

<b>Name:</b> Recto, Jon Jeous J.	<b>Date Performed:</b> Mar 4, 2024
<b>Course/Section:</b> CPE31S1	<b>Date Submitted:</b> Mar 5, 2024
<b>Instructor:</b> Dr. Taylar	<b>Semester and SY:</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>  <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 name it site.yml. Follow the commands as shown in the image below. Make sure to save the file and exit.	

```
jonjeous@localmachine-VirtualBox: ~/Recto_HOA6
GNU nano 6.2 site.yml
---
- hosts: all
  become: true
  tasks:

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

  - name: install apache and php for CentOS
    yum:
      name:
        - httpd
        - php
      state: latest
      update_cache: yes
      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:

```
jonjeous@localmachine-VirtualBox: ~/Recto_HOA6
GNU nano 6.2 inventory
[web_servers]
192.168.56.118 #Server1
192.168.56.119 #Server2

[db_servers]
192.168.56.116 #ManagedNode

[file_servers]
192.168.56.117 #centOS
```

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.

```

jonjeous@localmachine-VirtualBox:~/Recto_H0A6$ ansible-playbook --ask-become-pass
s site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.116]
ok: [192.168.56.118]
ok: [192.168.56.119]
ok: [192.168.56.117]

TASK [install apache and php for Ubuntu] *****
skipping: [192.168.56.117]
ok: [192.168.56.116]
ok: [192.168.56.119]
ok: [192.168.56.118]

TASK [install apache and php for CentOS] *****
skipping: [192.168.56.118]
skipping: [192.168.56.119]
skipping: [192.168.56.116]
ok: [192.168.56.117]

PLAY RECAP *****
192.168.56.116      : ok=2    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.117      : ok=2    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.118      : ok=2    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.119      : ok=2    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
jonjeous@localmachine-VirtualBox:~/Recto_H0A6$

```

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

Make sure to save the file and exit.

```
jonjeous@localmachine-VirtualBox: ~/Recto_HOA6
GNU nano 6.2 site.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
```

```
jonjeous@localmachine-VirtualBox: ~/Recto_HOA6
GNU nano 6.2 site.yml *
tasks:

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

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

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.

```
jonjeous@localmachine-VirtualBox:~/Recto_HOAc$ ansible-playbook --ask-become-pass site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.119]
ok: [192.168.56.116]
ok: [192.168.56.117]
ok: [192.168.56.118]

TASK [install updates (CentOS)] *****
skipping: [192.168.56.118]
skipping: [192.168.56.119]
skipping: [192.168.56.116]
ok: [192.168.56.117]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.117]
ok: [192.168.56.119]
ok: [192.168.56.118]
ok: [192.168.56.116]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.118]
ok: [192.168.56.119]

TASK [install apache and php for Ubuntu] *****
ok: [192.168.56.119]
ok: [192.168.56.118]

TASK [install apache and php for CentOS] *****
skipping: [192.168.56.118]
skipping: [192.168.56.119]

PLAY RECAP *****
192.168.56.116      : ok=2    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.117      : ok=2    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.118      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.119      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
```

- 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.

```
jonjeous@localmachine-VirtualBox: ~/Recto_HOA6
GNU nano 6.2 site.yml *

- 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.

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

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

TASK [install mariadb package (CentOS)] *****
skipping: [192.168.56.116]

TASK [install mariadb package (Ubuntu)] *****
changed: [192.168.56.116]

TASK [Mariadb- Restarting/Enabling] *****
changed: [192.168.56.116]

PLAY RECAP *****
192.168.56.116      : ok=5    changed=2    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.117      : ok=2    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.118      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.119      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
jonjeous@localmachine-VirtualBox:~/Recto_HOA6$
```

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.

```
jonjeous@localmachine-VirtualBox:~$ systemctl status mariadb
● mariadb.service - MariaDB 10.6.16 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor prese
   Active: active (running) since Tue 2024-03-05 11:04:45 PST; 1min 17s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 10527 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /va
   Process: 10529 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_S
   Process: 10531 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] &&
   Process: 10571 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_
   Process: 10573 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0
 Main PID: 10560 (mariabdb)
   Status: "Taking your SQL requests now..."
     Tasks: 9 (limit: 4598)
    Memory: 60.9M
       CPU: 457ms
    CGroup: /system.slice/mariadb.service
           └─10560 /usr/sbin/mariabdb

jonjeous@localmachine-VirtualBox:~$ ssh jonjeous@192.168.56.117
Last login: Mon Mar  4 22:03:28 2024 from 192.168.56.116
[jonjeous@server3 ~]$ systemctl status mariadb
Unit mariadb.service could not be found.
[jonjeous@server3 ~]$
```

*MariaDB is not found on CentOS server because the IP address for CentOS is not included in the `db_servers` inventory*

6. Edit the `site.yml` again. This time we will append the code to configure installation on the `file_servers` group. We can add the following on our file.

```
- hosts: file_servers
  become: true
  tasks:

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

Make sure to save the file and exit.

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

```

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.117]

TASK [install samba package] *****
changed: [192.168.56.117]

PLAY RECAP *****
192.168.56.116      : ok=5    changed=1    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.117      : ok=4    changed=1    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.118      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.119      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
jonjeous@localmachine-VirtualBox:~/Recto_H0Ac$

```

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"

```

```

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

- hosts: file_servers
  become: true
  tasks:

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

```

Make sure to save the file and exit.

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

```
jonjeous@localmachine-VirtualBox:~/Recto_H0A6$ ansible-playbook --ask-become-pass site.yml
BECOME password:
```

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

```
TASK [Gathering Facts] *****
```

```
ok: [192.168.56.118]
ok: [192.168.56.119]
ok: [192.168.56.116]
ok: [192.168.56.117]
```

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

```
skipping: [192.168.56.118]
skipping: [192.168.56.119]
skipping: [192.168.56.116]
ok: [192.168.56.117]
```

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

```
skipping: [192.168.56.117]
ok: [192.168.56.119]
ok: [192.168.56.118]
ok: [192.168.56.116]
```

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

```
TASK [Gathering Facts] *****
```

```
ok: [192.168.56.119]
ok: [192.168.56.118]
```

```
TASK [install apache and php for Ubuntu] *****
```

```
ok: [192.168.56.119]
ok: [192.168.56.118]
```

```
TASK [install apache and php for CentOS] *****
```

```
skipping: [192.168.56.118]
skipping: [192.168.56.119]
```

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

```
TASK [Gathering Facts] *****
```

```
ok: [192.168.56.116]
```

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

```
skipping: [192.168.56.116]
```

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

```
ok: [192.168.56.116]
```

```
TASK [Mariadb- Restarting/Enabling] *****
```

```
changed: [192.168.56.116]
```

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

```
TASK [Gathering Facts] *****
```

```
ok: [192.168.56.117]
```

```
TASK [install samba package] *****
```

```
ok: [192.168.56.117]
```

```
PLAY RECAP *****
192.168.56.116      : ok=5    changed=1    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.117      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.118      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.119      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
jonjeous@localmachine-VirtualBox:~/Recto_H0A6$
```

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

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

```
jonjeous@localmachine-VirtualBox:~/Recto_H0A6$ 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, ubuntu]

  play #4 (file_servers): file_servers    TAGS: []
    TASK TAGS: [samba]
jonjeous@localmachine-VirtualBox:~/Recto_H0A6$
```

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

```
jonjeous@localmachine-VirtualBox:~/Recto_H0A6$ ansible-playbook --tags centos --
ask-become-pass site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.119]
ok: [192.168.56.118]
ok: [192.168.56.116]
ok: [192.168.56.117]

TASK [install updates (CentOS)] *****
skipping: [192.168.56.118]
skipping: [192.168.56.119]
skipping: [192.168.56.116]
ok: [192.168.56.117]
```

```

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.117]
ok: [192.168.56.118]
ok: [192.168.56.119]
ok: [192.168.56.116]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.118]
ok: [192.168.56.119]

TASK [install apache and php for CentOS] *****
skipping: [192.168.56.118]
skipping: [192.168.56.119]

PLAY [db_servers] *****

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

```

```

TASK [install mariadb package (CentOS)] *****
skipping: [192.168.56.116]

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.117]

PLAY RECAP *****
192.168.56.116      : ok=3    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.117      : ok=3    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.118      : ok=3    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.119      : ok=3    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0

```

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

```

jonjeous@localmachine-VirtualBox:~/Recto_H0Ac$ ansible-playbook --tags db --ask-
become-pass site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.119]
ok: [192.168.56.118]
ok: [192.168.56.116]
ok: [192.168.56.117]

TASK [install updates (CentOS)] *****
skipping: [192.168.56.118]
skipping: [192.168.56.119]
skipping: [192.168.56.116]
ok: [192.168.56.117]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.117]
ok: [192.168.56.119]
ok: [192.168.56.118]
ok: [192.168.56.116]

```

```

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.118]
ok: [192.168.56.119]

PLAY [db_servers] *****

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

TASK [install mariadb package (CentOS)] *****
skipping: [192.168.56.116]

TASK [install mariadb package (Ubuntu)] *****
ok: [192.168.56.116]

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.117]

PLAY RECAP *****
192.168.56.116      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.117      : ok=3    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.118      : ok=3    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.119      : ok=3    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
jonjeous@localmachine-VirtualBox:~/Recto_H0A6$

```

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

```

jonjeous@localmachine-VirtualBox:~/Recto_H0A6$ ansible-playbook --tags apache --ask-become-pass site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.119]
ok: [192.168.56.118]
ok: [192.168.56.116]
ok: [192.168.56.117]

TASK [install updates (CentOS)] *****
skipping: [192.168.56.118]
skipping: [192.168.56.119]
skipping: [192.168.56.116]
ok: [192.168.56.117]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.117]
ok: [192.168.56.119]
ok: [192.168.56.118]
ok: [192.168.56.116]

```

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

TASK [Gathering Facts] *****
ok: [192.168.56.118]
ok: [192.168.56.119]

TASK [install apache and php for Ubuntu] *****
ok: [192.168.56.118]
ok: [192.168.56.119]

TASK [install apache and php for CentOS] *****
skipping: [192.168.56.118]
skipping: [192.168.56.119]

PLAY [db_servers] *****

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

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.117]

PLAY RECAP *****
192.168.56.116      : ok=3    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.117      : ok=3    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.118      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.119      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0

jonjeous@localmachine-VirtualBox:~/Recto_H0A6$
```

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

```
jonjeous@localmachine-VirtualBox:~/Recto_H0A6$ ansible-playbook --tags "apache,d
b" --ask-become-pass site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.119]
ok: [192.168.56.118]
ok: [192.168.56.117]
ok: [192.168.56.116]

TASK [install updates (CentOS)] *****
skipping: [192.168.56.118]
skipping: [192.168.56.119]
skipping: [192.168.56.116]
ok: [192.168.56.117]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.117]
ok: [192.168.56.118]
ok: [192.168.56.119]
ok: [192.168.56.116]
```

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

TASK [Gathering Facts] *****
ok: [192.168.56.118]
ok: [192.168.56.119]

TASK [install apache and php for Ubuntu] *****
ok: [192.168.56.119]
ok: [192.168.56.118]

TASK [install apache and php for CentOS] *****
skipping: [192.168.56.118]
skipping: [192.168.56.119]

PLAY [db_servers] *****

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

TASK [install mariadb package (CentOS)] *****
skipping: [192.168.56.116]
```

```
TASK [install mariadb package (Ubuntu)] *****
ok: [192.168.56.116]

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.117]

PLAY RECAP *****
192.168.56.116      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.117      : ok=3    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.118      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.119      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
jonjeous@localmachine-VirtualBox: ~/Recto_H0A6$
```

*You can use tags to skip certain tasks during playbook execution. This is helpful when you want to exclude certain tasks temporarily or in specific scenarios.*

### 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
  tags: apache,centos,httpd
  yum:
    name:
      - httpd
      - php
    state: latest
    update_cache: yes
  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.

```

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

```

Figure 3.1.2

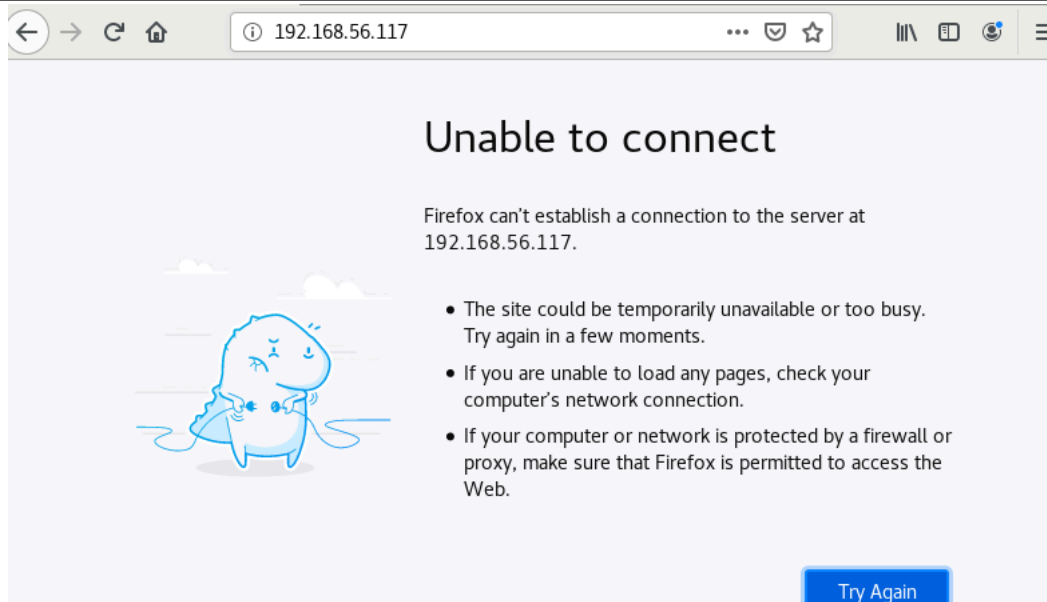
This is because in CentOS, installed packages' services are not run automatically. Thus, we need to create the module to run it automatically.

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.

```

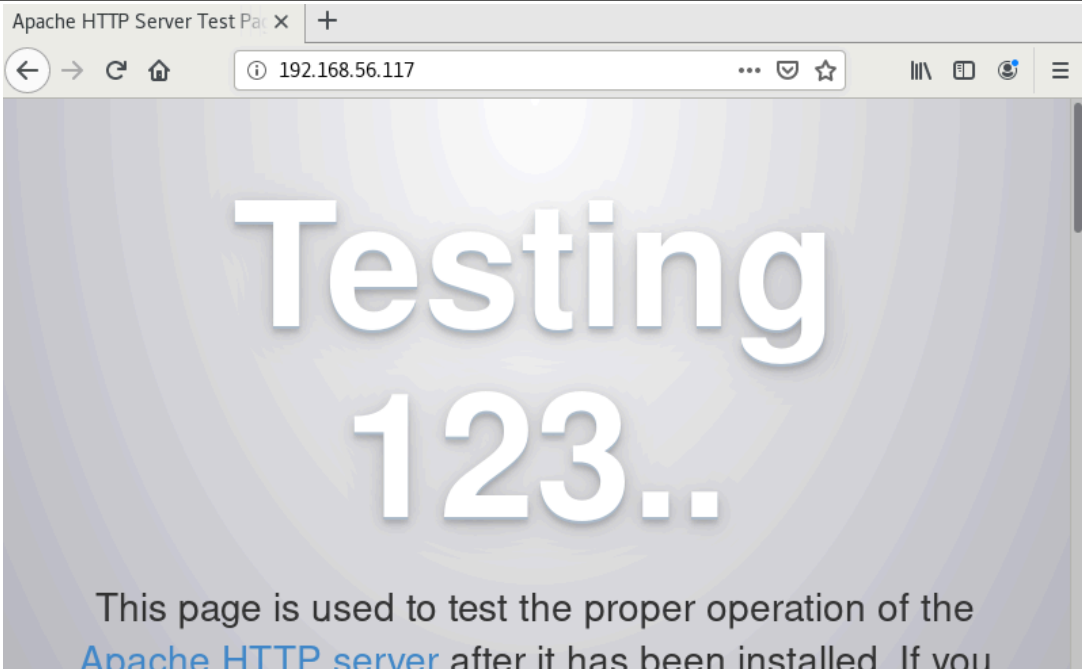
jonjeous@localmachine-VirtualBox:~$ ssh jonjeous@192.168.56.117
Last login: Mon Mar  4 22:31:58 2024 from 192.168.56.116
[jonjeous@server3 ~]$ sudo systemctl stop httpd
[sudo] password for jonjeous:
sudo: systemctl: command not found
[jonjeous@server3 ~]$ sudo systemctl stop httpd
[jonjeous@server3 ~]$

```



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.

```
PLAY RECAP *****
192.168.56.116      : ok=5    changed=1    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.117      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.118      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=3    rescued=0    ignored=0
192.168.56.119      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=3    rescued=0    ignored=0
jonjeous@localmachine-VirtualBox:~/Recto_H0A6$
```



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?

- Grouping servers in Ansible is like putting them into folders. This helps keep things organized, making it easier to manage and understand your computer systems. It's like sorting files into different folders on your computer. This organization allows you to do things more efficiently, like applying changes to only certain groups of servers or running tasks on similar servers at the same time. It's a practical way to handle and control your computer infrastructure.

#### 2. What is the importance of tags in playbooks?

- Tags in Ansible playbooks are like labels that help you pick and choose which tasks to run. They make playbooks organized, easy to read, and simplify maintenance. Tags also allow for efficient task execution, customization, and help when dealing with different server roles. Overall, they provide a flexible way to control and manage infrastructure automation.

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

- Automating service management through playbooks ensures consistent and efficient configuration across servers, reduces the risk of errors, and allows for quick adaptation to changing requirements. It provides a systematic and documented approach, saving time and enhancing reliability in maintaining services within IT systems. Overall, automation simplifies complex tasks and promotes uniformity in service management.

**Conclusion:**

In conclusion, this activity is all about making each server unique by giving them specific jobs. We use tags to pick and choose what actions to take, and we're fixing things so that services start automatically, like when we install a web server. The idea is to make everything run smoother and fit just right for each server in our setup.