Name: Aducal, John Mark S.	**Date Performed: 10 / 08 / 2022**
Course/Section: CPE232-CPE31S24	**Date Submitted: 10 / 08 / 2022**
Instructor: Engr. 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.

```yaml
---

- 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"
```

Screenshot:

liveserver [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

GNU nano 6.2 site.yml

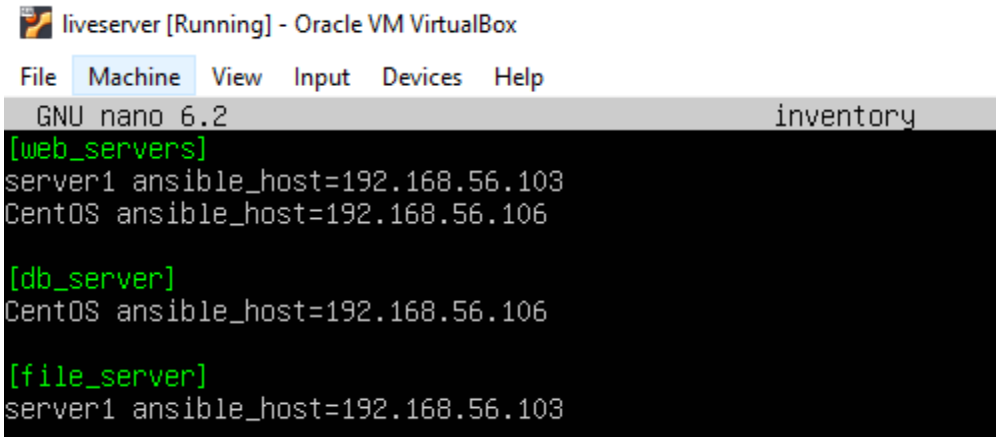
```yaml
---

- hosts: all
  become: true
  pre_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
```

Make sure to save the file and exit.

Screenshot:



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.

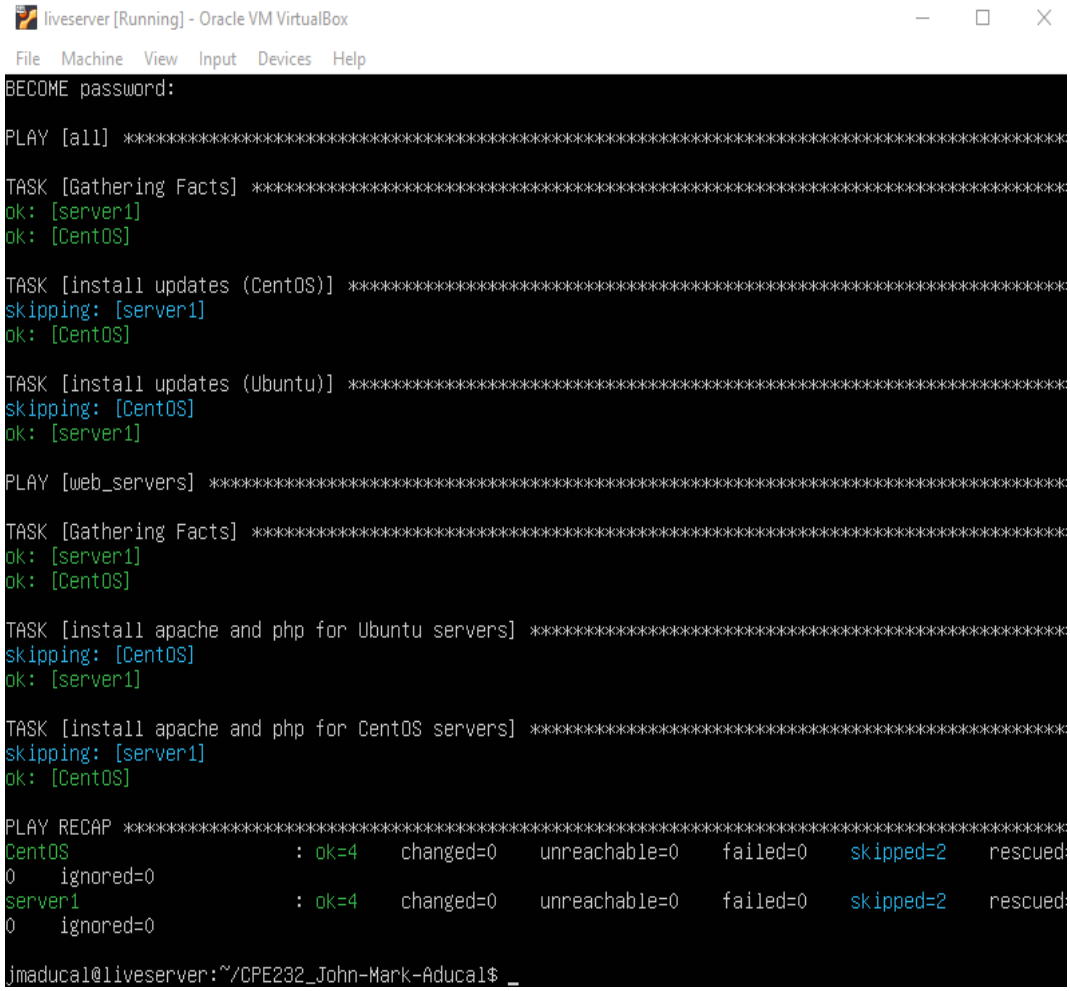
Screenshot:

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

- hosts: all
  become: true
  pre_tasks:

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

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

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

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

  - name: install apache and php for CentOS servers
    dnf:
      name:
        - httpd
        - php
      state: latest
    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.



The tasks for installing updates, apache and php for server1 and CentOS are successful for [web_servers].

4. Let's try to edit 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.

Screenshot:

```
- hosts: db_server
  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"
```

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

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

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

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

PLAY RECAP ****************************************************************************************
CentOS                     : ok=6    changed=2    unreachable=0    failed=0    skipped=3    rescued=
0    ignored=0
server1                    : ok=3    changed=0    unreachable=0    failed=0    skipped=2    rescued=
0    ignored=0

jmaducal@liveserver:~/CPE232_John-Mark-Aducal$
```

I have successfully installed the mariadb package for CentOS server and restarted/enabled the mariadb service.

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.

CentOS

```
jmaducal@CentOS:~
File  Edit  View  Search  Terminal  Help
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: dis
abled)
   Active: active (running) since Fri 2022-10-07 22:21:17 EDT; 3min 11s ago
  Process: 15163 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited, s
tatus=0/SUCCESS)
  Process: 15075 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, stat
us=0/SUCCESS)
 Main PID: 15162 (mysqld_safe)
    Tasks: 20
   CGroup: /system.slice/mariadb.service
           ├─15162 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
           └─15327 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --plu...
```

Describe the output.

The status of mariadb service in CentOS is now active (running).

I have successfully installed mariadb in db_server(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.

Screenshot:

```
- hosts: file_server
  become: true
  tasks:

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

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

```
TASK [install samba package] ***********************************************************************
changed: [server1]

PLAY RECAP *****************************************************************************************
CentOS                     : ok=6    changed=1    unreachable=0    failed=0    skipped=3    rescued=
0    ignored=0
server1                    : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=
0    ignored=0

jmaducal@liveserver:~/CPE232_John-Mark-Aducal$
```

I have successfully installed the samba package for the hosts: [file_servers] (Ubuntu).

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"
```

Screenshot:

```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"
```

```
- 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"
```

Screenshot:

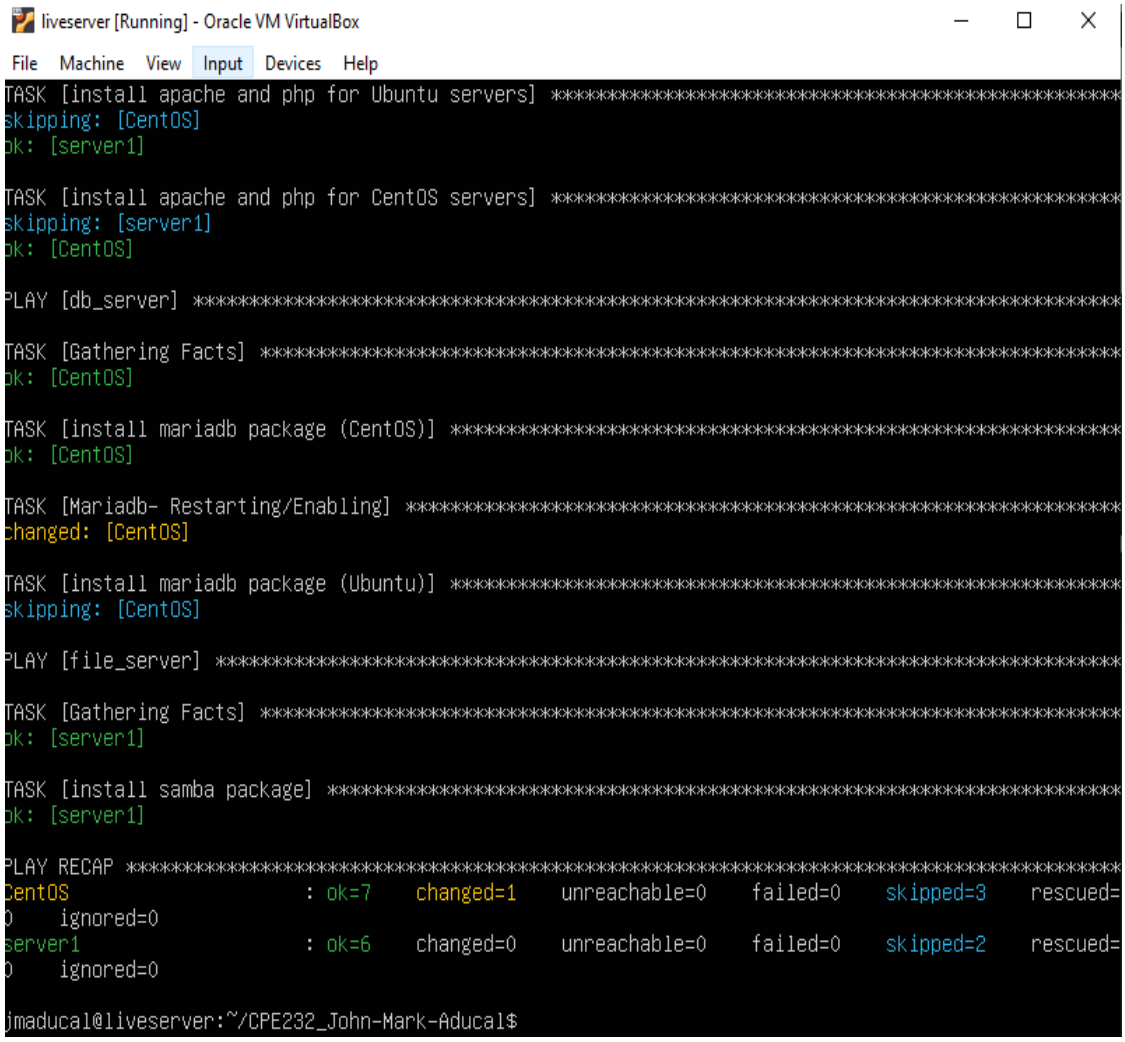
```
- hosts: web_servers
  become: true
  tasks:

  - 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"
```

```
- 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.

```
- hosts: db_server
  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_server
  become: true
  tasks:

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

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



Based on what I've observed the use of tags is to selectively target certain tasks at runtime. Tags avoid repetition and optimize playbook execution time.

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

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

liveserver [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
jmaducal@liveserver:~/CPE232_John-Mark-Aducal$ 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_server): db_server         TAGS: []
      TASK TAGS: [centos, db, mariadb, ubuntu]

  play #4 (file_server): file_server     TAGS: []
      TASK TAGS: [samba]
jmaducal@liveserver:~/CPE232_John-Mark-Aducal$
```

The command shows the list of all tags in our ansible playbook.

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

liveserver [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
skipping: [server1]
ok: [CentOS]

TASK [install updates (Ubuntu)] ***********************************************************************
skipping: [CentOS]
ok: [server1]

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

TASK [Gathering Facts] ********************************************************************************
ok: [server1]
ok: [CentOS]

TASK [install apache and php for CentOS servers] ******************************************************
skipping: [server1]
ok: [CentOS]

PLAY [db_server] **************************************************************************************

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

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

PLAY [file_server] ************************************************************************************

TASK [Gathering Facts] ********************************************************************************
ok: [server1]

PLAY RECAP ********************************************************************************************
CentOS                     : ok=6    changed=0    unreachable=0    failed=0    skipped=1    rescued=
0    ignored=0
server1                    : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=
0    ignored=0

jmaducal@liveserver:~/CPE232_John-Mark-Aducal$ _
```

The command selects only to run the tasks with the tags of CentOS. It plays only the task for CentOS, the execution of play is fast

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

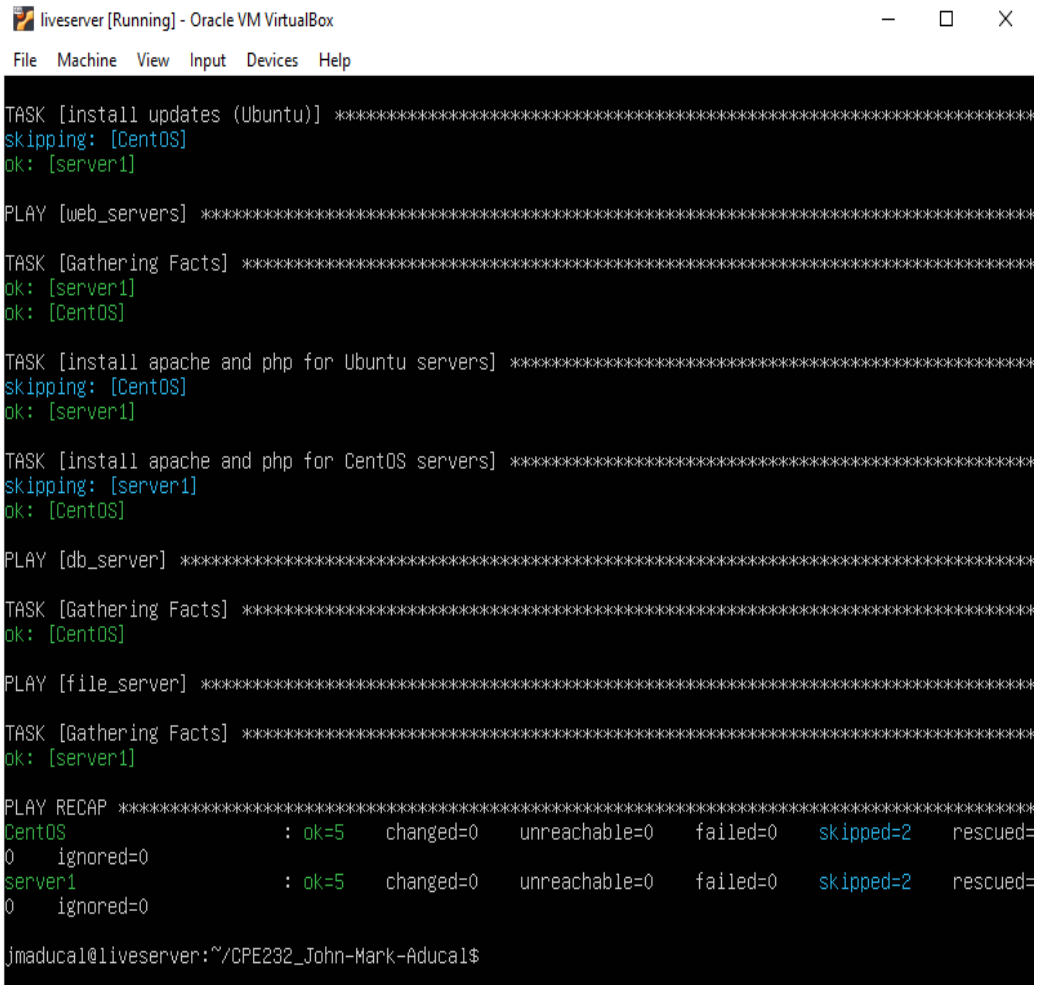
liveserver [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
TASK [install updates (CentOS)] ***********************************************
skipping: [server1]
ok: [CentOS]

TASK [install updates (Ubuntu)] ***********************************************
skipping: [CentOS]
ok: [server1]

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

TASK [Gathering Facts] ********************************************************
ok: [server1]
ok: [CentOS]

PLAY [db_server] **************************************************************

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

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

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

PLAY [file_server] ************************************************************

TASK [Gathering Facts] ********************************************************
ok: [server1]

PLAY RECAP ********************************************************************
CentOS                     : ok=5    changed=0    unreachable=0    failed=0    skipped=2    rescued=
0    ignored=0
server1                    : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=
0    ignored=0

jmaducal@liveserver:~/CPE232_John-Mark-Aducal$
```

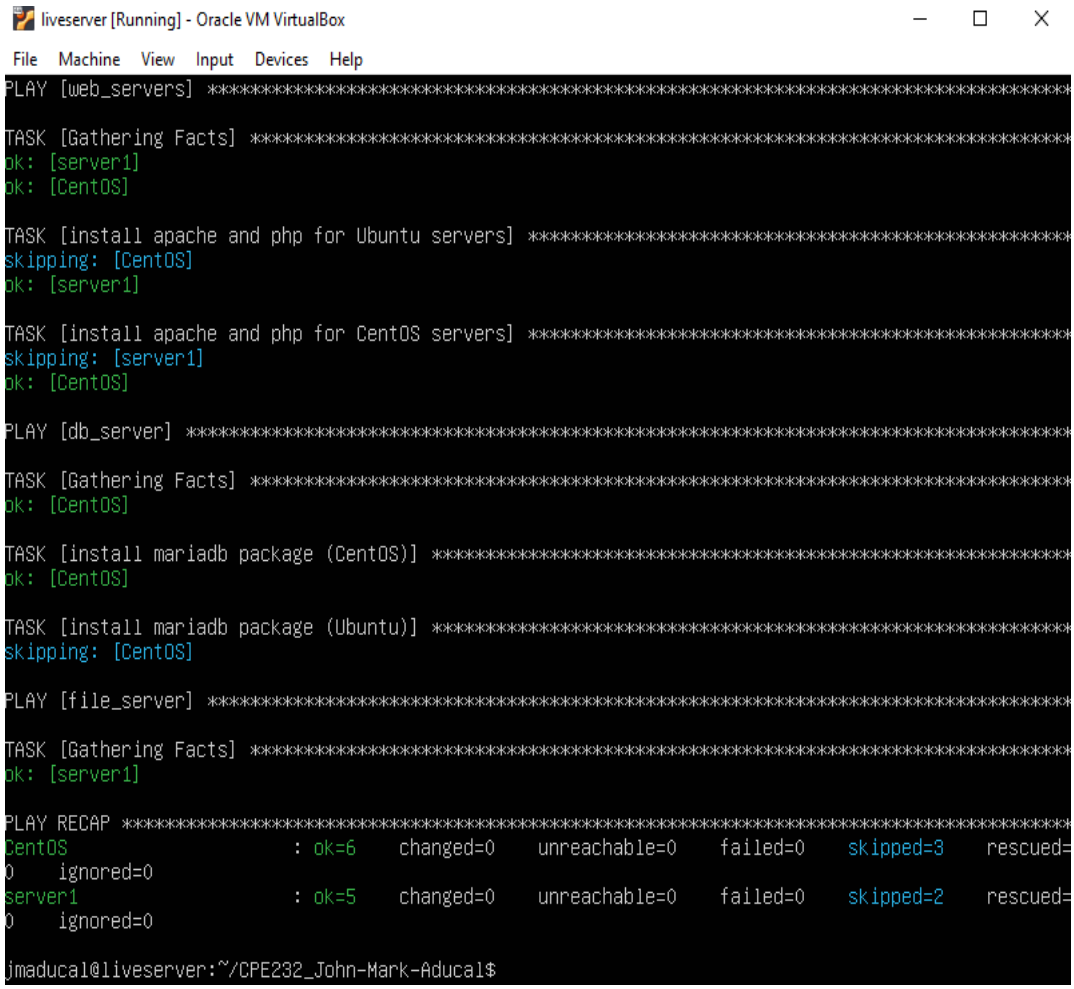
The command selects only to run the tasks with the tags of db. It plays only the task for db, the execution of play is fast.

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



The command selects only to run the tasks with the tags of apache. It only plays the task for apache, the execution of play is fast.

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



The command selects only to run the tasks with the tags of “apache,db”. It plays only the task for apache and db, the execution of play is fast.

Task 3: Managing Services

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

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

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

Figure 3.1.1

Screenshot:

```
- 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.

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
```

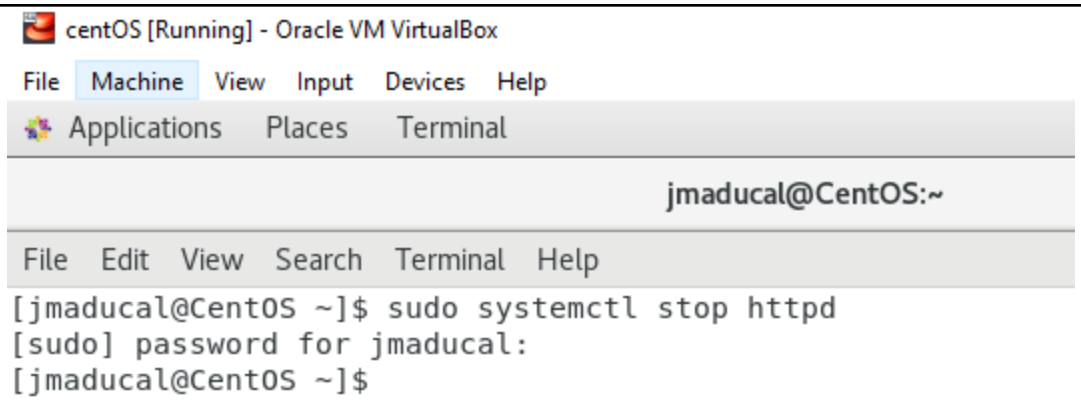
Figure 3.1.2

Screenshot:

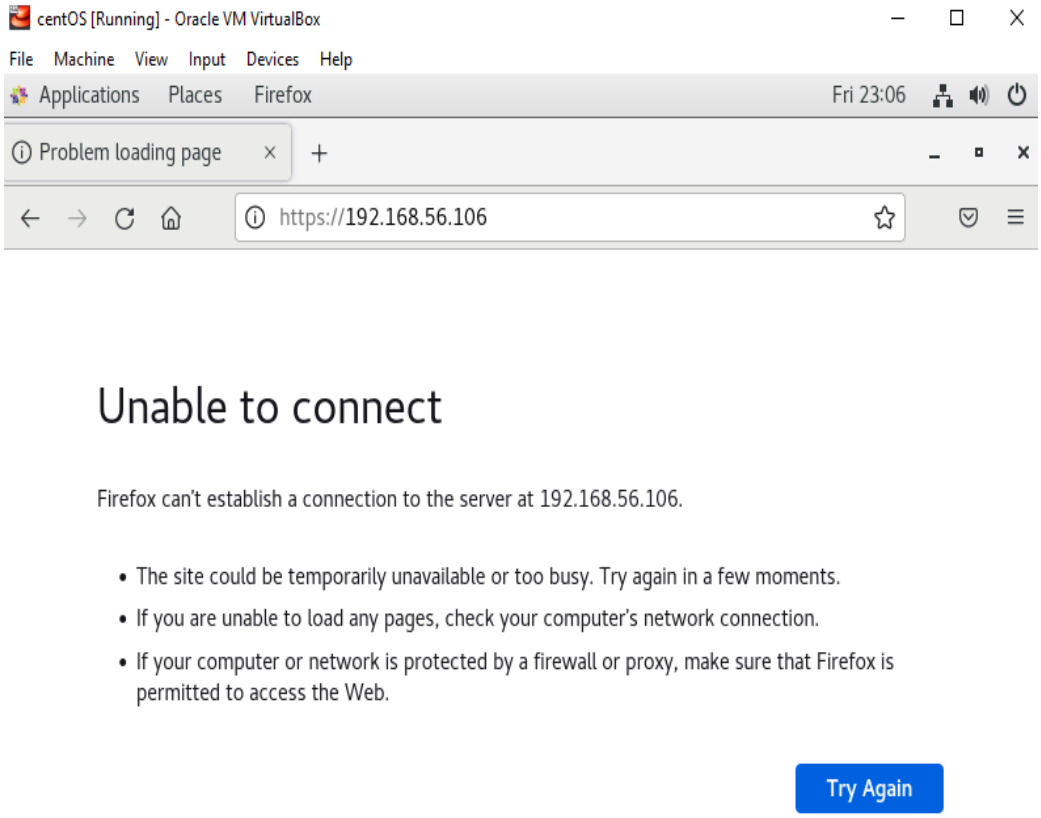
```
- hosts: db_server
  become: true
  tasks:

  - 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.



I have stopped the currently running httpd using the command sudo systemctl stop httpd.



I have already stopped the httpd that is currently running so we do not get a display.

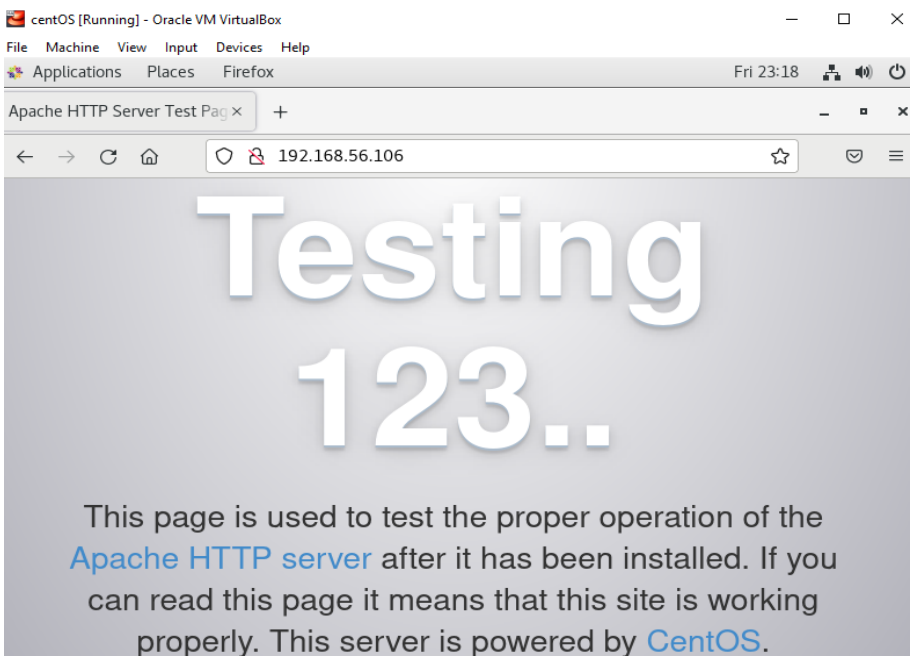
3. Go to the local machine and this time, run the *site.yml* file. Then after running the file, go again to the CentOS server and enter its IP address on the browser.

Describe the result.

```
TASK [start httpd (CentOS)] ****************************************************************
skipping: [server1]
changed: [CentOS]

PLAY [db_server] ***************************************************************************

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

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

PLAY [file_server] *************************************************************************

TASK [Gathering Facts] *********************************************************************
ok: [server1]

PLAY RECAP *********************************************************************************
CentOS                     : ok=7    changed=1    unreachable=0    failed=0    skipped=1    rescued=
0    ignored=0
server1                    : ok=4    changed=0    unreachable=0    failed=0    skipped=3    rescued=
0    ignored=0

jmaducal@liveserver:~/CPE232_John-Mark-Aducal$ _
```

The task start httpd for CentOS has been started and changed from inactive to active.

Result:



The httpd is now running (active) therefore when I try to type the IP Address of CentOS In the browser it displays the Apache HTTP server.

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? Putting our remote servers into a group could be important especially if we only want to have a service for only specific hosts like for example we don't want to have httpd or apache in all our servers. Because for instance, we have different servers like web servers, file servers or database servers.

2. What is the importance of tags in playbooks? Tags are very useful especially if we want to select or target a certain task to run. Tags tell ansible to run or not run a certain task and it can optimize the execution time of playbook. It will execute the play time faster

3. Why do I think some services need to be managed automatically in playbooks? Because not all services run automatically in playbooks, like for example when we install web servers or httpd for CentOS, I have noticed that the service did not start automatically.

Honor Pledge:

"I affirm that I shall not give or receive any unauthorized help on this activity and all the work shall be my own.