
192.168.56.108      : ok=3    changed=0    unreachable=0    failed=0
skipped=0         rescued=0    ignored=0
```

It displays the 3 successfully executed tasks from both groups which are one from the db\_servers and one from the file\_servers also. It also displays the package installed from the Ubuntu server.

## Task 3: Managing Services

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

```

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

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

- hosts: file_servers
  become: true
  tasks:

    - 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.

```

PLAY [db_servers] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.101]

TASK [install apache and php for CentOS servers] *****
*
ok: [192.168.56.101]

PLAY [file_servers] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.101]

TASK [start httpd (CentOS)] *****
*
changed: [192.168.56.101]

PLAY RECAP *****
*
192.168.56.101      : ok=4    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0

```

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

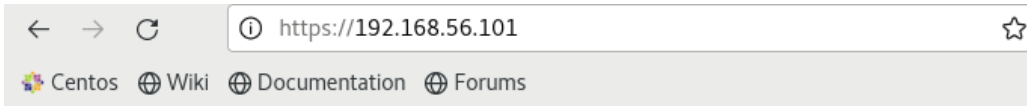
Figure 3.1.2

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.



```
[jozshua@localhost cpe_ansible_Alonzo]$ sudo systemctl stop httpd
[sudo] password for jozshua:
[jozshua@localhost cpe_ansible_Alonzo]$
```



## Unable to connect

Firefox can't establish a connection to the server at 192.168.56.101.

- 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.

```
jozshua@workstation-VirtualBox: ~/CPE232_Jozshua/cpe_ansible_Alonzo$ ansible-playbook --ask-become-pass site.yml
BECOME password:

PLAY [db_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.101]

TASK [install mariadb package (CentOS)] *****
ok: [192.168.56.101]

TASK [Mariadb- Restarting/Enabling] *****
changed: [192.168.56.101]

PLAY [file_servers] *****

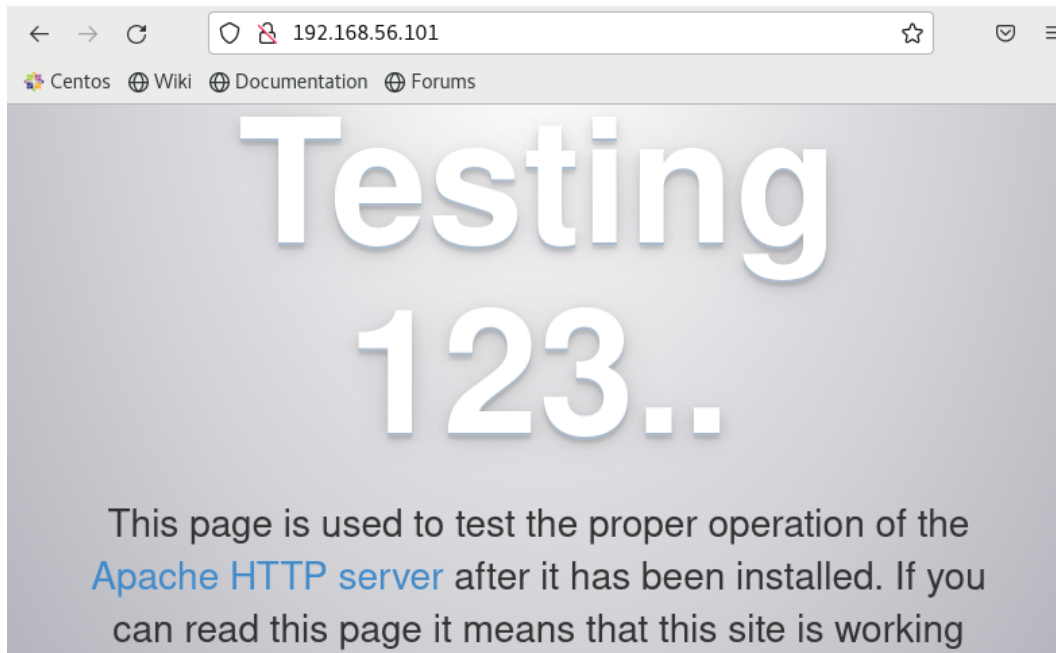
TASK [Gathering Facts] *****
ok: [192.168.56.101]

TASK [install samba package] *****
changed: [192.168.56.101]

PLAY RECAP *****
192.168.56.101 : ok=5 changed=2 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

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.



After editing the site.yml file with the command ***enabled: true*** on it. I observed that it automatically enabled the httpd which we disabled before. And I checked it on the web browser from the CentOS server from being unable to connect it now has displayed that HTTP is now installed on the CentOS server. Also, the Playbook shows 5 successful tasks executed on it and 2 changed states on it.

### Reflections:

Answer the following:

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

I think to call it a one hosts when you are going to give a task for them instead of calling each one of them servers per task for the playbooks.

2. What is the importance of tags in playbooks?

The tags were important to the playbooks because it makes them organized and simplified the tasks executed in the playbook.

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

I think it is because some services are not able to automatically open. That's why we need to do a step-by-step on it to be able to open or enable it, which is automatically able to manage by the playbook.