Name: Torrecampo, Juan Piolo S.	**Date Performed:** Oct 4, 2022
Course/Section: CPE 232 / CPE31S22	**Date Submitted:** Oct 5, 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"
```

Make sure to save the file and exit.

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

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

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

Figure 1.1. The screenshot above shows the content of the site.yml.

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
```

Make sure to save the file and exit.

```
[web_servers]
192.168.122.88 ansible_user=userver1
192.168.122.13 ansible_user=cserver1

[db_servers]
192.168.122.164 ansible_user=userver2

[file_servers]
192.168.122.131  ansible_user=userver3
```

Figure 1.2. The screenshot above shows the content of the inventory file.

```
~/D/r/C/ansible (main|+1) $ ansible all -m ping
192.168.122.164 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
192.168.122.131 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
192.168.122.13 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
192.168.122.88 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

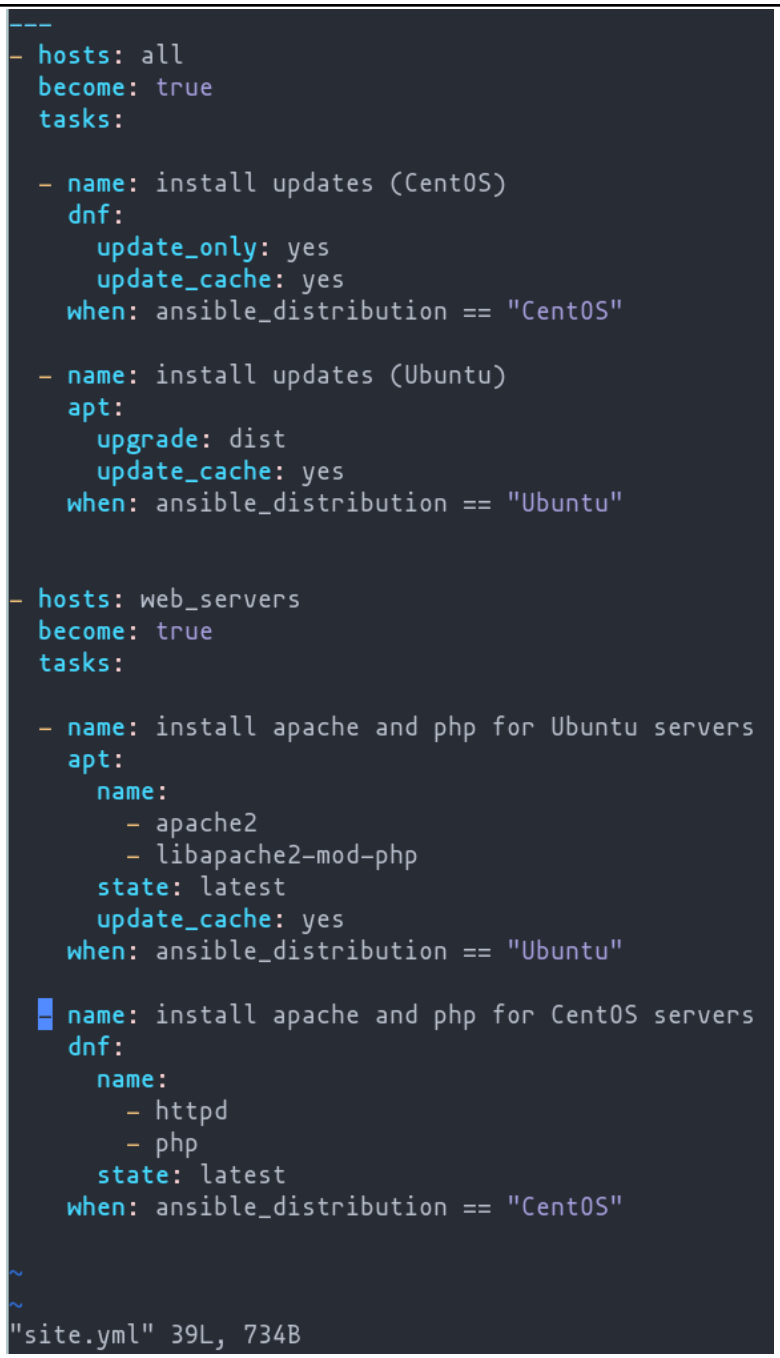
Figure 1.3. The screenshot above shows the ping test of every server stated inside of the inventory file.

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.



```
---
- hosts: all
  become: true
  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
    when: ansible_distribution == "CentOS"

~
~
"site.yml" 39L, 734B
```

Figure 1.4. The screenshot above shows the "web_servers" tasks section appended to the site.yml ansible playbook file.

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.

```
~/D/r/C/ansible (main|+1) $ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.122.131]
ok: [192.168.122.13]
ok: [192.168.122.88]
ok: [192.168.122.164]

TASK [install updates (CentOS)] ************************************************
skipping: [192.168.122.88]
skipping: [192.168.122.164]
skipping: [192.168.122.131]
ok: [192.168.122.13]

TASK [install updates (Ubuntu)] ************************************************
skipping: [192.168.122.13]
ok: [192.168.122.88]
ok: [192.168.122.164]
ok: [192.168.122.131]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.122.88]
ok: [192.168.122.13]

TASK [install apache and php for Ubuntu servers] *******************************
skipping: [192.168.122.13]
ok: [192.168.122.88]

TASK [install apache and php for CentOS servers] *******************************
skipping: [192.168.122.88]
changed: [192.168.122.13]

PLAY RECAP *********************************************************************
192.168.122.13             : ok=4    changed=1    unreachable=0    failed=0    skipped=2    rescued=0
 ignored=0
192.168.122.131            : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0
 ignored=0
192.168.122.164            : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0
 ignored=0
192.168.122.88             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0
 ignored=0
```

Figure 1.5. The screenshot above results in deploying the site.yml.

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"
```

Make sure to save the file and exit.

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

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

Figure 1.6. The screenshot above shows the "db_servers" set of tasks appended to the site.yml.

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

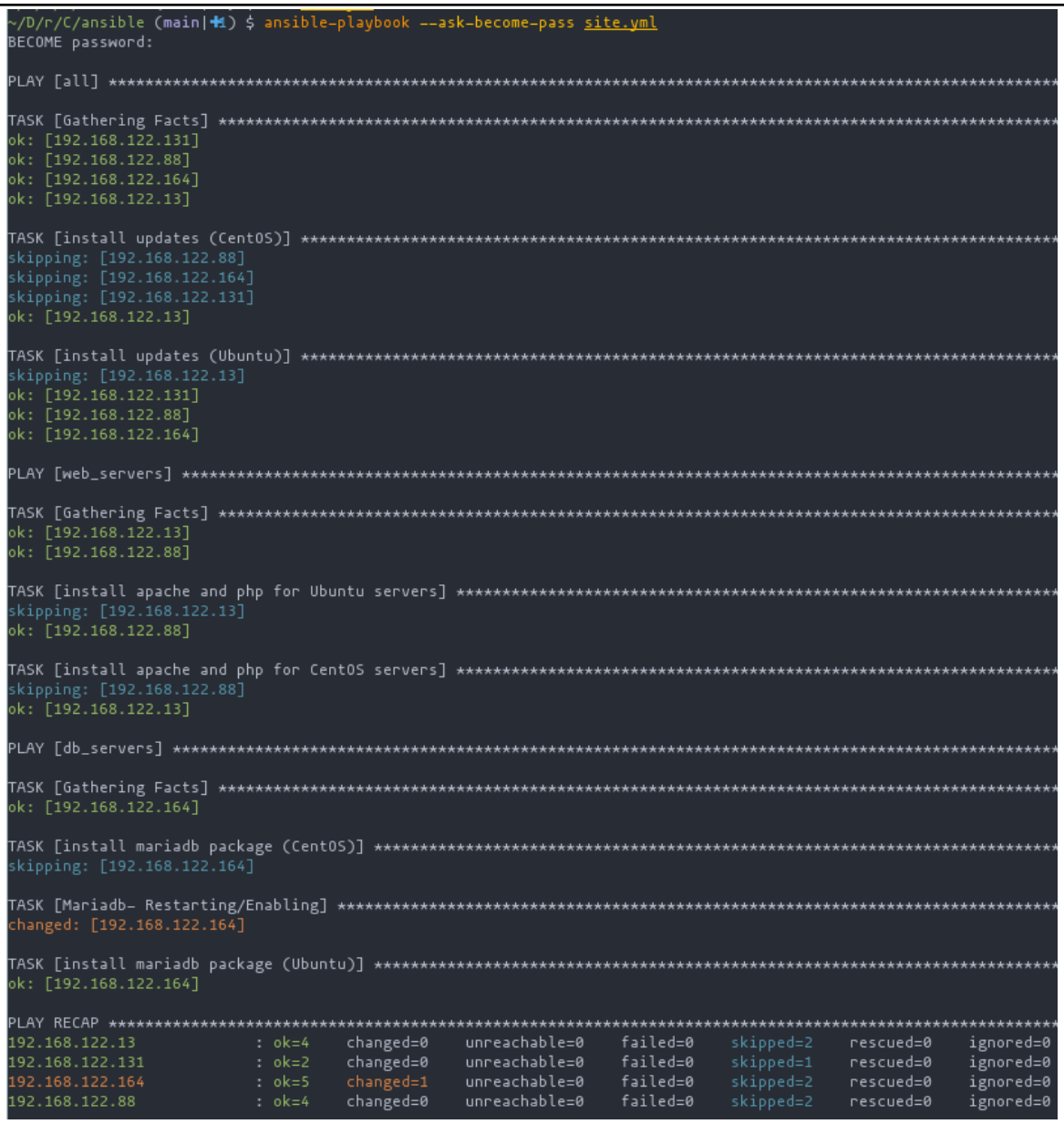


Figure 1.7. The screenshot above shows the result in executing the new site.yml ansible playbook file.

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

   Describe the output.

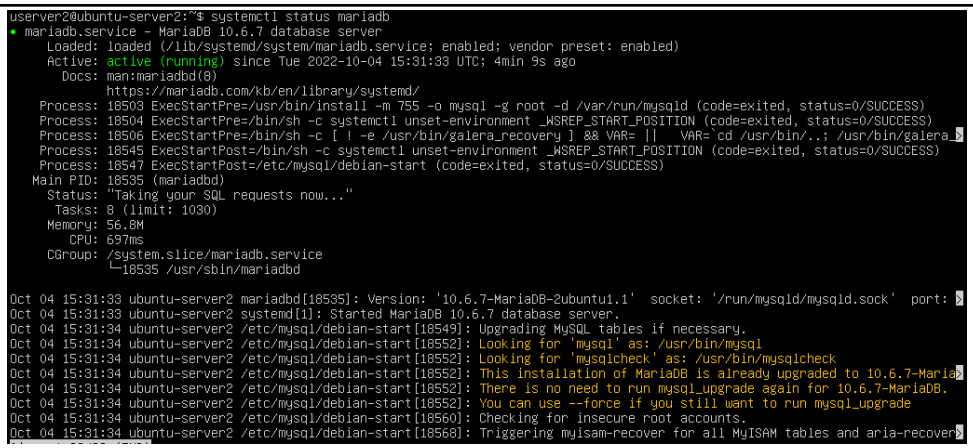


Figure 1.8. The screenshot above shows the mariadb running service inside of server 2.

```
[cserver1@localhost ~]$ sudo systemctl status mariadb
[sudo] password for cserver1:
Unit mariadb.service could not be found.
```

Figure 1.9.The screenshot above shows the output of the command inside of CentOS. Since the inventory only specifies the server 2 and does not include the CentOS.

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.

```
---

- hosts: all
  become: true
  tasks:

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

```yaml
- 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
    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

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


- hosts: file_servers
  become: true
  tasks:

  - name: install samba package
    package:
      name: samba
```

Figure 1.10 The screenshot above shows the "file_servers" set of tasks appended to the site.yml.

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

```
~/D/r/C/ansible (main|+1) [2]$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.164]
ok: [192.168.122.13]
ok: [192.168.122.131]
ok: [192.168.122.88]

TASK [install updates (CentOS)] ************************************************************************************
skipping: [192.168.122.88]
skipping: [192.168.122.164]
skipping: [192.168.122.131]
ok: [192.168.122.13]

TASK [install updates (Ubuntu)] ************************************************************************************
skipping: [192.168.122.13]
ok: [192.168.122.164]
ok: [192.168.122.131]
ok: [192.168.122.88]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.13]
ok: [192.168.122.88]

TASK [install apache and php for Ubuntu servers] *******************************************************************
skipping: [192.168.122.13]
ok: [192.168.122.88]

TASK [install apache and php for CentOS servers] *******************************************************************
skipping: [192.168.122.88]
ok: [192.168.122.13]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.164]

TASK [install mariadb package (CentOS)] ****************************************************************************
skipping: [192.168.122.164]

TASK [Mariadb- Restarting/Enabling] ********************************************************************************
changed: [192.168.122.164]

TASK [install mariadb package (Ubuntu)] ****************************************************************************
ok: [192.168.122.164]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.131]

TASK [install samba package] ***************************************************************************************
changed: [192.168.122.131]

PLAY RECAP *********************************************************************************************************
192.168.122.13             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.122.131            : ok=4    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.122.164            : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.122.88             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

Figure 1.11. The screenshot above shows the result after deploying the new configured site.yml ansible playbook file.

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 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.

```yaml
---

- hosts: all
  become: true
  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
    when: ansible_distribution == "CentOS"
```

```yaml
- 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

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

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

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

Figure 2.1. The screenshots above shows the implementation of tags in ansible playbook.

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

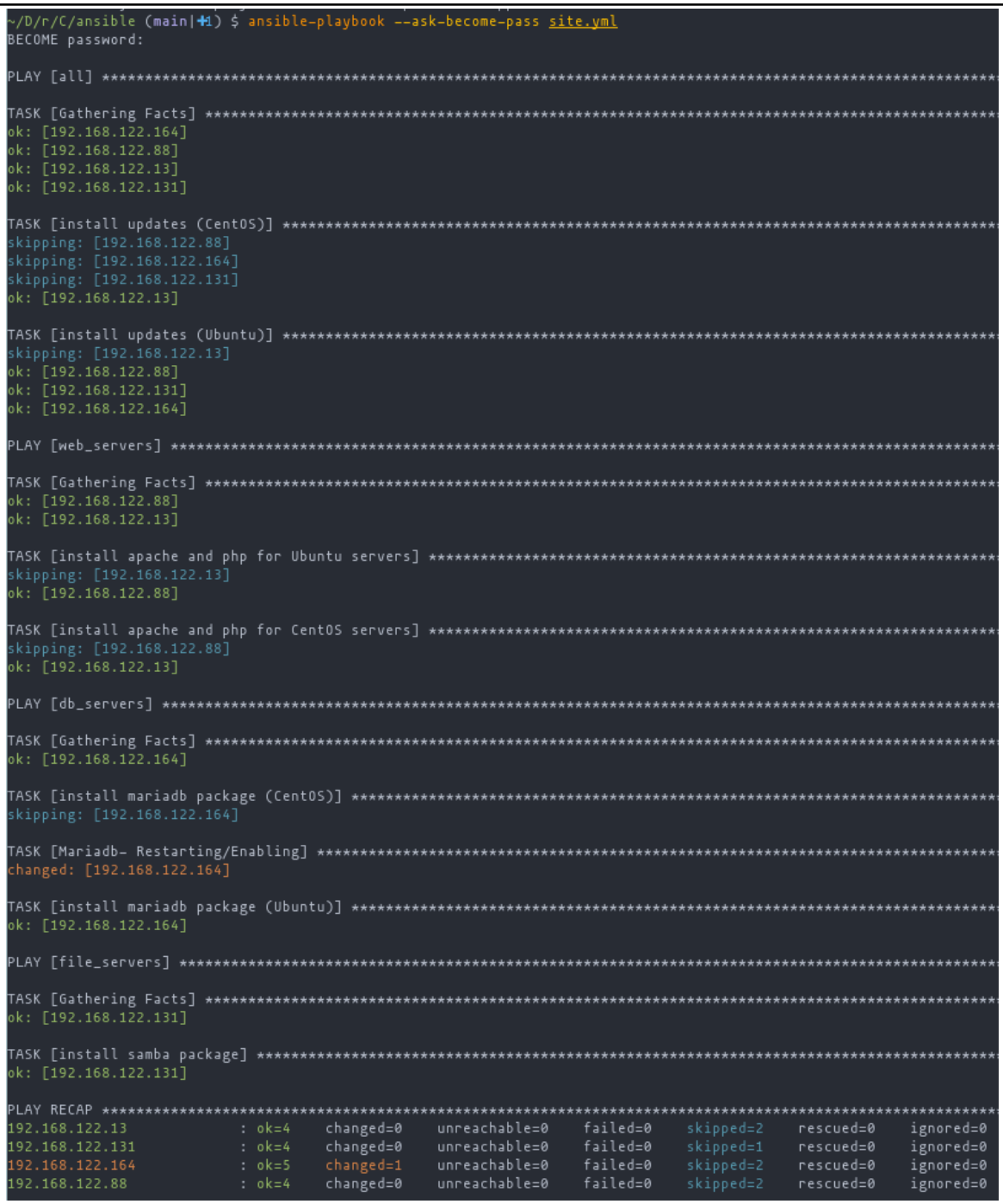


Figure 2.2. The screenshot above shows the site.yml result after deploying it to the servers.

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

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

```
~/D/r/C/ansible (main|+1) $ 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]
```

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

```
~/D/r/C/ansible (main|+1) $ ansible-playbook --tags centos --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.131]
ok: [192.168.122.13]
ok: [192.168.122.164]
ok: [192.168.122.88]

TASK [install updates (CentOS)] ************************************************************************************
skipping: [192.168.122.88]
skipping: [192.168.122.164]
skipping: [192.168.122.131]
ok: [192.168.122.13]

TASK [install updates (Ubuntu)] ************************************************************************************
skipping: [192.168.122.13]
ok: [192.168.122.88]
ok: [192.168.122.131]
ok: [192.168.122.164]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.88]
ok: [192.168.122.13]

TASK [install apache and php for CentOS servers] *******************************************************************
skipping: [192.168.122.88]
ok: [192.168.122.13]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.164]

TASK [install mariadb package (CentOS)] ****************************************************************************
skipping: [192.168.122.164]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.131]

PLAY RECAP *********************************************************************************************************
192.168.122.13             : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.122.131            : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.122.164            : ok=3    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.122.88             : ok=3    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0

~/D/r/C/ansible (main|+1) $
```

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

```
~/D/r/C/ansible (main|+1) $ ansible-playbook --tags db --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.13]
ok: [192.168.122.131]
ok: [192.168.122.88]
ok: [192.168.122.164]

TASK [install updates (CentOS)] ************************************************************************************
skipping: [192.168.122.88]
skipping: [192.168.122.164]
skipping: [192.168.122.131]
ok: [192.168.122.13]

TASK [install updates (Ubuntu)] ************************************************************************************
skipping: [192.168.122.13]
ok: [192.168.122.88]
ok: [192.168.122.164]
ok: [192.168.122.131]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.13]
ok: [192.168.122.88]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.164]

TASK [install mariadb package (CentOS)] ****************************************************************************
skipping: [192.168.122.164]

TASK [install mariadb package (Ubuntu)] ****************************************************************************
ok: [192.168.122.164]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.131]

PLAY RECAP *********************************************************************************************************
192.168.122.13             : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.122.131            : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.122.164            : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.122.88             : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

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

```
~/D/r/C/ansible (main|+1) $ ansible-playbook --tags apache --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.131]
ok: [192.168.122.164]
ok: [192.168.122.13]
ok: [192.168.122.88]

TASK [install updates (CentOS)] ************************************************************************************
skipping: [192.168.122.88]
skipping: [192.168.122.164]
skipping: [192.168.122.131]
ok: [192.168.122.13]

TASK [install updates (Ubuntu)] ************************************************************************************
skipping: [192.168.122.13]
ok: [192.168.122.131]
ok: [192.168.122.164]
ok: [192.168.122.88]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.13]
ok: [192.168.122.88]

TASK [install apache and php for Ubuntu servers] *******************************************************************
skipping: [192.168.122.13]
ok: [192.168.122.88]

TASK [install apache and php for CentOS servers] *******************************************************************
skipping: [192.168.122.88]
ok: [192.168.122.13]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.164]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.131]

PLAY RECAP *********************************************************************************************************
192.168.122.13             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.122.131            : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.122.164            : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.122.88             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

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

```
~/D/r/C/ansible (main|+1) $ ansible-playbook --tags "apache,db" --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.13]
ok: [192.168.122.164]
ok: [192.168.122.131]
ok: [192.168.122.88]

TASK [install updates (CentOS)] ************************************************************************************
skipping: [192.168.122.88]
skipping: [192.168.122.164]
skipping: [192.168.122.131]
ok: [192.168.122.13]

TASK [install updates (Ubuntu)] ************************************************************************************
skipping: [192.168.122.13]
ok: [192.168.122.164]
ok: [192.168.122.88]
ok: [192.168.122.131]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.13]
ok: [192.168.122.88]

TASK [install apache and php for Ubuntu servers] *******************************************************************
skipping: [192.168.122.13]
ok: [192.168.122.88]

TASK [install apache and php for CentOS servers] *******************************************************************
skipping: [192.168.122.88]
ok: [192.168.122.13]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.164]

TASK [install mariadb package (CentOS)] ****************************************************************************
skipping: [192.168.122.164]

TASK [install mariadb package (Ubuntu)] ****************************************************************************
ok: [192.168.122.164]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.131]

PLAY RECAP *********************************************************************************************************
192.168.122.13             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.122.131            : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.122.164            : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.122.88             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

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"
```

Make sure to save the file and exit.

```
- 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
    when: ansible_distribution == "CentOS"

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

Figure 3.1. The screenshot above shows the new configuration for site.yml where I add a service in starting the httpd inside of CentOS.

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
```

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.

```
[cserver1@localhost ~]$ sudo systemctl stop httpd
[sudo] password for cserver1:
[cserver1@localhost ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor prese
t: disabled)
   Active: inactive (dead)
     Docs: man:httpd(8)
           man:apachectl(8)

Oct 04 13:02:45 localhost.localdomain systemd[1]: Starting The Apache HTTP Se...
Oct 04 13:02:46 localhost.localdomain httpd[16421]: AH00558: httpd: Could not...
Oct 04 13:02:46 localhost.localdomain systemd[1]: Started The Apache HTTP Ser...
Oct 04 13:08:33 localhost.localdomain systemd[1]: Stopping The Apache HTTP Se...
Oct 04 13:08:34 localhost.localdomain systemd[1]: Stopped The Apache HTTP Ser...
Hint: Some lines were ellipsized, use -l to show in full.
[cserver1@localhost ~]$
```

Figure 3.2. The screenshot above shows the command used in stopping the service inside of CentOS.

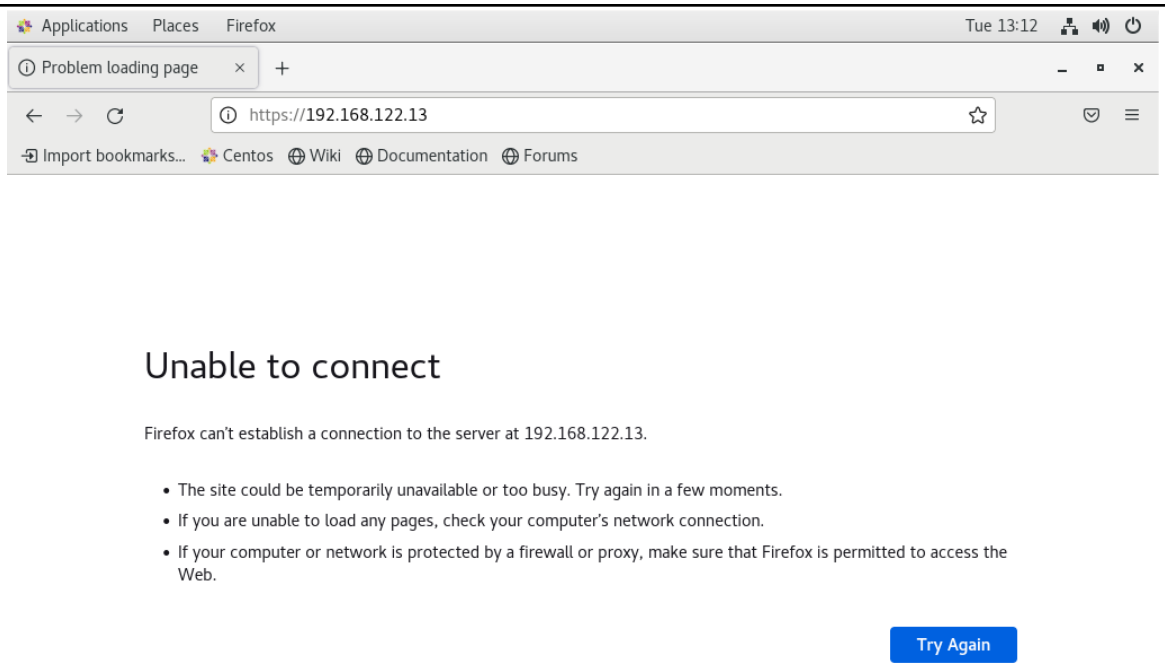


Figure 3.3. The screenshot above shows in validating whether the apache service is running.

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.

```
~/D/r/C/ansible (main|+1) $ ansible-playbook --tags "httpd" --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.88]
ok: [192.168.122.131]
ok: [192.168.122.164]
ok: [192.168.122.13]

TASK [install updates (CentOS)] ************************************************************************************
skipping: [192.168.122.88]
skipping: [192.168.122.164]
skipping: [192.168.122.131]
ok: [192.168.122.13]

TASK [install updates (Ubuntu)] ************************************************************************************
skipping: [192.168.122.13]
ok: [192.168.122.88]
ok: [192.168.122.164]
ok: [192.168.122.131]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.88]
ok: [192.168.122.13]

TASK [install apache and php for CentOS servers] *******************************************************************
skipping: [192.168.122.88]
ok: [192.168.122.13]

TASK [start httpd (CentOS)] ****************************************************************************************
skipping: [192.168.122.88]
changed: [192.168.122.13]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.164]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.131]

PLAY RECAP *********************************************************************************************************
192.168.122.13             : ok=5    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.122.131            : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.122.164            : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.122.88             : ok=3    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
```

Figure 3.4. The screenshot above shows the result after running the site.yml playbook.

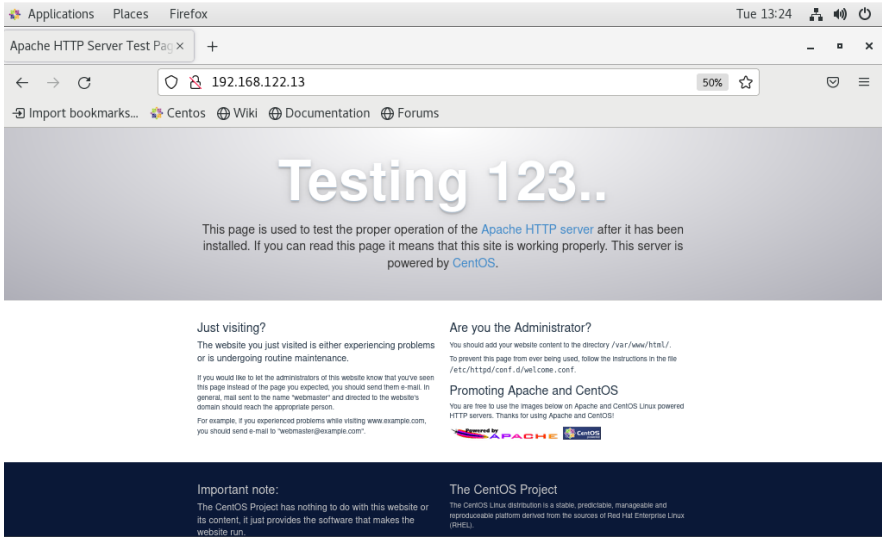


Figure 3.5. The screenshot above shows the validation whether the apache server is successfully started and in our case it does.

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.

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

Figure 3.6. The screenshot above shows the appended command in site.yml under the apache service of CentOS.

```
~/D/r/C/ansible (main|+1) $ ansible-playbook --tags "httpd" --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.164]
ok: [192.168.122.88]
ok: [192.168.122.131]
ok: [192.168.122.13]

TASK [install updates (CentOS)] ************************************************************************************
skipping: [192.168.122.88]
skipping: [192.168.122.164]
skipping: [192.168.122.131]
ok: [192.168.122.13]

TASK [install updates (Ubuntu)] ************************************************************************************
skipping: [192.168.122.13]
ok: [192.168.122.164]
ok: [192.168.122.88]
ok: [192.168.122.131]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.88]
ok: [192.168.122.13]

TASK [install apache and php for CentOS servers] *******************************************************************
skipping: [192.168.122.88]
ok: [192.168.122.13]

TASK [start httpd (CentOS)] ****************************************************************************************
skipping: [192.168.122.88]
changed: [192.168.122.13]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.164]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.122.131]

PLAY RECAP *********************************************************************************************************
192.168.122.13             : ok=5    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.122.131            : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.122.164            : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.122.88             : ok=3    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
```

Figure 3.7. The screenshot above shows the result after running the site.yml playbook.

Reflections:

Answer the following:

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

   It organizes the tasks that are issued to the different servers. In this method we can manage the task that can be implemented in each of the servers. As I observed in this activity that in the part of the inventory file there are three groups specifying different functions of a server. There is a server for a website, database and for file transfer. Inside of the site.yml file are where the groups are called in the hosts variable. This is where the tasks are issued and execute only the specified server/s in the group. This means that putting the servers into groups means we are grouping depending on their functionality and similarities to its neighboring servers.

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

   The importance of tags in a playbook is that it can specify and execute a specific task in a server whether it is in another group or not as long as the tags are specified to the target task. In task 2, we put a bunch of tags per task. These tags are similar to other tags in different tasks. When we run the ”ansible-playbook -tags “tag_name” site.yml” then it will run all of the tasks that have tags that are specified in the command. Ansible supports running multiple tags by separating it my comma and should be inside of a quotation marks.

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

   I do think that some services need to be managed automatically. For the reason that some services are stopped when there are updates or upgrades performed in the computer or the computer did not have that package and therefore it needed to start. This will also provide an assurance that the service is successfully started. As we can see at the last part of this activity, we tried to stop the service httpd and run it using the ansible playbook with a configured task to start the httpd server of the CentOS. We also specify the “enable: true” in the task to automatically run the service when the playbook is executed.

Honor Pledge:

“I affirm that I will not give or receive unauthorized help on this activity and that all will be my own.”