Name: Hermitaño, Johnny C.	Date Performed: Oct 6, 2022
Course/Section: CPE31S23	Date Submitted: Oct 8, 2022
Instructor: Engr. Jonathan 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.

Task 1: Targeting Specific Nodes

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

```
 GNU nano 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
      update_cache: yes
    when: ansible_distribution == "CentOS"
```

In this screenshot, shows a playbook that will install apache and php for both ubuntu and centos servers at the same time.

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]
192.168.56.105
192.168.56.115
192.168.56.106

[db_servers]
192.168.56.115
192.168.56.105

[file_servers]
192.168.56.106
```

This is the inventory I used for the entire activity.

Make sure to save the file and exit.

```
jhermitano@Workstation:~/SecondSemRepository$ ansible-playbook --ask-become-pas
s site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.105]
ok: [192.168.56.106]
ok: [192.168.56.115]

TASK [install apache and php for Ubuntu Servers] *******************************
*
skipping: [192.168.56.115]
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [install apache and php for CentOS Servers] *******************************
*
skipping: [192.168.56.105]
skipping: [192.168.56.106]
changed: [192.168.56.115]
```

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

In this screenshot, it shows the output for the playbook I run. The recap shows that the ip address 192.168.56.115 was changed, because the playbook just changed the server by installing apache and php into its system. You might be confused why the other servers aren't changed, it is because the other two servers I used are a server sitting for ages in my virtual machine and the centos server only which has the ip address of 192.168.56.115 is the one I just installed during this activity.

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:

Make sure to save the file and exit.

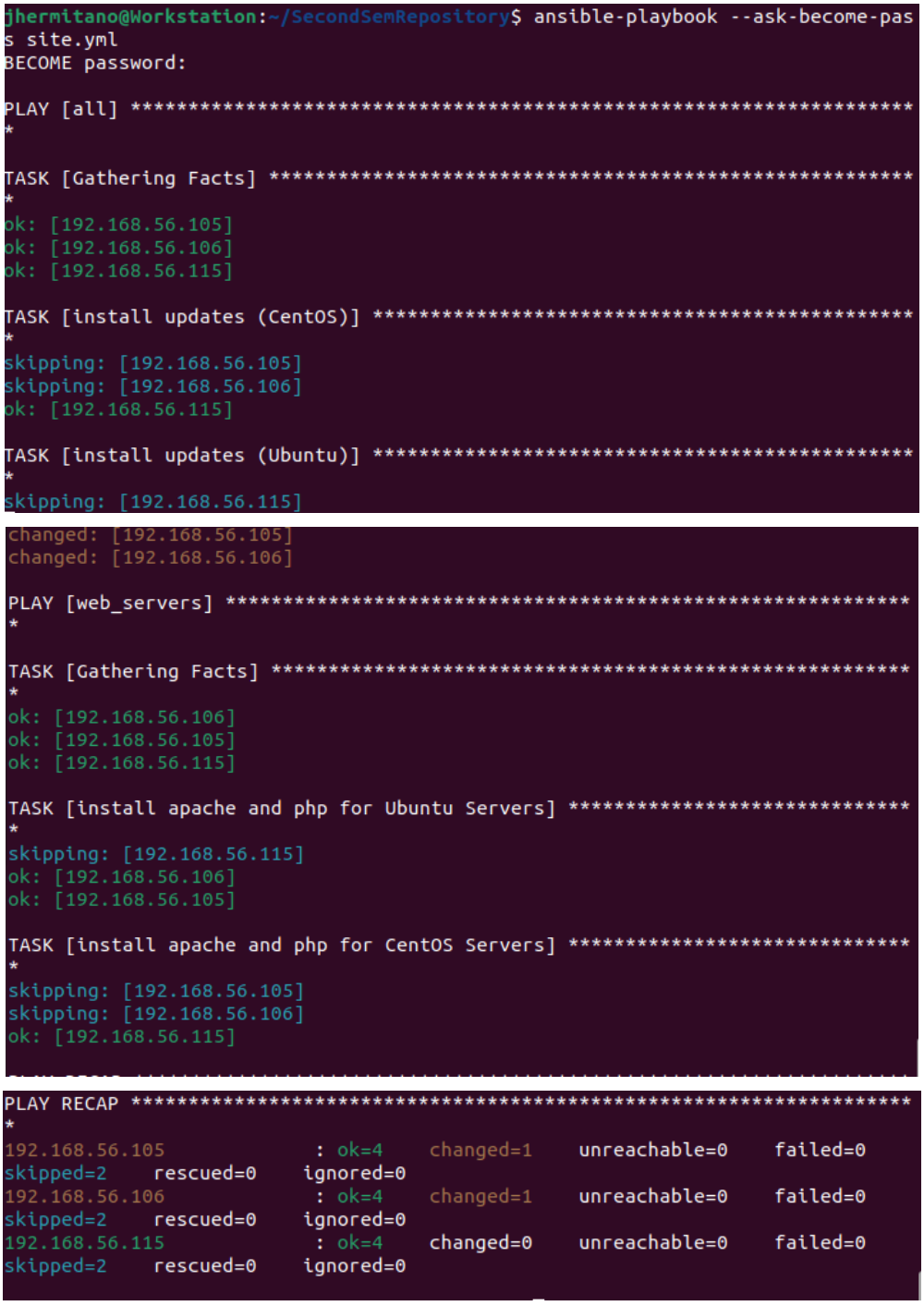
```
 GNU nano 6.2                                site.yml
---

- hosts: all
  become: true
  pre_tasks:

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

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

- hosts: web_servers
  become: true
  tasks:

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

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

In these screenshots shown above, we added/ changed some play in it. Like the pre-tasks added at the very top of the playbook that updates ubuntu and centos servers.

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.



In this screenshot above, is the output after running the playbook with the added play of updating the servers. The play recap shows that the playbook made changes on some servers and are successful.

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

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

- hosts: all
  become: true
  pre_tasks:

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

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

- hosts: web_servers
  become: true
  tasks:

  - name: install apache and php for Ubuntu Servers
    apt:
      name:
```

```
        - apache2
        - libapache2-mod-php
      state: latest
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: install apache and php for CentOS Servers
    dnf:
      name:
        - httpd
        - php
      state: latest
      update_cache: yes
    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: "Mariadb- Restarting/Enabling"
    service:
      name: mariadb
      state: restarted
      enabled: true
```

Make sure to save the file and exit.

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

```
jhermitano@Workstation:~/SecondSemRepository$ ansible-playbook --ask-become-pas
s site.yml
BECOME password:

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.106]
ok: [192.168.56.105]
ok: [192.168.56.115]

TASK [install updates (CentOS)] ***********************************************
*
skipping: [192.168.56.105]
skipping: [192.168.56.106]
ok: [192.168.56.115]

TASK [install updates (Ubuntu)] ***********************************************
*
skipping: [192.168.56.115]
ok: [192.168.56.106]
ok: [192.168.56.105]

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

```
TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.105]
ok: [192.168.56.106]
ok: [192.168.56.115]

TASK [install apache and php for Ubuntu Servers] ******************************
*
skipping: [192.168.56.115]
ok: [192.168.56.106]
ok: [192.168.56.105]

TASK [install apache and php for CentOS Servers] ******************************
*
skipping: [192.168.56.105]
skipping: [192.168.56.106]
ok: [192.168.56.115]

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.115]
```

```
TASK [install mariadb package (CentOS)] ************************************
*
ok: [192.168.56.115]

TASK [Mariadb- Restarting/Enabling] ****************************************
*
changed: [192.168.56.115]

PLAY RECAP *****************************************************************
*
192.168.56.105             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.106             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.115             : ok=7    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
```

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.

```
[jhermitano@localhost ~]$ systemctl status mariadb
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: dis
abled)
   Active: active (running) since Sat 2022-10-08 02:07:12 EDT; 3h 11min ago
  Process: 10783 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited, s
tatus=0/SUCCESS)
  Process: 10744 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, stat
us=0/SUCCESS)
 Main PID: 10782 (mysqld_safe)
    Tasks: 20
   CGroup: /system.slice/mariadb.service
           ├─10782 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
           └─10947 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --plu...

Oct 08 02:07:10 localhost.localdomain systemd[1]: Stopped MariaDB database server.
Oct 08 02:07:10 localhost.localdomain systemd[1]: Starting MariaDB database server...
Oct 08 02:07:10 localhost.localdomain mariadb-prepare-db-dir[10744]: Database MariaD...
Oct 08 02:07:10 localhost.localdomain mysqld_safe[10782]: 221008 02:07:10 mysqld_saf...
Oct 08 02:07:10 localhost.localdomain mysqld_safe[10782]: 221008 02:07:10 mysqld_saf...
Oct 08 02:07:12 localhost.localdomain systemd[1]: Started MariaDB database server.
Hint: Some lines were ellipsized, use -l to show in full.
[jhermitano@localhost ~]$
```

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.

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

- hosts: all
  become: true
  pre_tasks:

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

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

- hosts: web_servers
  become: true
  tasks:

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

  - name: install apache and php for CentOS Servers
    dnf:
      name:
        - httpd
        - php
      state: latest
      update_cache: yes
    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: "Mariadb- Restarting/Enabling"
    service:
      name: mariadb
      state: restarted
      enabled: true

- 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.

```
jhermitano@Workstation:~/SecondSemRepository$ ansible-playbook --ask-become-pas
s site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.106]
ok: [192.168.56.105]
ok: [192.168.56.115]

TASK [install updates (CentOS)] ************************************************
*
skipping: [192.168.56.105]
skipping: [192.168.56.106]
ok: [192.168.56.115]

TASK [install updates (Ubuntu)] ************************************************
*
skipping: [192.168.56.115]
ok: [192.168.56.106]
ok: [192.168.56.105]

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

```
TASK [Gathering Facts] ***********************************************************
*
ok: [192.168.56.105]
ok: [192.168.56.106]
ok: [192.168.56.115]

TASK [install apache and php for Ubuntu Servers] *****************************
*
skipping: [192.168.56.115]
ok: [192.168.56.106]
ok: [192.168.56.105]

TASK [install apache and php for CentOS Servers] *****************************
*
skipping: [192.168.56.105]
skipping: [192.168.56.106]
ok: [192.168.56.115]

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

TASK [Gathering Facts] ***********************************************************
*
ok: [192.168.56.115]
```

```
TASK [install mariadb package (CentOS)] ******************************************
*
ok: [192.168.56.115]

TASK [Mariadb- Restarting/Enabling] **********************************************
*
changed: [192.168.56.115]

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

TASK [Gathering Facts] ***********************************************************
*
ok: [192.168.56.106]

TASK [install samba package] *****************************************************
*
changed: [192.168.56.106]

PLAY RECAP ***********************************************************************
*
192.168.56.105             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.106             : ok=6    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.115             : ok=7    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
```

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
      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
      update_cache: yes
    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: "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
```

Make sure to save the file and exit.
Run the *site.yml* file and describe the result.

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

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

```
jhermitano@Workstation:~/SecondSemRepository$ ansible-playbook --list-tags site
.yml

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]

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

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

```
jhermitano@Workstation:~/SecondSemRepository$ ansible-playbook --tags centos --
ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.106]
ok: [192.168.56.105]
ok: [192.168.56.115]

TASK [install updates (CentOS)] ************************************************
*
skipping: [192.168.56.105]
skipping: [192.168.56.106]
ok: [192.168.56.115]

TASK [install updates (Ubuntu)] ************************************************
*
skipping: [192.168.56.115]
ok: [192.168.56.106]
ok: [192.168.56.105]

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

```
TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.105]
ok: [192.168.56.106]
ok: [192.168.56.115]

TASK [install apache and php for CentOS Servers] *******************************
*
skipping: [192.168.56.105]
skipping: [192.168.56.106]
ok: [192.168.56.115]

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.115]

TASK [install mariadb package (CentOS)] ****************************************
*
ok: [192.168.56.115]

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

```
TASK [Gathering Facts] ***************************************************
*
ok: [192.168.56.106]

PLAY RECAP ***************************************************************
*
192.168.56.105             : ok=3    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.106             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.115             : ok=6    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
```

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

```
jhermitano@Workstation:~/SecondSemRepository$ ansible-playbook --tags db --ask-
become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] ***************************************************
*
ok: [192.168.56.106]
ok: [192.168.56.105]
ok: [192.168.56.115]

TASK [install updates (CentOS)] ******************************************
*
skipping: [192.168.56.105]
skipping: [192.168.56.106]
ok: [192.168.56.115]

TASK [install updates (Ubuntu)] ******************************************
*
skipping: [192.168.56.115]
ok: [192.168.56.105]
ok: [192.168.56.106]

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

```
TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.105]
ok: [192.168.56.106]
ok: [192.168.56.115]

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.115]

TASK [install mariadb package (CentOS)] ****************************************
*
ok: [192.168.56.115]

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.106]
```

```
PLAY RECAP *********************************************************************
*
192.168.56.105             : ok=3    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.106             : ok=4    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.115             : ok=5    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
```

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

```
jhermitano@Workstation:~/SecondSemRepository$ ansible-playbook --tags apache --
ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.105]
ok: [192.168.56.106]
ok: [192.168.56.115]

TASK [install updates (CentOS)] ***********************************************
*
skipping: [192.168.56.105]
skipping: [192.168.56.106]
ok: [192.168.56.115]

TASK [install updates (Ubuntu)] ***********************************************
*
skipping: [192.168.56.115]
ok: [192.168.56.106]
ok: [192.168.56.105]

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.105]
ok: [192.168.56.106]
ok: [192.168.56.115]

TASK [install apache and php for Ubuntu Servers] ******************************
*
skipping: [192.168.56.115]
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [install apache and php for CentOS Servers] ******************************
*
skipping: [192.168.56.105]
skipping: [192.168.56.106]
ok: [192.168.56.115]

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.115]
```

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.106]

PLAY RECAP ********************************************************************
*
192.168.56.105             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.106             : ok=5    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.115             : ok=5    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
```

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

```
jhermitano@Workstation:~/SecondSemRepository$ ansible-playbook --tags "apache,d
b" --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.106]
ok: [192.168.56.105]
ok: [192.168.56.115]

TASK [install updates (CentOS)] ***********************************************
*
skipping: [192.168.56.105]
skipping: [192.168.56.106]
ok: [192.168.56.115]

TASK [install updates (Ubuntu)] ***********************************************
*
skipping: [192.168.56.115]
ok: [192.168.56.106]
ok: [192.168.56.105]

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

```
TASK [Gathering Facts] ***************************************************
*
ok: [192.168.56.115]
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [install apache and php for Ubuntu Servers] *****************************
*
skipping: [192.168.56.115]
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [install apache and php for CentOS Servers] *****************************
*
skipping: [192.168.56.105]
skipping: [192.168.56.106]
ok: [192.168.56.115]

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

TASK [Gathering Facts] ***************************************************
*
ok: [192.168.56.115]
```

```
TASK [install mariadb package (CentOS)] **************************************
*
ok: [192.168.56.115]

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

TASK [Gathering Facts] ***************************************************
*
ok: [192.168.56.106]

PLAY RECAP ***************************************************************
*
192.168.56.105             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.106             : ok=5    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.115             : ok=6    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
```

In this task, we added tags on some of the plays on the playbook. After editing the playbook; 2.1 tasks, shows/lists every tag that are on the playbook; 2.2 - 2.5 tasks, running the playbook depends on the tags. In conclusion, tags help you to run a certain play on the playbook without running the play you might not need.

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.

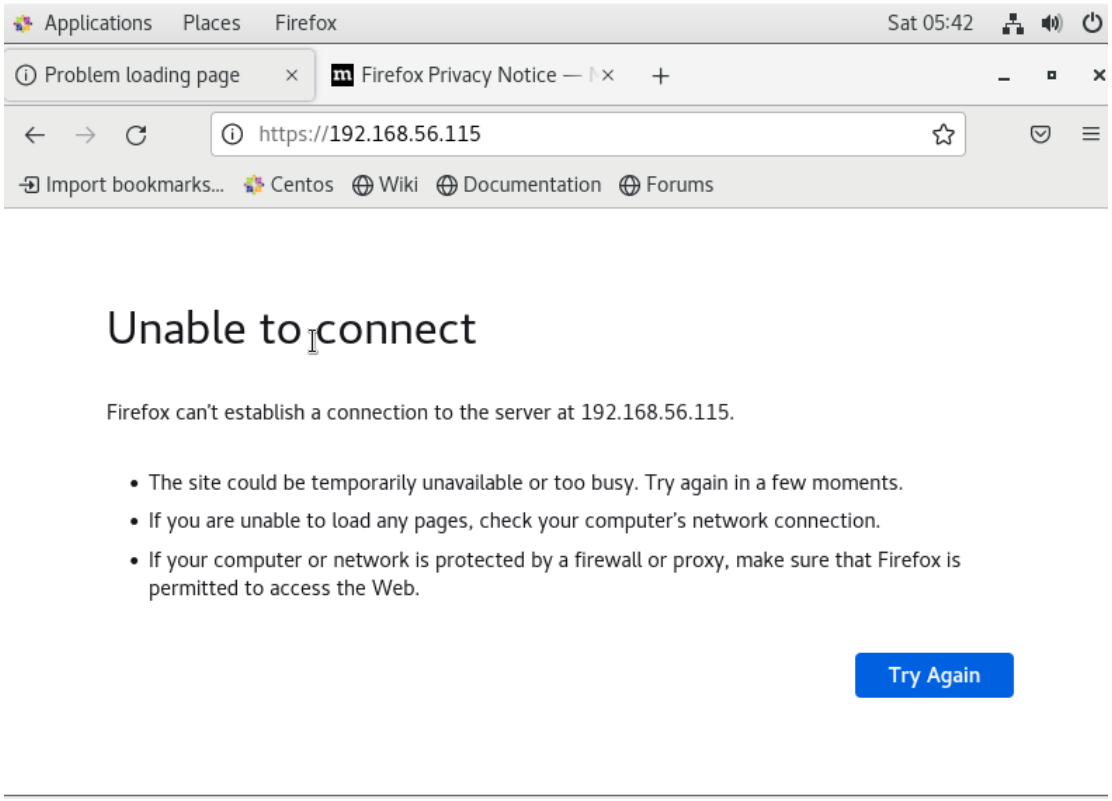
```
- hosts: db_servers
  become: true
  tasks:

  - 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.

```
[jhermitano@localhost ~]$ sudo systemctl stop httpd
[sudo] password for jhermitano:
[jhermitano@localhost ~]$
```

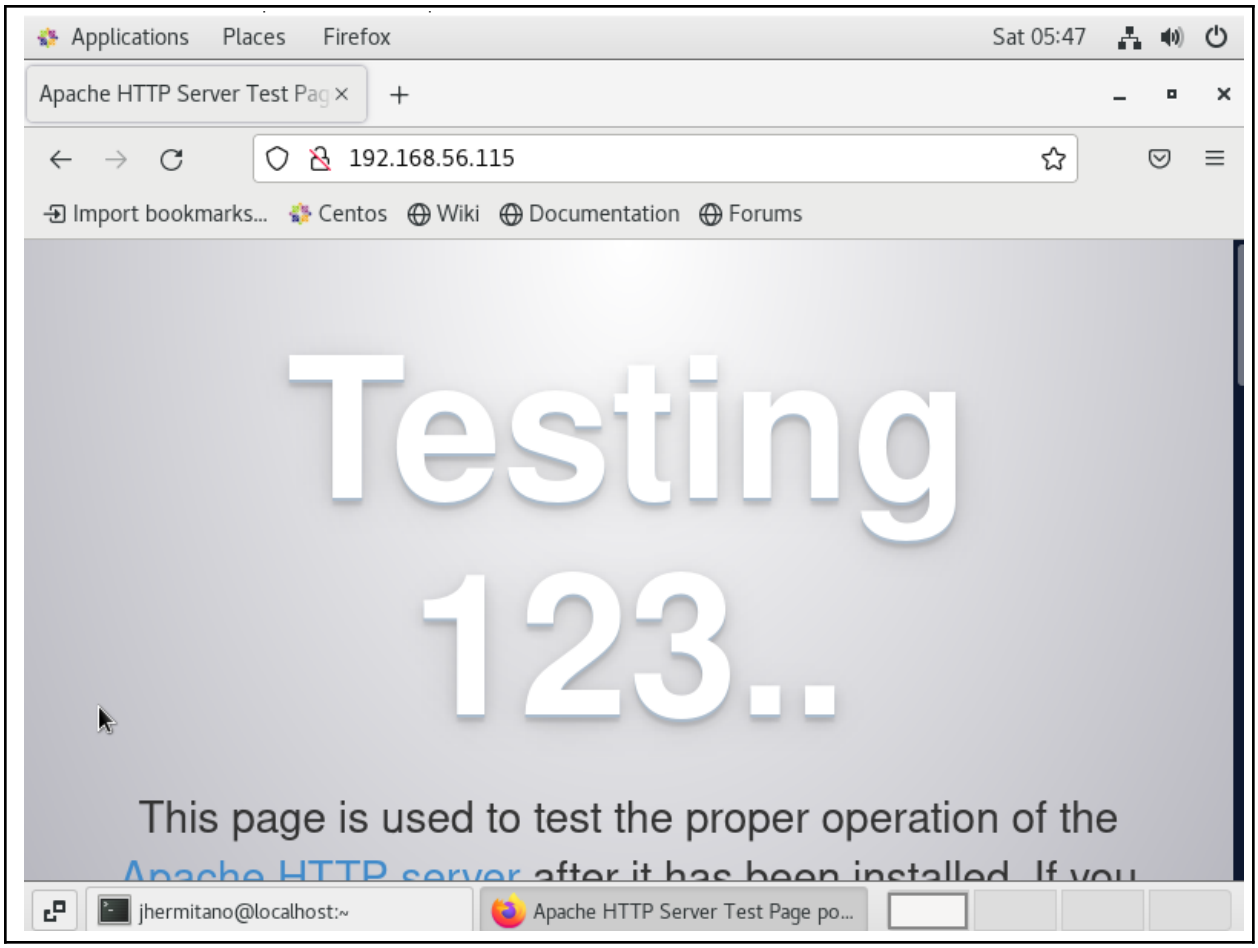


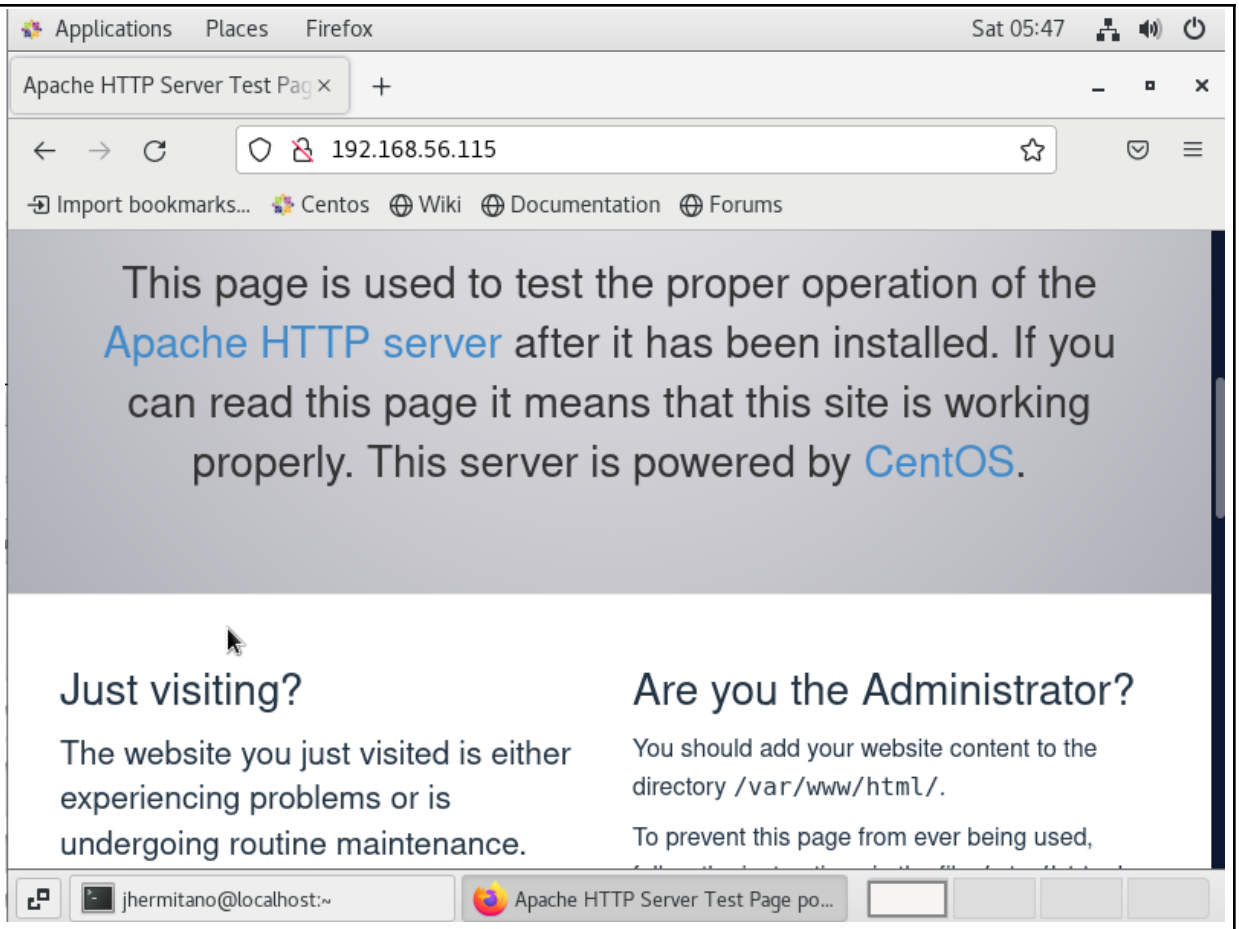
In this command, I stopped the currently running httpd to see the CentOS' httpd run automatically using playbook.

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.

```
TASK [start httpd (CentOS)] ****************************************************
*
skipping: [192.168.56.105]
skipping: [192.168.56.106]
changed: [192.168.56.115]
```


Applications
Places
Firefox
Sat 05:47
Apache HTTP Server Test Pag
192.168.56.115
Import bookmarks...
Centos
Wiki
Documentation
Forums
Testing
123..
This page is used to test the proper operation of the
jhermitano@localhost:~
Apache HTTP Server Test Page po...



You can see here that after running the command, httpd on centos server opens automatically.

Reflections:

In this activity, we learned to individualize hosts by putting remote servers into groups. Putting remote servers into groups will help you make changes to any group of servers you want without making changes to the other servers that doesn't need to be handled.

I also learned by applying tags on every playbook's play, it helps you to select plays you wanted to run without including other plays that aren't necessary on your project. *"Always" tags means it will be running whether you command it or not.

I think managing some services automatically is to help you make it easier, like running the httpd on centos automatically through playbooks.