Name: Davonn Escobilla	Date Performed: 10/08/2022
Course/Section: CPE31S24	Date Submitted: 10/08/2022
Instructor: Dr. Jonathan Taylar	Semester and SY: 1st Sem, 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.

```
---

- 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"
```

davonn@workstation: ~/CPE232_Davonn

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:

```
[web_servers]
192.168.56.120
192.168.56.121

[db_servers]
192.168.56.122


[file_servers]
192.168.56.123
```

```
davonn@workstation: ~/CPE232_Davonn
  GNU nano 6.2                          inventory *
[web_servers]
192.168.56.102
192.168.56.104

[db_servers]
192.168.56.102

[file_servers]
192.168.56.104
```

Make sure to save the file and exit.

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

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

```yaml
---

- 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"
```

davonn@workstation: ~/CPE232_Davonn

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

- hosts: all
  become: true
  tasks:
  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:
```

```
davonn@workstation: ~/CPE232_Davonn
GNU nano 6.2                                site.yml *
      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
Rhythmbox : 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"
```

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.

```
davonn@workstation: ~/Escobilla_PrelimExam
davonn@workstation:~/CPE232_Davonn$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] ***********************************************************
*
ok: [192.168.56.102]
[DEPRECATION WARNING]: Distribution ubuntu 22.04 on host 192.168.56.104 should
use /usr/bin/python3, but is using /usr/bin/python for backward compatibility
with prior Ansible releases. A future Ansible release will default to using the
 discovered platform python for this host. See https://docs.ansible.com/ansible
/2.10/reference_appendices/interpreter_discovery.html for more information.
This feature will be removed in version 2.12. Deprecation warnings can be
disabled by setting deprecation_warnings=False in ansible.cfg.
ok: [192.168.56.104]

TASK [install updates (CentOS)] **************************************************
*
skipping: [192.168.56.104]
fatal: [192.168.56.102]: FAILED! => {"changed": false, "cmd": "dnf install -y p
ython3-dnf", "msg": "Could not import the dnf python module using /usr/bin/pyth
on (3.6.8 (default, Nov 16 2020, 16:55:22) [GCC 4.8.5 20150623 (Red Hat 4.8.5-4
4)]). Please install `python3-dnf` package or ensure you have specified the cor
rect ansible_python_interpreter.", "rc": 1, "results": [], "stderr": "Error: Un
able to find a match: python3-dnf\n", "stderr_lines": ["Error: Unable to find a
 match: python3-dnf"], "stdout": "Last metadata expiration check: 0:04:45 ago o
n Saturday, 08 October, 2022 09:45:21 AM PST.\nNo match for argument: python3-d
```

Before I proceed to run the playbook, it seems that an error occurred in which I have to remove and reinstall python 3 again on CentOS as the IP address of Ubuntu Server 1 is skipped.

```
davonn@workstation: ~/CPE232_Davonn

davonn@workstation:~/CPE232_Davonn$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
[DEPRECATION WARNING]: Distribution ubuntu 22.04 on host 192.168.56.104 should
use /usr/bin/python3, but is using /usr/bin/python for backward compatibility
with prior Ansible releases. A future Ansible release will default to using the
 discovered platform python for this host. See https://docs.ansible.com/ansible
/2.10/reference_appendices/interpreter_discovery.html for more information.
This feature will be removed in version 2.12. Deprecation warnings can be
disabled by setting deprecation_warnings=False in ansible.cfg.
ok: [192.168.56.104]
[WARNING]: Distribution centos 7.9.2009 on host 192.168.56.102 should use
/usr/bin/python, but is using /usr/bin/python2.7, since the discovered platform
python interpreter was not present. See https://docs.ansible.com/ansible/2.10/r
eference_appendices/interpreter_discovery.html for more information.
ok: [192.168.56.102]

TASK [install updates (CentOS)] ************************************************
*
skipping: [192.168.56.104]
ok: [192.168.56.102]

TASK [install updates (Ubuntu)] ************************************************
*
```

davonn@workstation: ~/CPE232_Davonn

```
TASK [install updates (CentOS)] ********************************************
*
skipping: [192.168.56.104]
ok: [192.168.56.102]

TASK [install updates (Ubuntu)] ********************************************
*
skipping: [192.168.56.102]
ok: [192.168.56.104]

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

TASK [Gathering Facts] *****************************************************
*
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [install apache and php for Ubuntu servers] ****************************
*
skipping: [192.168.56.102]
ok: [192.168.56.104]

TASK [install apache and php for CentOS servers] ****************************
*
skipping: [192.168.56.104]
ok: [192.168.56.102]

PLAY RECAP *****************************************************************
```

Show Applications

```
davonn@workstation: ~/CPE232_Davonn
skipping: [192.168.56.102]
ok: [192.168.56.104]

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [install apache and php for Ubuntu servers] ******************************
*
skipping: [192.168.56.102]
ok: [192.168.56.104]

TASK [install apache and php for CentOS servers] ******************************
*
skipping: [192.168.56.104]
ok: [192.168.56.102]

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

davonn@workstation:~/CPE232_Davonn$
```

The solution to the problem is removing the python3 installed on the CentOS and then I proceeded to run the site.yml. The results are applied on the remote servers with the specified node of web_servers.

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

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

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

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

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

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

```
davonn@workstation: ~/CPE232_Davonn
GNU nano 6.2                                site.yml *
      state: latest
    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: install mariadb package (Ubuntu)
    apt:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "Ubuntu"

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

Make sure to save the file and exit.

```
TASK [Mariadb- Restarting/Enabling] ******************************************
*
fatal: [192.168.56.104]: FAILED! => {"changed": false, "msg": "Could not find t
he requested service mariadb: host"}
```

I have encountered this problem because I need to switch the order of install mariadb package and the restarting/enabling.

```
TASK [Mariadb- Restarting/Enabling] ******************************************
*
fatal: [192.168.56.104]: FAILED! => {"changed": false, "msg": "Unsupported para
meters for (ansible.legacy.systemd) module: enable Supported parameters include
: daemon_reexec, daemon_reload, enabled, force, masked, name, no_block, scope,
state, user"}

PLAY RECAP *********************************************************************
*
192.168.56.102             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.104             : ok=6    changed=1    unreachable=0    failed=1
skipped=3    rescued=0    ignored=0
```

I have solved this problem by changing the enable word in the last task to enabled.

```
davonn@workstation: ~/CPE232_Davonn
ok: [192.168.56.102]

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

TASK [Gathering Facts] ******************************************************
*
ok: [192.168.56.104]

TASK [install mariadb package (CentOS)] *************************************
*
skipping: [192.168.56.104]

TASK [install mariadb package (Ubuntu)] *************************************
*
ok: [192.168.56.104]

TASK [Mariadb- Restarting/Enabling] *****************************************
*
changed: [192.168.56.104]

PLAY RECAP ******************************************************************
*
192.168.56.102             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.104             : ok=7    changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0

davonn@workstation:~/CPE232_Davonn$
```

After running the playbook, I have successfully applied the changes on remote servers inside db_servers.

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

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

```
davonn@server1: ~
● mariadb.service - MariaDB 10.6.7 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor pres>
     Active: active (running) since Sat 2022-10-08 10:45:47 PST; 3min 8s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 33696 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /v>
    Process: 33697 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_>
    Process: 33699 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] &>
    Process: 33739 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP>
    Process: 33741 ExecStartPost=/etc/mysql/debian-start (code=exited, status=>
   Main PID: 33728 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 8 (limit: 1640)
     Memory: 56.8M
        CPU: 494ms
     CGroup: /system.slice/mariadb.service
             └─33728 /usr/sbin/mariadbd

Oct 08 10:45:47 server1 mariadbd[33728]: Version: '10.6.7-MariaDB-2ubuntu1.1' >
Oct 08 10:45:47 server1 systemd[1]: Started MariaDB 10.6.7 database server.
Oct 08 10:45:47 server1 /etc/mysql/debian-start[33743]: Upgrading MySQL tables>
Oct 08 10:45:47 server1 /etc/mysql/debian-start[33746]: Looking for 'mysql' as>
Oct 08 10:45:47 server1 /etc/mysql/debian-start[33746]: Looking for 'mysqlchec>
Oct 08 10:45:47 server1 /etc/mysql/debian-start[33746]: This installation of M>
Oct 08 10:45:47 server1 /etc/mysql/debian-start[33746]: There is no need to ru>
Oct 08 10:45:47 server1 /etc/mysql/debian-start[33746]: You can use --force if>
Oct 08 10:45:47 server1 /etc/mysql/debian-start[33754]: Checking for insecure >
Oct 08 10:45:47 server1 /etc/mysql/debian-start[33759]: Triggering myisam-reco>
lines 1-28
```

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

Describe the output.

Mariadb is installed on server 1 as the display says active (running), while on CentOS it says it could not be found.

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
```

```
davonn@workstation: ~/CPE232_Davonn
GNU nano 6.2                                site.yml *
    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

- hosts: file_servers
  become: true
  tasks:

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

Make sure to save the file and exit.

```
TASK [install samba package] ***************************************************
*
fatal: [192.168.56.102]: FAILED! => {"changed": false, "cmd": "/bin/yum -d 2 -y
 check-update", "msg": "[Errno 2] No such file or directory", "rc": 2}
```

An error occurred while installing the samba package on CentOS, so my solution is that inside the inventory the IP addresses of Ubuntu and CentOS will be changed.

```
davonn@workstation: ~/CPE232_Davonn
ok: [192.168.56.102]

TASK [install mariadb package (Ubuntu)] ************************************
*
skipping: [192.168.56.102]

TASK [Mariadb- Restarting/Enabling] ****************************************
*
changed: [192.168.56.102]

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

TASK [Gathering Facts] *****************************************************
*
ok: [192.168.56.104]

TASK [install samba package] ***********************************************
*
changed: [192.168.56.104]

PLAY RECAP *****************************************************************
*
192.168.56.102             : ok=7    changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
192.168.56.104             : ok=6    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
```

The output says it is already changed after I switched the IP's, meaning the task is successfully executed.

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

The testing of the *file_servers* is beyond the scope of this activity, and as well as our topics and objectives. However, in this activity we were able to show that we can target hosts or servers using grouping in ansible playbooks.

Task 2: Using Tags in running playbooks

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

1. Edit the *site.yml* file. Add tags to the playbook. After the name, we can place the tags: *name_of_tag*. This is an arbitrary command, which means you can use any name for a tag.

```yaml
---

- hosts: all
  become: true
  pre_tasks:

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

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

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

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

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

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

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

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

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

- hosts: file_servers
  become: true
  tasks:

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

Make sure to save the file and exit.

```
davonn@workstation: ~/CPE232_Davonn
ok: [192.168.56.102]

TASK [install mariadb package (Ubuntu)] ************************************
*
skipping: [192.168.56.102]

TASK [Mariadb- Restarting/Enabling] ****************************************
*
changed: [192.168.56.102]

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

TASK [Gathering Facts] *****************************************************
*
ok: [192.168.56.104]

TASK [install samba package] ***********************************************
*
ok: [192.168.56.104]

PLAY RECAP *****************************************************************
*
192.168.56.102             : ok=7    changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
192.168.56.104             : ok=6    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0

davonn@workstation:~/CPE232_Davonn$
```

The output is pretty much the same as before.

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

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

```
davonn@workstation:~/CPE232_Davonn$ 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, maria, mariadb, ubuntu]

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

The outputs are the tags present on the site.yml

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

```
davonn@workstation: ~/CPE232_Davonn
skipping: [192.168.56.104]
ok: [192.168.56.102]

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

TASK [Gathering Facts] ******************************************************
*
ok: [192.168.56.102]

TASK [install mariadb package (CentOS)] *************************************
*
ok: [192.168.56.102]

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

TASK [Gathering Facts] ******************************************************
*
ok: [192.168.56.104]

PLAY RECAP ******************************************************************
*
192.168.56.102             : ok=6    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.104             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
```

The task stopped at db_servers since the tag presented on that task is centos.

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

```
davonn@workstation: ~/CPE232_Davonn
*

TASK [Gathering Facts] ***********************************************************
*
ok: [192.168.56.102]

TASK [install mariadb package (CentOS)] ******************************************
*
ok: [192.168.56.102]

TASK [install mariadb package (Ubuntu)] ******************************************
*
skipping: [192.168.56.102]

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

TASK [Gathering Facts] ***********************************************************
*
ok: [192.168.56.104]

PLAY RECAP ***********************************************************************
*
192.168.56.102             : ok=5    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.104             : ok=4    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
```

As presented in the first task, the db_servers have a tag of db so when we executed the command it performed the task up until the db_server task.

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

```
davonn@workstation: ~/CPE232_Davonn
Terminal
ok: [192.168.56.104]

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.104]
ok: [192.168.56.102]

TASK [install apache and php for Ubuntu servers] ******************************
*
skipping: [192.168.56.102]
ok: [192.168.56.104]

TASK [install apache and php for CentOS servers] ******************************
*
skipping: [192.168.56.104]
ok: [192.168.56.102]

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.102]
```

```
davonn@workstation: ~/CPE232_Davonn
ok: [192.168.56.104]

TASK [install apache and php for CentOS servers] ******************************
*
skipping: [192.168.56.104]
ok: [192.168.56.102]

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.102]

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.104]

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

It only execute the task on web_servers since it is the only one with tag of apache.

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

```
davonn@workstation: ~/CPE232_Davonn

ok: [192.168.56.104]

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

TASK [Gathering Facts] *******************************************
*
ok: [192.168.56.104]
ok: [192.168.56.102]

TASK [install apache and php for Ubuntu servers] *****************
*
skipping: [192.168.56.102]
ok: [192.168.56.104]

TASK [install apache and php for CentOS servers] *****************
*
skipping: [192.168.56.104]
ok: [192.168.56.102]

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

TASK [Gathering Facts] *******************************************
*
ok: [192.168.56.102]

TASK [install mariadb package (CentOS)] **************************
*
```

```
davonn@workstation: ~/CPE232_Davonn
*

TASK [Gathering Facts] ***********************************************************
*
ok: [192.168.56.102]

TASK [install mariadb package (CentOS)] ******************************************
*
ok: [192.168.56.102]

TASK [install mariadb package (Ubuntu)] ******************************************
*
skipping: [192.168.56.102]

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

TASK [Gathering Facts] ***********************************************************
*
ok: [192.168.56.104]

PLAY RECAP ***********************************************************************
*
192.168.56.102             : ok=6    changed=0    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
192.168.56.104             : ok=5    changed=0    unreachable=0    failed=0
                           cued=0    ignored=0
Show Applications
```

The tasks with a tag of apache and db are successfully executed.

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

Task 3: Managing Services

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

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

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

```
davonn@workstation: ~/CPE232_Davonn
GNU nano 6.2                                site.yml *
- name: install apache and php for Ubuntu servers
  tags: apache,apache2,ubuntu
  apt:
    name:
      - apache2
      - libapache2-mod-php
    state: latest
    update_cache: yes
  when: ansible_distribution == "Ubuntu"

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

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

Figure 3.1.1
Make sure to save the file and exit.

You would also notice from our previous activity that we already created a module that runs a service.

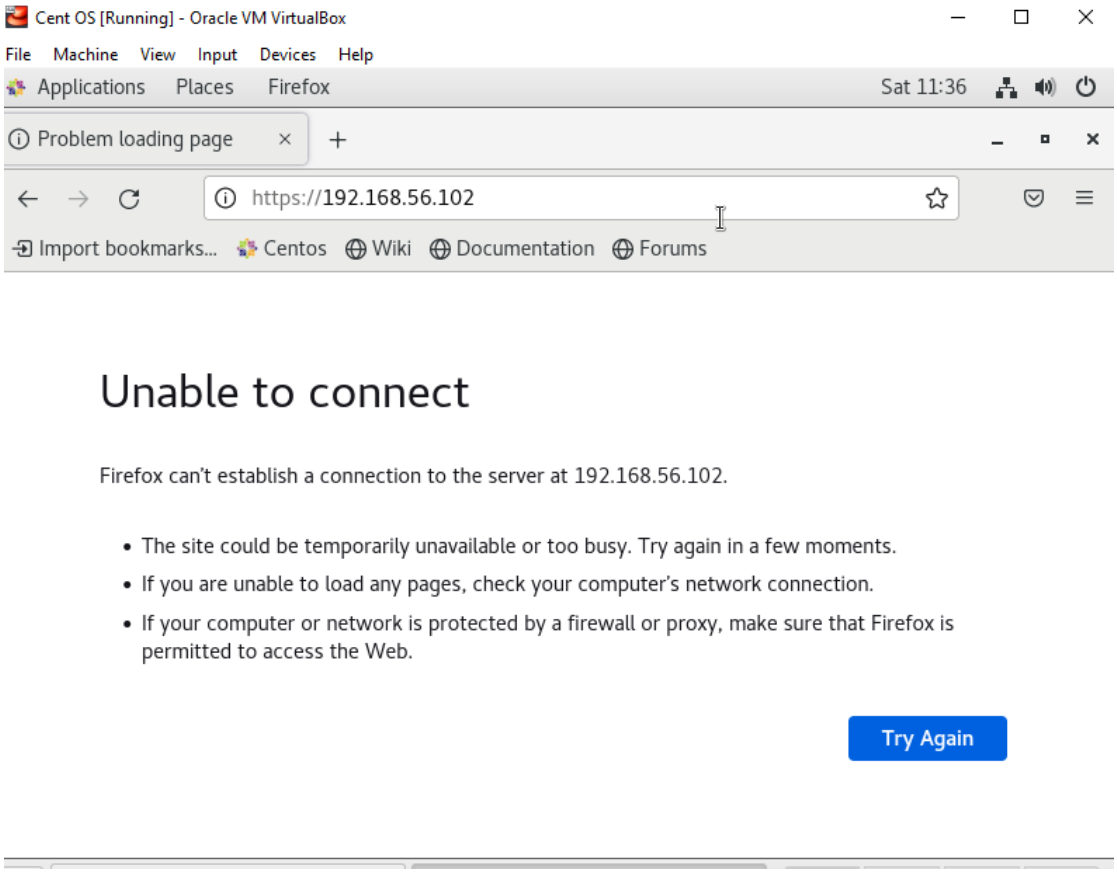
```
- hosts: db_servers
  become: true
  tasks:

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

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

Figure 3.1.2

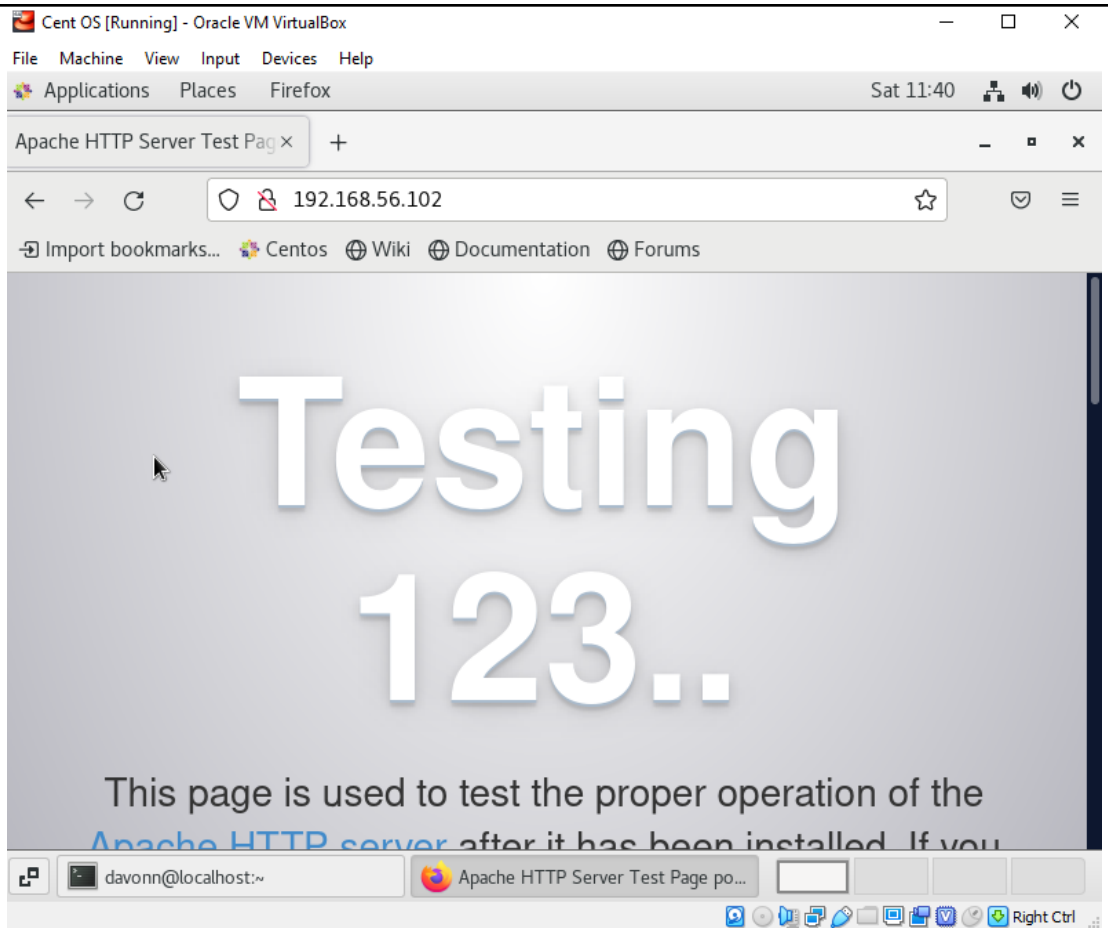
This is because in CentOS, installed packages' services are not run automatically. Thus, we need to create the module to run it automatically.

2. To test it, before you run the saved playbook, go to the CentOS server and stop the currently running httpd using the command *sudo systemctl stop httpd.* When prompted, enter the sudo password. After that, open the browser and enter the CentOS server's IP address. You should not be getting a display

because we stopped the httpd service already.



3. Go to the local machine and this time, run the *site.yml* file. Then after running the file, go again to the CentOS server and enter its IP address on the browser. Describe the result.
   To automatically enable the service every time we run the playbook, use the command *enabled: true* similar to Figure 7.1.2 and save the playbook.



Reflections:

Answer the following:

1. What is the importance of putting our remote servers into groups?
   The importance of it is that we can more clearly organize the tasks that we want to implement on some specific servers, therefore when a problem occurs it can be easily resolved.
2. What is the importance of tags in playbooks?
   Tags can be used to specifically perform certain tasks and install packages of that group.
3. Why do think some services need to be managed automatically in playbooks?
   I think it is for convenience so we can easily automate all the packages that we need to install for specific needs, thus it can improve the efficiency of work and reduce time consumption.